

Управление дисплеем MT-12864J средствами микроконтроллера K1986BE92QI

Андрей Шаронов, Валерий Володин (г. Пермь)

Графический дисплей MT-12864J компании МЭЛТ входит в комплект поставки отладочной платы микроконтроллера K1986BE92QI компании «Миландр». Поэтому вопрос управления им интересует многих пользователей отладочного комплекта. Настоящая статья посвящена разработке дисплейного драйвера, совместимого с API драйверов графических дисплеев компании Keil и описываемого заголовочным файлом GLCD.h.

До недавнего времени авторы этой статьи в своих разработках не использовали графический дисплей MT-12864J, полагая его возможности избыточными и с успехом применяя вместо него символьные дисплеи MT-16S2D и MT-20S4M. Наличие вариантов с напряжением питания 3 В и Windows-совместимая встроенная кодовая страница на базе CP1251 делали дисплеи MT-16S2D и MT-20S4M удобным решением для большинства приборов.

Однако со временем возникла необходимость формировать не только текстовые меню, но и графические изображения. Поэтому в перспективное изделие был заложен дисплей MT-12864J.

Для знакомства с возможностями дисплея для отладочной платы компании «Миландр» был разработан драйвер дисплея и ряд демонстрационных проектов. В процессе разработки драйвера и демонстрационных программ были выявлены возможности дисплея, которые могут быть полезны и для прибора с полностью текстовым меню:

- возможность вывода на дисплей изображения, хранящегося в формате битового поля (например, логотипа компании);
- возможность инверсии текстовой строки (светлый текст на тёмном фоне), что может быть использовано для выделения активного пункта меню (в дисплеях MT16S2D

и MT20S4M для этих целей используется символ указателя);

- большая вместимость дисплея: при использовании шрифта с размерами знака 8 × 5 пикселей на дисплей возможно вывести 8 строк по 25 символов, а при использовании шрифта 8 × 6 пикселей возможно вывести 8 строк по 21 символу.

К сожалению, более широкие возможности дисплея ведут к усложнению программного драйвера, и задача его разработки не выглядит столь тривиальной, как для символьных дисплеев.

ОБЩИЕ СВЕДЕНИЯ О ДИСПЛЕЕ MT-12864J

Согласно спецификации на сайте производителя [1], основу дисплея MT-12864J составляют два микроконтроллера K145BG10 производства компании «Ангстрем». Каждый из них отвечает только за часть изображения. То есть для вывода изображения на левую половину дисплея (64 × 64 пикселя) необходимо выбрать один контроллер, а для вывода изображения на правую – другой. Выбор соответствующего контроллера осуществляется установкой сигнала логического нуля на выводе E1 или E2.

Для программиста каждый из контроллеров представляет собой подобие статического ОЗУ, разделённого на 8 страниц по 64 байта каждая. Каждый байт может быть записан либо считан с помощью определённой последовательности команд.

Обмен микроконтроллера с дисплеем производится по восьмиразрядной параллельной шине, дополненной рядом управляющих сигналов. Назначение сигналов указано в таблице 1. Там же указаны и порты микроконтроллера K1986BE92QI (маркировка MDR32F9Q2I), к которым подключены соответствующие выводы дисплея.

Система команд одинакова для обоих контроллеров K145BG10 и представлена в таблице 2.

Программисту доступны следующие команды:

- включение и выключение дисплея;
- установка верхней строки дисплея;

Таблица 1. Назначение выводов дисплея MT-12864J

Обозначение вывода	Назначение вывода	Номер вывода дисплея	Порт микроконтроллера K1986BE92QI на отладочной плате (Rev. 2)
U _{cc}	Напряжение питания дисплейного модуля (цифровая часть)	1	–
GND	Общий вывод	2	–
U _o	Напряжение питания ЖК-панели	3	–
D0	Младший бит шины данных	4	PA0
D1	Первый бит шины данных	5	PA1
D2	Второй бит шины данных	6	PA2
D3	Третий бит шины данных	7	PA3
D4	Четвёртый бит шины данных	8	PA4
D5	Пятый бит шины данных	9	PA5
D6	Шестой бит шины данных	10	PF2
D7	Старший бит шины данных	11	PF3
E1	Выбор первого контроллера K145BG10 (левая сторона дисплея)	12	PB7
E2	Выбор второго контроллера K145BG10 (правая сторона дисплея)	13	PB8
RES	Сброс дисплейного модуля	14	PB9
R/W	Выбор режима: чтение или запись	15	PB10
A0	Выбор режима: команды или данные	16	PC0
E	Стробирование данных	17	PC1
U _{ss}	Выход DC/DC-преобразователя	18	–
A	«Плюс» питания подсветки	19	–
K	«Минус» питания подсветки	20	–

- установка активной страницы ОЗУ контроллера дисплея;
- установка активного байта ОЗУ контроллера дисплея.

В режиме данных доступны функции чтения и записи. После чтения данных из активного байта активной страницы ОЗУ контроллера дисплея происходит инкремент адреса байта. Поэтому есть возможность последовательной записи нескольких байтов в дисплей, что может ускорить определённые процедуры чтения и записи.

Как и символьные дисплеи MT-16S2 и MT-20S4, дисплей MT-12864J имеет регистр статуса, который считывается в режиме команд. Назначения битов регистра показаны в таблице 3.

Дисплей MT-12864J не имеет встроенного знакогенератора, поэтому обеспечение дисплея шрифтами – задача программиста. Это, с одной стороны, усложняет управляющую программу, но с другой – позволяет создавать собственные шрифты, не ограничиваясь ни размерами, ни начертанием, ни кодовой таблицей (в символьных дисплеях компании МЭЛТ присутствуют только две кодовые таблицы: совместимая с кодовой таблицей дисплеев на контроллере HD44700 и альтернативная, совместимая с CP1251). Кроме того, существует возможность инверсного выделения текста.

API ДРАЙВЕРА ГРАФИЧЕСКОГО ДИСПЛЕЯ ОТ KEIL

Для подключения драйверов графических дисплеев компанией Keil разработан специальный программный интерфейс. Прототипы функций драйвера дисплея определены в заголовочном файле GLCD.h и представлены в таблице 4.

Данный файл можно найти, например, в каталоге типа *C:\Каталог установки Keil\ARM\Boards\Keil\MCBSTM32C\LCD_Blinky* (информация справедлива для версий Keil MDK 4.x. В Keil MDK 5 этот файл, возможно, содержится в пакете поддержки микроконтроллеров STM32 или AT91SAM3x). Изначально, как можно увидеть из примеров программ (листингов), данный API предполагался для управления цветными графическими дисплеями, что следует и из наличия кодов, определяющих различные цвета. Однако для монохромных дисплеев также возможно создание драйверов, содержащих практически все функции, определённые данным API и заголовочным фай-

Таблица 2. Система команд контроллера K145BF10

Команда	Код команды								Описание	
	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Display ON/OFF Control	0	0	1	1	1	1	1	LCD On/Off	Включение и выключение дисплея – определяется битом DB0: 0 – дисплей выключен, 1 – дисплей включён	
Display Start Line	1	1	Display Start Line (0...63)							Определяет строку ОЗУ, которая будет отображаться в верхней строке дисплея
Set Page	1	0	1	1	1	Page (0...7)			Установка активной страницы	
Set Address	0	1	Address (0...63)							Установка активной ячейки памяти выбранной страницы

Таблица 3. Структура регистра статуса дисплея

№ бита	Название	Значение по умолчанию	Описание
0	–	0	Зарезервировано
1	–	0	Зарезервировано
2	–	0	Зарезервировано
3	–	0	Зарезервировано
4	Reset	0	Сброс дисплея: 1 – состояние сброса, 0 – нормальное состояние
5	On/Off	0	Включение и выключение дисплея: 1 – дисплей включён, 0 – дисплей выключен
6	–	0	Зарезервировано
7	Busy	0	Флаг занятости дисплея: 1 – дисплей занят, 0 – дисплей готов к работе

Таблица 4. Функции драйвера графического дисплея, определённые в файле GLCD.h

Функция	Описание
void GLCD_init (void)	Инициализация графического дисплея
void GLCD_WindowsMax (void)	Установка максимального рабочего пространства дисплея
void GLCD_PutPixel (unsigned int x, unsigned int y)	Закраска пикселя с координатами x и y в цвет, определённый, как цвет текста
void GLCD_SetTextColor (unsigned short color)	Установка цвета, переданного в параметре color, как цвета выводимого на дисплей текста
void GLCD_SetBackColor (unsigned short color)	Установка цвета, переданного в параметре color, как цвета фона
void GLCD_Clear (unsigned short color)	Очистка дисплея и заливка его цветом, определённым в параметре color
void GLCD_DrawChar (unsigned int x, unsigned int y, unsigned int cw, unsigned int ch, unsigned char *c)	Вывод на дисплей символа высотой ch и шириной cw в район точки, определённой координатами x и y (верхний левый край символа). Параметр *c является указателем на массив, определяющий начертание символа
void GLCD_DisplayChar (unsigned int ln, unsigned int col, unsigned char fi, unsigned char c)	Вывод на дисплей символа в строке ln и знакоместе col. Параметр fi определяет шрифт, параметр c – код выводимого символа
void GLCD_DisplayString (unsigned int ln, unsigned int col, unsigned char fi, unsigned char s)	Вывод на дисплей строки. Параметр ln определяет строку, параметр col – первое знакоместо, параметр fi – шрифт, параметр s – указатель на начало строки
void GLCD_ClearLn (unsigned int ln, unsigned char fi)	Очистка строки ln и покраска её цветом, определённым как цвет фона. Параметр fi определяет кегль очищаемой строки
void GLCD_Bargraph (unsigned int x, unsigned int y, unsigned int w, unsigned int h, unsigned int val)	Вывод на дисплей прямоугольника с началом в точке, определяемой параметрами x и y, с длиной w и высотой h. Параметр val определяет, какая часть прямоугольника будет закрашена в цвет текста, остальная часть красится в цвет фона
void GLCD_Bitmap (unsigned int x, unsigned int y, unsigned int w, unsigned int h, unsigned char bitmap)	Вывод на дисплей рисунка из битового поля. Параметры x и y – координаты начальной точки, w и h – длина и высота выводимого рисунка, bitmap – указатель на начало массива битового поля
void GLCD_ScrollVertical (unsigned int dy)	Вертикальный сдвиг выведенной на дисплей информации на промежуток dy (в пикселях)
void GLCD_WrCmd (unsigned char cmd)	Передача контроллеру дисплея команды cmd
void GLCD_WrReg (unsigned char reg, unsigned short val)	Запись в регистр контроллера reg значения val

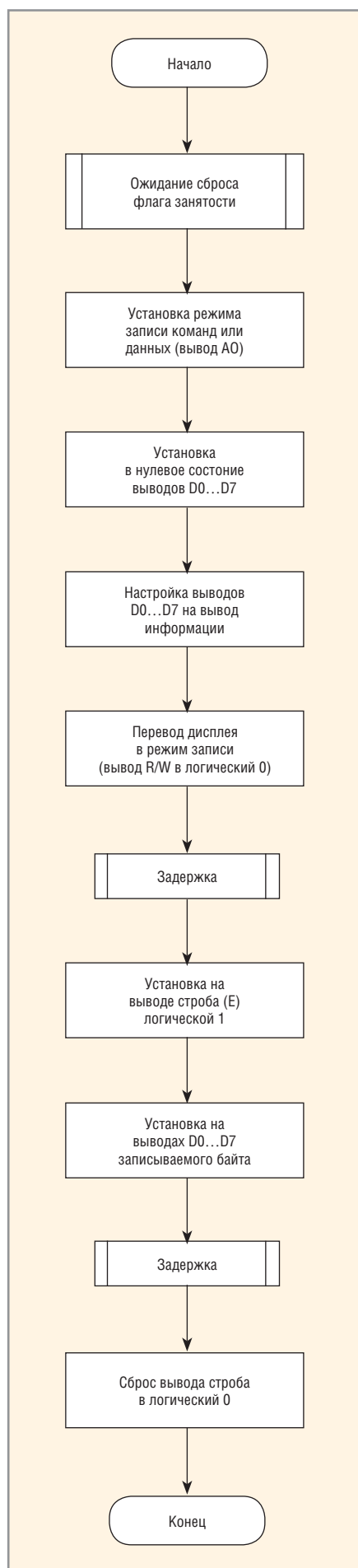


Рис. 1. Блок-схема алгоритма записи байта в контроллер дисплея

лом. Название файла драйвера дисплея должно начинаться с GLCD. Например, GLCD_SPI_STM32.c. Текст драйвера дисплея MT-12864J находится в файле GLCD_MT12864.c.

Разработанный драйвер имеет ряд ограничений и особенностей, отличающих его от прототипа:

- все цвета фона или текста, отличные от «белого», отображаются на дисплее «чёрным» цветом;
- драйвер поддерживает работу только в горизонтальном режиме;
- в параметрах функции рисования прямоугольника переменная `val` отражает закраску в процентах, а не в долях $1/1024$, как в прототипе;
- функция вывода рисунка предполагает вывод чёрно-белых рисунков, имеющих 256 градаций серого;
- невозможна работа со стандартными шрифтами, прилагающимися к драйверу графического дисплея от Keil (организация памяти дисплея MT-12864J отличается от графических дисплеев, использованных в отладочных платах Keil и Atmel, поэтому более эффективно использовать специально адаптированные шрифты);
- отсутствуют команды установки размера окна и записи регистра.

Дисплей MT-12864J получил широкое распространение среди радиолюбителей и профессионалов. Для различных микроконтроллеров были сделаны собственные драйверы данного дисплея, на которые авторы данной статьи опирались в процессе разработки. В частности, драйвер для близкого по архитектуре дисплея MT-12864A, описанный в статье «Подключаем MT-12864-2YLG» [2].

ОСНОВНЫЕ ФУНКЦИИ ВЗАИМОДЕЙСТВИЯ С ДИСПЛЕЕМ

Основу драйвера дисплея составляют следующие функции:

- начальная инициализация;
- передача дисплею команд и данных;
- чтение регистра состояния дисплея;
- чтение данных из памяти дисплея.

Данные функции позволяют реализовать обмен данными с дисплеем для вывода на него различной информации – как текстов, так и графических изображений, что доступно для разработчика через функции драйвера.

В спецификации на дисплей [1] начальная настройка описывается как установка на входе RES состояния логического нуля, удержание его в течение 1 мкс и ожидание сброса флагов BUSY и RESET в регистре состояния дисплея.

Однако драйвер, представленный автором статьи «Подключаем MT-12864-2YLG» [2], предполагает более сложную процедуру инициализации:

Установка логического нуля на выводе RES с последующим его сбросом через 100 мс.

1. Выбор первого контроллера дисплея (K145BF10) установкой логической единицы на входе E1.
2. Отправка последовательности команд: включение дисплея, установка нулевой ячейки текущей страницы ОЗУ в качестве активной, установка нулевой строки, как самой верхней (команды `0x3F`, `0x40`, `0xC0` соответственно).
3. Снятие сигнала выбора с первого контроллера и выбор второго контроллера K145BF10 (установка логической единицы на входе E2 и логического нуля на входе E1).
4. Повторение последовательности команд.

С программной реализацией данной функции можно ознакомиться в файле GLCD_MT12864.c любого из проектов, выложенных в дополнительных материалах к статье на сайте www.soel.ru (функция `GLCD_init()`).

В самом начале функции производится настройка всех портов, на выводах которых формируются сигналы управления дисплеем (см. табл. 1). В дальнейшем все управляющие сигналы устанавливаются в состояние логического нуля, в том числе и сигнал сброса. Через 100 мс сигнал сброса восстанавливается в состояние логической единицы. После этого установкой логической единицы на выводе E1 выбирается первый контроллер K145BF10 дисплея, отвечающий за левую половину экрана. Контроллеру передаются команды включения дисплея, установки активной ячейки памяти и первой отображаемой строки.

После передачи команд сигнал выбора контроллера снимается и выбирается второй контроллер, отвечающий за правую половину дисплея. Функция ожидает снятия сигнала занятости контроллера и снова передаёт те же команды.

Временные диаграммы сигналов на выводах дисплея показаны в спецификации [1]. Гораздо интереснее посмотреть, как это выглядит с точки зрения программиста. Блок-схема алгоритма записи байта в контроллер показана на рисунке 1, программная реализация содержится в функции `LCD_write()`

файла GLCD_MT12864.c из дополнительных материалов к статье.

Как видно из листинга, в функции отсутствуют процедуры ожидания сброса сигнала занятости и перевода дисплея в режим команд или режим данных. Эти операции вынесены из функции и реализуются в отдельных функциях: передачи команд и передачи данных, описанных в листингах 1 и 2.

Как видно из листингов и блок-схемы алгоритма, ожидание сброса сигнала занятости вынесено в отдельную функцию. Блок-схема алгоритма ожидания сброса сигнала занятости показана на рисунке 2, а программная реализация данного алгоритма представлена в функциях файла GLCD_MT12864.c – LCD_read_status() и LCD_wait_busy() соответственно.

Как и в случае записи, считывание байта и ожидание сброса флага реализованы в разных функциях. Аналогично чтению регистра статуса реализована функция чтения данных из дисплея. В этот раз авторы не стали делать одну функцию чтения байта. Единственное отличие функции чтения данных (LCD_read_data()) в том, что дисплей переводится в режим данных сигналом на выводе A0.

Ещё одна функция – очистка дисплея, которая заливает весь дисплей цветом, указанным, как цвет фона. Фактически, всё сводится к записи в каждую ячейку памяти значения 0x00 или 0xFF. С программной реализацией данной функции также можно ознакомиться в файле GLCD_MT12864.c – функция GLCD_Clear().

ФУНКЦИИ РАБОТЫ С ТЕКСТОМ

Основной задачей дисплея является вывод текстовой информации. Поэтому следующими по важности после функций инициализации и обмена данными с контроллером являются функции вывода на экран текстовой информации. Так как дисплей не имеет собственного знакогенератора, то актуальным является вопрос выбора шрифтов, которые будут использоваться для вывода текстов.

В составе примеров Keil поставляются два шрифта: 6×8 и 12×16 пикселей. Однако структура данных шрифтов для дисплея MT-12864J не является оптимальной. На рисунке 3 показана структура символа «1» для шрифта 8×6 , предлагаемого компанией Keil (файл Font_6x8h.h). Как видно из рисунка, символ представляет собой массив значений, каждое из которых описывает расстановку пикселей в одной из восьми строк.

Для дисплея MT-12864J более подходящим является шрифт, структура которого показана на рисунке 4. Здесь значения массива описывают расстановку пикселей в столбцах. Такой структуре соответствует шрифт в проекте «Подключаем MT-12864-2YLG» [2] (размер символа 5×8 пикселей), однако таблица шрифта несовместима с CP1251, что ведёт к усложнению функции вывода символов на дисплей.

В обсуждении на форуме разработчиков электроники [3] был выложен проект с комплектом из четырёх шрифтов, совместимых с кодовой таблицей CP1251 [4]. Каждый шрифт находится

Листинг 1

```
static void LCD_write_cmd (unsigned char cmd)
{
    LCD_wait_busy(); //Ожидание сброса сигнала
    //занятости дисплея
    MDR_PORTC->RXTX &= ~A0; //Перевод дисплея в режим команд
    LCD_write(cmd); //Запись байта в дисплей
}
```

Листинг 2

```
static void LCD_write_data (unsigned char data)
{
    LCD_wait_busy(); //Ожидание сброса сигнала
    //занятости дисплея
    MDR_PORTC->RXTX |= A0; //Перевод дисплея в режим данных
    LCD_write(data); //Запись байта в дисплей
}
```

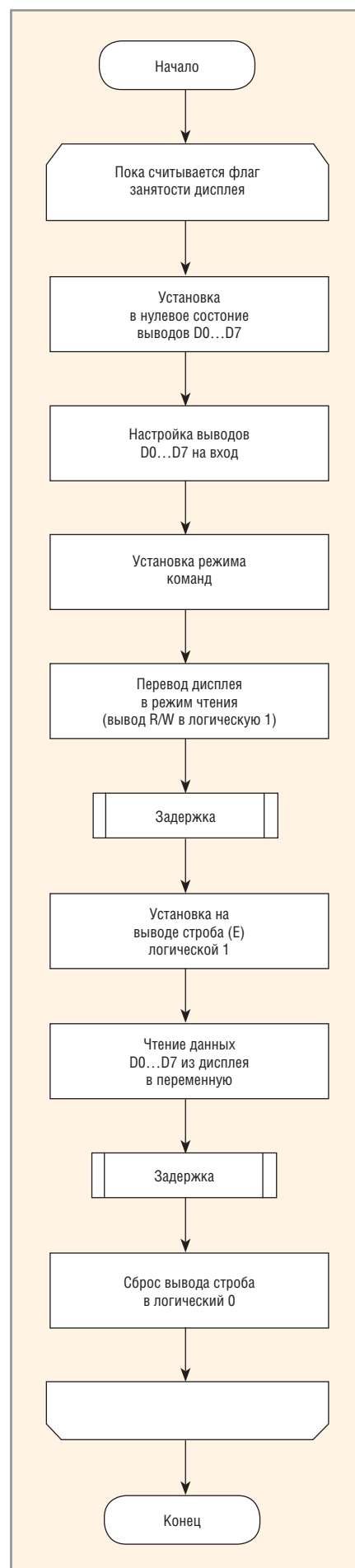


Рис. 2. Блок-схема алгоритма ожидания сброса сигнала занятости

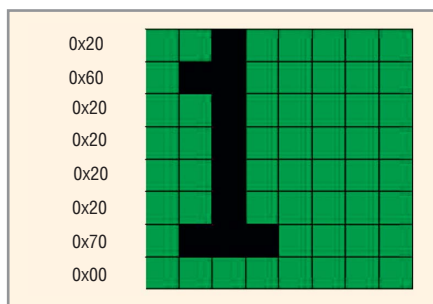


Рис. 3. Структура символа шрифта из файла Font_6x8h.h, входящего в примеры Keil

в отдельном заголовочном файле. Пользователю доступны шрифты размером символа 6 × 8, 12 × 16 пикселей, а также два шрифта размером 7 × 10 пикселей: нормальный и полужирный.

Для использования данных шрифтов вместе с программой в файлы были внесены определённые изменения:

- удалена структура FONT, описывающая параметры шрифта;
- удалено подключение заголовочного файла display.h;
- тип массива шрифтов изменён с `_flash BYTE` на `const unsigned char`.

После внесения изменений заголовочные файлы с таблицами знакогенератора были подключены к проекту. Непосредственный вывод символа на дисплей осуществляется функцией `GLCD_DrawChar()`, среди параметров которой передаются координаты начальной точки символа, ширина и высота символа, а также указатель на массив, описывающий начертание символа в таблице знакогенератора (см. табл. 4). В качестве прототипа была взята функция вывода символов на дисплей из проекта «Подключаем MT-12864-2YLG» [2].

Данная функция позволяет выводить на дисплей символы с произвольным расположением начальной точки, поэтому предусмотрена возможность вывода символа, не впи-

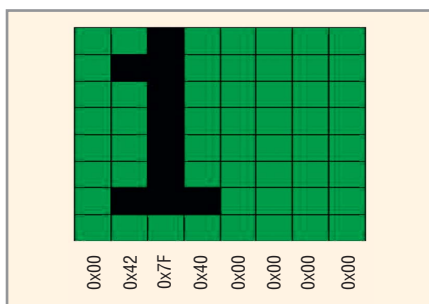


Рис. 4. Структура символа шрифта, рекомендуемого для дисплеев типа MT-12864

сывающегося в одну страницу. Для символов размером 7 × 10 пикселей предусмотрена также возможность вывода символа, не вписывающегося в две страницы. Также драйвер предусматривает функции вывода символа с указанием строки и знакоместа и вывода символьной строки: `GLCD_DisplayChar()` и `GLCD_DisplayString()` соответственно. В качестве параметра `fi` передаётся тип шрифта, используемого при выводе символа. В прототипе данный параметр может принимать значение «0» для шрифта 6 × 8 пикселей и «1» для шрифта 12 × 16 пикселей. В разработанном драйвере значение «0» соответствует шрифту 6 × 8 пикселей, значение «1» соответствует нормальному шрифту 7 × 10 пикселей, значение «2» соответствует полужирному шрифту 7 × 10 пикселей, а значение «3» – шрифту 12 × 16 пикселей. На рисунке 5 показаны все четыре шрифта на экране дисплея.

Также для работы с текстовой информацией предназначена функция очистки строки. В примерах Keil данная задача решается выводом на дисплей строки с пробелами. Так как разработанный драйвер использует шрифты с шириной, не кратной 8 (что не позволяет заполнить всю строку целым количеством пробелов), был применён метод, сходный с очисткой дисплея: заполне-



Рис. 5. Доступные шрифты на экране дисплея

ние всех ячеек памяти выбранной страницы памяти значениями 0x00 или 0xFF в зависимости от установленного цвета фона. Для шрифта 7 × 10 пикселей была предусмотрена ситуация, когда очищаемая строка занимает страницу только частично. Фрагмент кода, иллюстрирующий очистку строки высотой 8 пикселей, приведён в листинге 3.

ГРАФИЧЕСКИЕ ФУНКЦИИ

Практически все графические возможности драйвера базируются на функции вывода на экран пикселя с заданными координатами. Данная функция также взята из проекта «Подключаем MT-12864-2YLG» [2] и сводится к следующей последовательности операций:

1. Выбор страницы и байта, где будет находиться значение, описывающее положение пикселя на экране.
2. Считывание байта.
3. Модификация байта.
4. Запись изменённого значения обратно в ячейку памяти.

Программно данная процедура реализуется функцией `GLCD_PutPixel()` драйвера `GLCD_MT12864.c`.

Данная функция используется для вывода на экран прямоугольника в функции `GLCD_Vargraph()`. Сначала в цикле производится вывод на экран границ прямоугольника, а после – заливка цветом текста части, определённой параметром `val` в процентах, и остальной части – цветом фона. С программной реализацией данной функции также можно ознакомиться в тексте драйвера `GLCD_MT12864.c`.

Наиболее интересной является функция вывода битового поля `GLCD_Bitmap()`. Данная функция позволяет вывести на дисплей рисунок, описанный в виде массива значений. Для получения такого массива значений в составе среды Keil uVision присутствует программа `Bitmap converter for`

Листинг 3

```

LCD_write_cmd(0xC0); //Установка начальной строки
//Установка страницы памяти, соответствующей стираемой строке
LCD_write_cmd(0xB8 | page);
//Стирание строки путем записи значений, соответствующих цвету фона
for (i=0; i<64; i++)
{
    if (Color[BG_COLOR]==White)
        LCD_write_data(0x00);
    else
        LCD_write_data(0xFF);
}

```

emWin (C:\Каталог установки Keil\ARM\Segger\emWin\Tool\BmpCvt.exe).

Данная программа позволяет конвертировать любое графическое изображение в различные цветовые форматы, а также сохранять его либо в формате BMP, либо в виде файла с расширением *.c, содержащего массив, описывающий изображение в виде битового поля.

На взгляд авторов, формат лучше всего подходит для преобразования изображений типа «256 градаций серого». В этом случае каждый пиксель изображения описывается одним байтом массива, что облегчает работу с ним.

Для преобразования необходимо открыть исходный графический файл командой File->Open. В окне программы откроется графический файл, а также будут выведены его параметры: разрешение, цветовой формат и коэффициент увеличения программой – zoom (см. рис. 6).

Преобразование цветовой гаммы осуществляется в меню, выпадающем по команде Image->Convert Into (см. рис. 7).

После преобразования в режим «256 градаций серого» (Gray256) изображение выглядит так, как это показано на рисунке 8.

Сохранить преобразованное изображение можно командой File->Save As. В выпадающем меню можно выбрать тип файла. В данном случае это будет «C» bitmap file. После нажатия на кнопку сохранения появится диалоговое окно выбора формата, в котором будет сохранён массив, описывающий изображение. Здесь можно выбрать другой формат, не соответствующий тому, в который было преобразовано изображение (например, можно преобразовать исходное изображение в формат с четырьмя градациями серого, но сохранить в формате «256 градаций серого»). После выбора формата и нажатия кнопки «ОК» файл будет сохранён.

Сохранённый файл содержит следующие элементы:

- палитру – массив цветов static GUI_CONST_STORAGE GUI_COLOR;
- структуру, описывающую параметры массива палитры static GUI_CONST_STORAGE GUI_LOGPALETTE;
- массив битового поля static GUI_CONST_STORAGE unsigned char;
- структуру, описывающую битовое поле GUI_CONST_STORAGE GUI_BITMAP.

Для демонстрационного проекта будет достаточно массива битового

поля. Однако структура, описывающая битовое поле, имеет ряд полезных параметров, которые желательно также сохранить в тексте программы.

Авторы поместили массив битового поля в заголовочный файл, изменив его тип на static unsigned char (возможно также указать тип const unsigned char). Размер изображения в пикселях функция вывода на дисплей битового поля получает в качестве параметров (см. табл. 4). В каждом байте массива находится значение, определяющее степень градации серого у описываемого пикселя. Значение 0x00 соответствует «чёрному» пикселю, а значение 0xFF – «белому». Остальные значения описывают различные степени градации серого. Так как дисплей MT-12864J может воспроизводить только «белые» и «чёрные» пиксели, то в функции производится сравнение значения элемента массива с числом 0x7F, на основании результатов которого определяется цвет пикселя. Если число в массиве больше 0x7F, то цвет пикселя «белый». В противном случае пиксель закрашивается «чёрным» цветом. Фрагмент функции вывода на дисплей битового поля показан в листинге 4.

Образец выведенного на дисплей преобразованного графического изображения показан на рисунке 9. В определённых случаях оказывалось, что с «границей белого», равной 0x7F, качество изображения страдало (см. рис. 10), а само изображение получалось более контрастным в режиме четырёх градаций серого (см. рис. 11).

Последней функцией драйвера, используемой для работы с графическими, а также текстовыми объектами

Листинг 4

```
for (i=0; i<lim_Y; i++)
{
    for (j=0; j<lim_X; j++)
    {
        if (bitmap[i*w+j]>0x7F)
            //if (bitmap[i*w+j]==0x03)
            {
                Color[TXT_COLOR]=White;
            }
        else
        {
            Color[TXT_COLOR]=Black;
        }
        GLCD_PutPixel(x+j, y+i);
    }
}
```



Рис. 6. Изображение, открытое в программе Bitmap converter for emWin

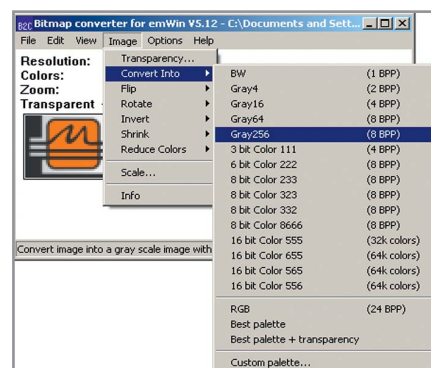


Рис. 7. Меню преобразования цветовой гаммы изображения

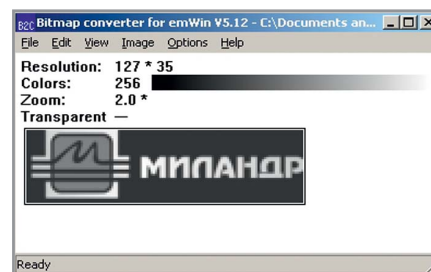


Рис. 8. Изображение после преобразования

ми, является функция вертикального скроллинга. В настоящем драйвере она реализована методом копирования одной области экрана в другую с помощью массива. Блок-схема алгоритма приведена на рисунке 12.

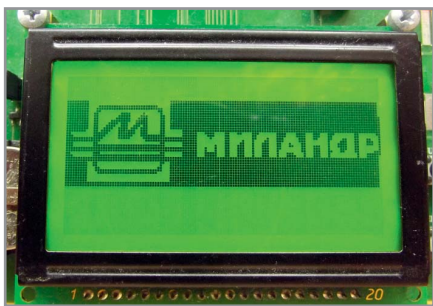


Рис. 9. Логотип «ПКК Миландр», выведенный на дисплей



Рис. 10. Пример неудачного вывода изображения в режиме «256 градаций серого»: некоторые светло-серые элементы не видны



Рис. 11. Тот же логотип в режиме четырёх градаций серого

Программно данный алгоритм реализуется функцией `GLCD_ScrollVertical()` драйвера дисплея.

Драйвер дисплея был испытан на отладочной плате микроконтроллера K1986BE92QI и показал хорошие результаты. В дальнейшем, с внесением определённых изменений, планируется его использование для реализации различных меню в разрабатываемых приборах и устройствах [5].

ЛИТЕРАТУРА

1. Описание индикатора MT-12864J. www.melt.com.ru/docs/MT-12864J.pdf.
2. Подключаем MT-12864-2YLG. www.radiokot.ru/lab/controller/30/.
3. Исходники программ и библиотек. Форум разработчиков электроники Electronix.ru. www.electronix.ru/forum/index.php?showtopic=10934&st=0.
4. Драйвер графического дисплея MT12864 с набором шрифтов. www.electronix.ru/forum/index.php?act=attach&type=post&id=3810.
5. www.milandr.ru.

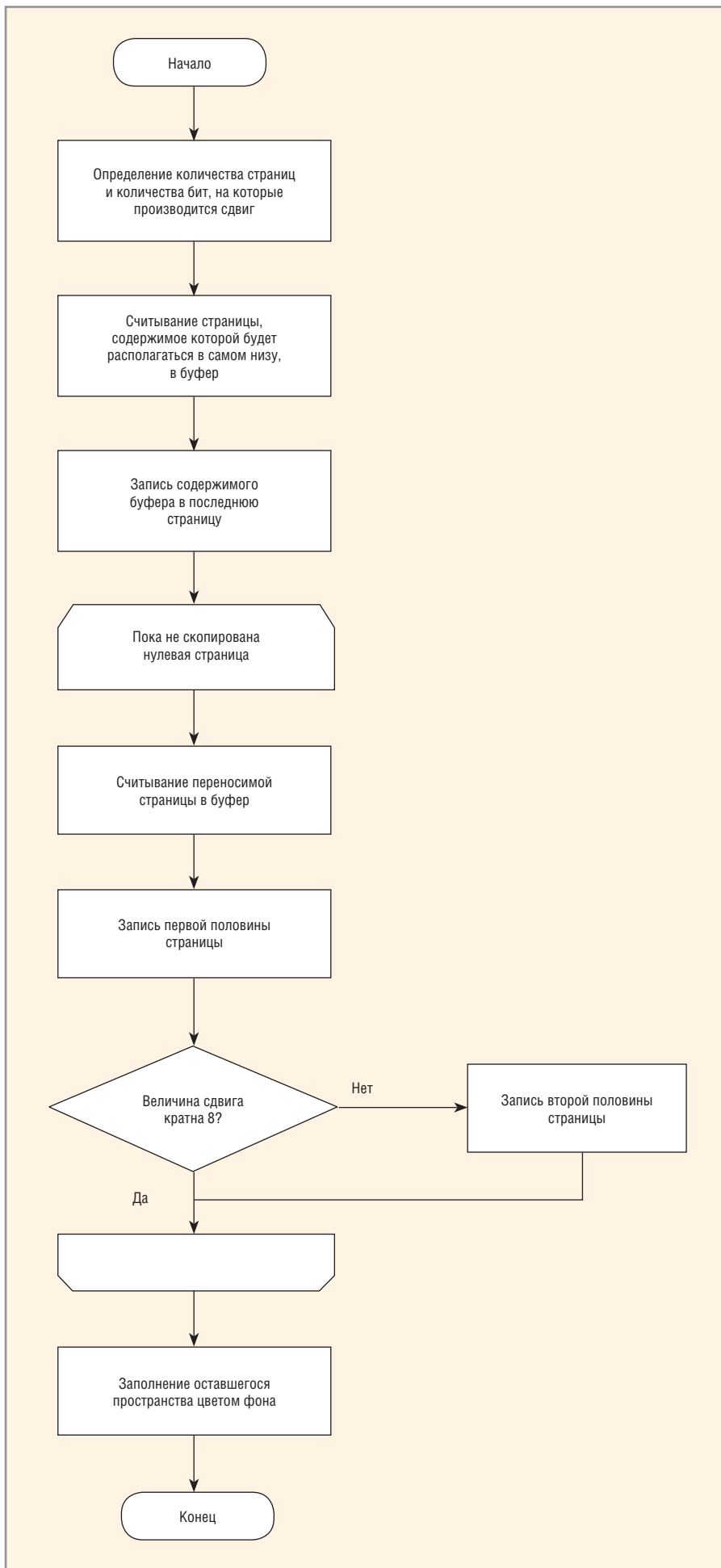


Рис. 12. Блок-схема алгоритма функции сдвига

Источники питания для монтажа на плату

- ✓ Низкая стоимость
- ✓ Высокая надёжность
- ✓ Короткое время выполнения заказа
- ✓ Стандартная площадь посадочного места
- ✓ Наивысшая удельная мощность



**5 Вт
AC/DC
площадь
1"×1"**



**20 Вт
DC/DC
площадь
1"×1"**

Серии DC/DC-преобразователей в стандартном исполнении

Маломощные источники питания AC/DC для установки на плату

Серия ESE

- От 5 до 40 Вт
- Ультракompактные площадью от 1"×1"
- Герметизированное исполнение
- Одно- и двухканальные модели
- Потребляемая мощность в режиме холостого хода < 0,3 Вт

Серия ESL

- От 5 до 30 Вт
- Ультракompактные
- Одно- и многоканальные модели
- Герметизированное исполнение и открытый корпус
- Потребляемая мощность в режиме холостого хода < 0,3 Вт

От 1 до 3 Вт

- Корпуса SIP и DIP
- Варианты со стабилизацией выходного напряжения и с низким коэффициентом стабилизации
- Входы 2:1, 4:1 и ±10%
- Гальваническая развязка вход-выход от 500 В до 6 кВ

От 2 до 6 Вт

- Площади оснований 1"×1", 1"×2", 2"×2" и DIP-24
- Одно-, двух- и трёхканальные модели со стабилизированными напряжениями
- Широкий диапазон входного напряжения 2:1 и 4:1
- Гальваническая развязка вход-выход до 3,5 кВ

Соответствие требованиям медицинских стандартов

- От 3 до 10 Вт
- IEC60601-1, 3-е издание
- Усиленная изоляция 4 кВ (переменный ток)
- Ток утечки на пациента 2 мкА

Драйверы для светодиодных систем освещения

- От 5 до 48 Вт
- Ток нагрузки до 1000 мА
- Режим генератора тока
- КПД 95%

Модели для поверхностного монтажа

- 1 и 2 Вт
- Варианты со стабилизацией выходного напряжения и с низким коэффициентом стабилизации
- Входы 2:1, 4:1 и ±10%
- Одно- и двухканальные модели

Загрузите
интерактивное руководство
по выбору преобразователей AC/DC и DC/DC
www.xppower.com/literature_downloads



Посетите наш сайт и загрузите последний выпуск каталога продукции:
www.prosoft.ru/xp_power



Селектор для загрузки в мобильные устройства



ОФИЦИАЛЬНЫЙ ДИСТРИБЬЮТОР ПРОДУКЦИИ XP POWER