

Практика применения в устройствах на микроконтроллерах дисплейных модулей от 4D Systems

Часть 1

Павел Редькин (г. Ульяновск)

В статье рассказано о применении в устройствах на микроконтроллерах интеллектуальных дисплейных модулей компании 4D Systems со встроенными графическими процессорами. Детально освещены вопросы вывода и отображения на дисплее текстовой и графической (в том числе видео) информации, построения с помощью модуля самодостаточного пользовательского интерфейса конечного устройства. Приведённая в статье информация поможет разработчикам в короткие сроки освоить применение дисплейных модулей от 4D Systems в своих проектах.

ВВЕДЕНИЕ

При создании электронной аппаратуры на базе встраиваемых микроконтроллеров (МК) разработчику, как правило, приходится проектировать её пользовательский интерфейс, включающий средства управления, настройки ввода-вывода, а также отображения пользовательской и сервисной информации. Зачастую аппаратные ресурсы и процессорное время, необходимые для организации сколько-нибудь привлекательного интерфейса, сопоставимы или даже превышают ресурсы и время, требуемые для работы устройства по его основному назначению. Это особенно актуально при использовании МК с невысокой разрядностью, малым количеством аппаратных интерфейсов и линий ввода-вывода. В некоторых случаях пользовательский интерфейс для собственно работы устройства не нужен, но необходим временно для его настройки или отладки. Также бывает, что уже имеющийся в устройстве интерфейс необходимо модернизировать, например, сделать отображение данных в цвете, а не монохромным, или добавить кроме текстового ещё и графический режим, а излишка свободных ресурсов для этого в системе нет.

Именно для перечисленных задач хорошо подходят интеллектуальные дисплейные модули австралийской компании 4D Systems с встроенными графическими процессорами GOLDELOX, PICASO и DIABLO16. Они обеспечивают построение вполне современного и, что особенно важно,

самодостаточного интерфейса, забирая при этом минимальное количество аппаратных ресурсов и процессорного времени управляющего МК. Фактически они берут на себя все ресурсные затраты по организации интерфейса, кроме низкоскоростного канала обмена между управляющим МК и дисплейным модулем. Под самодостаточным здесь понимается такой интерфейс, который включает в себя все необходимые и достаточные компоненты для функционирования устройства, а именно: средства сигнального и пользовательского ввода-вывода (аппаратный интерфейс с МК, собственные входы для подключения кнопок и/или сенсорная панель), средства графического вывода (LED- или LCD-дисплей), средства аудиовывода (собственный аудиовыход). Опционально модули имеют собственную энергонезависимую память большого объёма для хранения текстовых шрифтов, графики, видео и пользовательских данных.

К недостаткам дисплейных модулей от 4D Systems может быть отнесена их сравнительно высокая стоимость.

Актуальный на текущий момент перечень графических модулей от 4D Systems из нескольких десятков позиций можно увидеть на сайте производителя [1]. Для просмотра линейки модулей, о которых идёт речь в статье, на сайте необходимо выбрать категорию microLCD Display Modules и в ней подкатегорию Display Modules, как показано на рисунке 1. Помимо этого, весьма полезная и интересная инфор-

мация о номенклатуре, характеристиках, режимах и порядке применения модулей от 4D Systems содержится в других источниках [2, 3].

АППАРАТНАЯ ПЛАТФОРМА ДЛЯ СОЗДАНИЯ ИНТЕРФЕЙСА

В предлагаемой статье описывается применение графических модулей от 4D Systems только в одном из поддерживаемых ими режимов – Serial (работа под управлением МК). Для примера в статье рассмотрена совместная работа МК с одним из самых простых и дешёвых модулей начального уровня μ OLED-128-G2 [4] из состава линейки microLCD с самым слабым в линейке графическим процессором GOLDELOX. Однако возможности даже такого простого устройства весьма впечатляют. Модуль обеспечивает следующие базовые функции:

- двустороннюю связь с управляющим МК по интерфейсу UART с помощью набора из нескольких десятков простых команд. Поддерживается широкий набор стандартных скоростей;
- опрос до пяти внешних кнопок управления по команде МК. Кнопки подключаются непосредственно к дисплейному модулю. Результаты опроса передаются в МК в ответ на поданную им команду запуска опроса;
- генерация монофонических звуковых сигналов с задаваемыми длительностью и тональностью по команде МК;
- поддержка вывода на встроенный LED-дисплей цветной текстовой информации от МК в единственном доступном шрифте, но с возможностью задания «на лету» расположения, размеров и цветов текстовых символов. Возможен вывод одной командой как отдельных символов, так и их строк;
- поддержка вывода на встроенный LED-дисплей цветной графической информации по командам МК в виде графических примитивов, простых

геометрических фигур с возможностью задания «на лету» их расположения, размеров, цветов и режимов отображения. Возможен вывод одной командой одного графического объекта.

Помимо базовых функций модуль обеспечивает и опциональные, так называемые медийные интерфейсные функции. Их использование возможно при подключении к модулю карты памяти μ SD, поддерживающей интерфейс SPI, для чего имеется стандартный слот. При подключённой карте модулем поддерживается широкий набор шрифтов для текстового вывода, а также вывод сложных цветных изображений и воспроизведение цветного видео (непрерывно или по кадрам) с разрешением встроенного дисплея, но без звукового сопровождения. Все эти функции также реализуются по командам МК, причём вывод одного медийного объекта (изображения или видео) может быть осуществлён всего двумя-тремя командами. Кроме того, подключённую к модулю μ SD-карту МК может использовать в качестве энергонезависимой памяти, доступной для чтения и записи цифровых данных. Шрифты, изображения и видео предварительно необходимо записать на карту с помощью специального инструментального программного обеспечения, о чём будет сказано далее. Количество шрифтов, количество и размеры файлов изображений и файлов видео, других хранимых данных ограничиваются только ёмкостью используемой карты. Модулем μ OLED-128-G2 поддерживаются μ SD-карты объёмом до 4 Гб.

В модуле μ OLED-128-G2 используется цветной (65К цветов) OLED-дисплей с разрешением 128×128 точек, размером диагонали 1,5 дюйма, контрастностью 5000:1, яркостью 100 кд/м^2 и углом обзора 160° . Обмен с управляющим МК модуль осуществляет через UART с заданной по умолчанию скоростью 9600 бит/с, 8-N-1. Помимо работы под управлением внешнего МК модуль также поддерживает автономную работу под управлением программ, написанных на графически-ориентированном языке 4DGL и хранящихся во встроенной памяти графического процессора. Помимо UART модуль имеет две цифровые линии ввода-вывода общего назначения, одна из которых может быть сконфигурирована как аналоговый вход АЦП. Частота тактового генератора графического процессора

Рис. 1. Линейка дисплейных модулей microLCD Display Modules на сайте 4D Systems

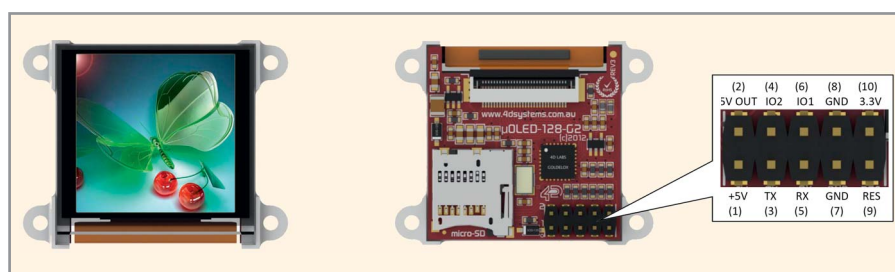


Рис. 2. Внешний вид дисплейного модуля μ OLED-128-G2 с детализацией контактов сигнального/питающего разъёма

GOLDELOX модуля равна 12 МГц, а его системная тактовая частота – 48 МГц. Типовой ток потребления процессора GOLDELOX от источника +3,3 В составляет 12 мА. Внешний вид дисплейного модуля μ OLED-128-G2 представлен на рисунке 2.

По умолчанию после включения питания μ OLED-128-G2 с заводскими настройками находится в состоянии SPE READY, предназначенном для работы в режиме Serial. В этом режиме графическим процессором выполняется заводская управляющая программа

SPE (Serial Platform Emulator). Указанная программа осуществляет преобразование поступающих от МК через UART команд в так называемые внутренние функции графического процессора GOLDELOX.

В качестве внешнего управляющего МК в предлагаемой статье использован 32-разрядный МК LM4F120H5QRFIGA3 семейства STELLARIS фирмы Texas Instruments, установленный на отладочной плате EK-LM4F120XL [5]. В настоящее время этот МК выпускается с названием TM4C1233H6PM.

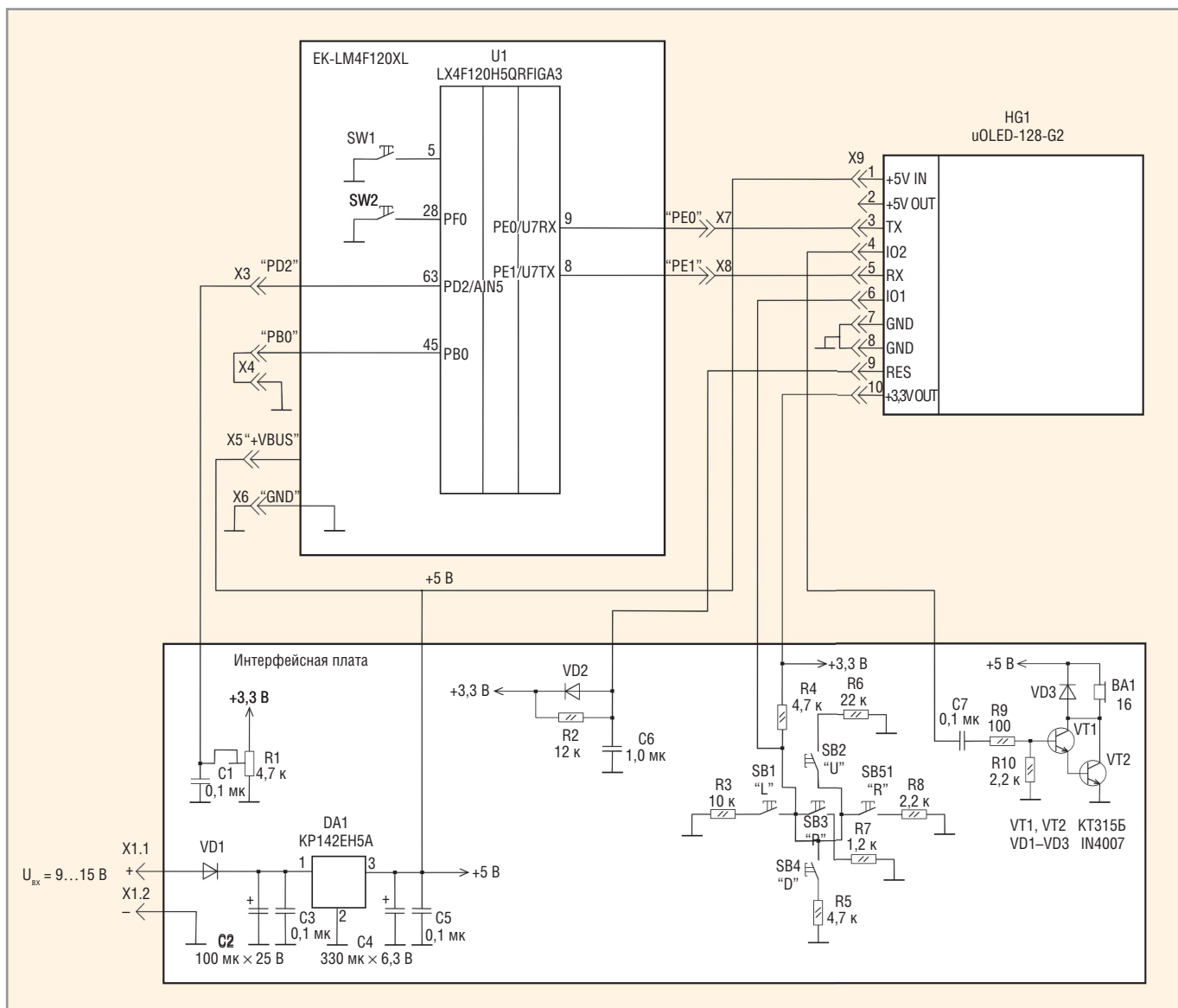


Рис. 3. Принципиальная схема связи «плата EK-LM4F120XL + модуль µOLED-128-G2» с элементами интерфейса, размещёнными на дополнительной плате

Семейство, в которое он входит, сейчас называется TIVA, а отладочная плата – EK-TM4C123GXL. МК имеет процессорное ядро ARM Cortex-M4F с тактовой частотой до 80 МГц, 256 КБ встроенной FLASH-памяти, 32 КБ SRAM и в числе прочих интерфейсов – 8 портов UART. Программирование и отладка МК осуществляется посредством интегрированного в плату JTAG-отладчика Stellaris ICDI через его USB-порт. Плата EK-LM4F120XL не имеет собственного дисплея и снабжена всего двумя пользовательскими кнопками и светодиодом, то есть, пользовательский интерфейс у неё практически отсутствует. Исходя из этого, применение для приложений на основе этой платы дисплейных модулей от 4D Systems представляется вполне обоснованным.

Принципиальная схема связи «плата EK-LM4F120XL + модуль µOLED-128-G2» показана на рисунке 3, а внешний вид

собранного по этой схеме макета – на рисунке 4. Внешние кнопки, усилитель звуковой частоты, первичный стабилизатор питания макета, цепь сброса графического модуля и другие вспомогательные цепи смонтированы на дополнительной интерфейсной плате, также показанной на рисунках 3 и 4. Из внутренней структуры платы EK-LM4F120XL на схеме показаны только задействованные в приложении выводы МК и пользовательские кнопки SW1, SW2.

Программа SPE по умолчанию конфигурирует линию ввода-вывода IO2 модуля µOLED-128-G2 как цифровой выход ШИМ, используемый для генерации звука, а линию IO1 – как аналоговый вход АЦП, используемый для опроса внешних кнопок. Все кнопки подключаются между этим входом и внешними подключёнными к общему проводу резисторами различного

(заданного производителем) сопротивления. С учётом наличия подтягивающего к плюсу питания +3,3 В резистора заданного сопротивления, на этой линии при нажатии каждой из кнопок образуется делитель напряжения с уникальным в пределах массива кнопок коэффициентом деления. Таким образом, нажатие на любую из кнопок создаёт на входе АЦП сопоставленный ей уровень напряжения, который распознаётся программой SPE и в виде соответствующего кода передаётся через UART в управляющий МК. Необходимое для работы кнопочного интерфейса напряжение +3,3 В вырабатывается встроенным в модуль отдельным стабилизатором с допустимым током нагрузки до 50 мА.

Цифровые сигналы со звукового выхода модуля поступают на вход усилителя звуковой частоты, нагруженного на электромагнитный звуковой излу-

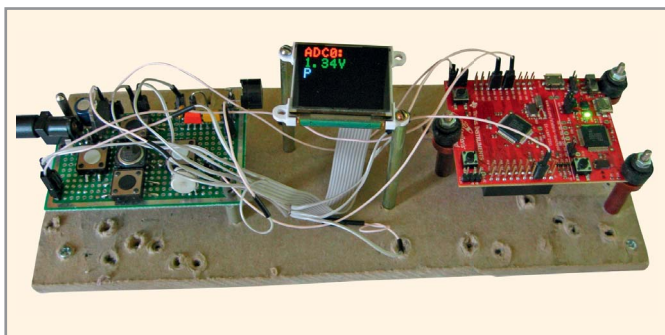


Рис. 4. Внешний вид макета, собранного по принципиальной схеме

тель «спикер», подобный тем, которые используются в материнских платах ПК. Нагрузочная способность цифрового (звукового) выхода IO2 не превышает 4 мА.

Модуль питается от внешнего источника постоянного напряжения +5 В (диапазон питающих напряжений составляет +4,0...+5,5 В), однако его линии ввода-вывода и линии UART используют уровни TTL +3,3 В, хотя толерантны и к напряжению +5 В. Линия сброса RES имеет внутренний подтянутый к уровню +3,3 В резистор 4,7 кОм. Внешний активный низкий уровень на ножке RES длительностью

более 2 мкс вызывает сброс модуля. С выхода модуля «+5 В OUT» при необходимости снимается входное питающее напряжение +5 В, пропущенное через внутренний защитный диод.

Все линии сигналов обмена с МК, цепей сброса, питания и общего провода сведены в модуле μ OLED-128-G2 в один стандартный двухрядный 10-контактный разъём, как показано на рисунке 2.

БАЗОВЫЕ ФУНКЦИИ ДИСПЛЕЙНОГО МОДУЛЯ

Через несколько секунд после включения питания модуля в случае, если

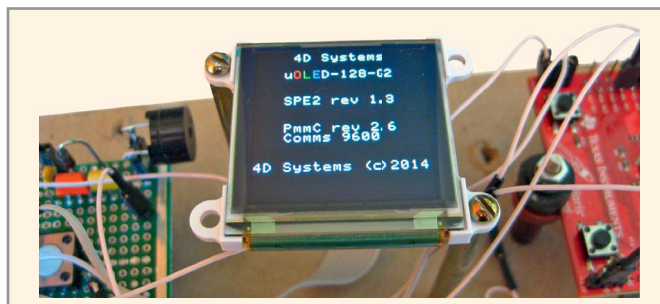


Рис. 5. Начальная экранная заставка дисплея μ OLED-128-G2 в режиме Serial

от управляющего МК не поступит никаких команд, программа SPE выводит на дисплей начальную экранную заставку, показанную на рисунке 5. В ней содержится информация о типе модуля, версиях загруженной в него программы SPE и его внутренней «прошивки» PmmC, а также актуальное в данный момент значение скорости обмена через UART. Выведенная заставка медленно прокручивается по дисплею. Управляющему МК надлежит захватить управление модулем, подавая ему через UART какие-нибудь команды. Например, команды очистки дисплея или запрещения прокрутки

Новое поколение GaN-транзисторов

WolfSpeed

<p>CGHV14250 Диапазон частот: 1200–1400 МГц Мощность: 250 Вт Коэффициент усиления: 18 дБ</p>	<p>CGHV14500 Диапазон частот: 1200–1400 МГц Мощность: 500 Вт Коэффициент усиления: 17 дБ</p>	<p>CGHV35150 Диапазон частот: 2900–3500 МГц Мощность: 150 Вт Коэффициент усиления: 13,5 дБ</p>
--	--	--

ОФИЦИАЛЬНЫЙ ДИСТРИБЬЮТОР ПРОДУКЦИИ WOLFSPPEED

ProChip
POWERED BY ProSoft

Активный компонент вашего бизнеса

ТЕЛ.: (495) 232-2522 / ФАКС: (495) 234-0640 / INFO@PROCHIP.RU / WWW.PROCHIP.RU

Реклама

Таблица 1. Основные текстовые и строковые команды режима Serial

Команда (библиотечная функция)	Назначение команды
txt_MoveCursor	Переместить курсор на заданную номером строки и номером столбца позицию дисплея
PutCH	Вывести на дисплей символ
Putstr	Вывести на дисплей строку символов
Charwidth	Задать ширину символа в пикселях
Charheight	Задать высоту символа в пикселях
txt_FGcolour	Задать цвет выводимого на дисплей текста
txt_BGcolour	Задать цвет фона для выводимого на дисплей текста
txt_FontID	Задать требуемый шрифт, используя его идентификатор (ID). Эта команда поддерживается только для шрифтов, загружаемых с µSD-карты. В этом случае ID = 7
txt_Width	Задать множитель ширины текста в диапазоне 1...16
txt_Height	Задать множитель высоты текста в диапазоне 1...16
txt_Xgap	Задать разрыв между символами по оси X (дистанцию) в пикселях в диапазоне 0...32
txt_Ygap	Задать разрыв между символами по оси Y (интервал) в пикселях в диапазоне 0...32
txt_Bold	Задать полужирный текстовый шрифт
txt_Inverse	Задать текстовую инверсию
txt_Italic	Задать текстовый курсив
txt_Underline	Задать текстовое подчёркивание
txt_Set	Задать временную задержку между символами, выводимыми на дисплей с помощью команд PutCH или Putstr, в диапазоне 0...255 мс

Таблица 2. Основные графические команды режима Serial

Команда (библиотечная функция)	Назначение команды
gfx_Cls	Очистить экран дисплея, используя текущий цвет фона
gfx_ChangeColour	Изменить цвет всех пикселей заданного (старого) цвета на другой заданный (новый) цвет
gfx_Circle	Нарисовать незакрашенную окружность заданного цвета с центром в заданной точке, с заданным радиусом. Все величины задаются в пикселях
gfx_CircleFilled	Нарисовать закрашенную окружность заданного цвета с центром в заданной точке, с заданным радиусом. Все величины задаются в пикселях
gfx_Line	Нарисовать прямую линию заданного цвета между двумя точками с заданными координатами. Все величины задаются в пикселях
gfx_Rectangle	Нарисовать незакрашенный прямоугольник заданного цвета с заданными координатами левого верхнего и правого нижнего углов. Все величины задаются в пикселях
gfx_RectangleFilled	Нарисовать закрашенный прямоугольник заданного цвета с заданными координатами левого верхнего и правого нижнего углов. Все величины задаются в пикселях
gfx_Polyline	Нарисовать ломаную линию заданного цвета в виде отрезков прямых между массивом точек с заданными координатами. Все величины задаются в пикселях
gfx_Polygon	Нарисовать многоугольник (замкнутую фигуру) заданного цвета в виде отрезков прямых между массивом точек с заданными координатами. Все величины задаются в пикселях
gfx_Triangle	Нарисовать треугольник (замкнутую фигуру) заданного цвета в виде отрезков прямых между тремя точками с заданными координатами. Все величины задаются в пикселях
gfx_Orbit	Вычислить координаты (x, y) удаленной точки относительно исходной точки, исходя из заданных параметров: угла и расстояния от исходной точки до удаленной. По умолчанию исходная точка имеет координаты (0, 0). Все величины задаются в пикселях
gfx_PutPixel	Закрасить пиксель с заданными координатами заданным цветом. Координаты задаются в пикселях
gfx_GetPixel	Прочитать значение цвета пикселя с заданными координатами. Координаты задаются в пикселях
gfx_MoveTo	Переместить исходную точку по заданным координатам. Координаты задаются в пикселях
gfx_LineTo	Нарисовать линию из исходной точки до точки с заданными координатами. После этого исходная точка перемещается по этим координатам. Координаты задаются в пикселях
gfx_Clippping	Включить/отключить возможность усечения (клиппирования) дисплея
gfx_ClipWindow	Задать координаты окна усечения дисплея. Любые объекты и текст, выведенные на дисплей, будут усечены и станут видны на дисплее только в области этого окна. Чтобы окно усечения было включено, необходимо передать команду gfx_Clippping
gfx_SetClipRegion	Расширить область усечения. Команда расширяет область усечения вплоть до размеров области расположения последнего выведенного на дисплей текста или изображения
gfx_BGcolour	Задать цвет фона для выводимых на дисплей графических объектов
gfx_OutlineColour	Задать цвет для прямоугольников и кругов
gfx_Contrast	Задать уровень контраста дисплея в диапазоне 1...15. При уровне контраста 0 дисплей отключается

выводимого изображения. Затем можно приступить к выводу на дисплей требуемых в приложении данных, опросу кнопок, генерации звука в нужном приложению контексте и т.д. В документации производителя на модуль [6] сказано, что МК может пытаться связываться с модулем приблизительно через три секунды после его сброса, а начальная заставка появляется примерно через 8 секунд. Однако по опыту работы с экземпляром µOLED-128-G2 оказалось, что модуль корректно реагирует на команды МК практически сразу после сброса.

Полный перечень из более чем семидесяти команд, поддерживаемых модулем в режиме Serial, с их подробным описанием содержится в [6]. Этот документ можно загрузить по ссылке с интернет-страницы [7]. Необходимо заметить, что каждая конкретная команда может поддерживаться не всеми моделями модулей, что отражено в документации.

Для облегчения использования своих модулей совместно с различными сторонними программными средствами и аппаратными платформами (ARM, AVR, PIC, Arduino и пр.) производитель распространяет перечень команд режима Serial в виде наборов библиотек: Arduino Serial Library, C Serial Library, Pascal Serial Library, PicAxe Serial Library, Linux Serial Library. Перечисленные библиотеки в виде архивных файлов могут быть загружены по соответствующим ссылкам с интернет-страницы [7].

Рассмотрим систему команд режима Serial подробнее. Группа текстовых и строковых команд управляет курсором дисплея при выводе текста, задает его цвет, размеры и атрибуты (жирный, курсив, подчёркивание, инверсия). При этом доступен только один текстовый шрифт. Основные из этих команд перечислены в таблице 1. Эта и другие таблицы, приведённые в статье, содержат только имена и краткие характеристики команд (библиотечных функций) без их кодов и параметров.

Цвета выводимых на дисплей текстовых символов и графических объектов задаются в командах двухбайтовой константой в соответствующем поле. Перечень соответствия этих констант цветам дисплея приведён в [8].

Группа графических команд управляет выводом на дисплей простых геометрических фигур (прямых и ломаных линий, окружностей, треугольни-

ков, прямоугольников и т.п.), задаёт их расположение, размеры, цвет и режимы отображения, например, заполнение цветом. Сюда же входит такая важная команда, как очистка экрана `gfx_Cls`. Основные графические команды перечислены в таблице 2.

Группа медийных команд управляет взаимодействием модуля с подключённой к нему μ SD-картой, её начальной инициализацией, записью-чтением данных с неё, выводом с неё на дисплей изображений, видеокладов, воспроизведением с неё видео. Основные из этих команд перечислены в таблице 3.

Группа команд доступа к памяти предназначена для обеспечения доступа для записи и чтения к управляющим системным регистрам графического процессора. Основные команды этой группы перечислены в таблице 4. В рамках данной статьи эти команды рассматриваться не будут.

Отдельная команда под названием `Joystick` обеспечивает опрос всех подключённых к модулю кнопок (до пяти кнопок) и обнаружение среди них нажатой. Код нажатой на момент опро-

Таблица 3. Основные медийные команды режима `Serial`

Команда (библиотечная функция)	Назначение команды
<code>media_Init</code>	Инициализировать носитель-карту памяти μ SD/SD/SDHC для проведения с ней дальнейших операций. μ SD-карта подключается к графическому процессору через интерфейс SPI
<code>media_SetAdd</code>	Задать в памяти носителя внутренний указатель адреса для доступа к не выровненному по границе сектора байту
<code>media_SetSector</code>	Задать в памяти носителя внутренний указатель адреса для доступа к сектору
<code>media_ReadByte</code>	Считать и вернуть значение байта из носителя по адресу, заданному командой <code>media_SetAdd</code>
<code>media_ReadWord</code>	Считать и вернуть значение слова (2 байта) из носителя по адресу, заданному командой <code>media_SetAdd</code>
<code>media_WriteByte</code>	Записать байт в носитель по адресу, заданному командой <code>media_SetSector</code> . Запись байтов или слов в носитель должна запускаться с начала сектора
<code>media_WriteWord</code>	Записать слово (2 байта) в носитель по адресу, заданному командой <code>media_SetSector</code> . Запись байтов или слов в носитель должна запускаться с начала сектора
<code>media_Flush</code>	Подача этой команды после записи любых данных в носитель гарантирует, что сектор, в который производилась запись, будет корректно сохранён
<code>media_Image</code>	Вывести на дисплей изображение, хранящееся в носителе по указанному адресу. Адрес изображения предварительно должен быть задан командой <code>media_SetAdd</code> или <code>media_SetSector</code>
<code>media_Video</code>	Воспроизвести на дисплее видеоклип, хранящийся в носителе по указанному адресу. Адрес видеоклипа предварительно должен быть задан командой <code>media_SetAdd</code> или <code>media_SetSector</code> . Во время воспроизведения видео блокируются все другие процессы
<code>media_VideoFrame</code>	Вывести на дисплей один или несколько кадров видеоклипа, хранящегося в носителе по указанному адресу. Адрес видеоклипа предварительно должен быть задан командой <code>media_SetAdd</code> или <code>media_SetSector</code>

са кнопки в ответе на эту команду возвращается в МК.

Отдельная команда под названием `Beep` обеспечивает генерацию модулем непрерывного звукового сигнала

с заданными тональностью и длительностью. Тональность (частота) задаётся в относительных единицах в диапазоне от 0 до 64. Длительность звучания задаётся в мс.

SHARP

LCD-ПАНЕЛИ SHARP ДЛЯ ПРОМЫШЛЕННЫХ ПРИМЕНЕНИЙ



Размер диагонали, дюйм	10,4	12,1	15...23	2,5...8,5	
Разрешение	VGA, SVGA	SVGA, WXGA	XGA, SXGA, USGA	Q-VGA, SVGA, WVGA	
Яркость, кд/м ²	420...450	400...1200	350...1000	280...500	
Контрастность : 1	600...800	800...1000	600...1000	300...800	
Входной видеосигнал	Digital RGB, LVDS	Digital RGB, LVDS	LVDS	Digital RGB, LVDS	
Задняя подсветка	LED	LED	LED	LED	
Предельная рабочая температура, °C	мин.	-30	-15	-20	-30
	макс.	+80	+75	+70	+70

ОФИЦИАЛЬНЫЙ ДИСТРИБЬЮТОР ПРОДУКЦИИ SHARP

ProCHIP
POWERED BY PROSOFT

Активный компонент вашего бизнеса

ТЕЛ.: (495) 232-2522 / ФАКС: (495) 234-0640 / INFO@PROCHIP.RU / WWW.PROCHIP.RU



Реклама

Таблица 4. Основные команды доступа к памяти режима Serial

Команда (библиотечная функция)	Назначение команды
peekB	Возвратить прочитанное значение младшего байта системного регистра графического процессора с заданным адресом
pokeB	Задать значение младшего байта системного регистра графического процессора с заданным адресом
peekW	Возвратить прочитанное значение слова (2 байта) системного регистра графического процессора с заданным адресом
pokeW	Задать значение слова (2 байта) системного регистра графического процессора с заданным адресом

Таблица 5. Основные системные команды режима Serial

Команда (библиотечная функция)	Назначение команды
sys_GetModel	Возвратить модель дисплея в форме текстовой строки без разделителя NULL
sys_GetVersion	Возвратить версию приложения SPE, загруженного в дисплей
sys_GetPmmC	Возвратить версию «прошивки» PmmC, загруженной в дисплей
SSTimeout	Задать значение тайм-аута экранной заставки в диапазоне 1...65535 мс. Значение 0 отключает экранную заставку
SSSpeed	Задать скорость прокрутки экранной заставки. Диапазоны возможных значений скорости различаются для различных моделей модулей. Для µOLED-128-G2 этот диапазон составляет от 0 (самая быстрая скорость) до 3 (самая медленная скорость)
SSMode	Задать направление прокрутки экранной заставки

Таблица 6. Соответствие между значением ответа на команду Joystick и кнопками на принципиальной схеме, показанной на рисунке 3 (позициями джойстика)

Значение байта ответа	Кнопка на схеме, изображённой на рисунке 3	Позиция джойстика
0	Ни одна кнопка не нажата	RELEASED
1	SB2 «U»	Вверх – UP
2	SB1 «L»	Влево – LEFT
3	SB4 «D»	Вниз – DOWN
4	SB5 «R»	Вправо – RIGHT
5	SB3 «P»	Центр – PRESS

Отдельная команда задания скорости обмена Set Baud Rate обеспечивает выбор скорости обмена через UART из набора стандартных значений в диапазоне от 110 до 600 000 бит/с.

Отдельная команда управления изображением blitComtoDisplay задаёт горизонтальную и вертикальную позиции верхнего левого угла изображения, которое будет выведено на дисплей, ширину и высоту этого изображения, а далее в своём теле содержит само изображение, передаваемое попиксельно в виде 16-разрядных слов, значение каждого из которых определяет цвет пикселя. Эта команда может быть очень длинной.

Группа системных команд обеспечивает вывод на дисплей информации о модели и версиях встроенного ПО (SPE, PmmC) модуля, а также задаёт режим, тайм-аут и скорость прокрутки изображения на дисплее. Основные системные команды перечислены в таблице 5. Чтобы отменить прокрутку любого выводимого на дисплей изображения, заданную в программе SPE по умолчанию, необходимо передать в модуль команду SSTimeout(0).

Заметим, что отобразив, например, на дисплее какие-либо графические изображения, можно тут же рядом с ними или поверх них вывести и текстовые данные. И наоборот. То есть текстовый и графический режимы не разделяются.

Рассмотрим формат команды на примере команды Joystick. Передача через UART всегда ведётся побайтно. Сначала МК передаёт так называемый cmd-код команды (для Joystick cmd = 0xFFD9). Первым всегда передаётся старший байт. Если команда корректно воспринята модулем, он возвращает байт подтверждения ACK, всегда равный 0x06. Какое-либо иное значение байта подтверждения говорит о некорректном приёме команды. Затем модуль возвращает ответ на команду – один байт с номером кнопки, нажатой на момент опроса, то есть на момент поступления команды Joystick. Соответствие между значением ответа и изображёнными на принципиальной схеме кнопками приведено в таблице 6. Заметим, что одновременное нажатие двух и более кнопок является некорректным и приведёт к неоднозначному результату.

Команда позиционирования текстового курсора в определённое положение на дисплее txt_MoveCursor будет выглядеть следующим образом. Сначала код cmd = 0xFFE4 (2 байта), затем параметры: номер строки (2 байта), номер столбца (2 байта). Ответ форматом этой команды не предусмотрен. Например, чтобы переместить курсор в строку 7 (0x07), ряд 12 (0x0C), МК должен передать в модуль последовательность байтов: 0xFF 0xE4 0x00 0x07 0x00 0x0C, а в подтверждении получить 0x06. Старшие байты в двухбайтовых параметрах должны быть заполнены незначащими нулями. По умолчанию после сброса для вывода на дисплей текстовых данных задан самый мелкий из возможных размеров шрифта – «4D SYSTEM (7x8)» (см. рис. 5). С учётом разрешения дисплея 128 × 128 точек, для такого шрифта доступно тестовое поле из 16 строк и 18 столбцов. Команды вывода на дисплей символа и символьной строки выводят данные, начиная с текущей позиции курсора.

Чтобы нарисовать на дисплее полый (не закрашенный) прямоугольник функцией gfx_Rectangle, нужно передать код cmd = 0xFFCF (2 байта), координаты позиции X1 (2 байта) в пикселях, позиции Y1 (2 байта) в пикселях, позиции X2 (2 байта) в пикселях, позиции Y2 (2 байта) в пикселях, цвет (2 байта) этого прямоугольника. Ответ форматом этой команды не предусмотрен. Например, для синего (код цвета 0x001F) прямоугольника, рисуемого из точки с координатами X1 = 10 (0x0A), Y1 = 20 (0x14) в верхнем левом углу дисплея в точку с координатами X2 = 80 (0x50), Y2 = 80 (0x50) в правом нижнем углу, МК должен передать в модуль последовательность байтов: 0xFF 0xCF 0x00 0x0A 0x00 0x14 0x00 0x50 0x00 0x50 0x00 0x1F, а в подтверждение получить 0x06.

Каждая переданная от МК команда требует определённого времени для её обработки модулем. Это время между моментами поступления команды и генерации подтверждения на неё или ответа, если последний предусмотрен форматом. Необходимо заметить, что передав в модуль команду, МК не должен передавать что-либо ещё, пока не дождётся от модуля подтверждения, а затем и ответа, если последний предусмотрен форматом. В противном случае логика модуля поведёт себя неоднозначно.

С целью отладки алгоритмов взаимодействия МК с модулями от 4D Systems автором была написана управляющая программа для МК платы EK-LM4F120XL, реализующая как базовые функции модуля (вывод на дисплей текстовых данных с заданным по умолчанию шрифтом), так и медийные (выбор шрифта, вывод на дисплей графики и видео) с использованием немного переработанной библиотеки функций C Serial Library. Исходный текст этой программы (и файл «прошивки» Flash-памяти МК) содержится в каталоге проекта UART_4D, архив которого доступен для загрузки с сайта журнала (www.soel.ru). Проект был подготовлен и отлажен в интегрированной среде разработки IDE μ Vision V5.15 от Keil Software (бесплатная версия с ограниченным размером кода). Программирование и отладка производились с помощью встроенного JTAG-отладчика Stellaris ICDI.

При включении питания макета программа передаёт в модуль команды очистки дисплея, запрета прокрутки изображения, задания размеров текстовых символов, их цветов, позици-

онирования курсора, а затем циклически (в моменты завершения АЦП МК) осуществляет вывод на дисплей цифровых данных – результата АЦП в вольтах. Входное измеряемое напряжение для АЦП формируется в интерфейсной плате из питающего напряжения +3,3 В цепью R1, C1 и поступает на линию МК PD2, сконфигурированную как аналоговый вход AIN5. Вид дисплея для этого случая показан на рисунке 4. Как видно из рисунка, при достаточно крупном размере символов шрифт на дисплее выглядит не очень эстетично. Восприятие картинки можно улучшить с помощью медийной функции задания шрифтов, о чём будет рассказано во второй части статьи.

Помимо вывода на дисплей результатов АЦП программа циклически (примерно два раза в секунду) передаёт в графический модуль команду Joystick и анализирует ответ на неё. В зависимости от наличия нажатой кнопки и её номера программа выводит на дисплей соответствующий ей символ: «U», «L», «D», «R», «P». Каждое нажатие на любую из кнопок макета (две штатные кноп-

ки платы EK-LM4F120XL и пять кнопок интерфейсной платы) сопровождается коротким звуковым сигналом высокого тона, генерируемым дисплейным модулем по команде Веер, поступающей из МК.

ЛИТЕРАТУРА

1. www.4dsystems.com.au/products.
2. Долгушин С. Графические контроллеры и дисплейные модули компании 4D Systems. Компоненты и технологии. 2013. №2.
3. Долгушин С. Графический интерфейс пользователя на базе готовых дисплейных модулей компании 4D Systems. Компоненты и технологии. 2013. №4.
4. www.4dsystems.com.au/product/uOLED_128_G2.
5. Stellaris LM4F120 LaunchPad Evaluation Board. User Manual. www.ti.com/general/docs/lit/getliterature.tsp?baseLiteratureNumber=spmu289&fileType=pdf&keyMatch=LM4F120H5QR&tisearch=Search-EN-TechDocs.
6. 4D Systems Goldebox Serial Environment Command Set. Part of the Workshop 4 IDE.
7. www.4dsystems.com.au/product/4D_Workshop_4_IDE.
8. 4D Systems Application Note 4D-AN-00043. General 4DGL Colour Constants. ©



bulgin
a brand of Elektron Technology

Виссaneer 6000

Первый разъём от Vulgin с «ЖЕЛЕЗНЫМ» характером



- Корпус из металла или пластика (на выбор)
- Степень защиты IP66, IP68 и IP69K
- Прочность и защита от ЭМИ благодаря металлическому корпусу
- 16 А, 277 В для силовых соединителей
- Силовые и интерфейсные разъёмы в одинаковых корпусах
- Удобный и быстрый механизм фиксации Push/pull

-40...+120°C

IP66

IP68

IP69K

ОФИЦИАЛЬНЫЙ ДИСТРИБЬЮТОР ПРОДУКЦИИ VULGIN НА ТЕРРИТОРИИ РОССИИ И СТРАН СНГ



POWERED BY PROSOFT

Активный компонент вашего бизнеса

ТЕЛ.: (495) 232-2522 / ФАКС: (495) 234-0640 / INFO@PROCHIP.RU / WWW.PROCHIP.RU



Реклама