



Министерство науки и высшего образования Российской Федерации
Мытищинский филиал
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Космический

КАФЕДРА «Прикладная математика, информатика и вычислительная техника» КЗ-МФ

Лабораторная работа №3

ПО ДИСЦИПЛИНЕ:

Организация ЭВМ и систем

Студент КЗ-66Б
(Группа)

_____ Чернов Владислав Дмитриевич
(Подпись, дата) (И.О.Фамилия)

Студент КЗ-66Б
(Группа)

_____ Братов Аким Романович
(Подпись, дата) (И.О.Фамилия)

Преподаватель

_____ Ефремов Николай Владимирович
(Подпись, дата) (И.О.Фамилия)

2025 г.

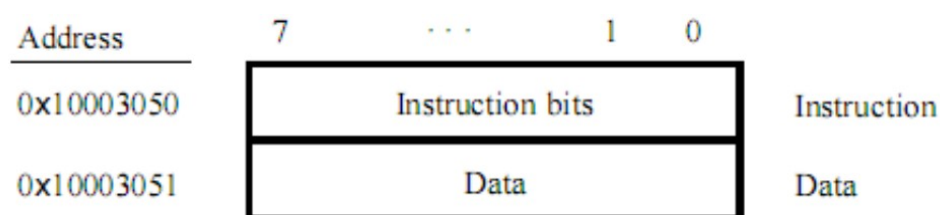
Цель работы

Изучить, как символьная информация представляется в памяти процессорной системы, директивы ассемблера для работы с символьной информацией, научиться выводить её на LCD дисплей стенда.

Приложенные материалы

Описание порта LCD дисплея

Процессорная система «**DE2-115(70) Media Computer**» включает порт для взаимодействия с жидкокристаллическим дисплеем LCD стенда. Программно доступными регистрами порта являются регистр команд, используемый для управления дисплеем LCD и регистр данных, используемый для отправки символьных данных для отображения их на дисплее. Форматы этих регистров, вместе с адресами показаны на рисунке **J-тип**. Данный формат используется для инструкций вызова подпрограмм Call. Поле IMMED26, дополненное справа двумя нулевыми битами и слева четырьмя старшими разрядами счетчика команд PC, используется для задания прямого адреса вызываемой процедуры.



В состав LCD дисплея входит встроенный блок управления, который обрабатывает поступающие в LCD порт команды и три компоненты памяти **DDRAM**, **CGROM**, **CGRAM**.

DDRAM (Display Data RAM) — это оперативная память для записи выводимого на экран дисплея текстового сообщения. Каждой позиции на экране соответствует один байт в этой памяти. Другими словами, в памяти DDRAM размещается буфер выводимых на экран символов. Все символы, записываемые в этот буфер, выводятся на экран LCD в соответствующую позицию.

Позиции на экране адресуются с использованием двух координат (y, x), как показано на рисунке 3.4. Верхней строке соответствует координата y, равная нулю, а нижней строке координата y, равная единице. Координата x меняется в диапазоне от 0 до 39 (0x27), что соответствует 40 символам, которые можно записать в символьный буфер DDRAM для каждой строки экрана. Однако видимыми на экране LCD являются только 16 символов. Содержимое невидимой части символьного буфера можно вывести на экран только в режиме бегущей строки, перемещая текст по экрану. Для этого предусмотрены специальные команды, о которых речь пойдет ниже.

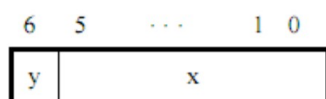


Рис. 3.4 Схема адресации позиций на экране символьного дисплея

Таким образом, для адресации 80 позиций на экране и, следовательно, байтов в символьном буфере, то есть в памяти DDRAM, используется семиразрядный счетчик адреса. Выделяемая курсором на экране LCD позиция, соответствует текущему

значению счетчика адреса DDRAM. Если записать по этому адресу ASCII код символа, то сам символ будет отображен в выделенной позиции экрана.

В соответствии с рис. 3.4, диапазон адресов в DDRAM для верхней и нижней строк дисплея начинается с адресов 0x00 и 0x40, как показано на рис. 3.5. Голубым цветом показаны позиции, видимые на экране.

	0	1	2	...	15	16	...	39
0	00	01	02	...	0F	+24 locations		
1	40	41	42	...	4F	+24 locations		

Рис. 3.5 Адреса позиций видимой и невидимой части LCD дисплея

Следующей компонентой памяти является постоянная память **CGROM** (Character Generator ROM). Эта память доступна только для чтения, и её изменить нельзя. В памяти CGROM хранится таблица с изображениями символов (таблица 3.1), то есть это память знакогенератора. Когда в какую-нибудь ячейку DDRAM записывается байт с кодом символа, то из этой таблицы берется изображение соответствующего символа и прорисовывается на экране LCD.

CGROM	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	40
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	80
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	100
100	101	102	103	104	105	106	107	108	109	10A	10B	10C	10D	10E	10F	110
110	111	112	113	114	115	116	117	118	119	11A	11B	11C	11D	11E	11F	120
120	121	122	123	124	125	126	127	128	129	12A	12B	12C	12D	12E	12F	130
130	131	132	133	134	135	136	137	138	139	13A	13B	13C	13D	13E	13F	140
140	141	142	143	144	145	146	147	148	149	14A	14B	14C	14D	14E	14F	150
150	151	152	153	154	155	156	157	158	159	15A	15B	15C	15D	15E	15F	160
160	161	162	163	164	165	166	167	168	169	16A	16B	16C	16D	16E	16F	170
170	171	172	173	174	175	176	177	178	179	17A	17B	17C	17D	17E	17F	180
180	181	182	183	184	185	186	187	188	189	18A	18B	18C	18D	18E	18F	190
190	191	192	193	194	195	196	197	198	199	19A	19B	19C	19D	19E	19F	200
200	201	202	203	204	205	206	207	208	209	20A	20B	20C	20D	20E	20F	210
210	211	212	213	214	215	216	217	218	219	21A	21B	21C	21D	21E	21F	220
220	221	222	223	224	225	226	227	228	229	22A	22B	22C	22D	22E	22F	230
230	231	232	233	234	235	236	237	238	239	23A	23B	23C	23D	23E	23F	240
240	241	242	243	244	245	246	247	248	249	24A	24B	24C	24D	24E	24F	250
250	251	252	253	254	255	256	257	258	259	25A	25B	25C	25D	25E	25F	260
260	261	262	263	264	265	266	267	268	269	26A	26B	26C	26D	26E	26F	270
270	271	272	273	274	275	276	277	278	279	27A	27B	27C	27D	27E	27F	280
280	281	282	283	284	285	286	287	288	289	28A	28B	28C	28D	28E	28F	290
290	291	292	293	294	295	296	297	298	299	29A	29B	29C	29D	29E	29F	300
300	301	302	303	304	305	306	307	308	309	30A	30B	30C	30D	30E	30F	310
310	311	312	313	314	315	316	317	318	319	31A	31B	31C	31D	31E	31F	320
320	321	322	323	324	325	326	327	328	329	32A	32B	32C	32D	32E	32F	330
330	331	332	333	334	335	336	337	338	339	33A	33B	33C	33D	33E	33F	340
340	341	342	343	344	345	346	347	348	349	34A	34B	34C	34D	34E	34F	350
350	351	352	353	354	355	356	357	358	359	35A	35B	35C	35D	35E	35F	360
360	361	362	363	364	365	366	367	368	369	36A	36B	36C	36D	36E	36F	370
370	371	372	373	374	375	376	377	378	379	37A	37B	37C	37D	37E	37F	380
380	381	382	383	384	385	386	387	388	389	38A	38B	38C	38D	38E	38F	390
390	391	392	393	394	395	396	397	398	399	39A	39B	39C	39D	39E	39F	400
400	401	402	403	404	405	406	407	408	409	40A	40B	40C	40D	40E	40F	410
410	411	412	413	414	415	416	417	418	419	41A	41B	41C	41D	41E	41F	420
420	421	422	423	424	425	426	427	428	429	42A	42B	42C	42D	42E	42F	430
430	431	432	433	434	435	436	437	438	439	43A	43B	43C	43D	43E	43F	440
440	441	442	443	444	445	446	447	448	449	44A	44B	44C	44D	44E	44F	450
450	451	452	453	454	455	456	457	458	459	45A	45B	45C	45D	45E	45F	460
460	461	462	463	464	465	466	467	468	469	46A	46B	46C	46D	46E	46F	470
470	471	472	473	474	475	476	477	478	479	47A	47B	47C	47D	47E	47F	480
480	481	482	483	484	485	486	487	488	489	48A	48B	48C	48D	48E	48F	490
490	491	492	493	494	495	496	497	498	499	49A	49B	49C	49D	49E	49F	500
500	501	502	503	504	505	506	507	508	509	50A	50B	50C	50D	50E	50F	510
510	511	512	513	514	515	516	517	518	519	51A	51B	51C	51D	51E	51F	520
520	521	522	523	524	525	526	527	528	529	52A	52B	52C	52D	52E	52F	530
530	531	532	533	534	535	536	537	538	539	53A	53B	53C	53D	53E	53F	540
540	541	542	543	544	545	546	547	548	549	54A	54B	54C	54D	54E	54F	550
550	551	552	553	554	555	556	557	558	559	55A	55B	55C	55D	55E	55F	560
560	561	562	563	564	565	566	567	568	569	56A	56B	56C	56D	56E	56F	570
570	571	572	573	574	575	576	577	578	579	57A	57B	57C	57D	57E	57F	580
580	581	582	583	584	585	586	587	588	589	58A	58B	58C	58D	58E	58F	590
590	591	592	593	594	595	596	597	598	599	59A	59B	59C	59D	59E	59F	600
600	601	602	603	604	605	606	607	608	609	60A	60B	60C	60D	60E	60F	610
610	611	612	613	614	615	616	617	618	619	61A	61B	61C	61D	61E	61F	620
620	621	622	623	624	625	626	627	628	629	62A	62B	62C	62D	62E	62F	630
630	631	632	633	634	635	636	637	638	639	63A	63B	63C	63D	63E	63F	640
640	641	642	643	644	645	646	647	648	649	64A	64B	64C	64D	64E	64F	650
650	651	652	653	654	655	656	657	658	659	65A	65B	65C	65D	65E	65F	660
660	661	662	663	664	665	666	667	668	669	66A	66B	66C	66D	66E	66F	670
670	671	672	673	674	675	676	677	678	679	67A	67B	67C	67D	67E	67F	680
680	681	682	683	684	685	686	687	688	689	68A	68B	68C	68D	68E	68F	690
690	691	692	693	694	695	696	697	698	699	69A	69B	69C	69D	69E	69F	700
700	701	702	703	704	705	706	707	708	709	70A	70B	70C	70D	70E	70F	710
710	711	712	713	714	715	716	717	718	719	71A	71B	71C	71D	71E	71F	720
720	721	722	723	724	725	726	727	728	729	72A	72B	72C	72D	72E	72F	730
730	731	732	733	734	735	736	737	738	739	73A	73B	73C	73D	73E	73F	740
740	741	742	743	744	745	746	747	748	749	74A	74B	74C	74D	74E	74F	750
750	751	752	753	754	755	756	757	758	759	75A	75B	75C	75D	75E	75F	760
760	761	762	763	764	765	766	767	768	769	76A	76B	76C	76D	76E	76F	770
770	771	772	773	774	775	776	777	778	779	77A	77B	77C	77D	77E	77F	780
780	781	782	783	784	785	786	787	788	789	78A	78B	78C	78D	78E	78F	790
790	791	792	793	794	795	796	797	798	799	79A	79B	79C	79D	79E	79F	800
800	801	802	803	804	805	806	807	808	809	80A	80B	80C	80D	80E	80F	810
810	811	812	813	814	815	816	817	818	819	81A	81B	81C	81D	81E	81F	820
820	821	822	823	824	825	826	827	828	829	82A	82B	82C	82D	82E	82F	830
830	831	832	833	834	835	836	837	838	839	83A	83B	83C	83D	83E	83F	840
840	841	842	843	844	845	846	847	848	849	84A	84B	84C	84D	84E	84F	850
850	851	852	853	854	855	856	857	858	859	85A	85B	85C	85D	85E	85F	860
860	861	862	863	864	865	866	867	868	869	86A	86B	86C	86D	86E	86F	870
870	871	872	873	874	875	876	877	878	879	87A	87B	87C	87D	87E	87F	880
880	881	882	883	884	885	886	887	888	889	88A	88B	88C	88D	88E	88F	890
890	891	892	893	894	895	896	897	898	899	89A						

байт, начиная с адреса 8. В последний байт с изображением символа рекомендуется записывать нули, чтобы использовать это место для корректного отображения на экране LCD мерцающего курсора в виде нижнего подчеркивания.



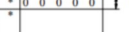
Character Codes (DDRAM data)								CGRAM Address				Character Patterns (CGRAM data)												
7	6	5	4	3	2	1	0	5	4	3	2	1	0	7	6	5	4	3	2	1	0			
High				Low				High				Low				High				Low				
0 0 0 0 * 0 0 0								0 0 0				0	0	*	*	*	*	0		Character pattern (1)				
												0	0	1	*	*	*	*			0	0	0	0
												0	1	0	*	*	*	*			0	0	0	0
												0	1	1	*	*	*	*			0	0	0	0
												1	0	1	*	*	*	*			0	0	0	0
												1	1	0	*	*	*	*			0	0	0	0
												1	1	1	*	*	*	*			0	0	0	0
												0	0	0	*	*	*	*			0	0	0	0
												0	0	1	*	*	*	*			0	0	0	0
												0	1	0	*	*	*	*			0	0	0	0
0 0 0 0 * 0 0 1								0 0 1				0	1	0	*	*	*	0		Character pattern (2)				
												0	1	1	*	*	*	0			0	0	0	
												1	0	0	*	*	*	0			0	0	0	
												1	1	0	*	*	*	0			0	0	0	
												1	1	1	*	*	*	0			0	0	0	
												0	0	0	*	*	*	0		Cursor pattern				
												0	0	1	*	*	*	0			0	0	0	
0 0 0 0 * 1 1 1								1 1 1				1	0	0	*	*	*							
												1	0	1	*	*	*							
												1	1	0	*	*	*							
												1	1	1	*	*	*							
															*	*	*							

Рис. 3.6 Принцип адресации CGRAM памяти и формирования изображения символов на экране LCD.

I/D — инкремент или декремент счетчика адреса. По умолчанию, этот бит равен нулю и выполняется декремент счетчика. То есть каждый следующий байт будет записываться в предыдущую, n-1 ячейку DDRAM. Если установить этот бит в единицу, то будет выполняться инкремент счетчика.

D — бит включения дисплея. Если очистить этот бит, то изображение исчезнет с экрана, а чтобы оно вновь появилось, в эту позицию надо записать 1.

B — бит включения курсора в виде мигающего черного прямоугольника.

S/C — сдвиг курсора или экрана. Если ноль, то сдвигается курсор. Если единица, то текст на экране. По одному разу за команду.

R/L — определяет направление сдвига. 0 — влево, 1 — вправо.

AG — адрес в памяти знакогенератора CGRAM.

AD — адрес в памяти символьного буфера DDRAM.

DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Значение
0	0	0	0	0	0	0	1	Очистка экрана. Счетчик адреса устанавливается в 0 позицию DDRAM.
0	0	0	0	0	0	1	—	Адресация в DDRAM. Отмена сдвигов, Счетчик адреса устанавливается в 0.
0	0	0	0	0	1	I/D	S	Настройка режима вывода текста на экран.
0	0	0	0	0	1	0	0	Поле для самостоятельного заполнения
0	0	0	0	0	1	0	1	Поле для самостоятельного заполнения
0	0	0	0	0	1	1	0	Поле для самостоятельного заполнения
0	0	0	0	0	1	1	1	Поле для самостоятельного заполнения
0	0	0	0	1	D	C	B	Настройка режима отображения
0	0	0	1	S/C	R/L	—	—	Сдвиг курсора или текста, в зависимости от установленных битов.
0	1	AG	AG	AG	AG	AG	AG	Переключить адресацию на CGRAM и задать адрес в CGRAM.
1	AD	AD	AD	AD	AD	AD	AD	Переключить адресацию на DDRAM и задать адрес в DDRAM.

Таблица 3.3. ASCII - коды си

	00	10	20	30	40	50	60	70
0		▢		0	1	2	3	4
1	▢	▢	▢	▢	▢	▢	▢	▢
2	▢	▢	▢	▢	▢	▢	▢	▢
3	▢	▢	▢	▢	▢	▢	▢	▢
4	▢	▢	▢	▢	▢	▢	▢	▢
5	▢	▢	▢	▢	▢	▢	▢	▢
6	▢	▢	▢	▢	▢	▢	▢	▢
7	▢	▢	▢	▢	▢	▢	▢	▢
8	▢	▢	▢	▢	▢	▢	▢	▢
9	▢	▢	▢	▢	▢	▢	▢	▢
A	▢	▢	▢	▢	▢	▢	▢	▢
B	▢	▢	▢	▢	▢	▢	▢	▢
C	▢	▢	▢	▢	▢	▢	▢	▢
D	▢	▢	▢	▢	▢	▢	▢	▢
E	▢	▢	▢	▢	▢	▢	▢	▢
F	▢	▢	▢	▢	▢	▢	▢	▢

В файле `address_map.s` содержатся поименованные константы. Первая из них задаёт адрес LCD порта, а остальные представляют собой часто используемые команды для управления LCD дисплеем. В дальнейшем при написании программ, работающих с LCD дисплеем, рекомендуется данный файл подключать с помощью директивы `.include "имя_файла"`.

Листинг 4

Листинг файла `address_map.s`

```
.equ lcd, 0x10003050 /*Адрес регистра lcd*/
.equ clear, 0x00000001 /*Код команды очистки экрана lcd*/
.equ set1, 0b10000000 /*Код команды установки курсора в начало первой строки*/
.equ set2, 0b11000000 /*Код команды установки курсора в начало второй строки*/
.equ off, 0b00001100 /*Код команды выключения курсора*/
.equ right, 0b00011100 /*Код команды сдвига текста вправо*/
.equ left, 0b00011000 /*Код команды сдвига текста влево*/
```

Листинг 5

Листинг программы `lab_lcd.s`

```
.include "address_map.s" /*Добавляем константы из файла*/
.global _start
.text /*Определяем секцию кода*/

_start:
    movia r2, lcd /*Адрес LCD в регистр r2*/
    movi r3, clear /*Команду очистки lcd в r3*/
    stbio r3, 0(r2) /*Очищаем экран lcd*/

    movi r3, set1
    stbio r3, 0(r2) /*Устанавливаем курсор в начало первой строки на LCD*/

    movia r4, String1 /*В регистр r4 записываем адрес первой строки в ОП*/
cikl:
    ldb r5, 0(r4) /*Считываем символ из строки (ОП) и помещаем его в r5*/
    beq r5, zero, met /*Если считанный символ равен 0, выход из цикла*/
    stbio r5, 1(r2) /*Пересылаем из r5 в регистр данных lcd*/
    addi r4, r4, 1 /*Инкрементируем адрес символа в строке*/
    br cikl /*Повторяем для остальных символов*/

met:
    movi r3, set2
    stbio r3, 0(r2) /*Устанавливаем курсор в начало второй строки на LCD*/

    movia r4, String2 /*Выводим вторую строку на LCD*/
cikl2:
    ldb r5, 0(r4)
    beq r5, zero, 1f
    stbio r5, 1(r2)
    addi r4, r4, 1
    br cikl2

1:
    movi r3, off
    stbio r3, 0(r2) /*Выключаем курсор*/

    movi r3, right
    stbio r3, 0(r2) /*Сдвигаем строки вправо*/

    br -.4 /*Повторяем сдвиг строк*/

2: movi r3, left
    stbio r3, 0(r2) /*Сдвигаем строки влево*/
    br b2+4 /*Повторяем сдвиг строк*/

.data /*Определяем секцию данных*/
String1:
.asciz "Familiya"
String2:
.asciz "Imya"
.end
    konec"
```

Программа, представленная в листинге 6, создаёт в CGRAM изображение заглавной буквы «Г» с кодом 02 и выводит её в начале первой строки на экран LCD.

```
.include "address_map.s"

movia r2, 0x10003050 /*Адрес lcd в R2*/
movia r9, 0b01010000 /*Готовим команду для выбора адреса 0x10 в CGRAM
*/
stbio r9, 0(r2) /*Отправка команды в регистр команд LCD */

movia r9, 0b00011111 /*Первый байт данных для символа Г */
stbio r9, 1(r2) /*Отправка байта в регистр данных*/
movia r9, 0b00010000 /*Второй байт данных для символа Г */
stbio r9, 1(r2)
movia r9, 0b00010000 /*Третий байт данных для символа Г */
stbio r9, 1(r2)
movia r9, 0b00010000 /*Четвертый байт данных для символа Г */
stbio r9, 1(r2)
movia r9, 0b00010000 /*Пятый байт данных для символа Г */
stbio r9, 1(r2)
movia r9, 0b00010000 /*Шестой байт данных для символа Г */
stbio r9, 1(r2)
movia r9, 0b00010000 /*Седьмой байт данных для символа Г */
stbio r9, 1(r2)
movia r9, 0b00000000 /*Восьмой байт данных для символа Г */
stbio r9, 1(r2)

movia r9, 0b10000000 /*Команда переключение адреса на DDRAM и указатель на адрес 00000000 — первый символ в первой строке */
stbio r9, 0(r2) /*Отправка команды в регистр команд*/

movia r9, 0b00000010 /*Данные, код 10 — именно он используется для кодирования «Г» */
stbio r9, 1(r2) /*Отправка данных в регистр данных*/
```

```
.text
.equ lcd, 0x10003050 /*Адрес регистра команд lcd*/
.equ clear, 0b00000001 /*Код команды очистки экрана lcd*/
.equ set1, 0b10000000 /*Код команды переключения указателя на адрес 00000000 — первый символ в первой строке*/
.global _start
_start:
    movia sp, 0x3ffffc /*Настройка стека*/
    movia r6, lcd /*Адрес lcd в R6*/
    movia r3, clear /*Команда очистки экрана в R3*/
    stbio r3, 0(r6) /*Отправка команды очистки экрана*/
    movia r3, set1 /*Команды переключение указателя на адрес 00000000 в регистр r3*/
    stbio r3, 0(r6) /*Отправка команды переключения*/
    movia r4, string1 /*Считать значение метки string1*/

loop:
    ldbu r7, 0(r4) /*Считать значение первого символа по адресу в r4*/
    beq r7, zero, STOP /*Выйти из цикла, когда считанное значение будет равно нулю*/
    call print /*Вызов процедуры print, которая будет печатать символ*/
    addi r4, r4, 1 /*Инкремент значения адреса в r4, чтобы перейти к следующему символу*/
    br loop /*Зацикливание*/
STOP:
    br STOP

.data
string1:
    .asciz "КАФЕДРА КЗ" /*Строка для печати*/
.end
Листинг процедуры print:

# нужно настроить стек
# передавать символ через регистр r7

.text
.equ lcd, 0x10003050 /*Адрес регистра lcd*/
.equ clear, 0b00000001 /*Код команды очистки экрана lcd*/
.equ set1, 0b10000000 /*Код команды установки курсора в первую позицию первой строки*/
```

```

.equ set2,0b11000000      /*Код команды установки курсора в начало 2 строки*/
.equ off,0b00001100      /*Код команды выключения курсора*/
.equ right,0b00011100     /*Код команды сдвига экрана вправо*/
.equ left,0b00011000      /*Код команды сдвига экрана влево*/
.equ create_c,0b01000000  /*переключить память на CGRAM и задать адрес*/
.equ com_bf,0b100000000   /*переключить память на DDRAM и задать адрес*/
# .equ address_bf,0x100

/* это ASCII коды букв русского алфавита */

.equ a,0x90 /*А */
.equ b,0x91 /*Б */
.equ v,0x92 /*В */
.equ g,0x93 /*Г */
.equ d,0x94 /*Д */
.equ e,0x95 /*Е */
.equ yo,0x81 /*Ё */
.equ jz,0x96 /*Ж */
.equ z,0x97 /*З */
.equ i,0x98 /*И */
.equ iy,0x99 /*Й */
.equ k,0x9a /*К */
.equ l,0x9b /*Л */
.equ m,0x9c /*М */
.equ n,0x9d /*Н */
.equ o,0x9e /*О */
.equ p,0x9f /*П */
.equ r,0xa0 /*Р */
.equ s,0xa1 /*С */
.equ t,0xa2 /*Т */
.equ u,0xa3 /*У */
.equ f,0xa4 /*Ф */
.equ h,0xa5 /*Х */
.equ c,0xa6 /*Ц */
.equ ch,0xa7 /*Ч */
.equ sh,0xa8 /*Ш */
.equ chsh,0xa9 /*Щ */
.equ hard_s,0xaa /*Ъ */
.equ hard_i,0xab /*Ы */
.equ soft_s,0xac /*Ь */
.equ ye,0xad /*Э */
.equ yu,0xae /*Ю */
.equ ya,0xaf /*Я */

```

.global print

print:

```

    stw r8,0(sp)      /*r8 в стек */

```

```

    subi sp,sp,4

```

```

    stw r5,0(sp)      /*r5 в стек */

```

```

    subi sp,sp,4

```

```

    stw r9,0(sp)      /*r9 в стек */

```



```
subi sp,sp,4
```

```
stw r2,0(sp)      /*r2 в стек */  
subi sp,sp,4
```

```
stw r3,0(sp)      /*r3 в стек */  
subi sp,sp,4
```

```
movia r8,0xd0      /* 0xd0 символ конца строки */  
beq r7, r8, the_end /* если в r7 символ конца строки */  
movia r2,lcd        /*Адрес команды lcd в R2 */  
movia r3,lcd+1      /*Адрес данных в lcd в R3 */
```

```
ldbio r8,0(r2)      /* читаем регистр команд lcd */  
and r5, r5, zero    /* обнуляем r5 */  
andi r5, r8, 0b111  /* выделяем младшие три разряда */
```

```
movia r9, a          /* в r9 записываем ASCII код буквы а */  
beq r7, r9, it_a     /* если в r7 код буквы а */  
movia r9, b  
beq r7, r9, it_b  
movia r9, v  
beq r7, r9, it_v  
movia r9, g  
beq r7, r9, it_g  
movia r9, d  
beq r7, r9, it_d  
movia r9, e  
beq r7, r9, it_e  
movia r9, yo  
beq r7, r9, it_yo  
movia r9, jz  
beq r7, r9, it_jz  
movia r9, z  
beq r7, r9, it_z  
movia r9, i  
beq r7, r9, it_i  
movia r9, iy  
beq r7, r9, it_iy  
movia r9, k  
beq r7, r9, it_k  
movia r9, l  
beq r7, r9, it_l  
movia r9, m  
beq r7, r9, it_m  
movia r9, n  
beq r7, r9, it_n  
movia r9, o  
beq r7, r9, it_o  
movia r9, p  
beq r7, r9, it_p  
movia r9, r  
beq r7, r9, it_r  
movia r9, s
```

```

beq  r7, r9, it_s
movia r9, t
beq  r7, r9, it_t
movia r9, u
beq  r7, r9, it_u
movia r9, f
beq  r7, r9, it_f
movia r9, h
beq  r7, r9, it_h
movia r9, c
beq  r7, r9, it_c
movia r9, ch
beq  r7, r9, it_ch
movia r9, sh
beq  r7, r9, it_sh
movia r9, chsh
beq  r7, r9, it_chsh
movia r9, hard_s
beq  r7, r9, it_hard_s
movia r9, hard_i
beq  r7, r9, it_hard_i
movia r9, soft_s
beq  r7, r9, it_soft_s
movia r9, ye
beq  r7, r9, it_ye
movia r9, yu
beq  r7, r9, it_yu
movia r9, ya
beq  r7, r9, it_ya
br another_s

```

it_a:

```

movia r5, str                /* если а, то используем латинскую букву */
ldb r5, 0(r5)
br printing

```

```

it_b:                        /* если Б, то */
slli r9, r5, 0x3            /* сдвигаем r5 на 3 разряда влево */
ori r9, r9, 0b01000000      /* устанавливаем 6 разряд r9 */
stbio r9, 0(r2)             /* отправляем в регистр команд */

movia r9, 0b00011111        /* выполняется запись байтов с изображением Б */
stbio r9, 0(r3)
movia r9, 0b00010000
stbio r9, 0(r3)
movia r9, 0b00010000
stbio r9, 0(r3)
movia r9, 0b00011110
stbio r9, 0(r3)
movia r9, 0b00010001
stbio r9, 0(r3)
movia r9, 0b00010001

```

```

    stbio r9, 0(r3)
    movia r9, 0b00011110
    stbio r9, 0(r3)
    movia r9, 0b00000000
    stbio r9, 0(r3)
br printing

it_v:                                /* если В, то используется латинская буква */

    movia r5, str
    ldb r5, 1(r5)
    br printing

it_g:                                /* если Г, то

    slli r9, r5, 0x3
    ori r9, r9, 0b01000000
    stbio r9, 0(r2)

    movia r9, 0b00011111    /* выполняется запись байтов с изображением Г */
    stbio r9, 0(r3)
    movia r9, 0b00010000
    stbio r9, 0(r3)
    movia r9, 0b00010000
    stbio r9, 0(r3)
    movia r9, 0b00010000
    stbio r9, 0(r3)
    movia r9, 0b00010000
    stbio r9, 0(r3)
    movia r9, 0b00010000
    stbio r9, 0(r3)
    movia r9, 0b00010000
    stbio r9, 0(r3)
    movia r9, 0b00000000
    stbio r9, 0(r3)
    br printing

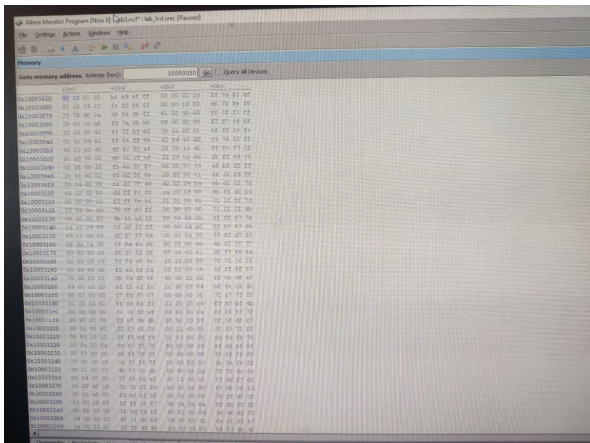
it_d:

    slli r9, r5, 0x3
    ori r9, r9, 0b01000000
    stbio r9, 0(r2)

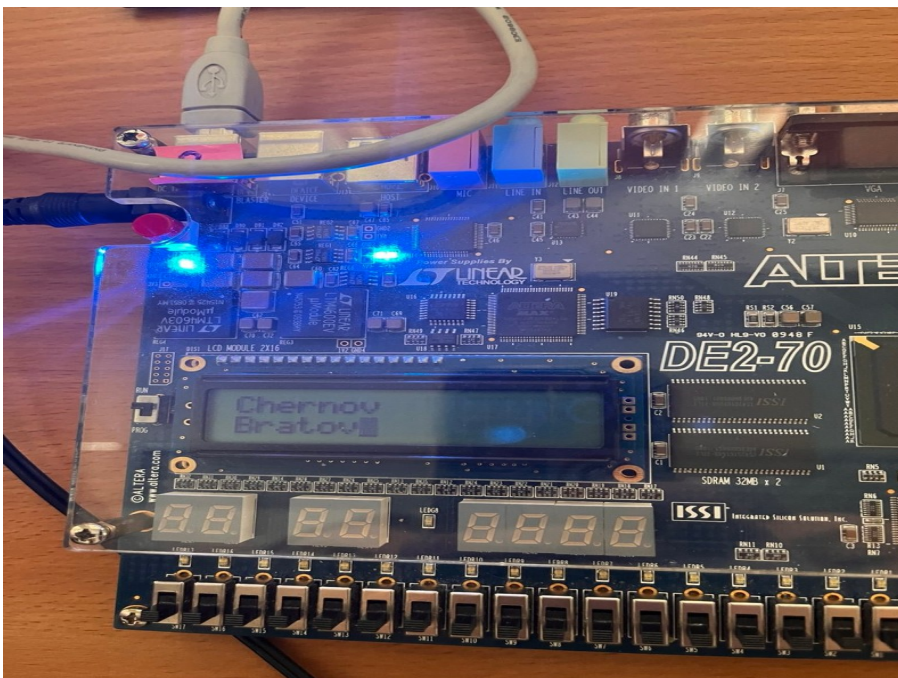
```

Выполнение заданий лабораторной работы

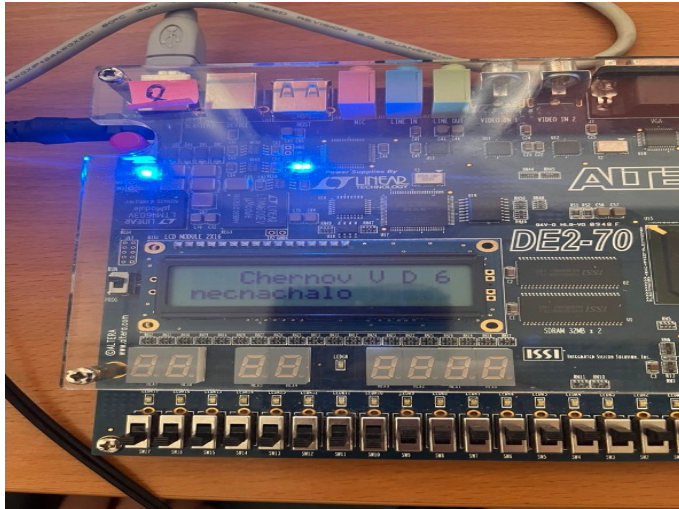
1 Выполним очистку LCD-дисплея, записывая соответствующую команду в регистр управления LCD (первая команда в таблице 3.2).



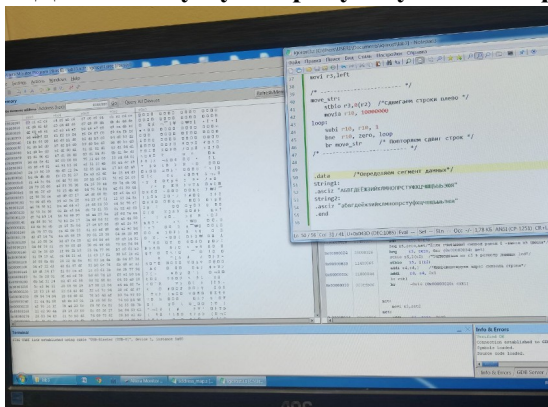
2 С помощью команд и ascii кодов вводим наши фамилии.



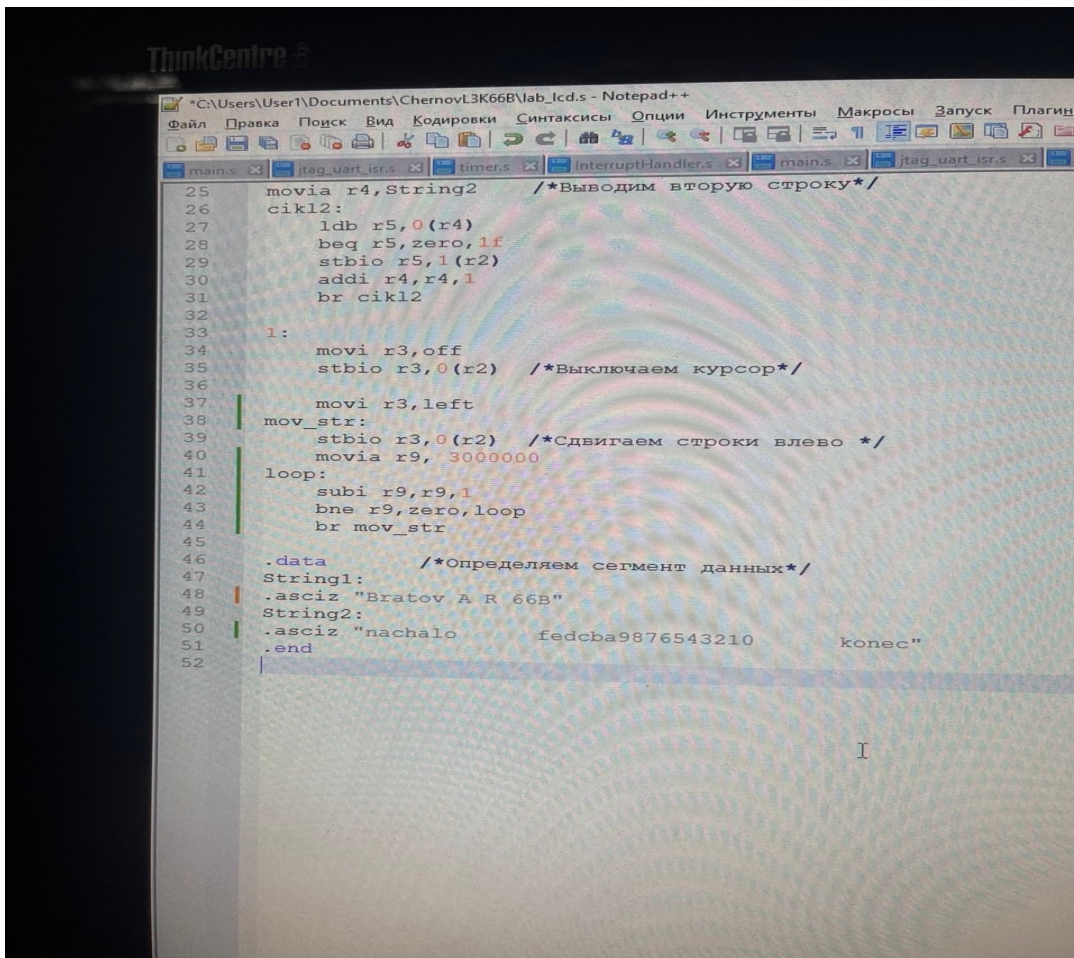
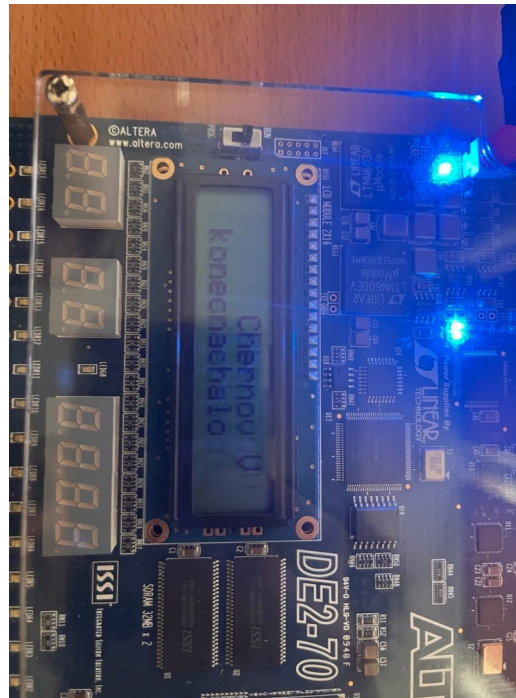
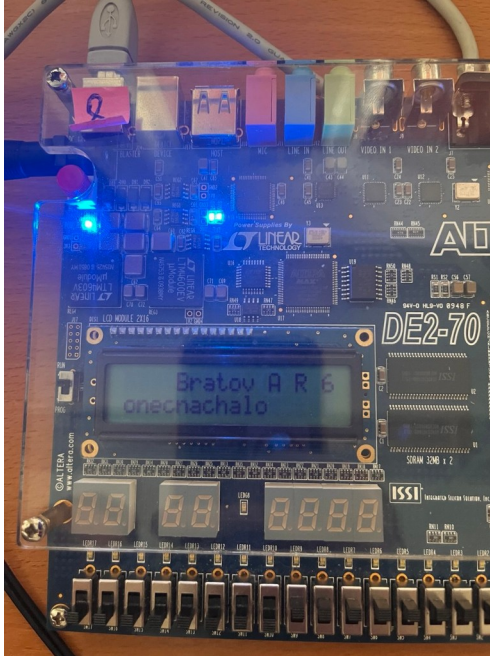
3. Экспериментируем с разными командами. Например, добавить символ и сдвинуть строку влево\вправо.



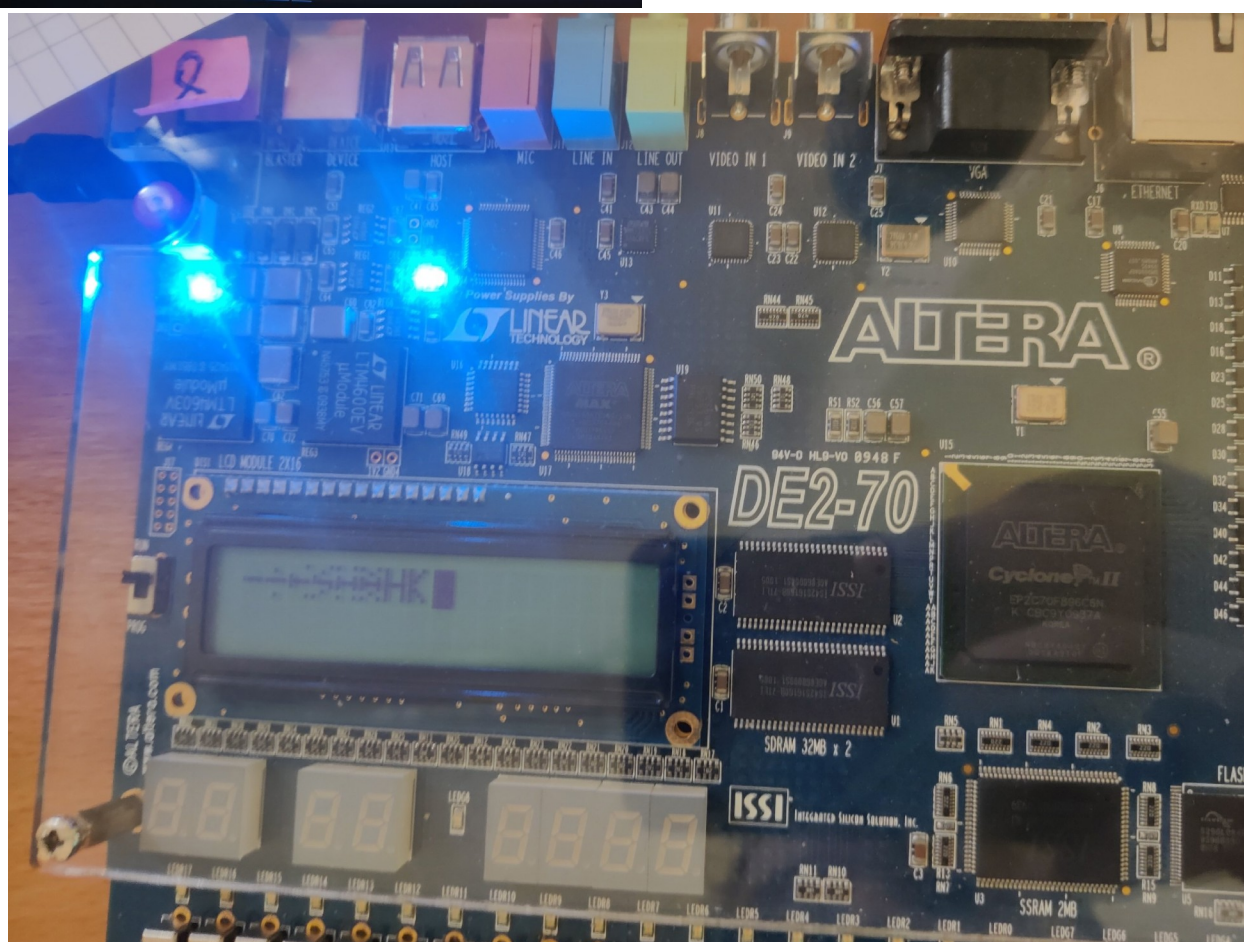
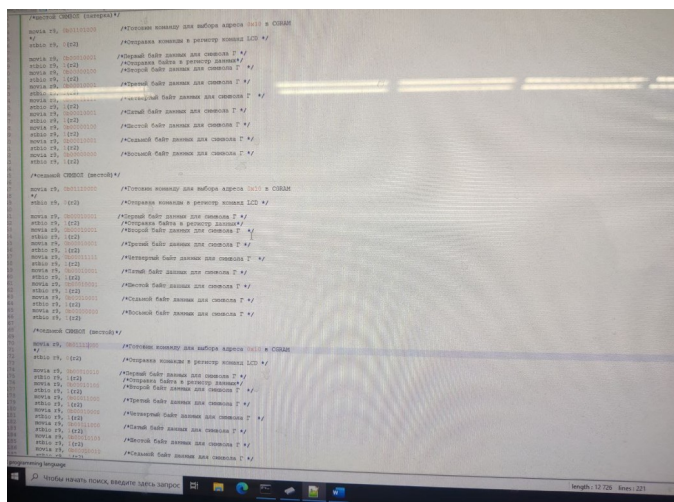
4. Делаем бегущую строку с буквами алфавита.



5. Делаем бегущую строку с нашими именами и фамилиями. В верхней строке «Ostapenkov Lev» и «Golik Andrey». В нижней строке «nachalo», буквы от f до a и цифры от 9 до 0 и «konets». Сама строка зациклена.



6. Делаем свои символы. Домино и буква «К».



Заключение

В ходе выполнения лабораторной работы №3, студенты обрели следующие навыки:

1. Использования символьной информации в своих программах;
2. Вывода символьной информации на индикаторы, типа LCD дисплея;