

8. Особенности работы с ЖКИ дисплеем NOKIA 5110

8.1. Введение

В настоящее время повсеместное применение получила цифровая техника и оборудование. От бытовых приборов, таких как электронные часы, цифровой термометр, ПК или сотовый телефон до промышленных систем управления производством и космических технологий. Во всех этих системах требуется обмен информацией с пользователем или оператором. Осуществляется этот обмен посредством устройств ввода и вывода информации.

В недалеком прошлом основным устройствами отображения информации на производстве являлись индикационные элементы, такие как светодиоды, а так же семисегментные индикаторы. Сегодня широкое распространение получили жидкокристаллические дисплеи. Небольшие размер и энергопотребление позволяют применять их в самых различных системах. С помощью ЖК-дисплея возможно организовать удобный интерфейс управления системой и обеспечить лучшее отображение различного вида графической информации, от текстовых и числовых данных до различных графиков и схем.

Зачастую в электронных устройствах применяются законченные модули ввода-вывода. Такие модули содержат блок индикации, реализованный на светодиодах или ЖКИ дисплея, блок ввода информации и управляющий микроконтроллер. В данной главе описывается создание и работа модуля ввода-вывода информации на базе отладочной платы STM32VLDISCOVERY и ЖКИ дисплея Nokia5110.

8.2. Дисплей Nokia5110

8.2.1. Основные характеристики

Для начала необходимо рассмотреть характеристики дисплея, на основе которого делается модуль ввода-вывода. Это монохромный ЖК-дисплей с разрешением экрана 84x48 точек. Управляется встроенным драйвером PCD8544. В дисплей так же встроены генераторы напряжения питания и напряжения смещения ЖК-элементов, светодиодная подсветка. Интерфейс ввода информации – SPI. Дисплей может работать в четырех режимах: нормальный, инверсия изображения, пустой экран, «все точки включены». Так же пользователю доступно управление температурным режимом, напряжением питания и смещения. Основные электрические параметры приведены ниже. Более подробную информацию можно получить из документации на дисплей [1].

- напряжение питания: 2,7 – 3,3В;
- ток: до 320мкА;
- частота тактирования: до 4МГц;

- время сброса: не менее 100нс;
- температурный диапазон: -25 – +70 °С.

8.2.2. Структура дисплея

Дисплей представляет собой матрицу ЖК-элементов и драйвер для их управления, размещенные в одном корпусе, который крепится к плате. На ней так же размещены четыре светодиода подсветки экрана. Внешний вид дисплея представлен на *Рис. 60*. Блок схема встроенного контроллера дисплея представлена *Рис. 61*.

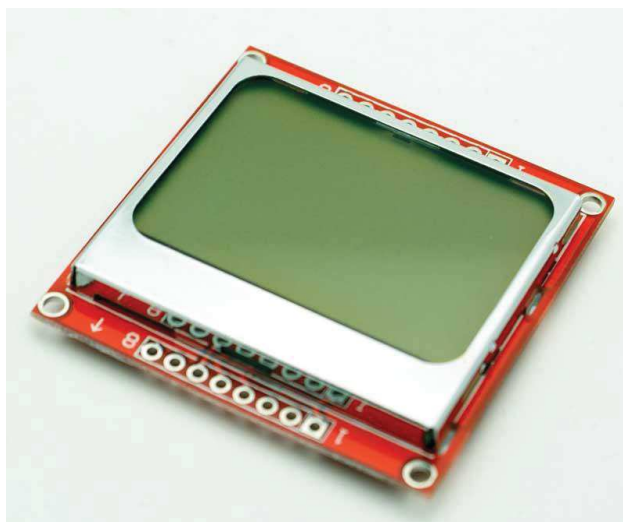


Рис. 60. Внешний вид дисплея NOKIA 5110

Как видно из блок-схемы, информация о состоянии точек дисплея хранится в оперативной памяти контроллера, каждой точке соответствует один бит памяти. Так же встроен счетчик адреса, который автоматически увеличивается при записи очередного байта информации в память. Подробнее об управлении дисплеем сказано в разделе «Команды управления».

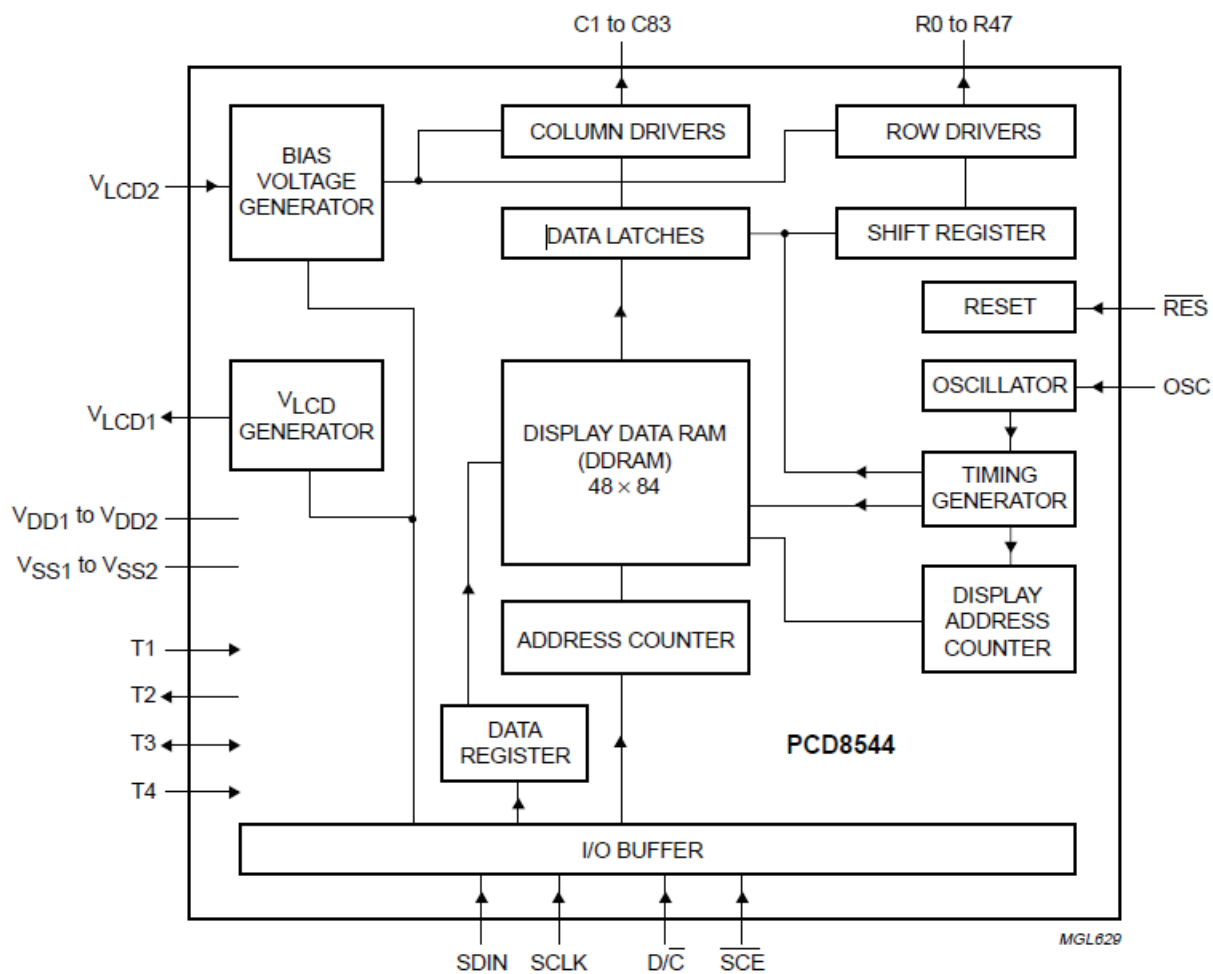


Рис. 61. Блок-схема дисплея

8.2.3. Особенности управления

Управление дисплеем осуществляется по интерфейсу SPI, дисплей является ведомым устройством. Однако, вместо обычных четырех линий управления в этом дисплее присутствуют лишь три. Это линия тактирования CLK, линия выбора кристалла SCE и линия входных данных MOSI. Линия выходных данных MISO отсутствует. Эта особенность приводит к необходимости применять специальные методы управления, подробнее об этом будет сказано ниже. Работает интерфейс в режимах SPI-0 или SPI-3. В дисплее присутствует также дополнительная линия управления Информация /Команда – D/C. Каждый байт, передаваемый в дисплей, может быть интерпретирован или как слово команды, или как информационный байт, в зависимости от уровня напряжения на линии D/C. Теперь рассмотрим особенности, связанные с управлением дисплеем.

Как уже было сказано выше, первая особенность состоит в том, что в SPI-интерфейсе отсутствует линия MISO. Это означает, что передача информации однонаправленная, и считать данные из памяти и регистров дисплея невозможно. Поэтому, в программе необходимо предусмотреть возможность контролировать состояния дисплея. Однако, есть еще одна

особенность, которая существенно усложняет управление. Эта особенность связано с организацией памяти дисплея. На *Рис. 62* приведена структура памяти. Память состоит из шести банков, в каждом из которых находится 84 ячейки емкостью один байт. Это означает, что запись информации в память осуществляется побайтно, а не побитно, и, следовательно, нет возможности управлять каждой точкой в отдельности, а лишь группами по восемь точек. Это в сочетании с тем, что информация из памяти дисплея не может быть считана, приводит к тому, что перед отправкой необходимо запоминать, какие данные в какой ячейке хранятся. В противном случае при отправке новых данных в дисплей можно потерять информацию. Эту особенность иллюстрирует *Рис. 63*. Поэтому, при написании управляющей программы необходимо предусмотреть возможность хранения данных.

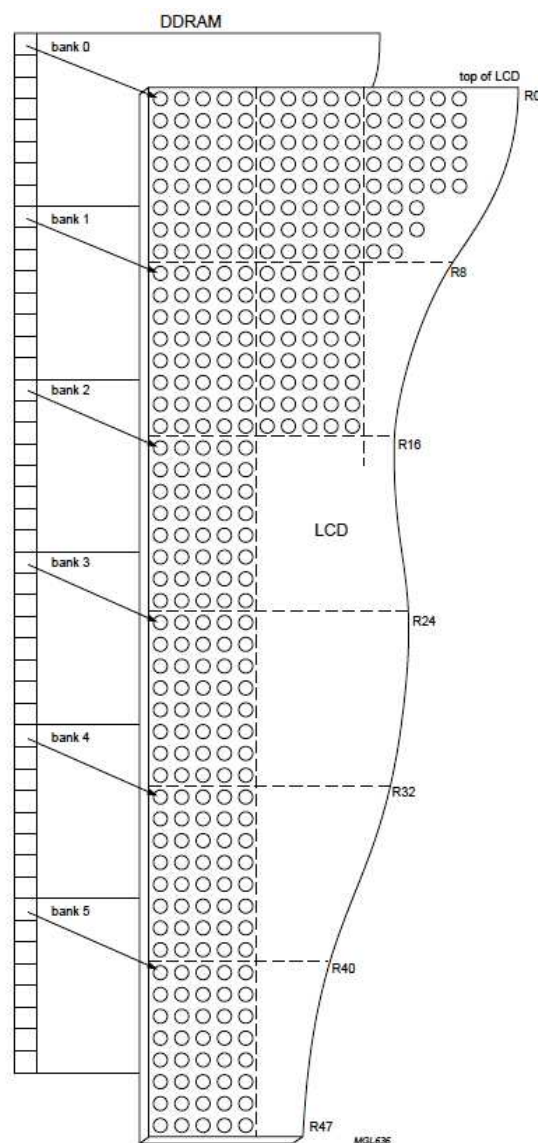


Рис. 62. Внешний вид дисплея NOKIA 5110

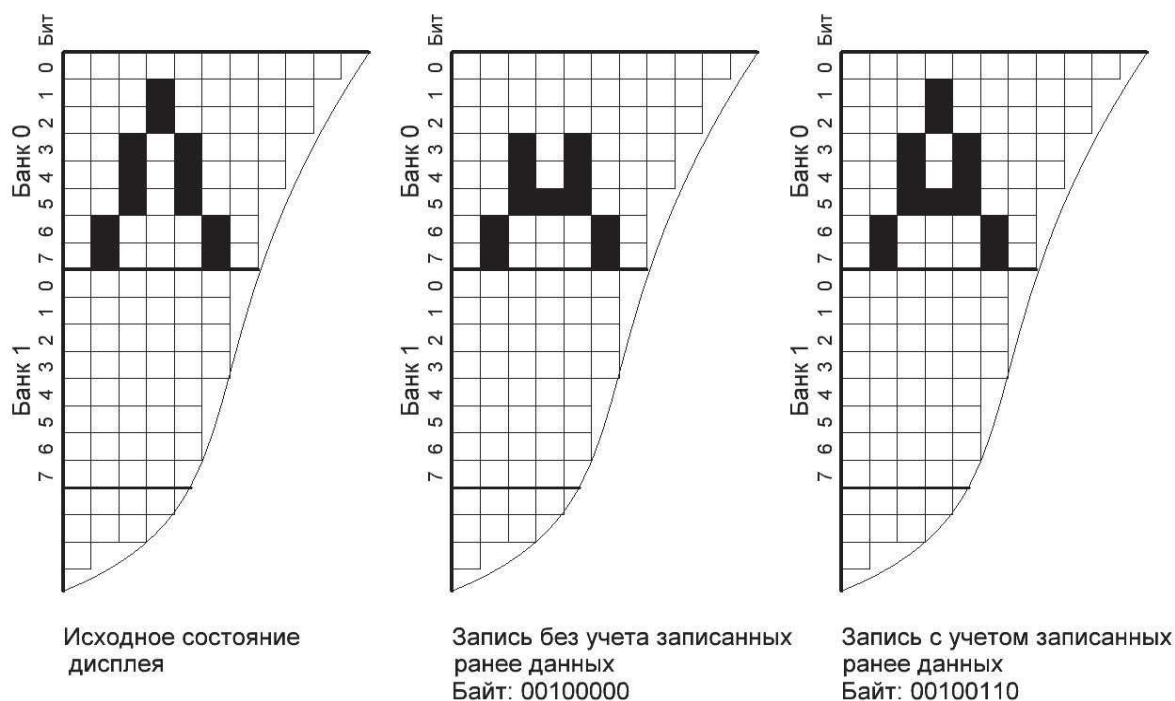


Рис. 63. Пояснение к работе дисплея. Замена символа «Л» на символ «А»

Например, можно использовать в качестве промежуточной памяти память управляющего контроллера, в которой будет храниться копия данных в дисплее. Перед отправкой необходимо корректировать данные, в зависимости от того, какая информация хранится в промежуточной памяти. Именно такой подход был реализован при написании библиотеки дисплея. Подробнее о ней сказано в разделе «Программная часть».

8.2.4. Команды управления

Управление дисплеем осуществляется путем отправки командного слова в дисплей. Размер слова – 1 байт. Все команды управления разделены на две категории – функции управления, основные функции и расширенные функции. Ниже приведен список функций дисплея и краткое описание каждой из них.

Функции управления

Установить тип функций

Устанавливает, с каким типом функций будет работать дисплей – основными или расширенными.

Установить режим питания

Включает или отключает питание дисплея.

Установить режим адресации

Определяет тип адресации памяти – вертикальный или горизонтальный. В первом случае после записи байта данных будет увеличен счетчик Y-адреса, то есть, запись будет идти по столбцам. Во втором – счетчик X-адреса, запись будет идти по строкам.

Основные функции

Установить режим отображения

Определяет режим отображения информации дисплеем – пустой экран, все точки экрана включены, нормальное отображение информации, инверсное отображение информации.

Установить X-адрес

Устанавливает X-адрес ячейки, куда будет записан следующий байт.

Установить Y-адрес

Устанавливает Y-адрес ячейки, куда будет записан следующий байт.

Расширенные функции

Установить температурный режим

Определяет температурный режим дисплея

Установить напряжение смещения

Определяет напряжение смещения ЖК-элементов дисплея.

Установить напряжения питания

Определяет напряжение питания ЖК-элементов дисплея.

Три функции управления определяются одним командным словом. Это слово необходимо отправить в дисплей в начале работы с ним. Формат слова следующий:

0 0 1 0 0 PD V H.

Здесь бит PD определяет режим питания, бит V – режим адресации, бит H – тип функций, с которыми будет идти дальнейшая работа – основные или расширенные. Как видно, необходимо запоминать текущее состояние дисплея, чтобы при установке нового значения какого-либо одного параметра не потерять информацию о значениях других.

Работа с основными и расширенными функциями проще, поскольку каждой из них соответствует отдельное командное слово. Необходимо только помнить, что для работы с определенным типом функций необходимо перевести дисплей в режим работы с этим типом функций. Иначе отправка слова команды приведет к некорректному выполнению этой команды. Более подробную информацию о командах управления можно получить в документации на дисплей со страницы 11 [1].

8.3. Аппаратная часть

8.3.1. Семейство отладочных плат STM32

Компания STMicroelectronics наряду с микроконтроллерами выпускает также и отладочные платы для них. На данный момент существуют платы для четырех семейств микроконтроллеров: STM32F1xx, STM32F2xx, STM32F3xx и STM32F4xx. Использование этих плат позволяет изучить возможности каждого семейства, выработать методику написания программ для той или иной периферии микроконтроллера. На платах помимо микроконтроллера установлены отладчик-программатор ST-Link, светодиоды, датчики, внешняя память и другие устройства.

8.3.2. STM32VLDISCOVERY

В устройстве используется отладочная плата для микроконтроллера STM32F100RBT6B. Загрузка программы осуществляется с ПК через USB-интерфейс. Непосредственно в контроллер программа загружается через ST-Link. Помимо контроллера плата содержит следующие элементы [2]:

- микросхема загрузки и отладки ST-Link;
- USB-разъем;
- выводы для внешнего источника питания 5В и 3,3В;
- кнопка сброса;
- пользовательская кнопка;
- пользовательские светодиоды;
- кварцевый резонатор на 8МГц

8.3.3. Общая схема модуля

Для подключения дисплея используются восемь выводов. Схема подключения и назначение выводов приведены ниже.

Согласно технической документации ток потребления дисплея не превышает 320 мкА, а максимально допустимый ток вывода микроконтроллера составляет 20 мА. Измерение величины тока светодиодной подсветки дают величину около 6 мА. Следовательно, все выводы дисплея можно напрямую подключить к выводам микроконтроллера. Вывод сброса следует «подтянуть» к земле. Схема подключения дисплея к плате приведена на *Рис. 64*.

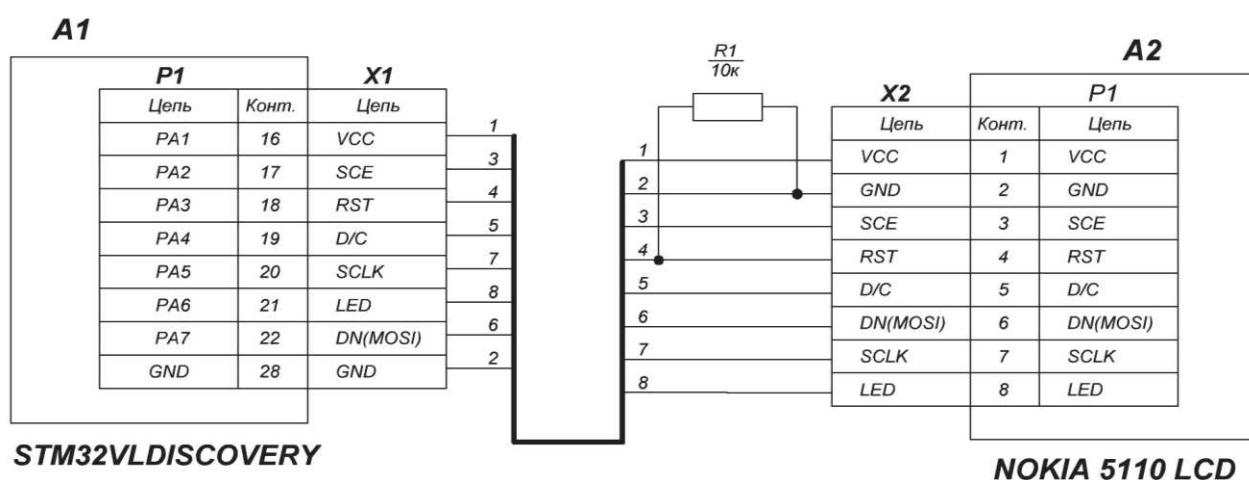


Рис. 64. Схема подключения дисплея к плате

8.4. Программная часть

8.4.1. Структура библиотеки

Для работы с дисплеем была написана библиотека на языке C++. В данной библиотеке определен класс дисплея. В качестве свойств класса использовались переменные, в которых хранятся данные о текущем состоянии дисплея, а так же массив, в котором хранится копия данных в ОЗУ дисплея. Необходимость такого подхода была обоснована в разделе Особенности управления дисплеем. Методы класса были разделены на две категории – вспомогательные и основные. Вспомогательные методы реализуют инициализацию дисплея, отправку данных по SPI и другие служебные функции. Основные же методы реализуют отображение графической информации трех типов: текстовая, растровая и векторная графика. Ниже более подробно описана реализация этих трех типов графики.

8.4.2. Векторная графика

На *Рис. 65* представлен пример отображения векторной графики. Реализован вывод четырех типов фигур: точки, линии, эллипса и прямоугольника. Вывод точки осуществляется достаточно просто. Сначала по заданному *y*-адресу определяется бит внутри байта, который соответствует точке на экране, затем вычисляется *x*- и *y*-адреса ячейки памяти, куда будет отправлен байт. Перед отправкой необходимо дополнить полученный байт битами, которые соответствуют точкам экрана из той же ячейки, чтобы не потерять информацию о них.

Линия выводится путем вычисления координат точек, из которых эта линия состоит, и вывода на экран этих точек. Однако необходимо учесть одно обстоятельство. Чтобы в линии не было разрывов, необходимо предварительно вычислить угол ее наклона. В случае если он меньше 45 градусов, необходимо задавать *x*-координаты и вычислять соответствующие им *y*-координаты точек. В противном случае все делается наоборот.

Прямоугольник формируется путем вывода четырех линий. Для эллипса использован следующий подход. Перед выводом точки ее координаты проверяются вхождение в область, которая описывается уравнением эллипса. Если координаты попадают в эту область, точка выводится, иначе – нет. В случае если эллипс или прямоугольник имеет не пустой фон, алгоритм этот несколько усложняется.

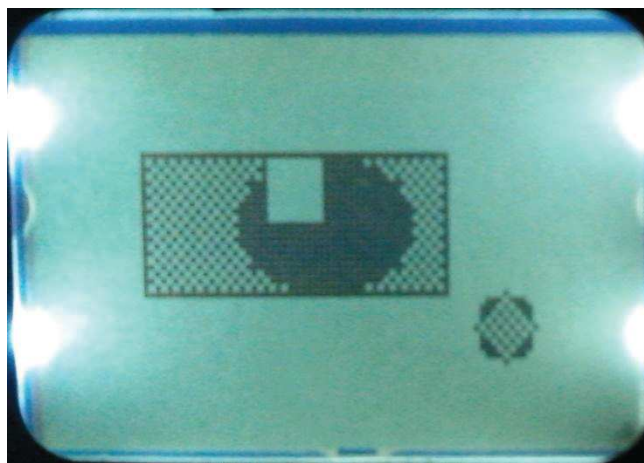


Рис. 65. Пример отображения векторной графики

8.4.3. Растровая графика

В режиме растровой графики реализован вывод изображения, размером 84x48 точек. Делается это путем побайтного заполнения памяти дисплея информацией, соответствующей точкам этого изображения. На Рис. 66 приведен пример отображения растровой графики.

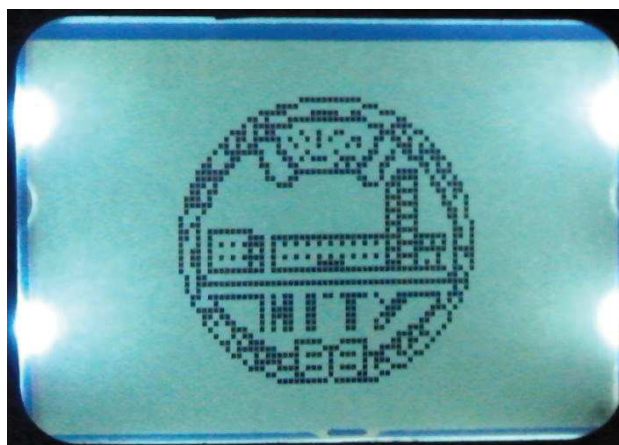


Рис. 66. Пример отображения растровой графики

8.4.4. Тестовая информация

Вывод текстовых символов реализован так же, как и вывод растрового изображения, с той лишь разницей, что размер его много меньше – всего 8 на 6 точек. Этот размер един для всех символов, что удобно. Единственная особенность возникает при отображении символов, где есть подстрочные элементы, такие как «Ц» или «Щ». Они по отношению к остальным как бы сдвинуты на одну точку вверх. Поэтому при выводе строки необходимо эти символы смещать на одну точку вниз. В библиотеке реализована функция вывода строки, этот сдвиг производит автоматически. На Рис. 67 приведен пример отображения текстовой информации.

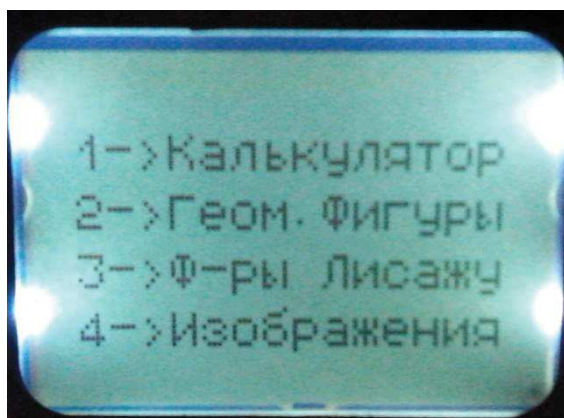


Рис. 67. Пример отображения тестовой информации

По результатам данной главы можно сделать следующие выводы:

1. ЖКИ является простым по подключению и управлению устройством. Он может применяться в различных системах как недорогое устройство отображения информации.
2. Разработана библиотека, реализующая отображение на ЖКИ различной информации – текстовой, а так же растровых и векторных изображений.
3. С помощью отладочной платы STM32VLDISCOVERY можно создать макет устройства на основе контроллера STM32 в небольшие сроки. Макет позволяет отработать алгоритм управления, реализацию функциональных возможностей будущего устройства перед созданием его окончательной версии.

8.5. Контрольные вопросы

1. Какое разрешение в точках у дисплея?
2. Какой интерфейс используется для связи?
3. Есть ли возможность прочитать данные с дисплея?

8.6. Литература к разделу 8

[1] Philips Semiconductors PCD8544 48 x 84 pixels matrix LCD controller/driver [Электронный ресурс]. – Режим доступа : <https://www.sparkfun.com/datasheets/LCD/Monochrome/Nokia5110.pdf>

[2] Официальный STMicroelectronics STM32VLDISCOVERY STM32 value line Discovery [Электронный ресурс]. – Режим доступа : http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/CD00267113.pdf