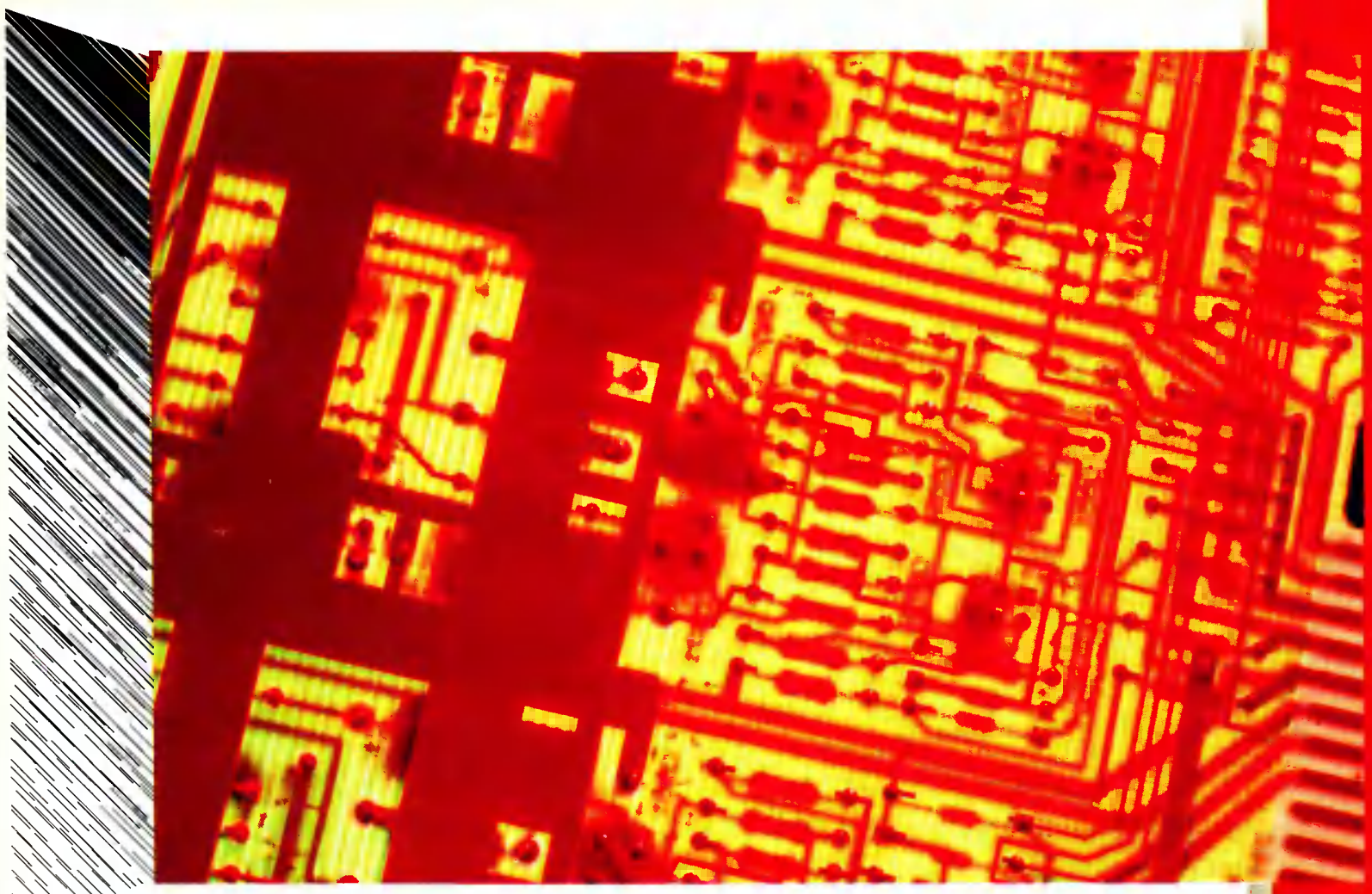
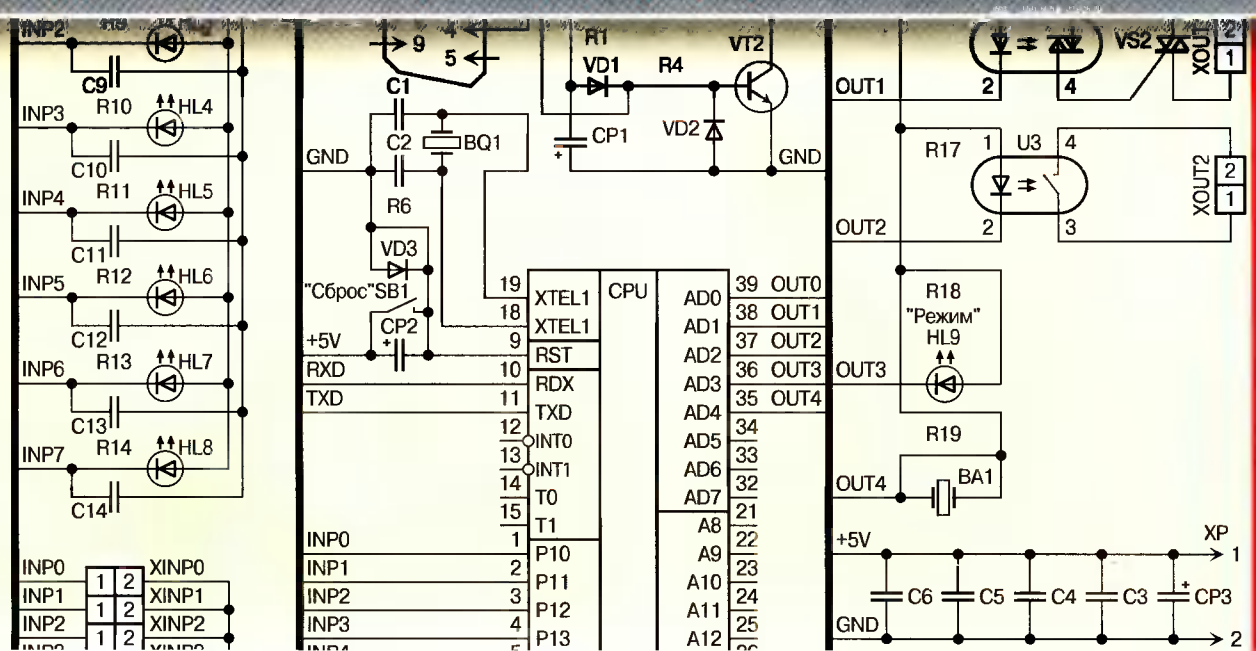
СХЕМОТЕХНИКА

№12
декабрь
2007



Магнетометры и компасы фирмы Honeywell
Математическая батарейка
Звездное небо на микроконтроллере PIC18F2525
Преобразователь сигналов дифференциального емкостного датчика



подписные индексы
80724 , 82117

НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

СХЕМОТЕХНИКА

№ 12 (86)
декабрь 2007

И. о. главного редактора
Юлия Асташкевич

Редакционная коллегия
Павел Асташкевич
Юлия Герасимова
Павел Чикин

Дизайн и верстка
Ирина Чикина

Отдел распространения
Марина Трофимова
Сергей Лукин
(495) 775-16-76
e-mail: sales@dian.ru

Отдел рекламы
Юлия Иванова

Адрес редакции:
Москва, Ленинградское ш., д. 18, оф. 342
Для писем: 125047 Москва, а/я 202
«ИД Скимен»
тел./факс: (495) 775-16-76
www.dian.ru
e-mail: editor@dian.ru
editordian@mail.ru

Издатель и учредитель
ООО "ИД Скимен"

Отпечатано в ОАО Ордена Трудового Красного Знамени "Чеховский полиграфический комбинат"
142300 г. Чехов Московской области,
Сайт www.chpk.ru
E-mail: marketing@chpk.ru
Факс 8(499)6-25-36,
факс 8(499)270-73-59
Отдел продаж услуг многоканальный:
8(499)270-73-59

Тираж 5 200 экз.
Заказ №

Журнал зарегистрирован в Министерстве РФ по делам печати, телерадиовещания и средств массовых коммуникаций
Рег. № ПИ77-5262

Журнал включен в Реферативный журнал и Базы данных ВИНИТИ. Сведения о журнале ежегодно публикуются в международной справочной системе по периодическим и продолжающимся изданиям «Ulrich's Periodicals Directory».

Редакция не несет ответственности за информацию, приведенную в рекламных материалах

За содержание статьи и ее оригинальность несет ответственность автор

Полное или частичное воспроизведение материалов допускается только с разрешения ООО "ИД Скимен"

Информацию о подписке см. на последней странице журнала

Содержание

Искусство схемотехники

О. Николайчук. Использование цветного графического LCD от телефонов Nokia в микроконтроллерных системах 2

Практика

А. Межлумян. Не слишком ли дорого обходятся дешевые компьютерные блоки питания? (Незамеченная революция в компьютерных блоках питания) 10

В. Селиванов. Преобразователь сигналов дифференциального емкостного датчика 12

Ю. Садиков. Экономичный светильник на мощных светодиодных лампах для радиолюбительского рабочего места 13

Компоненты

А. Зыбайло. Магнетометры и компасы фирмы Honeywell 16

М Ильин, А. Крук. Пассивные компоненты (эффективная защита от перенапряжения) 20

Софт

О. Вальпа. Методика быстрого создания программ для карманных персональных компьютеров и коммуникаторов 22

О. Петраков. Математическая батарейка 29

А. Брякин. Звездное небо на микроконтроллере PIC18F2525 32

Новые книги. А. Фрунзе. «Микроконтроллеры фирмы Philips семейства x51» (с. 31). О. Николайчук. «x51-совместимые микроконтроллеры фирмы Silicon Laboratires (Cygnal)» (с. 34). Сводное содержание журнала «Схемотехника» за 2007 г. (с. 35).

Использование цветного графического LCD от телефонов Nokia в микроконтроллерных системах

Настоящая статья посвящена проблемам использования цветных графических жидкокристаллических индикаторов от мобильных телефонов фирмы Nokia в различных микроконтроллерных системах. Описаны основные достоинства и недостатки используемых графических индикаторов, рассмотрены основные программные отличия.

Многие разработчики современных микроконтроллерных систем стремятся к использованию малогабаритных цветных графических жидкокристаллических индикаторов. Это обусловлено тем, что наличие в микроконтроллерной системе встроенного графического индикатора существенно повышает функциональные возможности системы, позволяет отображать большие объемы разнообразной информации, используя при этом различные формы представления данных. Наличие в устройстве цветного индикатора повышает его эргономические параметры, позволяет минимизировать количество используемых клавиш и организовать многоуровневые системы управления режимами с помощью вложенных меню. Конечно, при этом разработчики стремятся существенно не увеличивать стоимость устройства и аппаратные затраты.

Исходя из этих соображений, можно сформулировать основные требования к подсистеме (узлу) представления (отображения) данных для микроконтроллерных устройств:

- Подсистема отображения данных должна иметь достаточное разрешение для формирования качественного изображения или текста, т. е. индикатор должен иметь определенное минимальное количество активных управляемых точек изображения (пикселей) по осям X и Y. Для большинства микроконтроллерных систем считается достаточным разрешение 128×28 пикселей. Это позволяет выводить примерно 16 символов по 16 строк с размером шрифта 8×8 мм, строить на экране довольно качественные графики и диаграммы, а также отображать достаточно качественные картинки. Дальнейшее увеличение разрешения экрана приводит к значительному увеличению стоимости индикатора, размеров устройства и времени построения изображения.
- Желательно, чтобы подсистема отображения была оснащена индикатором, обеспечивающим высокое цветовое разрешение. Нормальным для современных систем считается цветовое разрешение — 256...4096 цветов. Конечно, наличие большего цветового разрешения приветствуется, однако формирование изображений при этом значительно замедляется.
- Желательно, чтобы индикатор имел встроенный контроллер, обеспечивающий достаточно простое управление.
- Индикатор должен иметь достаточно высокую яркость и формировать высококонтрастное изображение даже при солнечном освещении.
- Индикатор должен иметь низковольтное питание, желательно такое же, как и остальная элементная база устройства.
- Индикатор должен иметь малое энергопотребление.
- Индикатор цветной подсистемы отображения должен иметь минимальную стоимость.
- Индикатор должен быть легко доступным и широко распространенным.
- Индикатор должен иметь минимальное количество линий управления и достаточно простой интерфейс.
- Желательно, чтобы индикатор был оснащен стандартизованным соединителем (разъемом).
- Индикатор должен иметь минимально возможные размеры и вес.

Очевидно, что искать подходящий индикатор можно либо среди промышленных индикаторов, либо среди индикаторов современных мобильных устройств — телефонов, КПК, плейеров, так как именно эти устройства являются наиболее распространенными изделиями, содержащими цветные графические индикаторы. Даже поверхностного ознакомления с продукцией фирм, выпускающих промышленные цветные графические индикаторы (например, фирм Data Vision [1], «Мэлт» [2], Intech-LCD [3], Ampire [4], Bolymin [5] и др.) достаточно, чтобы понять, что все промышленные графические индикаторы имеют достаточно большие размеры, и соответственно, не подходят для наших целей. Таким образом, основной областью поиска являются индикаторы для мобильных устройств.

Цветные жидкокристаллические индикаторы для карманных компьютеров и различных плейеров являются довольно дорогими и дефицитными. С другой стороны, практически каждый современный мобильный телефон оснащается цветным графическим жидкокристаллическим индикатором. Поэтому, основное внимание в ходе поисков мы уделили индикаторам мобильных телефонов.

В результате ознакомления с особенностями производства мобильных телефонов выяснилось, что все разнообразие мобильных телефонов объединяется в ряд довольно больших групп по принципу платформ — некоторых однотипных схемотехнических и конструктивных решений. Это в свою очередь означает, что в таких группах используется один и тот же конструктивно совпадающий тип индикатора.

При рассмотрении различных групп жидкокристаллических индикаторов принимались во внимание следующие основные критерии:

- количество элементов изображения по осям X и Y;
- количество градаций цветовой гаммы;
- размеры области изображения;
- наличие и тип разъема;
- габаритные размеры индикатора и шлейфа с разъемом;
- наличие в телефонах, в которых используются выбранные индикаторы, конструктивных элементов

крепления индикатора к плате — крепежных рамок с амортизаторами;
- количество сигнальных линий и сложность протокола;
- номенклатура питающих напряжений;
- уровни напряжений сигнальных линий;
- доступность и стоимость индикатора.

В результате поисков наш выбор остановился на индикаторах семейства ML019/020 для телефонов Nokia [6, 7].

Выбранные индикаторы имеют следующие параметры:
- количество элементов изображения по осям X и Y — 132×132;
- количество градаций цветности кодируется одним байтом (256 цветов), либо 12 битами (4096 цветов). Разрядность цветового кодирования выбирается программно;
- размеры области изображения составляет примерно 28×28 мм;
- количество используемых сигнальных линий разъема, включая линии питания — 9. Протокол нестандартный трехпроводный, очень похожий на SPI;
- напряжение питания — 2,7…3,3 В;
- достаточно низкая стоимость — примерно 9 долл.;

Собственно жидкокристаллический индикатор закреплен в пластиковом держателе, с обратной стороны которого размещены четыре светодиода подсветки. Габаритные размеры рамки пластмассового держателя составляют 35×35 мм. Соединительный шлейф с разъемом выступают за габаритные размеры рамки и также защищены пластмассовой защитной рамкой меньшего размера. Общие габаритные индикатора с защитными рамками составляют 35×48×3,5 мм.

Соединительный разъем имеет 10 контактов. Кабельная часть установлена на шлейфе индикатора, а платная часть предназначена для поверхностного (SMT — Surface Mount Technology) монтажа.

В телефонах, в которых используются выбранные индикаторы, имеются очень удобные конструктивные элементы крепления индикатора к плате.

Выбранный индикатор широко распространен, так как используется примерно в 30 типах мобильных телефонов, самые известные из которых Nokia 2650, 3100, 3200, 5100, 5140, 6100, 6220, 6610, 6610i, 6800, 6810, 6820, 7210, 7210e, 7250, 7250i и др.

В результате поиска информации по выбранным жидкокристаллическим индикаторам выяснилось, что эти индикаторы выпускаются рядом производителей, причем в них используется несколько типов контроллеров. Наиболее распространенными являются жидкокристаллические индикаторы с контроллерами PCF8833 (Phillips) [8] и S1D15G00 (Epson) [9].

К сожалению, на многих индикаторах не указывается их тип, не говоря уже о типе контроллера, который в нем используется. Удалось выяснить, что индикаторы с контроллером PCF8833 фирмы Phillips имеют коричневый цвет печатной платы контроллера и шлейфа (рис. 1), а индикаторы S1D15G00 фирмы Epson имеют зеленый цвет печатной платы и шлейфа (рис. 2). Индикаторы фирмы Epson могут содержать и другие аналогичные контроллеры, например S1D15G10, S1D15G14, S1D15G17, имеющие незначительные отличия.

На рис. 1 и 2 индикаторы показаны с обеих сторон. В обоих индикаторах контроллеры установлены на дополнительных печатных платах со стороны разъема и заклеены бумажной наклейкой с производственными данными (рис. 2). На индикаторе, показанном на рис. 1, наклейка с платы контроллера снята. На каждом из рисунков показана также платная часть разъема.

Хотя ЖКИ с этими контроллерами и могут использоваться в одном и том же телефоне, они имеют контроллеры, не совместимые по системе команд. Программное обеспечение мобильных телефонов определяет тип подключенного индикатора, точнее, тип используемого в индикаторе контроллера с помощью команд чтения ID (Identification Code) дисплеев. На различных форумах разработчиков микроконтроллерных систем тема использования выбранных жидкокристаллических индикаторов широко обсуждается. Многие отдают предпочтение индикаторам с контроллерами фирмы Phillips. Как правило, отмечают следующие достоинства этих индикаторов. Во-первых, документация на контроллер PCF8833 фирмы Phillips более понятная и читаемая. Во-вторых, если с обратной стороны индикатора аккуратно приподнять защитную наклейку, покрывающую печатную плату контроллера, а затем приподнять шлейф, то откроются 11 больших контактов, к которым очень удобно подпаять обычные провода (рис. 1). В третьих, на описываемом индикаторе проще получить нормальную цветовую гамму, чем на индикаторе S1D15G00 фирмы Epson. Однако проведенные автором эксперименты показали, что если не принимать во внимание наличие у индикатора от фирмы Phillips печатных контактов на плате и пользоваться предусмотренными разъемами, остальные доводы можно считать субъективными. На самом деле достаточно хорошо программируются и ведут себя оба типа индикаторов.

Жидкокристаллические индикаторы крепятся к печатной плате телефона специальной рамкой, состоящей из двух частей (рис. 3).

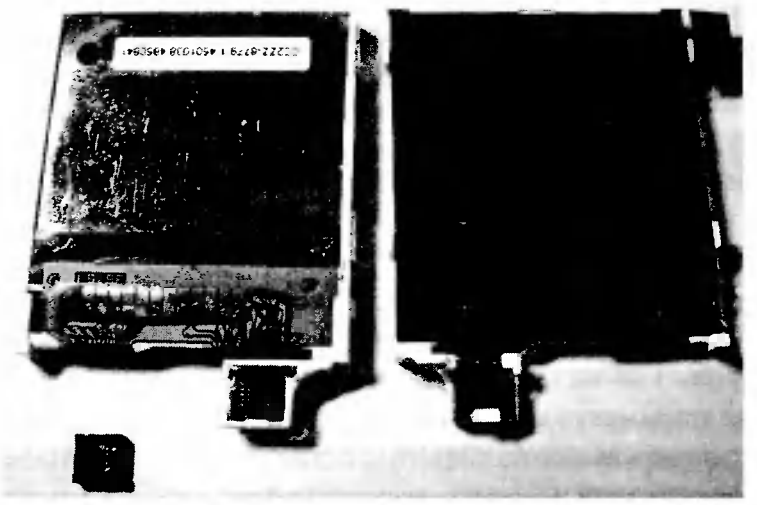


Рис. 1. Жидкокристаллический индикатор с контроллером PCF8833 фирмы Phillips

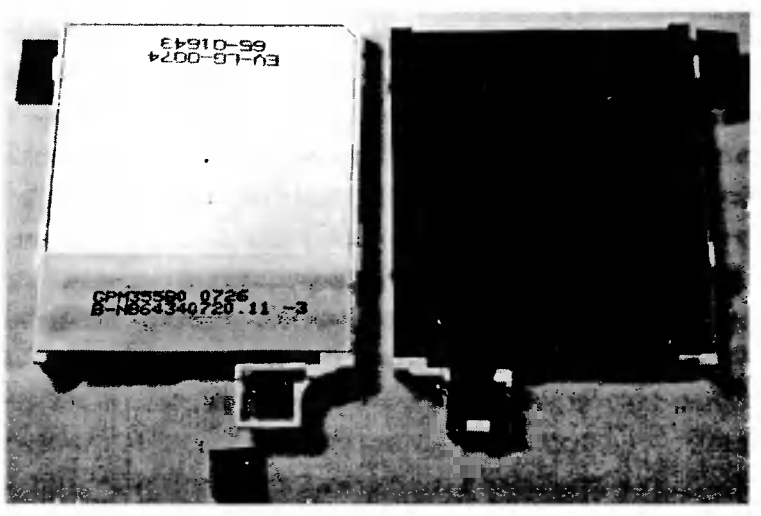


Рис. 2. Жидкокристаллический индикатор с контроллером S1D15G00 фирмы Epson

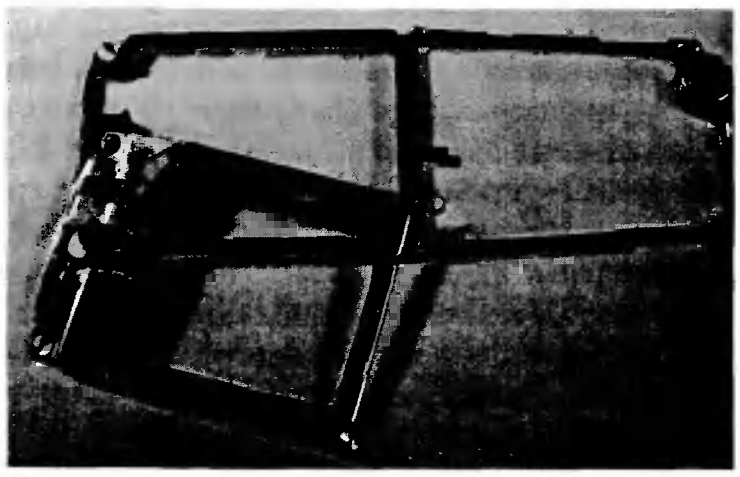


Рис. 3. Рамка для крепления жидкокристаллического индикатора к печатной плате

Нижняя часть рамки — пластмассовая. Она, кроме стягивающей рамки индикатора, имеет еще и рамку для крепления клавиатуры телефона и гнездо для крепления микрофона (на рис. 3 справа). Вторая верхняя часть рамки — металлическая с резиновым амортизатором. При использовании этой рамки для крепления индикатора в микроконтроллерных системах нижнюю рамку можно укоротить справа до крепежных отверстий (при этом она получится такого же размера, как и верхняя металлическая рамка). На рис. 4 показан жидкокристаллический индикатор с крепежной рамкой, укрепленный на печатной плате контроллера.

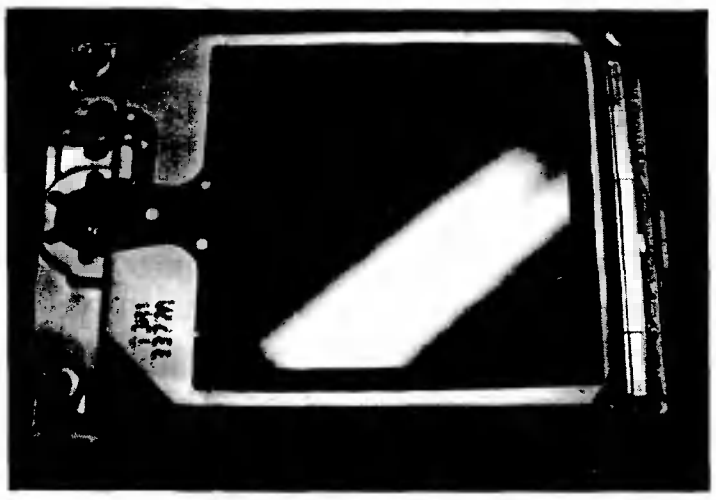


Рис. 4. Жидкокристаллический индикатор с крепежной рамкой, укрепленный на печатной плате контроллера

Многих разработчиков микроконтроллерных систем, конечно же, заинтересует вопрос о достижимом предельном (максимальном) быстродействии описываемых индикаторов. Так вот, внимательное ознакомление с временными параметрами и системой команд обоих основных контроллеров позволяет посчитать максимальное быстродействие при последовательном формировании полных кадров изображения с размерами 128×128 пикселей. Время формирования полного кадра изображения на максимальной скорости составляет примерно 36 мкс. Это означает, что за одну секунду может быть полностью обновлено примерно 27,7 кадров. При этом следует подчеркнуть, что расчет производился при последовательной прорисовке кадра из массива оперативной памяти, в которой этот кадр хранился, и совершенно не учитывались никакие другие операции. Кроме того, при расчете считалось, что используется однобайтное цветовое кодирование, т. е. каждый пиксель задается одним байтом.

Отметим, что в реальных микроконтроллерных системах такое быстродействие довольно сложно достичь. Во-первых, для хранения одного кадра изображения необходимо достаточно много памяти. Очевидно, что эта величина составляет 128×128 байтов или 16 Кбайт, что довольно значительная величина для микроконтроллерных систем. Во-вторых, алгоритм передачи данных многих современных микроконтроллеров предполагает сначала чтение данных из оперативной памяти (например, в аккумулятор), а затем — передачу байта данных в жидкокристаллический индикатор. Для повышения быстродействия желательно использование механизма прямого доступа в память (DMA — Direct Memory Access), который является достаточно большой редкостью в микроконтроллерных системах. Если система не использует механизм DMA, при расчете максимального быстродействия необходимо еще учитывать время, необходимое для операций модификации двухбайтного адреса данных. Таким образом, в микроконтроллерных системах, не имеющих аппаратных ускорителей (например, DMA), реально достижима частота смены полных кадров изображения примерно 15 Гц (15 кадров в секунду).

На рис. 1 и 2 хорошо видны кабельная и платная части соединительного разъема. Следует отметить, что платная часть разъема не имеет ключа. Если смотреть на экран дисплея спереди, то нумерация контактов при счете сверху вниз будет 1—5 в левом вертикальном ряду и 10—6 в правом вертикальном ряду. Нумерация контактов показана также на рис. 5.

Назначение выводов соединительного разъема показано в табл. 1.

Отметим, что на практике выводы 1 и 6, как правило, объединяются.

Описываемые жидкокристаллические индикаторы имеют встроенные светодиоды подсветки, расположенные с обратной стороны индикатора под пластиковым корпусом с зеркальным покрытием. Используется четыре последо-

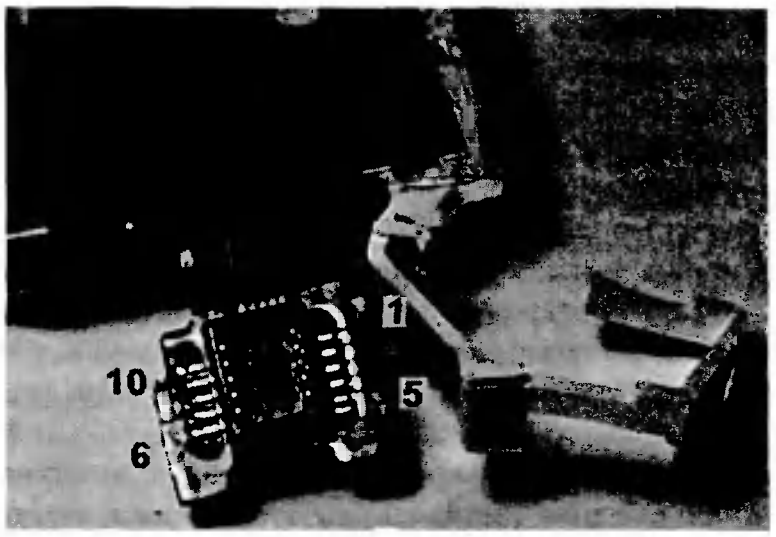


Рис. 5. Нумерация контактов соединительного разъема

Таблица 1

Номер вывода	Название связи	Назначение связи
1	Digital Power	Напряжение питания контроллера от +2,7 до +3,3 В
2	Reset/	Линия сброса (активный уровень «0»)
3	Serial Data	Линия данных
4	Serial Clock	Линия тактирования
5	Chip Select/	Линия выборки кристалла (активный уровень «0»)
6	Display Power	Напряжение питания индикатора от +2,7 до +3,3 В
7	NC	Вывод не используется
8	GND	Общий вывод
9	LED–	Минусовой вывод светодиодов подсветки
10	LED+	Плюсовой вывод светодиодов подсветки

вательно соединенных светодиода. Рекомендуемое значение тока — 15...19 мА при напряжении около 5,8...6,2 В. Очевидно, что это напряжение достаточно «неудобное» для микроконтроллерных систем, поэтому для его генерации используются специальные преобразователи, которые обычно называют White — LED Drivers. Специализированные микросхемы для этих преобразователей выпускаются многими фирмами, например, фирма Maxim выпускает 38 типов микросхем [10], предназначенных для таких преобразователей. Однако многие производители мобильных телефонов используют более дешевые и хорошо себя зарекомендовавшие микросхемы TK11850/TK11851 фирмы Toko [11]. Типовая схема включения для указанных микросхем и наших индикаторов приведена на рис. 6. Вход LCD_LED позволяет отключать подсветку индикатора («1» — включено). Ток подсветки задается с помощью резистора R1, а частота преобразования — емкостью конденсатора C1. Указанные на схеме номиналы рекомендованы производителем для жидкокристаллических индикаторов с четырьмя последовательными светодиодами подсветки, рассчитанными на ток примерно 20 мА. Микросхема TK11851 является дальнейшей модификацией использованной на рис. 6 микросхемы TK11850. Отличие заключается в том, что резистор R2 и диод VD1 на рис. 6, выполнены в интегральном исполнении внутри микросхемы TK11851. Иными словами, при использовании микросхемы TK11851 элементы R2 и VD1 устанавливать не нужно.

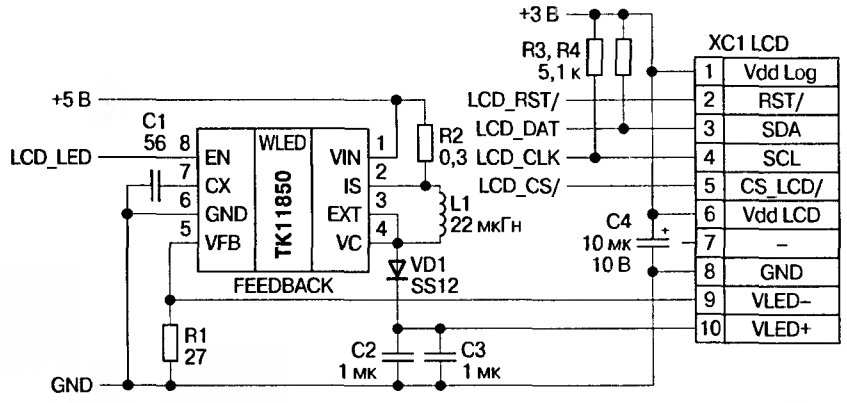


Рис. 6. Подсистема LCD на микросхеме TK11850

Рассмотрим основные принципы программирования жидкокристаллических индикаторов на примере совместной работы с микроконтроллерами фирмы Silicon Laboratories серии C8051F342[12], написанные на языке C для компилятора фирмы Keil [13].

Как уже отмечалось выше, описываемые жидкокристаллические индикаторы имеют разные системы команд. Обычно систему команд каждого из контроллеров описывают в отдельном файле описаний.

Файл описаний PCF8833.h системы команд контроллера PCF8833 от Phillips [8] выглядит следующим образом:

```
CODE
// ************************************************
// File PCF8833.h
// ************************************************

#ifdef _PCF8833_
// ************************************************
#define   NOP                                0x00 // None
#define SOFT_RESET                   0x01 // None
#define BOOSTER_OFF                  0x02 // None
#define BOOSTER_ON          0x03 // None
#define READ_ID                      0x04 // 4 Bytes Read
#define READ_STATUS                  0x09 // 4 Bytes Read
#define SLEEP_IN                     0x10 // None
#define SLEEP_OUT                    0x11 // None
#define PARTIAL_ON                   0x12 // None
#define DISPLAY_NORMAL               0x13 // None
#define DISPLAY_INV_OFF              0x20 // None
#define DISPLAY_INVERSE              0x21 // None
#define ALL_PIXELS_OFF               0x22 // None
#define ALL_PIXELS_ON                0x23 // None
#define SET_CONTRAST                 0x25 // 1 Byte
#define DISPLAY_OFF                  0x28 // None
#define DISPLAY_ON                   0x29 // None
#define SET_COLUMN_ADDRESS  0x2A // 2 Bytes
#define SET_PAGE_ADDRESS    0x2B // 2 Bytes
#define WRITING_TO_MEMORY   0x2C // 1 Byte
#define RGB_SET                      0x2D // 20 Bytes
#define PARTIAL_AREA                 0x30 // 2 Bytes
#define VERTICAL_SCROLL_AREA0x33 // 3 Bytes
#define TEARING_LINE_OFF             0x34 // None
#define TEARING_LINE_ON              0x35 // 1 Byte
#define MEM_CONTROL                  0x36 // 1 Byte
#define ROLLING_SCROLL_MODE 0x37 // 1 Byte
#define IDLE_MODE_OFF                0x38 // None
#define IDLE_MODE_ON                 0x39 // None
#define COLOR_INTERFACE              0x3A // 1 Byte
#define SET_VOP                      0xB0 // 2 Bytes
#define BOTTOM_ROW_SWAP     0xB4 // None
#define TOP_ROW_SWAP                 0xB6 // None
#define FRAME_INVERSION              0xB9 // None
#define DATA_ORDER          0xBA // None
#define EN_DIS_TEMP_CONTROL 0xBD // None
#define EN_DIS_VOP_TEMP              0xBF // None
#define INT_EXT_OSC                  0xC0 // None
#define SET_MULTIPLICATION  0xC2 // 1 Byte
#define SET_SLOPES_A_B               0xC3// 1 Byte
#define SET_SLOPES_C_D               0xC4 // 1 Byte
#define SET_DIVIDER                        0xC5 // 4 Bytes
#define SET_DIVIDER_8                0xC6 // 1 Byte
#define SET_BIAS                     0xC7 // 1 Byte
#define TEMP_READ_BACK               0xC8 // 1 Byte Read
#define NLINE_INVERSION              0xC9 // 1 Byte
#define READ_ID1                     0xDA // 1 Byte Read
#define READ_ID2                     0xDB // 1 Byte Read
#define READ_ID3                     0xDC // 1 Byte Read
#define FACTORY_DEFAULT     0xEF // None
#define CALIBRATION                  0xF0 // None
// ************************************************
#endif // _PCF8833_
```

Файл описаний S1D15G00.h системы команд контроллера S1D15G00 от Epson [9]:

```
CODE
// ************************************************
// File S1D15G00.h
// ************************************************

#ifdef _S1D15G00_
// ************************************************
#define DISPLAY_ON                   0xAF // None
#define DISPLAY_OFF                  0xAE // None
#define DISPLAY_NORMAL               0xA6 // None
#define DISPLAY_INVERSE              0xA7 // None
#define COMMON_SCAN_DIRECTION 0xBB // 1 Byte
#define DISPLAY_CONTROL              0xCA // 3 Bytes
#define SLEEP_IN                     0x95 // None
#define SLEEP_OUT                    0x94 // None
#define SET_PAGE_ADDRESS    0x75 // 2 Bytes
#define SET_COLUMN_ADDRESS  0x15 // 2 Bytes
#define DATA_CONTROL                 0xBC // 3 Bytes
#define RGB_SET                      0xCE // 20 Bytes
#define WRITING_TO_MEMORY   0x5C // 1 Byte
#define READING_FROM_MEMORY 0x5D // 1 Byte
#define PARTIAL_DISPLAY_IN  0xA8 // 2 Bytes
```

```
#define PARTIAL_DISPLAY_OUT 0xA9 // None
#define READ_AND_MODIFY_WRITE 0xE0 // None
#define END                                    0xEE // None
#define SET_SCROLL_AREA              0xAA // 4 Bytes
#define INT_OSCILL_ON                0xD1 // None
#define INT_OSCILL_OFF               0xD2 // None
#define POWER_CONTROL                0x20 // 1 Byte
#define ELEC_VOLUME_CONTROL 0x81 // 2 Bytes
#define INC_ELEC_CONTROL    0xD6 // None
#define DEC_ELEC_CONTROL    0xD7 // None
#define SET_TEMP_GRADIENT   0x82 // 1 Byte
#define EEPROM_CONTROL               0xCD // 1 Byte
#define EEPROM_CONTROL_CONCEL 0xCC // None
#define EEPROM_WRITE                 0xFC // None
#define EEPROM_READ                  0xFD // None
#define READ_REGISTER_1              0x7C // None
#define READ_REGISTER_2              0x7D // None
#define NOP                                    0x25 // None
// ***********************************************
#endif // _S1D15G00_
```

Для организации интерфейса с жидкокристаллическими индикаторами обычно используется 5 линий микроконтроллера. Для выбранного микроконтроллера C8051F342 произведем следующие определения:

```
CODE
sbit     LCD_RST_Bit       = P1^0;
sbit     LCD_DAT_Bit       = P1^1;
sbit     LCD_CLK_Bit       = P1^2;
sbit     LCD_CS_Bit        = P1^3;
sbit     LCD_LED_Bit       = P1^4;
```

Номера линий и номер порта ввода/вывода могут быть выбраны произвольно. Эти определения используются для подпрограмм доступа к назначенным входам/выходам. Следует помнить, что эти подпрограммы должны быть в том же программном модуле, в котором приведены соответствующие sbit определения.

```
CODE
void     LCD_CS   (bit A)  {LCD_CS_Bit=A;}
void     LCD_RST  (bit A)  {LCD_RST_Bit=A;}
void     LCD_CLK  (bit A)  {LCD_CLK_Bit=A;}
void     LCD_DAT_Set (bit A)        {LCD_DAT_Bit=A;}
bit      LCD_DAT_Get (void) {return LCD_DAT_Bit;}
void     LCD_LED  (bit A)  {LCD_LED_Bit=A;}
```

Подпрограммы нижнего уровня одинаковы для обоих типов контроллеров. Следует отметить, что эти подпрограммы используют две внешние подпрограммы: WDT() — подпрограмма перезапуска охранного таймера (если он используется) и Time (unsigned uS) — подпрограмма задержки примерно на uS мкс.

```
CODE
// ***********************************************
// File Low_Level_Functions.C
// ***********************************************
extern void       WDT (void);
extern void        Time (unsigned uS);

extern void       LCD_CS (bit A);
extern void       LCD_RST (bit A);
extern void       LCD_CLK (bit A);
extern void       LCD_DAT_Set (bit A);
extern bit        LCD_DAT_Get (void);
extern void       LCD_LED (bit A);
```

```
// ************************************
// Подпрограмма выполняет сброс LCD контроллера
// ************************************
void LCD_Reset (void)
{
 LCD_CS (1);               // Установить CS
 LCD_RST (1);              // Установить RST
 LCD_CLK (1);              // Установить CLK
 LCD_DAT_Set (1);          // Установить данные
 LCD_CLK (0);              // Сбросить CLK
 Time (200);                       // Задержка примерно 200
мкс
 LCD_RST (0);              // Сбросить RST
 Time (10);                        // Задержка примерно 10
мкс
 LCD_RST (1);              // Установить RST
}

// ************************************
// Генерация тактового импульса
// ************************************
void LCD_Pulse (void)
{
 _nop_();                  // Задержка на 1 такт MCU примерно
50 нс
 LCD_CLK(1);               // Установить CLK
 _nop_();                  // Задержка на 1 такт MCU примерно
50 нс
 LCD_CLK(0);               // Сбросить CLK
}


// ************************************
// Пересылка одного бита
// ************************************
void LCD_BitSend (bit B)
{
 _nop_();                  // Задержка на 1 такт MCU примерно
50 нс
 LCD_DAT_Set (B);          // Установить бит данных
 LCD_Pulse ();             // Сгенерировать тактовый импульс
}

// ************************************
// Пересылка байта в LCD
// Если CB=0, пересыка команды
// Если CB=1, пересылка данных
// ************************************
void LCD_Send (byte DAT, bit CB)
{
register byte i;
register byte mask = 0x80;

 LCD_CS (0);               // Сбросить CS
 _nop_();                  // Задержка на 1 такт MCU примерно
50 нс
 LCD_BitSend(CB);          // Переслать бит признака операции
      // «0» - если далее следует байт команды
      // «1» - если далее следует байт данных
 for (i=0;i<8;i++)
 {
  LCD_BitSend ( (DAT&mask)?1:0 ); // Переслать очередной бит
  mask=mask>>1;            // Модифицировать маску (бит)

 }
 LCD_CS (1);               // Установить CS
}

// Переслать байт команды
void LCD_COM (byte COM)    {LCD_Send(COM,0);}

// Переслать байт данных
void LCD_DAT (byte DAT)    {LCD_Send(DAT,1);}
```

```
// ************************************
// Чтение данных из LCD
// ************************************
byte LCD_Read (void)
{
register byte i;
register byte mask = 0x80;
register byte REZ=0;

 LCD_CS (0);                // Сбросить CS
 _nop_();                   // Задержка на 1 такт MCU примерно
50 нс
 for (i=0;i<8;i++)
 {
  LCD_CLK(1);               // Установить CLK
  _nop_();                          // Задержка на 1 такт MCU
примерно 50 нс
  if (LCD_DAT_Get())        // Взять бит данных
  REZ|=mask;
  mask=mask>>1;             // Модифицировать маску (бит)
 }
 LCD_CS (1);                // Установить CS
 return REZ;                        // Вернуть байт данных
}
```

К сожалению, на этом общее в программах для различных жидкокристаллических индикаторов, т .е. индикаторов с различными контроллерами, заканчивается. Все остальные команды более высокого уровня располагаются в индивидуальных файлах.

Рассмотрим файл, содержащий файл GraphLCD.h, с помощью которого определяются цвета некоторых величин и функции инициализации, одинаковый для обоих контроллеров:

```
CODE
 / ************************************************
// File GraphLCD.h
// ************************************************

// ВЫБОР ТИПА ИСПОЛЬЗУЕМОГО КОНТРОЛЛЕРА LCD

#define           _S1D15G00_ // Выбран LCD с контроллером EP-
SON
//#define _PCF8833_

// ************************************************
#ifdef            _S1D15G00_

#include "S1D15G00.h"

#else

#include "PCF8833.h"

#endif
// ************************************************



// Определение типа byte
#ifndef __BYTE__
#define __BYTE__
typedef unsigned char       byte;
#endif /* __BYTE__ */

// ************************************************
// Основные макроопределния
// ************************************************

#define LCD_Mem_Write()     LCD_COM(WRITING_TO_MEMORY)
#define LCD_Clear()                 LCD_FillRect (0,0,SCREEN_
WIDTH-1,SCREEN_HEIGHT-1,BG_COLOR)
```

```
// ************************************************
// Определения основных цветов
// ************************************************
#define RGB(r,g,b)                  ((r&0xE0)|((g&0xE0)>>3)|(b
>>6))
#define NONE       RGB(0x00, 0x20, 0x00)
#define BLACK      RGB(0x00, 0x00, 0x00)
#define WHITE      RGB(0xFF, 0xFF, 0xFF)
#define RED        RGB(0xFF, 0x00, 0x00)
#define GREEN      RGB(0x00, 0xFF, 0x00)
#define BLUE       RGB(0x00, 0x00, 0xFF)
#define YELLOW     RGB(0xFF, 0xFF, 0x00)
#define MAGENTA             RGB(0xFF, 0x00, 0xFF)
#define CYAN       RGB(0x00, 0xFF, 0xFF)
#define GRAY       RGB(0x80, 0x80, 0x40)
#define SILVER     RGB(0xA0, 0xA0, 0x80)
#define GOLD       RGB(0xA0, 0xA0, 0x40)

// ************************************************
// Определения основных размеров
// ************************************************
#define SCREEN_WIDTH        132
#define SCREEN_HEIGHT       132
#define SCREEN_LEFT         4
#define SCREEN_TOP          4
#define SCREEN_RIGHT        (SCREEN_WIDTH-4)
#define SCREEN_BOTTOM (SCREEN_HEIGHT-4)
// ************************************************
// Внешние подпрограммы
// ************************************************
extern void LCD_CS                  (bit A);
extern void LCD_RST         (bit A);
extern void LCD_CLK         (bit A);
extern void LCD_DAT_Set     (bit A);
extern bit LCD_DAT_Get      (void);
extern void LCD_LED         (bit A);
// ************************************************
extern void Time            (unsigned mkS);
extern void Delay           (unsigned mS);
extern void WDT             (void);
// ************************************************
extern void LCD_Reset       (void);
extern void LCD_Pulse       (void);
extern void LCD_BitSend     (bit B);
extern void LCD_Send        (byte DAT, bit CB);
extern byte LCD_Read        (void);
extern void LCD_COM         (byte COM);
extern void LCD_DAT         (byte DAT);
// ************************************************
void LCD_Init (void);
```

Далее приводится файл PCF8833.C, содержащий подпрограммы, необходимые для инициализации LCD с контроллером от Phillips:

```
CODE
// ************************************
// * PCF8833.C
// ************************************
#include "GraphLCD.h"

#ifdef _PCF8833_
// ************************************************
#define LCD_NOP()                   LCD_COM(NOP)
#define LCD_Soft_Reset()            LCD_COM(SOFT_RESET)
#define LCD_Booster_ON()            LCD_COM(BOOSTER_ON)
#define LCD_Booster_OFF()           LCD_COM(BOOSTER_OFF)
#define LCD_ON()                    LCD_COM(DISPLAY_ON)
#define LCD_OFF()                   LCD_COM(DISPLAY_OFF)
#define LCD_Sleep_In()              LCD_COM(SLEEP_IN)
#define LCD_Sleep_Out()             LCD_COM(SLEEP_OUT)
#define LCD_Idle_Mode_OFF()         LCD_COM(IDLE_MODE_OFF)
```

```
#define LCD_Idle_Mode_ON()  LCD_COM(IDLE_MODE_ON)
#define LCD_All_Pixels_OFF()          LCD_COM(ALL_PIXELS_OFF)
#define LCD_All_Pixels_ON()           LCD_COM(ALL_PIXELS_ON)
#define LCD_Normal_Mode()             LCD_COM(DISPLAY_NORMAL)
#define LCD_Inverse_Mode()            LCD_COM(DISPLAY_INVERSE)
#define LCD_Inverse_OFF()             LCD_COM(DISPLAY_INV_OFF)
#define LCD_Frame_Inversion()         LCD_COM(FRAME_INVERSION)
#define LCD_Tearing_Line_OFF()        LCD_COM(TEARING_LINE_OFF)
#define LCD_Partial_Mode_ON()         LCD_COM(PARTIAL_ON)
#define LCD_Oscill(BIT)               LCD_COM(INT_EXT_OSC|BIT)
#define LCD_Data_Order(BIT)           LCD_COM(DATA_ORDER|BIT)
#define LCD_Factory_Default(BIT)      LCD_COM(FACTORY_
DEFAULT|BIT)
#define LCD_Top_Row_Swap(BIT)         LCD_COM(TOP_ROW_SWAP|BIT)
#define LCD_Bottom_Row_Swap(BIT) LCD_COM(BOTTOM_ROW_SWAP|BIT)
#define LCD_Temp_VQP(BIT)   LCD_COM(EN_DIS_VQP_TEMP|BIT)
#define LCD_Temp_Control(BIT)         LCD_COM(EN_DIS_TEMP_
CONTROL|BIT)

// ***********************************************
// Выбор режима 8, 12 или 16 бит / пиксель
// ***********************************************
static void LCD_Color_Interface (byte COL)
{
register byte ch=0;

 switch(COL)
 {
  case 12:        ch=0x03;
  break;
  case 16: ch=0x05;
  break;
  default:
  case 8: ch=0x02;
  break;
 }
 LCD_COM (COLOR_INTERFACE);
 LCD_DAT (ch);
}

// ***********************************************
// Установка кода контраста
// ***********************************************
static void LCD_Set_Contrast (byte CONTR)
{
 LCD_COM (SET_CONTRAST);
 LCD_DAT (CONTR);
}
// ***********************************************
// Установка направлений заполнения экрана
// ***********************************************
static void LCD_Memory_Control (bit MirrorY, bit MirrorX, bit
DirY, bit LAO_BT, bit BGR)
{
register byte ch=0;

 if (MirrorY)     ch|=0x80;
 if (MirrorX) ch|=0x40;
 if (DirY) ch|=0x20;
 if (LAO_BT) ch|=0x10;
 if (BGR) ch|=0x08;
 LCD_COM (MEM_CONTROL);
 LCD_DAT (ch);
}

// ***********************************************
// Подпрограмма инициализации
// ***********************************************
void LCD_Init (void)
{
 LCD_Reset ();                    // Аппаратный сброс LCD
 LCD_Soft_Reset();                // Программный сброс LCD
 LCD_Sleep_Out();                 // Активировать LCD
 LCD_ON();                              // Включить LCD
 LCD_Booster_ON();          // Включить «горячий» старт
 LCD_Memory_Control (0,1,0,0,0); // Задать режим 8-битного RGB
 Delay (200);               // Задержка примерно 200 мс
 LCD_Set_Contrast (65);     // Установить рекомендованное
          // значение контраста
 LCD_Color_Interface (8);   // Интерфейс цветности - 8 бит
 LCD_8Bit_Palette_Set();    // Установить 8 битную палитру
 LINE_WIDTH=1;              // Установить текущую толщину
          // линии для рисования
 LCD_Set_BG_Color (WHITE); // Установить текущий цвет фона
 LCD_Clear();               // Очистить экран
}
// ***********************************************
#endif // _PCF8833_
```

Аналогичный файл S1D15G00.C для инициализации LCD с контроллером от Epson выглядит следующим образом:

```
CODE
 / ************************************
// * S1D15G00.C
// ************************************
#include "GraphLCD.h"

#ifdef            _S1D15G00_

// ***********************************************
#define           LCD_ON()          LCD_COM(DISPLAY_ON)
#define           LCD_OFF()         LCD_COM(DISPLAY_OFF)
#define           LCD_Normal_Mode()  LCD_COM(DISPLAY_NORMAL)
#define           LCD_Inverse_Mode() LCD_COM(DISPLAY_INVERSE)
#define           LCD_Sleep_In()    LCD_COM(SLEEP_IN)
#define           LCD_Sleep_Out()   LCD_COM(SLEEP_OUT)

#define           LCD_Oscill_ON()   LCD_COM(INT_OSCILL_ON)
#define           LCD_Oscill_OFF()  LCD_COM(INT_OSCILL_OFF)
#define           LCD_END()         LCD_COM(END)
#define           LCD_NOP()         LCD_COM(NOP)
#define           LCD_Read_RG1()    LCD_COM(READ_REGISTER_1)
#define           LCD_Read_RG2()    LCD_COM(READ_REGISTER_2)
#define           LCD_Elec_Control_Inc() LCD_COM(INC_ELEC_CON-
TROL)
#define           LCD_Elec_Control_Dec() LCD_COM(DEC_ELEC_CON-
TROL)
#define           LCD_Partial_Display_Out() LCD_COM(PARTIAL_
DISPLAY_OUT)
#define           LCD_Read_and_Modify_Write() LCD_COM(READ_AND_
MODIFY_WRITE)
#define           LCD_EEPROM_Control_Concel() LCD_COM(EEPROM_
CONTROL_CONCEL)
#define           LCD_EEPROM_Write() LCD_COM(EEPROM_WRITE)
#define           LCD_EEPROM_Read() LCD_COM(EEPROM_READ)

// ***********************************************
// Установка частотного режима
// ***********************************************
static void LCD_Display_Control (byte DIV, byte DUTY, byte FRINV)
{
 LCD_COM (DISPLAY_CONTROL); // Установка времен задержки и
          // частот, необходимых для работы
 LCD_DAT (DIV);                              // Передать код
делителя частоты
 LCD_DAT (DUTY);                             // Передать ко-
эффициент заполнения
 LCD_DAT (FRINV);                  // Установить флаг инвер-
сии
 LCD_DAT (0x00);                             // Завершить пе-
редачу частотных
          // параметров
}
```

```
// ************************************************
// Установить направление сканирования выводов контроллера
// ************************************************
static void LCD_Pin_Scan_Direction (byte B)
{
 LCD_COM (COMMON_SCAN_DIRECTION); // Установка направления
 LCD_DAT (B);                                  // Выбрать байт
направления
}

// ************************************************
// Установить яркость и контрастность
// ************************************************
static void LCD_Electric_Control (byte CONTRAST, byte BRIGHT-
NESS)
{
 LCD_COM (ELEC_VOLUME_CONTROL); // Установить значения
 LCD_DAT (CONTRAST);                            // Рекомендуе-
мый контраст =5
 LCD_DAT (BRIGHTNESS);              // Рекомендуемая яркость
=2
}

// ************************************************
// Установить параметры питания
// ************************************************
static void LCD_Power_Control (byte B)
{
 LCD_COM (POWER_CONTROL);   // Включить все цепи питания LCD
 LCD_DAT (B);                          // Передать код
}

// ************************************************
// Установить эксплуатационные параметры
// ************************************************
static void LCD_Data_Control_Set (byte PAGE_N_I, byte COLUMN_N_
I, byte DIR_C_P,
        byte RGB_Type, byte GRAY_SCALE)
{
register byte ch=0;

 LCD_COM (DATA_CONTROL);    // Установка параметров
 if (PAGE_N_I)     ch|=0x01; // Направление рядов
 if (COLUMN_N_I)   ch|=0x02; // Направление столбцов
 if (DIR_C_P)      ch|=0x04; // Направление заполнения

 LCD_DAT (ch);                          // Передать данные
 LCD_DAT (RGB_Type);                    // Задать цветность
 LCD_DAT (GRAY_SCALE);      // Задать градации серого
 LCD_DAT (0x00);                        // Завершить передачу

}

// ************************************************
// Подпрограмма инициализации
// ************************************************
void LCD_Init (void)
{
 LCD_Reset ();                          // Сбросить LCD
 LCD_Display_Control(0x03,0x20,0x0c); // Установить временные
параметры
 LCD_Pin_Scan_Direction (0x01);       // Установить параметры
сканирования выводов
 LCD_Oscill_ON();                       // Включить внутренний
генератор
 LCD_Sleep_Out();                       // Активировать LCD
 LCD_Electric_Control (0x3f,0x01);    // Установить рекомендуе-
мые яркость и контраст
 LCD_Power_Control (0x0F);              // Включить цепи питания
 Delay (100);                           // Задержка примерно 100 мс
 LCD_Inverse_Mode();                    // Включить инверсный режим
 LCD_Data_Control_Set(0,1,0,0,0x01); // Установить параметры
данных
 LCD_8Bit_Palette_Set();                // Установить палитру
 LCD_ON();                              // Включить LCD
 Delay (200);                           // Задержка примерно 200
мс
 LINE_WIDTH=1;                          // Установить текущую тол-
щину
          // линии для рисования
 LCD_Set_BG_Color (WHITE);  // Установить текущий цвет фона
 LCD_Clear();      // Очистить экран
}
// ************************************************
#endif //          _S1D15G00_
```

После проведения инициализации дисплей полностью готов к работе. Теперь пользователь может с помощью простейших функций строить различные изображения. В заключение приведем еще две простейшие функции, которые позволят разработчику быстро создать (нарисовать) простейшие изображения:

```
CODE
// ************************************************
// Подпрограмма задания адреса точки или области
// ************************************************
void LCD_SetAddr (byte x1, byte y1, byte x2, byte y2)
{
 LCD_COM(SET_PAGE_ADDRESS); // Передать код задания
                            // диапазона по строке
                            // величина X1 должна быть меньше
X2
 LCD_DAT(X1);               // Адрес начала зоны
 LCD_DAT(X2);               // Адрес конца зоны
 LCD_COM(SET_COLUMN_ADDRESS);        // Передать код задания
                                     // диапазона по столбцу
                                                  // ве-
личина Y1 должна быть меньше Y2
 LCD_DAT(Y1);               // Адрес начала зона
 LCD_DAT(Y2);               // Адрес конца зоны
}

// ************************************************
// Подпрограмма вывода цветовой точки
// ************************************************
void LCD_PutPixel (byte x, byte y, byte color)
{
 LCD_SetAddr (x, y, x+1, y+1);       // Задать координаты точки
 LCD_Mem_Write ();                   // Передать код записи в
память
 LCD_DAT (color);                    // Записать байт цвета
}
```

Автор надеется, что статья будет полезна многим разработчикам микроконтроллерных систем, использующих цветные графические жидкокристаллические индикаторы от мобильных телефонов.

Олег Николайчук,
г. Кишинев, Молдова

ЛИТЕРАТУРА:

1. *http://www.datavision.com.tw*
2. *http://www.melt.com.ru/*
3. *http://www.intech-lcd.com*
4. *http://www.ampire.com.tw*
5. *http://www.bolymin.com.tw*
6. *http://www.huton-cellphone.com/lcd/ML019-001091.htm*
7. *http://www.huton-cellphone.com/lcd/ML020-001092.htm*
8. *http://www.nxp.com/acrobat_download/datasheets/PCF8833_1.pdf*

http://www.shellyinc.com/techsup/S1D15G00_REV1_0.pdf

Не слишком ли дорого обходятся дешевые компьютерные блоки питания?

Незамеченная революция в компьютерных блоках питания

Компьютерные блоки питания (БП) нового поколения появились несколько лет назад, сейчас уже имеются их улучшенные варианты и совершенствование продолжается. Но для большинства «обычных» пользователей это событие прошло незамеченным и они по привычке выбирают дешевые «классические» БП, исходя из принципа «работают, ну и ладно». Увы, но такая экономия может обойтись слишком дорого. Дело в том, что БП нового поколения наряду с другими достоинствами имеют одно принципиально важное — с ними компьютеры не выгорают.

Истину, что скупой платит дважды, сегодня знают все. Правда, в основном, теоретически, поскольку при ее применении на практике возникают некоторые проблемы как объективного, так и чисто субъективного свойства. В том числе и по причине отсутствия надежной информации.

Все компьютерные комплектующие можно разделить на три группы:

- на которых можно и нужно экономить;
- на которых экономить можно, но осторожно;
- на которых ни в коем случае нельзя экономить.

Покажем это на примерах. Самым ценным в компьютере безусловно является винчестер и попытки сэкономить на нем могут привести к обратным результатам, ведь восстановление данных с «погибшего» винчестера обходится существенно дороже нового, если это вообще удастся сделать. А вот экономить на корпусе не только можно, но и нужно, ведь для компьютера он не является необходимым, поскольку его без проблем можно собрать и на стенке, приколотив к ней гвоздями, а дисковод, винчестер и приводы CD разместить рядом на полочке. При этом компьютер будет работать даже лучше, потому что условия охлаждения в таком варианте лучше, чем в корпусе. Корпус — это некий конструктив, позволяющий собрать удобную и относительно компактную конструкцию. И для компьютера неважно, будет ли он выполнен из оцинкованного кровельного железа или из трехмиллиметровой нержавейки. Конечно, к корпусу есть еще вполне определенные требования по конструкции и внешнему виду, как объективные, так и чисто субъективные, определяемые вкусом владельца, но это уже вопрос дизайна, который от применяемого материала практически не зависит. Поэтому платить за лишние килограммы, которые потом еще и мешать будут, вряд ли целесообразно. Впрочем, сбыт и таких корпусов практически обеспечен, поскольку всегда найдутся люди, у которых количество денег превышает количество знаний по компьютерному железу.

Примерно такое же мнение было у автора и о компьютерных блоках питания. Действительно, схемотехника и конструкции БП были хорошо отработаны за многие годы массового производства. Все они выполнялись на основе давно ставшей классической микросхемы ШИМ-контроллера типа TL494 (или ее разновидностей). Разработанная фирмой Texas Instruments микросхема оказалась настолько удачной, что стала тормозить создание новых микросхем ШИМ-контроллеров, поскольку разработчики занимались в основном ее совершенствованием, внося непринципиальные улучшения. Схемы и конструкции БП разных фирм также имели лишь небольшие и непринципиальные различия. Дальнейшее их упрощение стало практически невозможным (разве что «замена» в цепи фильтров сетевого питания дросселей на перемычки, а конденсаторов на дырки в печатной плате). Впрочем, и блоки с такими «заменами» работали вполне нормально, а большинство пользователей даже не подозревали об этом. «Дорогие» БП отличались от своих дешевых собратьев в основном лишь более сложными фильтрами в цепях питания и несколько лучшим качеством изготовления, т. е. тем, что обычный пользователь в принципе заметить не в состоянии.

Но время шло, технический прогресс также не стоял на месте, и были созданы микросхемы ШИМ контроллеров нового поколения. Затем уже на их основе были созданы и БП нового поколения, существенно отличающиеся от «старых» БП. Цена их, естественно, была несколько выше, чем у «классических».

Однако для большинства пользователей, в том числе и для автора, эта тихая «революция» в компьютерных БП по вполне понятным причинам осталась незамеченной, и для них оставалось неясным, за что же платить лишние деньги. Только когда у автора случайно появились несколько неисправных БП нового поколения, и в процессе ремонта пришлось познакомиться с ними достаточно близко, автору стало ясно, что он не настолько богат, чтобы пользоваться старыми БП. Конечно те, кто зарабатывает сборкой компьютеров (назвать их профи язык не поворачивается, поскольку такой профессии не существует, как и профессии землекопа) и особо любознательные пользователи безусловно заметили не только появление более дорогих БП, но и довольно четкое разделение их на три ценовые группы. По аналогии с существовавшим в СССР разделением аппаратуры на классы, их также можно обозначить как 1, 2 и 3 класс. К классу 3 относятся БП старого поколения и с ними все достаточно ясно. Блоки класса 1 ввиду специфики потребителей этой продукции, о чем говорилось выше, оказываются как бы «вне конкуренции», что позволяет резко завышать цены. Для «обычных» потребителей продукция этого класса интереса не представляет и рассматривать ее не имеет смысла. А вот о блоках, которые условно можно отнести ко классу 2, следует поговорить подробнее, поскольку именно они и относятся к БП нового поколения.

Конечно, привести в небольшой статье подробные описания и характеристики всех имеющихся сейчас в продаже блоков питания нового поколения — задача нереальная. Впрочем, в этом нет необходимости. Желающие могут найти их подробные описания и характеристики в Интернете. Можно лишь отметить, что усовершенствования в линейке БП нового поколения не являются принципиальными и в основном касаются примененной элементной базы. Поэтому рассмотрим лишь их основные особенности на примере БП типа Power Man производства фир-

мы In Win мощностью 300 Вт, которые сейчас уже можно считать классикой.

Эти БП выполнены по типовой полумостовой схеме с двумя силовыми ключевыми транзисторами и действительно нормально «тянут» такую мощность, в отличие от большинства дешевых блоков, который лишь маркируются как 300-ватные. Но главное, конечно, в том, что в них применены микросхемы ШИМ-контроллеров типа SG6105D, а в некоторых партиях блоков — аналогичные контроллеры собственного производства с маркировкой IW1688. Именно эта микросхема, принципиально отличающаяся от TL494, и обеспечивает принципиально новые качества БП. В первую очередь, по мнению автора, это относится к функциям защиты. Многие пользователи знают о случаях выгорания компьютеров. Ситуация, конечно, не из приятных, даже если такое произошло не у тебя, а у твоих знакомых. Причин выгорания может быть множество: броски сетевого напряжения, которые не смог «отработать» блок питания, возникшие неисправности в отдельных узлах и блоках компьютера, в том числе и в самом БП и т. п. Во всех этих случаях идеальный БП должен просто отключить питание компьютера, после чего остается только найти конкретную неисправность. Другими словами, БП, как хороший боксер, обязан «держать удар». У обычных блоков питания полную защиту обеспечивает лишь плавкий предохранитель, но до его сгорания компьютер успевает благополучно выгореть. В итоге даже определить истинную причину неисправности, т. е. с чего все началось, становится невозможным.

В этом отношении защитные функции блоков питания нового поколения принципиально иные, в первую очередь по скорости срабатывания защиты, хотя говорить об абсолютной защите все же рискованно.

Пример из практики: к компьютеру был подключен винчестер с коротким замыканием по цепи питания 12 В. После включения компьютера сгорел обычный БП и его заменили БП IN WIN, после чего компьютер просто перестал включаться. После замены винчестера компьютер работал нормально. Понятно, что те, кто столкнулся с подобным хотя бы раз, даже понаслышке, вряд ли уже захотят рисковать со старыми блоками питания. Ну а поскольку в функции БП стала входить и защита компьютера от выгорания, в том числе и винчестера, вполне понятно, что теперь именно они вышли на первое место по ценности, опередив по этому критерию даже винчестер. И экономия на блоках питания сейчас может оказаться наиболее дорогостоящей. На всякий случай стоит пояснить, что именно по ценности, а не по стоимости, поскольку это суть разные понятия.

Улучшенные функции защиты являются хотя и самым важным, но не единственным достоинством новых БП. Рассмотрим одну особенность, достаточно важную для компьютера. В обычных блоках питания стабилизируется напряжение +5 В (наличие аналоговых стабилизаторов в цепях –5 В или –12 В не суть важно). Это вполне понятно, поскольку цепь обратной связи может быть замкнута лишь по одному параметру. Поэтому изменения тока нагрузки по цепи +12 В не отрабатываются и изменение тока нагрузки в цепи +5 В приводит к изменению напряжения в цепи +12 В. Например, возрастание потребления тока в цепи +5 В при неизменном токе в цепи +12 В приведет к повышению напряжения в цепи +12 В. В блоках IN WIN обратная связь замыкается по усредненному значению напряжений +5 В и +12 В. Такое решение, конечно же, не является идеальным, но все же лучше обычного. Примененные в классической микросхеме TL494 (и ее аналогах) операционные усилители (ОУ), через которые замыкается цепь обратной связи, являются инерционными, поэтому бросок тока в цепи +5 В приводит к появлению противоположного по знаку скачка напряжения. Быстродействующие ОУ микросхемы SG6105D совместно с форсирующими конденсаторами в цепи обратной связи отрабатывают броски тока в нагрузке практически в течение одного периода импульсного преобразователя, т. е. за 15...30 мкс. За это время напряжение на конденсаторах в цепях питания не успевает существенно измениться, поэтому и броски напряжения появиться не могут. Это быстродействие примерно соответствует скорости выключения БП при появлении в системном блоке неисправностей по питанию.

Конечно, эти преимущества напрямую для пользователя также незаметны. Возникает вопрос: нужны ли они? Попробуем ответить.

Все пользователи так или иначе сталкивались с необъяснимыми случаями сбоев и зависаний компьютера. По традиции во всем принято винить Билли Гейтса, подсунувшему нам такую замечательную «винду». Однако реально часть таких сбоев является чисто аппаратной и вызвана именно бросками напряжения в цепях питания компьютера. Чем современней и сложней компьютер, тем чувствительней он к «качеству» питания.

Пожалуй единственное, что действительно может заметить пользователь в новых БП — это более тихие вентиляторы. На самом деле вентиляторы в них стоят вполне обычные, но в состав БП входит небольшая платка управления частотой вращения вентилятора. В блоках IN WIN критерием является потребляемая компьютером мощность, которая определяется по скважности импульсов во вторичной цепи питания. Чем больше нагрузка,– тем быстрее вращается вентилятор. В некоторых блоках других фирм критерием является температура радиатора, на котором установлены выпрямительные диоды. Стоит отметить, что наибольшая мощность выделяется именно на них, а не в силовых транзисторах, как это принято считать, хотя для ее снижения и применяют сборки на диодах Шотки. На первый взгляд такое решение кажется более удачным, поскольку действительно важно не допустить перегрева выпрямительных диодов. Но в силу инерционности тепловых процессов нормальная работа может быть обеспечена лишь при медленном изменении нагрузки. В этом отношении критерий потребляемой мощности позволяет работать «на опережение» — при резком возрастании потребляемой мощности быстро увеличивается и частота вращения вентилятора, «не дожидаясь», когда нагреются транзисторы (или диоды), а затем и радиатор. Понятно, что лучшим решением является совместное применение двух этих критериев и в некоторых современных БП действительно именно так и делается.

Конечно, некоторые пользователи могут возразить: «мой компьютер и со старым БП длительное время работает вполне нормально»... Действительно, как уже говорилось, заметить разницу в работе компьютера с разными БП, как, например, при замене видеокарты на более мощную, практически невозможно. И возразить на это действительно нечего, да и незачем — если работает нормально, то пусть работает и дальше. Надо просто еще раз пояснить, что все изложенное выше является чистой статистикой. Другими словами, это некая лотерея, выигрышем в которой может быть небольшая экономия на блоке питания. О проигрыше уже говорилось. И стоит ли играть в эту лотерею, решать только владельцу компьютера.

Александр Межлумян,
г. Москва

Преобразователь сигналов дифференциального емкостного датчика

Описывается преобразователь сигналов дифференциального емкостного датчика компенсационного типа, обеспечивающий существенную независимость от диэлектрической проницаемости, а также линеаризацию передаточной характеристики преобразователя при технологической нелинейности датчика.

Для измерения линейных и угловых перемещений в системах автоматизации технологических процессов применяют емкостные датчики, характеризующиеся высокой чувствительностью к измеряемому параметру. При этом питание датчика от импульсного источника напряжения не обеспечивает дистанционность измерений в силу повышенного влияния паразитных емкостей линии связи и требует применения выносных преобразователей. Преобразователь [1] с применением линейно изменяющихся напряжений питания датчика обеспечивает большую дистанционность измерений, однако он имеет метрологические характеристики, соответствующие методу прямого преобразования, с использованием которого он построен.

Известно, что в последовательно соединенной цепи датчик — измерительный преобразователь точность всего измерительного канала определяет датчик. Сделать точный измерительный канал — это значит сделать точный датчик. Дифференциальный емкостной датчик должен иметь достаточно хорошую линейность в функции от измеряемого параметра, что не всегда выполнимо при практической реализации. Например, в датчике перемещения с переменной площадью S для обеспечения хорошей линейности необходимо обеспечить постоянство зазора d и диэлектрической проницаемости ε, иначе обе эти величины будут искажать передаточную характеристику датчика. Функция преобразования преобразователя [1]

$$U = K(C2 - C1)$$

напрямую зависит от диэлектрической проницаемости ε, а следовательно и от ее изменений и, кроме того, «повторит» возможные технологические нелинейности фактического датчика, допущенные при его изготовлении.

Ниже описан преобразователь дифференциального емкостного датчика, лишенный указанных недостатков. Преимущества предлагаемого преобразователя достигаются построением преобразователя с функцией преобразования, которая обеспечивает существенную независимость от изменений окружающей среды и позволяет «сглаживать» технологические нелинейности фактического датчика, так как в точках нелинейности (т. X* на рис. 1) помимо нелинейной разности (C2 – C1) нелинейной будет и сумма (C1 + C2), а общая ПФ канала линеаризуется как результат деления нелинейности на нелинейность.

Для идеального датчика сумма (C1 + C2) = const и в ПФ она входит как коэффициент.

В состав преобразователя входят дифференциальный емкостной датчик, усилитель неравновесия (C3, R1, DA1.1), дискретный интегратор (C6, C9, DA4, DA1.2), масштабирующий функциональный преобразователь (МФП), выполненный на основе ОУ DA6, синхронные детекторы на DA5, формирователи линейно изменяющихся напряжений на ОУ DA2, источник опорного напряжения DA8 и цепь установки нуля на DA9.2. Указанные элементы образуют замкнутую систему с обратной связью, при этом сам датчик входит в петлю обратной связи.

Принцип действия преобразователя основан на уравновешивании токов, протекающих через емкости C1, C2 датчика. Токи имеют вид меандра и задаются симметричными треугольными напряжениями, формируемыми на выходах ОУ DA2. Разбаланс токов, вызванный изменением измеряемой величины X, обнаруживается усилителем неравновесия. Цепь уравновешивания токов включает усилитель неравновесия, синхронный детектор, интегратор. Масштабирующий функциональный преобразователь формирует два дифференциальных меняющихся напряжения:

$$U1 = к1 \times U0 \pm к2 \times U$$

$$U2 = -к1 \times U0 \pm к2 \times U,$$

где к1, к2 — масштабные коэффициенты МФП соответственно по опорному напряжению U0 и напряжению U интегратора. Знаки «+», «–» соответствуют положительным и отрицательным значениям измеряемой величины X.

Эти напряжения задают симметричные треугольные напряжения на выходах ОУ DA2. При этом в емкостях C1, C2 датчика протекают разнополярные токи.

По цепи ОС преобразователя интегратор развивает на своем выходе такое напряжение, при котором напряжение на его входе стремится к нулю, что достигается равенством токов в емкостях датчика.

При дифференциальных изменениях C1, C2, вызванных изменением X, схема автоматически поддерживает равенство токов независимо от изменений C1, C2. Результат прохождения опорного напряжения U0 на вход интегратора уравновешивается результатом прохождения на тот же вход напряжения U интегратора. При этом, поскольку уравновешивание происходит по цепи, являющейся общей как для опорного напряжения, так и напряжения интегратора, то стабильность преобразователя определяется стабильностью опорного источника и является очень высокой.

На элементах DD1, DD2 формируются последовательности импульсов W1, W2. Последовательность W1 обеспечивает работу ключей DA4, DA5. Последовательность W2 — детектирование разбаланса токов по окончании переходных процессов в линии и усилителе неравновесия.

Как и для всякой системы с ОС параметры преобразователя определяются петлевым усилением, в частности дополнительным преимуществом преобразователя следует считать его меньшую чувствительность к дестабилизирующим факторам (например, температуре) как результат реализации преобразователя с замкнутой ОС.

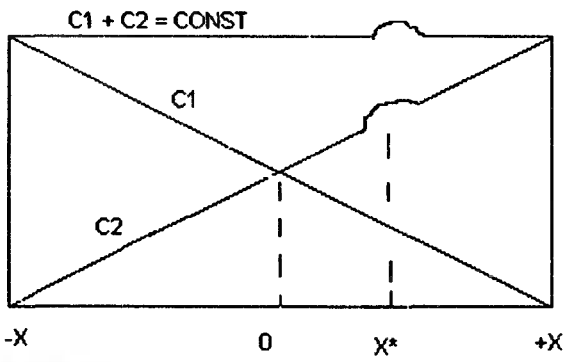


Рис. 1

Кроме того, датчик, следовательно, и его кабели подключения находятся в петле обратной связи и петлевым усилением ослабляются значения паразитных емкостей кабеля. «Согласование» преобразователя с датчиком необходимо производить при определении того напряжения МФП, при котором МФП обеспечит поддержание равенства токов во всем диапазоне изменений ΔC датчика.

Требуемое напряжение МФП находится из соотношения:

$$\frac{\Delta U}{UM0} = \frac{\Delta C}{C0},$$

где ΔC, C0 = C10 = C20 — соответственно диапазон изменения и начальное значение емкости датчика; ΔU = к2×U, UM0 = к1×U0 — требуемые значения приращения и начального значения напряжений МФП для обеспечения поддержания им равенства токов датчика во всем диапазоне изменений ΔC.

Резисторы R2, R3 обеспечивают устойчивость по постоянному току ОУ DA2, а именно задают нулевую ось треугольного напряжения.

Значения параметров элементов R1, C1 должны быть таковы, чтобы по отношению к линейно изменяющимся напряжениям усилитель неравновесия производил их дифференцирование, для чего постоянную времени T = R1× C1 необходимо выбирать меньше периода частоты. Необходимые операции «установка 0» и масштаба выходного сигнала выполняются резисторами R20 и R24 соответственно для нуля и максимального значения измеряемого параметра. Максимальное значение выходного напряжения выбрано равным типовому значению диапазона современных АЦП (2,5 В). Диапазон выходного напряжения — ± 2,5 В.

Подключение датчика можно производить по трехпроводной, либо по пятипроводной схеме. Последний вариант предпочтительнее, поскольку симметричные треугольные напряжения формируются непосредственно на самом датчике.

Все резисторы — С2-29В – 0,125 с допуском ±0,1 %, кроме R25: С2-33Н – 0,125 – 100к ± 5 %. Могут быть применены и резисторы R1-16П и R1-12. Переменные резисторы — СП5 – 2 ВБ-0,5. Конденсаторы — К10-17а- … (К10-69В- …), К53- 56А- 20 В, 10 мкФ. ОУ — AD822AR, (AD824AR), ключи DA4, DA5 — MAX4519. ОУ можно заменить на отечественные: DA1.1 на 544УД2А, остальные — на К140УД17А.

Каких-либо дополнительных требований преобразователь не требует.

Преобразователь использовался с датчиком с диапазоном изменения ΔC = ± 100 пФ при начальной емкости порядка 500 пФ и длине кабеля L = 7 м.

ЛИТЕРАТУРА

1. Ю. Пахоменков. Преобразователь сигналов дифференциального емкостного датчика. — Схемотехника, 2006, № 10, с. 42—45.

Вячеслав Селиванов,
г. Жуковский
Московской обл.

Экономичный светильник на мощных светодиодных лампах для радиолюбительского рабочего места

В статье описан простой и очень эффективный светильник на основе матрицы из трех мощных светодиодных ламп белого свечения.

Если вы когда-либо видели фонари на сверхъярких светодиодах — забудьте, мощные светодиодные лампы не имеют с ними ничего общего! Мощные светодиодные лампы построены на основе полупроводниковых кристаллов из карбида кремния (SiC) и обладают рядом преимуществ по сравнению с аналогичными кремниевыми изделиями: максимальное время наработки на отказ; низкое напряжение питания; мгновенное зажигание; полное гашение; отсутствие свинца и ртути; устойчивость к электростатическим воздействиям более 2 киловольт; встроенная в корпус линза, позволяющая легко сопрягать кристалл с любой оптической системой; малая площадь основания корпуса. Внешний вид светодиодной лампы приведен на рис. 1.

Преимущества мощных светодиодных ламп по сравнению с традиционными лампами накаливания:

- рекордная эффективность за счет максимального числа люменов светового потока на 1 Вт потребляемой мощности; что позволяет работать от аккумулятора несколько часов непрерывно с полным ощущением включенной лампы накаливания;
- используется безопасное напряжение питания постоянного тока: 12...25 В;
- создается ровный, немерцающий свет, успокаивающий глаза; при этом яркость можно плавно ре-

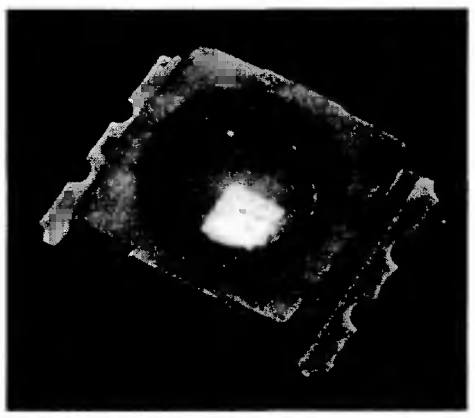


Рис. 1. Внешний вид светодиодной лампы 1 Вт

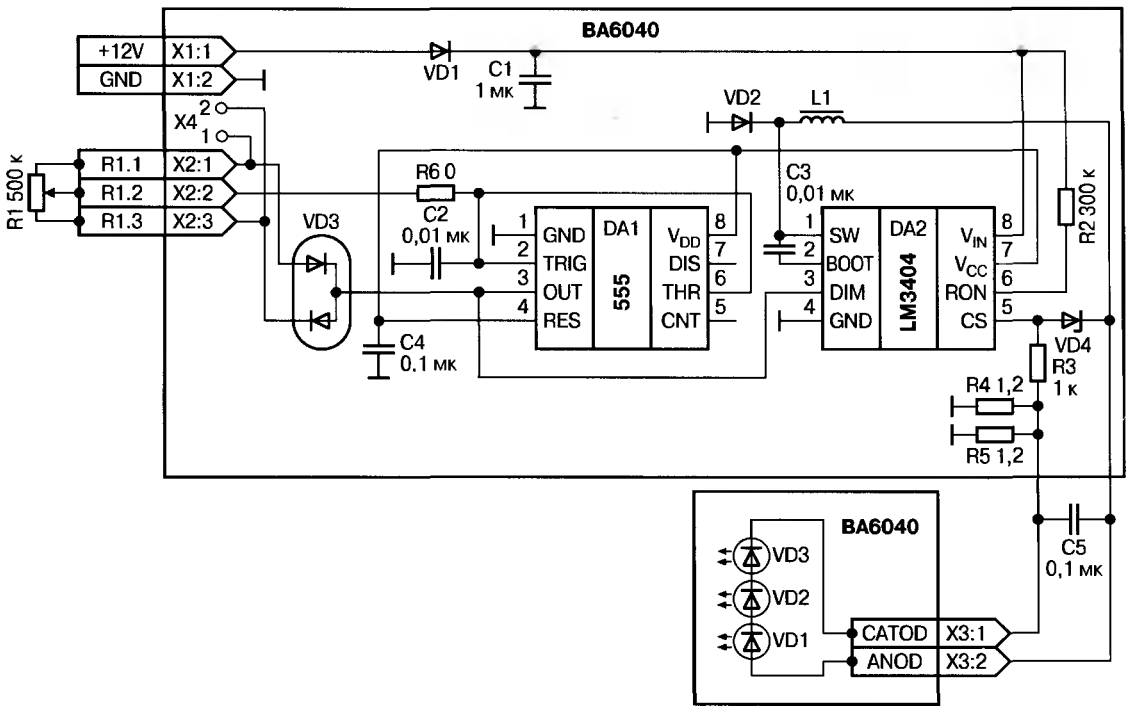


Рис. 3. Принципиальная схема светильника

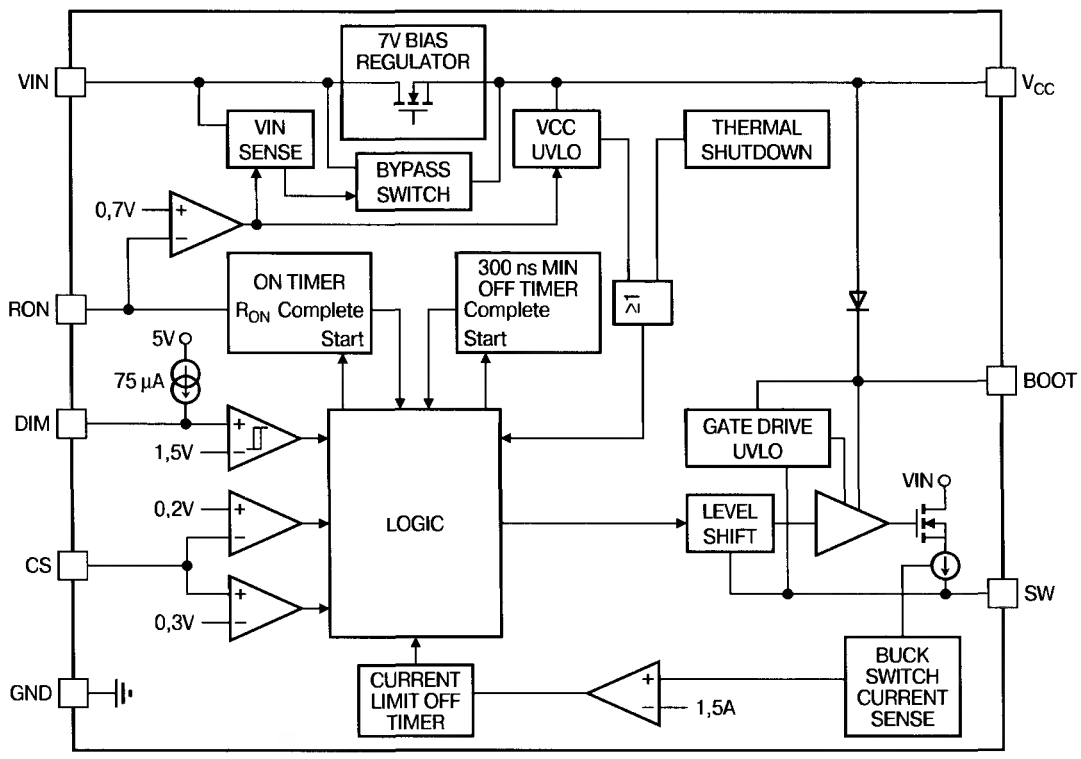


Рис. 4. Блок-схема микросхемы LM3404

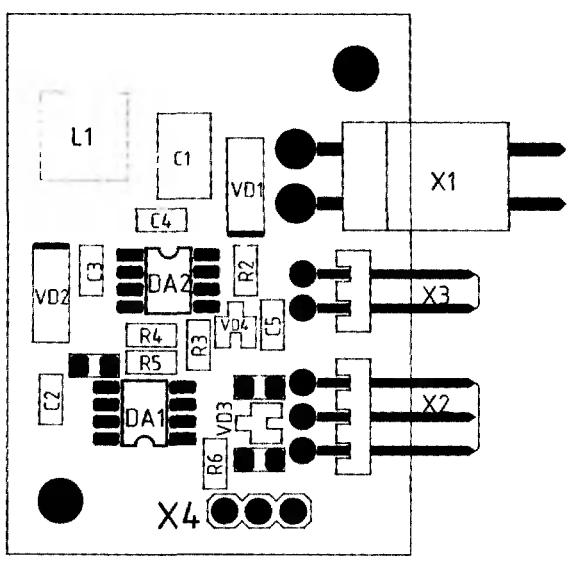


Рис. 5. Монтажная схема регулятора

струкция позволяет подключение дополнительной светодиодной матрицы из трех светодиодных ламп, и включить ее последовательно с имеющейся (отпаяйте белый провод от одного из модулей и подпаяйте к нему черный провод от дополнительной матрицы, белый же провод от дополнительного провода подпаяйте к освободившемуся контакту штатной матрицы светодиодных ламп). В этом случае при напряжении питания 24 В. Вы получите удвоенный световой поток — до 500 Лм.

Чтобы избавить Вас от поиска электронных компонентов, изготовления печатных плат и проведения монтажа, МАСТЕР КИТ предлагает готовый блок **BM6120** «Светильник на мощных светодиодных лампах» и дополнительный светодиодный модуль **BM6020**.

Более подробно ознакомиться с ассортиментом нашей продукции можно с помощью каталога "МАСТЕР КИТ" и на нашем сайте masterkit.ru, где представлено много полезной информации по электронным наборам и модулям МАСТЕР КИТ, а также приведены адреса магазинов, где их можно купить.

Наборы, блоки и модули МАСТЕР КИТ, бумажный каталог «МАСТЕР КИТ-2008» и журналы «Схемотехника» можно купить в магазинах радиодеталей вашего города.

Юрий Садиков
г. Москва

Магнетометры и компасы фирмы Honeywell

Фирма Honeywell Inc. является одной из немногих компаний, которая производит достаточно широкую линейку магниточувствительных датчиков, начиная от простейших пороговых датчиков на эффекте Холла и заканчивая сложными микропроцессорными устройствами с программным управлением. Одними из таких устройств являются магнетометры и компасы с выходом в цифровом коде.

В общем случае и компас, и магнетометр представляют собой схему, собранную на одной плате и состоящую из чувствительного элемента на основе магниторезистора с мостовой схемой, усилителя сигнала и микропроцессора, который осуществляет связь по какому-либо интерфейсному протоколу (RS-232/ USART /RS-485), оцифровку и пересчет сигнала с датчика. Общая блок схема подобного устройства приведена на рис. 1.

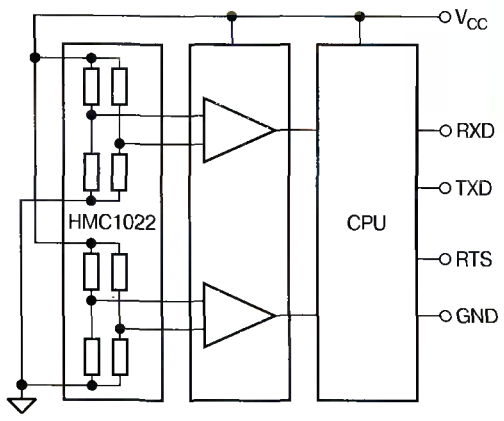


Рис. 1. Блок-схема цифрового компаса

В реальной схеме присутствуют дополнительные элементы, такие как стабилизаторы питания, элементы калибровки, источники опорного напряжения и элементы электромагнитной совместимости.

Младшим представителем семейства цифровых компасов Honeywell можно считать решение под названием HMR3100 (рис. 2 и 3), которое

Рис. 2. Сторона элементов платы HMR3100

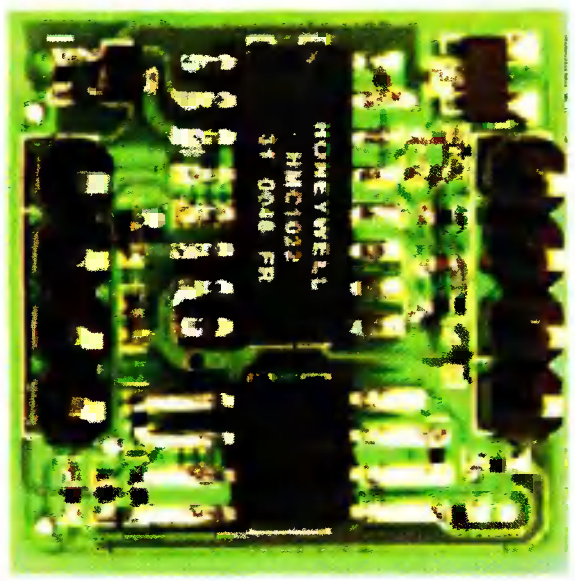


Рис. 3 Сторона печати HMR3100

представляет собой двухстороннюю плату с размерами 19×19×4,5 мм с элементами и клеммами подключения.

Устройство собрано с использованием магниторезистора HMC1022 и имеет следующие технические характеристики:

- точность — 5°;
- разрешение — 0,5°;
- повторяемость— 3°;
- чувствительность — 6 мГаусс;
- измерение по двум осям;
- встроенный калибровочный элемент;
- скорость обмена данными — 2400/4800/9600/19200;
- интерфейс USART, кодирование ASCII;
- устройство работоспособно в диапазоне температур 0…70 °С и напряжении питания 2,6…5 В DC;
- потребляемый ток — 0,5 мА (рабочий режим), 1 мкА (режим ожидания), 17,3 мА (режим калибровки);
- вес — не более 1,5 г.

Основные области применения: навигационные системы, автомобильные компасы, позиционирование телескопов и антенн, а также в портативной аппаратуре.

Устройство измеряет напряженность магнитного поля земли в двух взаимно перпендикулярных направлениях с использованием двухмостовой схемы HMC1022. В режиме измерения устройство должно находиться в горизонтальном положении. Аналоговые данные от двух мостов поступают на аналоговые усилители, а потом на АЦП микроконтроллера, который обрабатывает их с учетом напряжения питания, сдвига нуля и данных по калибровке.

Устройство получает питание от литиевой батареи. Электрическая схема спроектирована с общей землей для питания, логики и аналоговой части. Расположение и распиновка внешних разъемов устройства приведена на рис. 4. Стрелка указывает направление на «Север».

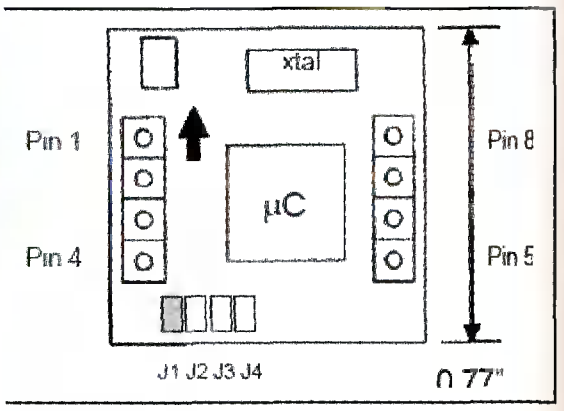


Рис. 4. Внешние разъемы HMR3100

Назначение выводов приведено в табл. 1.

Заводская установка протокола USART 9600.N.8.1. Скорость обмена может устанавливаться джамперами J1&J3. Заводская установка подразумевает установку перемычки в J1 со скоростью 9600. Установка перемычек на J1&J3 переводит скорость обмена на 19200. При скорости 2400 перемычек нет. При скорости 4800 перемычка на J3.

J2 — служебный. J4 служит для инвертирования сигнала по оси Y при установке устройства «стрелкой назад» в рабочем положении.

Таблица 1. Назначение выводов внешних разъемов

Номер	Наименование	Назначение
1	Vcc	Вход напряжения питания
2	NC	Не используется
3	RTS	Готовность к передаче (входной)
4	NC	Не используется
5	TXD	Передаваемые данные
6	RXD	Принимаемые данные (входной)
7	GND	Общая земля
8	NC	Не используется

Передача/прием осуществляется по стандартной USART процедуре для 3-проводного соединения. Активное состояние сигнала — низкий логический уровень. В зависимости от управляющих сигналов устройство может находиться в одном из трех режимов:

- 1 — нормальное состояние (измерение);
- 2 — состояние передачи;
- 3 — состояние калибровки.

НОРМАЛЬНОЕ СОСТОЯНИЕ

RX-в состоянии 1 (высокий уровень). При появлении на выводе RTS нулевого импульса от внешнего устройства HMR3100 начинает передавать по линии TXD байты состояния/данных. При этом внешнее устройство должно удерживать линию RXD в высоком состоянии. Линия RTS должна удерживаться внешним устройством в высоком состоянии. HMR3100 посылает пакет состояния/данных три раза и если в течение этих посылок линия RTS ни разу не перешла в низкое состояние, HMR3100 возвращается в состояние ожидания и низкого энергопотребления. Временная диаграмма этого режима показана на рис. 5.

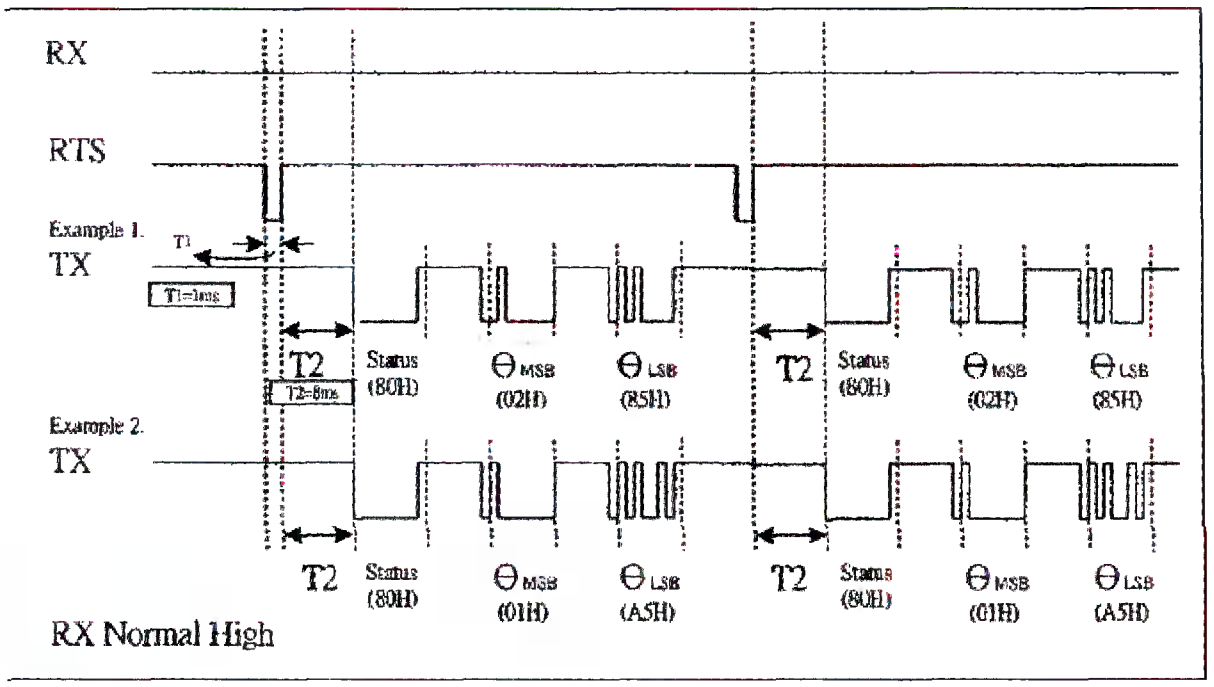


Рис. 5. Временная диаграмма нормального состояния HMR3100

СОСТОЯНИЕ ПЕРЕДАЧИ

Когда внешнее устройство удерживает линию RTS в низком состоянии, HMR3100 непрерывно посылает пакеты данных на вывод TXD. Внешнее устройство должно удерживать линию RXD в высоком состоянии в течение всего режима. Частота передачи пакетов — порядка 2 Гц. HMR3100 возвращается в состояние ожидания и низкого энергопотребления при переходе сигнала RTS в высокое состояние. Временная диаграмма этого режима показана на рис. 6.

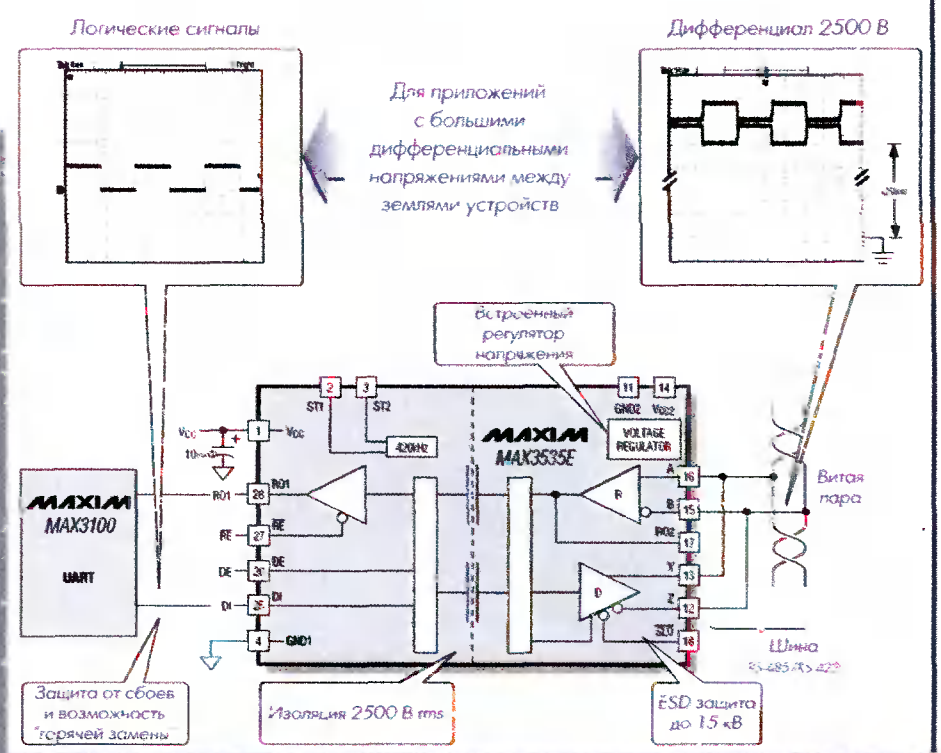
РЕЖИМ КАЛИБРОВКИ

Для перевода HMR3100 в режим калибровки необходимо установить на линии RX низкий уровень и подать импульс низкого уровня на линию RTS. Калибровка производится по образцу магнитного материала, который закреплен на плате HMR3100. После перевода устройства в режим калибровки необходимо произвести два оборота ус-

MAXIM DALLAS SEMICONDUCTOR
muRata
Infineon technologies
EPCOS
Honeywell
SICK
VISHAY
BOURNS
MITSUBISHI ELECTRIC
Tyco Electronics
Panasonic ideas for life
DATA VISION
Kingbright
CRYDOM
IOR

Изолированный трансивер RS-485/RS-422 MAX3535E

- Внутренний емкостной изоляционный барьер с уровнем изоляции ±2500 В rms
- Гальваническая развязка
- Ограничение скорости нарастания выходного напряжения для снижения уровня помех
- Возможность использования «горячей замены»
- Повышенная нагрузочная способность, до 256 устройств
- Быстродействие до 1000 кБит/с
- Защита от электростатики ± 15 кВ
- Питание от 3 до 5.5 В, сверхнизкое энергопотребление
- Не требует внешней оптопары



www.platan.ru
ПЛАТАН
• Москва, ул. Ивана Франко, д. 40, стр. 2
Тел. (495) 97 000 99 E-mail: platan@aha.ru
• Москва, 1-й Щемиловский пер., д. 16, стр. 2
Тел. (495) 744 70 70 E-mail: platan@platan.ru

тройства, расположив его на плоской немагнитной поверхности. Каждый оборот необходимо производить за время не более 5 с. Во время вращения устройство производит измерение и калибровку. После завершения этой процедуры необходимо вернуть линию RXD в высокое состояние, при этом HMR3100 перейдет в состояние ожидания. Начало процедуры калибровки подтверждается передачей от HMR3100 кода 53 54 41 hex, завершение — передачей 52 44 59 hex. Временная диаграмма режима калибровки HMR3100 представлена на рис. 7.

ФОРМАТ ПЕРЕДАВАЕМЫХ ДАННЫХ

Устройство передает измеряемые параметры в двоичном коде по линии RTS. Активное состояние — низкий уровень. Начало передачи слова состояния/данных предваряется передачей кода 80hex или 81hex. Код 81hex указывает на то, что возможно уменьшение точности измерения вследствие ослабления сигнала или наличие дополнительной ошибки. Если необходима максимальная точность измерения, необходимо провести калибровку.

Передача данных о текущем курсе (угле между направлением на север и направлением, указанным на плате стрелкой) ведется в двоичном коде старшим и младшим байтами, например, 80 02 85 hex соответствуют курсу 322,5°. При этом 80 hex — стартовый байт, 02 hex переводится в десятичный формат и умножается на 256. Младший байт — 85 hex — переводится в десятичный формат (соответственно это будет 133) и прибавляется к 512. В результате получается сумма 133 + + 512 = 645. Результат делится на 2, что дает 322,5°.

Данная форма представления данных позволяет достичь точности в 0,5°, за счет наличия перевода из двоичного в десятичный формат и деления на 2.

Демонстрационная плата для HMR3100 включает в себя печатную плату для установки компаса с двумя разъемами под батарею 9 В и RS-232/9pin, Windows совместимое программное обеспечение и руководство пользователя. На рис. 8 показан внешний вид демонстрационной платы.

Honeywell выпускает цифровые компасы с более высокой точностью — HMR3200 и HMR3300. Эти компасы собраны с использованием магниторезистора HMC1022 и HMC1022 и имеют следующие технические характеристики:

- точность — 1°;
- разрешение — 0,1°;
- повторяемость — 0,5°;
- чувствительность — 6 мГаусс;

напряжении питания 6…15 В DC или 5 В AC;

- потребляемый ток — 20/24 мА;
- вес — не более 7,5 г.

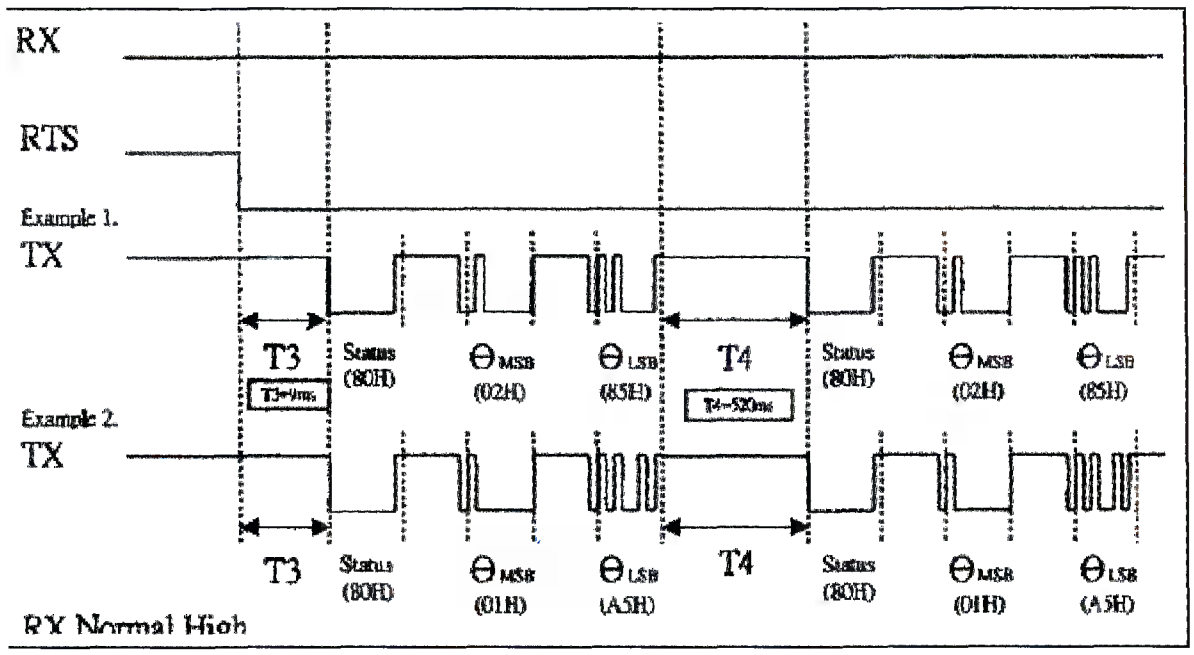


Рис. 6. Временная диаграмма режима передачи HMR3100

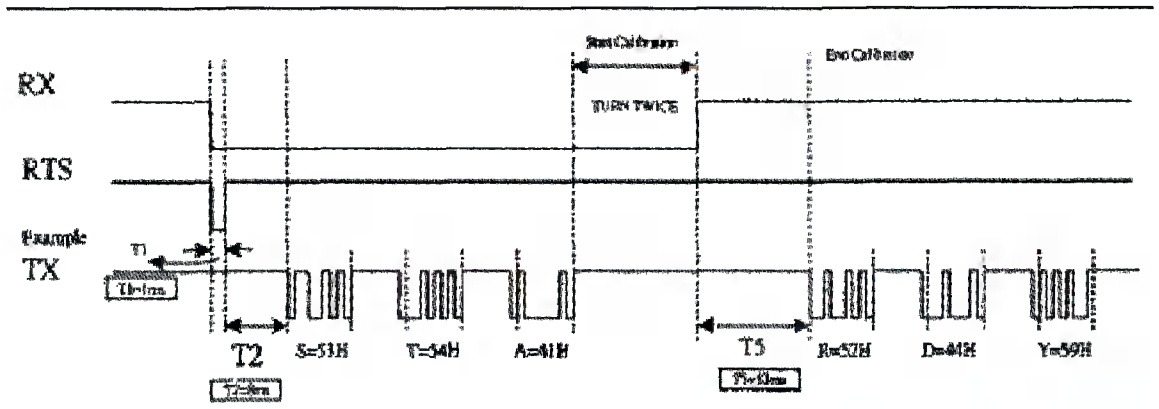


Рис. 7. Временная диаграмма режима калибровки HMR3100

- измерение по двум осям (HMR3200) и трем осям (HMR3300);
- встроенный калибровочный элемент;
- частота обмена данными — 15 Гц;
- интерфейс USART/SPI, кодирование ASCII;
- USART 2400/4800/9600/19200
- SPI CKE = 0, CKP = 0 (псевдомастер);
- устройство работоспособно в диапазоне температур –40…85 °С и

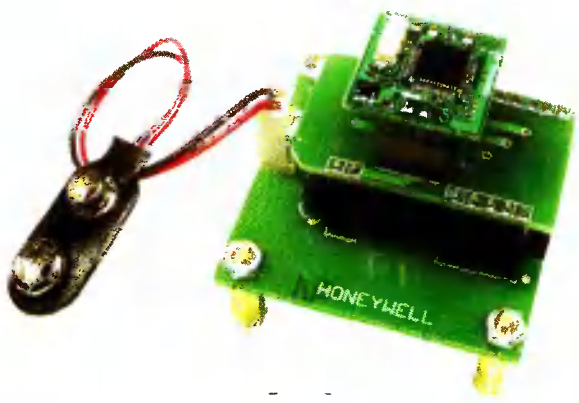


Рис. 8. Внешний вид демонстрационной платы HMR3100

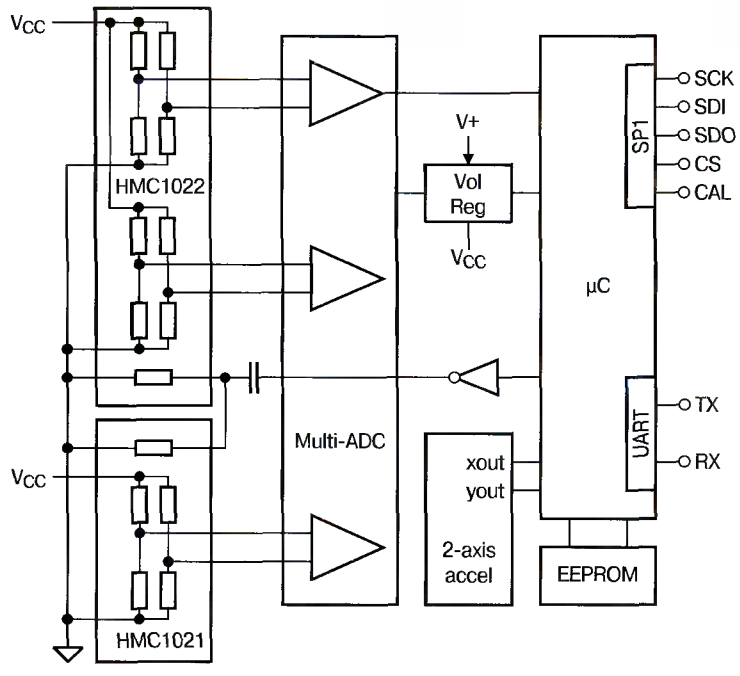


Рис. 9. Блок-схема цифровых компасов HMR3200 и HMR3300

Основные области применения: навигационные системы, позиционирование телескопов и антенн, портативная аппаратура, позиционирование платформ, GPS устройства, лазерные системы наведения.

HMR3200 представляет собой цифровой компас, измеряющий напряженность магнитного поля по двум осям, и может быть ориентирован при измерении или горизонтально или вертикально.

В состав HMR3300 для увеличения точности дополнительно включен акселерометр, измеряющий ускорение по двум осям. На рис. 9 представлена блок-схема цифровых компасов HMR3200 и HMR3300. Назначение выводов приведено в табл. 2.

Расположение элементов на плате и нумерация внешних выводов показана на рис. 10.

По интерфейсу UART можно подать на модуль целый ряд команд:

- Выбор режима выдачи результатов *H<cr><lf>)
- Перевод магнетометра в режим выдачи курса, тангажа и рысканье (*M<cr><lf>)
- Старт и стоп передачи данных (*S<cr><lf>)
- Запрос (*Q<cr><lf>)
- Обнуление величины поворота (*O<cr><lf>)
- Обнуление величины тангажа (*P<cr><lf>)
- Запрос средней величины (*A<cr><lf>)
- Сброс (*R<cr><lf>)
- Калибровка (*C<cr><lf>)

При калибровке платформа с компасом вращается с постоянной скоростью вокруг оси на 360°. Скорость вращения должна быть небольшой. Процесс необходимо продолжать в течение 1 мин. При HMR3200 вращение необходимо проводить в плоскости платы. В случае с HMR3300 вращение можно проводить и при отклонении от плоскости платы. Процесс калибровки начинается подачей на модуль команды перехода в командный режим (*X<cr><lf>) и команды собственно калибровки (*C<cr><lf>) по интерфейсу UART. После завершения необходимых процедур для выхода из режима калибровки необходимо подать ту же команду (*C<cr><lf>).

Ответ модуля на переданные команды представляет последовательность:

#Dxxx<cr><lf> — данные;

#I<cr><lf> — признак неидентифицируемости предыдущей команды.

Таблица 2

Номер	Наименование	Назначение
1	SCK	Выходной тактовый сигнал SPI интерфейса
2	RX/SDI	UART принимаемые данные/SPI входные данные
3	TX/SDO	UART передаваемые данные/SPI выходные данные
4	CS	Выбор кристалла (для режима SPI, активируется задним фронтом)
5	CAL	Включение/выключение калибровки (активируется задним фронтом)
6	+5VDC	Вход 5В постоянного тока, стабилизированного
7	GND	Общая земля
8	+V	Вход нерегулированного постоянного напряжения +6...+15 В

ИНТЕРФЕЙС SPI

Модули HMR3200 и HMR3300 контролируют выдачу синхронного тактового сигнала (SCK) и синхронного сигнала выходных данных (SDO). Внешний процессор контролирует следующие сигналы — выбор кристалла (CS) и сигнала входных данных (SDI). Для активации интерфейса SPI внешнее устройство должно на 20 мкс подать сигнал низкого уровня на линию CS. Модуль посылает в ответ символ 's' в ASCII, после чего внешнее устройство должно выдать разрешенную команду. Модуль декодирует команду, выдает соответствующие данные и завершает SPI передачу. Если команда не декодирована или неизвестна, то модуль посылает в ответ символ 'e' в ASCII .

Временные характеристики интерфейса SPI приведены на рис. 11.

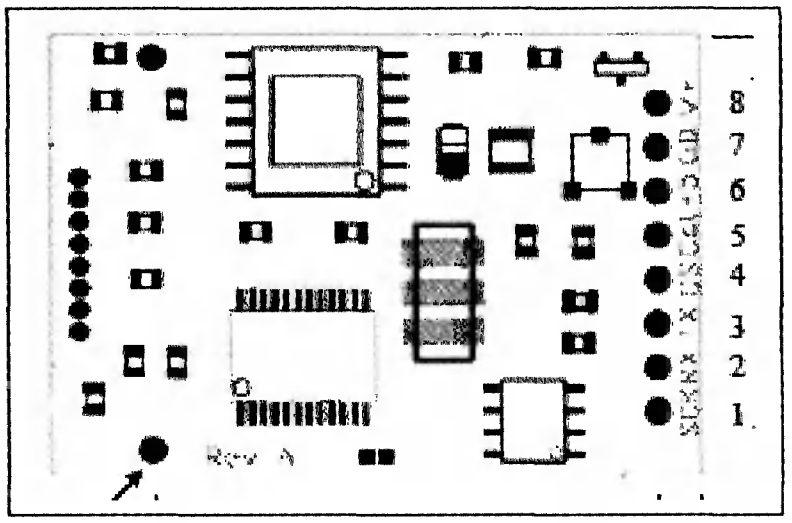


Рис. 10. Расположение элементов и нумерация выводов на плате HMR3200 и HMR3300

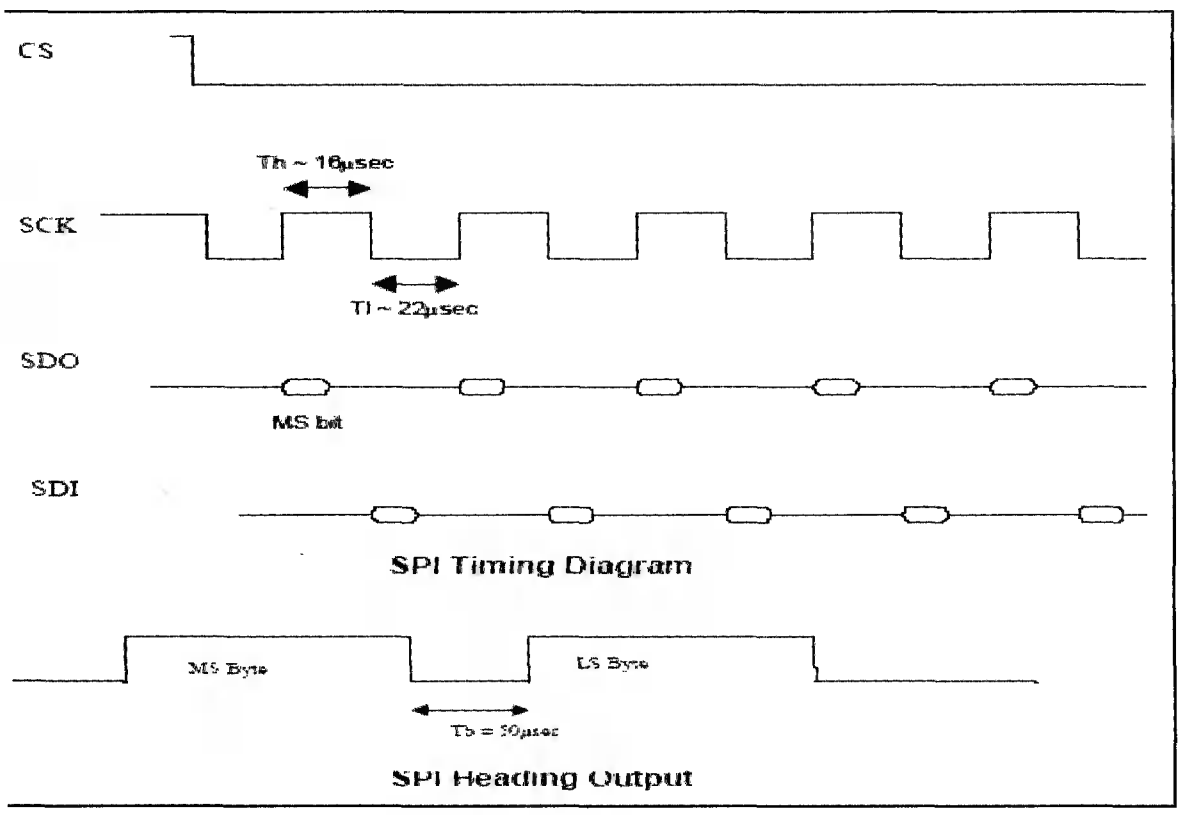


Рис. 11. Временные характеристики интерфейса SPI

ЛИТЕРАТУРА:

1. *http://www.magneticsensors.com/datasheets/hmr32003300.pdf*
2. *http://www.magneticsensors.com/datasheets/hmr3100.pdf*
3. *http://www.magneticsensors.com/datasheets/hmr32003300.pdf*
4. *http://www.magneticsensors.com/datasheets/HMR32003300_manual.pdf*

Александр Зыбайло,
г. Москва

Пассивные компоненты

Эффективная защита от перенапряжения

Энергетические варисторы используют как незаменимый конструктивный элемент для защиты от перенапряжения сетей энергоснабжения. В статье рассказывается о преимуществах варисторов Epcos.

Непогода становится причиной колебаний и масштабных сбоев электроснабжения.

В зависимости от расположения, качества и структуры сетей организаций электроснабжения их необходимо обезопасить от перенапряжения, чтобы обеспечить бесперебойное электроснабжение потребителей. Перенапряжение может возникать при прямом ударе молнии, подаче энергии или процессах коммутации. Чтобы гарантировать энергетическую безопасность потребителей, возникающее перенапряжение нужно отводить. Для этого отлично подходят металлооксидные энергетические варисторы производства компании Epcos AG (Германия). Epcos является ведущим производителем варисторов и предлагает полный спектр дисковых, пластинчатых, блочных и энергетических варисторов.

Варисторы представляют собой керамические детали, которые изменяют свое электрическое сопротивление в зависимости от прилагаемого напряжения. Они относятся к керамическим полупроводникам. Классификация энергетических варисторов определяется IEC 60099-4.

Исходный материал для изготовления варисторов — ZnO (рис. 1). Помимо системы ZnO есть и другие системы варисторов, например, на базе SiC или $SrTiO_3$. В качестве варисторной системы Epcos использует исключительно ZnO, потому что ZnO отличается своими электрическими свойствами и высокой доступностью и чистотой.

ZnO проходит химическую подготовку и с другими оксидами, например, Bi2O3, Co3O4, Mn3O4, NiO перерабатывается в гранулят. Доля ZnO в грануляте составляет около 95 моль %. После того как гранулят при соответствующем технологическом процессе прессуется и спекается при температуре около 1400…1500 °C, в спеченном теле образуется поликристаллическая структура. Величина частиц при этом составляет около 15—20 µм (рис. 1). На концах зерен образуется запирающий слой (рис. 2). Если электрический потенциал на границе зерен превышает 3,2 В, запирающий слой ликвидируется и переходит в проводящее состояние (рис. 3). Переход из запирающего состояния в проводящее происходит в течение 10^{-9} с. В области пропускания для тока действительна зависимость $I \sim U^{\alpha}$ (значение α при этом составляет от 40 до 80).

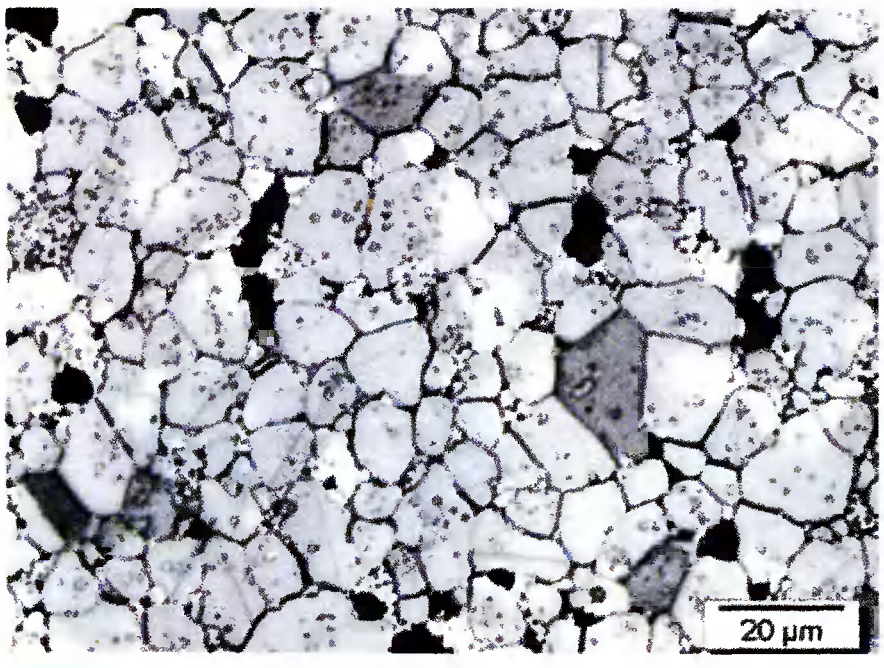


Рис. 1. Вид шлифованной поверхности ZnO-керамика

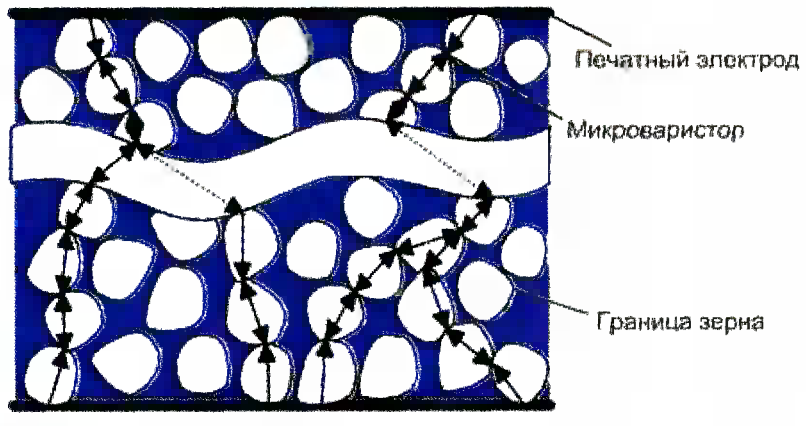


Рис. 2. Границы зерен и запирающие слои (микровaристор)

Варистор как идеальный защитный элемент против перенапряжения отличает зависимость сопротивления от напряжения.

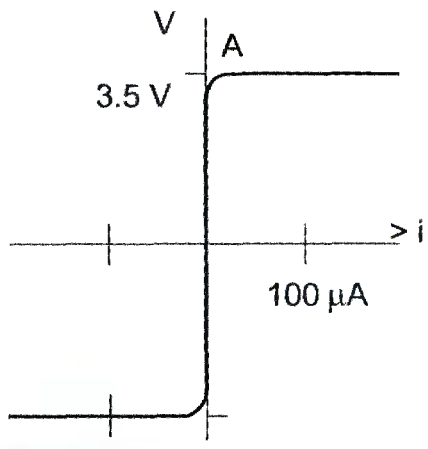


Рис. 3. Эффект полупроводника

Механические размеры варистора (высота и диаметр) позволяют масштабировать напряжение от нескольких вольт до нескольких тысяч вольт. Предельно допустимая нагрузка (сила тока) охватывает 10 порядков. Химический состав гранулята и технологические особенности позволяют задавать электрические свойства. Варисторы называют энергетическими потому, что они характеризуются высокой предельно допустимой нагрузкой (силой тока) и высоким номинальным напряжением.

Металлооксидные энергетические варисторы Epcos находят применение в качестве ключевых деталей в разрядниках для защиты от перенапряжений. Они также используются в фарфоровых и полимерных ограничителях перенапряжений (ОПН) для открытых установок. ОПН являются незаменимыми компонентами сетей электропередач и энергораспределения, они служат для защиты установок, трансформаторов, генераторов и железнодорожного транспорта. предлагает широкий спектр металлооксидных энергетических варисторов для низко-, средне- и высоковольтных сетей (рис. 4).

Компания Epcos производит энергетические варисторы с 1978 г. и обладает большим практическим опытом в промышленном производстве энергетических варисторов. Владея технологией производства от создания рецептуры керамики непосредственно до выпекания варисторов Epcos способна изготовить варисторы с параметрами, требуемыми для конкретного применения.

В то же время компания Epcos, пожалуй, единственная компания в мире, которая изготавливает только высоковольтные варисторы и не занимается производством ОПН. Благодаря умелой деятельности фирмы «Платан», являющейся генеральным партнером Epcos в России и странах СНГ варисторы Epcos применяют ведущие производители ОПН: «Феникс-88», «Полимер-Аппарат», «ЛМ-Электро», «РК Таврида-Электро», НИИ ЗАИ, ЗЭТО, СЗП и другие. Используя варисторы из стандартной программы поставок или же совместно создавая специальные типы варисторов, клиенты смогут производить не только самые современные, но и, если необходимо, адаптированные к *локальным сетям* ОПН, *не уступающие лучшим мировым* аналогам!

В заключение приведем некоторые отзывы компетентных специалистов об этой продукции:

«Высоконелинейные резисторы (варисторы) производства фирмы Epcos по своим техническим и потребительским характеристикам на сегодня лучшие на российском рынке ограничителей перенапряжений из всех нами испытанных», — зам. генерального директора по науке Государственного научного центра РФ «Всероссийский электротехнический институт им. В. И. Ленина», директор испытательного центра высоковольтного электрооборудования к. т. н. Г. Лаврентьев.

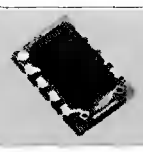
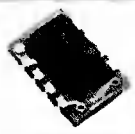
«Благоприятная ценовая политика, отличная дисциплина выполнения поставок по праву позволяет позиционировать Epcos как лидера среди производителей силовых варисторов для российского рынка», — директор Липецкого Электротехнического завода «РК Таврида Электрик» А. А. Родин.

Рис. 4. Примеры варисторов компании Epcos

Ильин Михаил
Крук Александр,
г. Москва

ЦИФРОВЫЕ
датчики температуры и влажности

Компания **Sensirion** выпускает цифровые датчики влажности со встроенными датчиками температуры. Инновационная технология **CMOSens**, используемая компанией, позволила достичь небывалой долговременной стабильности измерений (0.5% в год) при максимальном удобстве эксплуатации.

SHT10		Малобюджетные датчики влажности для промышленных применений Диапазон измеряемой влажности 0...100% с точностью ±4,5% RH, ±0.5°C. Время отклика 4 с. Напряжение питания 2.4....5.5 В.
SHT11		Малобюджетные датчики влажности для промышленных применений Диапазон измеряемой влажности 0...100% с точностью ±3% RH, ±0.4°C. Время отклика 4 с.
SHT15		Высокоточные датчики влажности в SMD корпусах Диапазон измеряемой влажности 0...100% с точностью ±2% RH, ±0.3°C. Время отклика 4 с. Напряжение питания 2.4....5.5 В.
SHT71		Высококачественные датчики влажности для монтажа в отверстие Диапазон измеряемой влажности 0...100% с точностью ±1,8% RH, ±0.3°C. Время отклика менее 3 с. Взаимозаменяемы без калибровки. Напряжение питания 2.4....5.5 В.
SHT75		Высококачественные инновационные датчики влажности для монтажа в отверстие Диапазон измеряемой влажности 0...100% с точностью ±3% RH, ±0.4°C. Время отклика менее 3 с. Взаимозаменяемы без калибровки. Напряжение питания 2.4....5.5 В.

СЕНСОРИКА Москва 1-й Щемиловский пер., д. 16, стр. 2
Тел./факс: 495 223 0038, e-mail: info@sensorica.ru **www.sensorica.ru**

Методика быстрого создания программ для карманных персональных компьютеров, смартфонов и коммуникаторов

Здесь описывается способ быстрой разработки программ для мобильных устройств, таких как карманные персональные компьютеры, смартфоны и коммуникаторы,на языке программирования С с помощью свободно распространяемой программной среды разработки «Pelles C for Windows».

Карманные персональные компьютеры (КПК), смартфоны и коммуникаторы активно входят в жизнь современных людей. Сочетая в себе множество полезных функций, таких как часы с будильниками, калькулятор, календарь с органайзером, справочники, GPS навигатор, GPRS модем и др., они помогают людям организовать свое время, быстро произвести расчеты, оперативно найти необходимую информацию, изучать иностранные языки, отправлять и получать электронную почту, посетить Интернет, развлечься игрой и т. д. И, конечно же, основным преимуществом для пользователя КПК, является возможность совершать данные операции, находясь практически в любом географическом месте Земли, благодаря энергонезависимости и компактности этих мобильных устройств.

КПК могут работать под управлением различных операционных систем, таких как Palm OS, Symbian, Windows CE (Pocket PC), которые позволяют разрабатывать приложения, активно использующие мультимедийные возможности устройства.

В мире создано большое количество программ для мобильных устройств, благодаря которым функциональность этих компактных устройств значительно возрастает. Тем не менее, в жизни очень часто возникают задачи, для решения которых требуется специальная программа.

Рассмотрим методику создания программ для КПК, смартфонов и коммуникаторов с самой распространенной операционной системой Windows CE, разновидности которой имеют названия Windows Mobile и Pocket PC различных версий. Отличительной чертой Windows CE является поддержка шрифтов Microsoft True Type. Этот популярный формат позволяет удовлетворить практически любые дизайнерские потребности. Кроме того, операционная система Windows стала привычной для миллионов людей на нашей планете.

Существует множество как платных, так и бесплатных средств разработки программ для КПК. Эти средства позволяют разрабатывать программы на разных языках программирования, таких как ассемблер, Visual C, Visual Basic, Ford, Perl и др.

Мощное средство разработки бесплатно предоставляет компания Microsoft на своем сайте [1]. Однако оно довольно сложное и требует установки на компьютер с операционной системой Windows XP громоздких средств разработки Visual Studio 2005 Professional и MSDN объемом более 3 Гбайт, а также Windows Mobile 5.0 SDK объемом 174 Мбайт и эмуляторов.

Помимо названного средства разработки, в табл. 1 приведен краткий перечень других средств, позволяющих создавать программы для КПК.

Как видно из таблицы, эти средства имеют довольно высокую стоимость. К тому же они довольно сложны и требуют для своей работы приличной производительности компьютера, на котором будут создаваться программы, и немалый объем дисковой памяти для их установки.

Описываемый в статье способ, позволяет обойтись установкой на компьютер простого программного средства для разработки объемом чуть более 7 Мбайт. Это средство разработки имеет название Pelles C for Windows и свободно доступно на сайте его разработчиков [2]. Программы, созданные с его помощью, были успешно протестированы автором на КПК Pocket PC Loox N520 с операционной системой Windows Mobile 5.0.

Данное средство разработки, позволяет создавать разнообразные программы не только для КПК, но и для обычных стационарных персональных компьютеров, а также для смартфонов и коммуникаторов.

Автором среды разработки Pelles C for Windows является программист' по имени Pelle (Швеция) [3]. Он живет в г. Стокгольме и начал программировать в 1979 г. на шведском компьютере ABC80, но вскоре освоил компьютеры на Z80. Профессионально он работает с 1984 г. По большей части создает программы с кодом X86 на языке С.

Пакет разработки Pelles C for Windows включает в себя интерфейс среды разработки IDE, компилятор языка про-

Таблица 1. Краткий перечень средств разработки

Название	Краткое описание	Ориентировочная стоимость, руб.
C++ Builder 6.0 Enterprise	Средство быстрой разработки приложений, позволяющее создавать приложения на языке C++, используя при этом среду разработки и библиотеку компонентов Delphi	50 000
Delphi 2005 Architect	Среда Delphi 2005, поддерживающая несколько языков, SDK Win32 и NET	100 000
Delphi 7 Studio Professional Full System	Имеет возможности для создания бизнес-логики приложений и внедрения моделей в проект посредством визуальных инструментов моделирования и их взаимодействия со средой разработки	28 000
PL/SQL Developer 5.1.4 Single User	Включает в себя следующие компоненты: PL/SQL Editor, Integrated Debbuger, Query Builder, PL/SQL Beautifier, Reporter, Object Browser	4 500

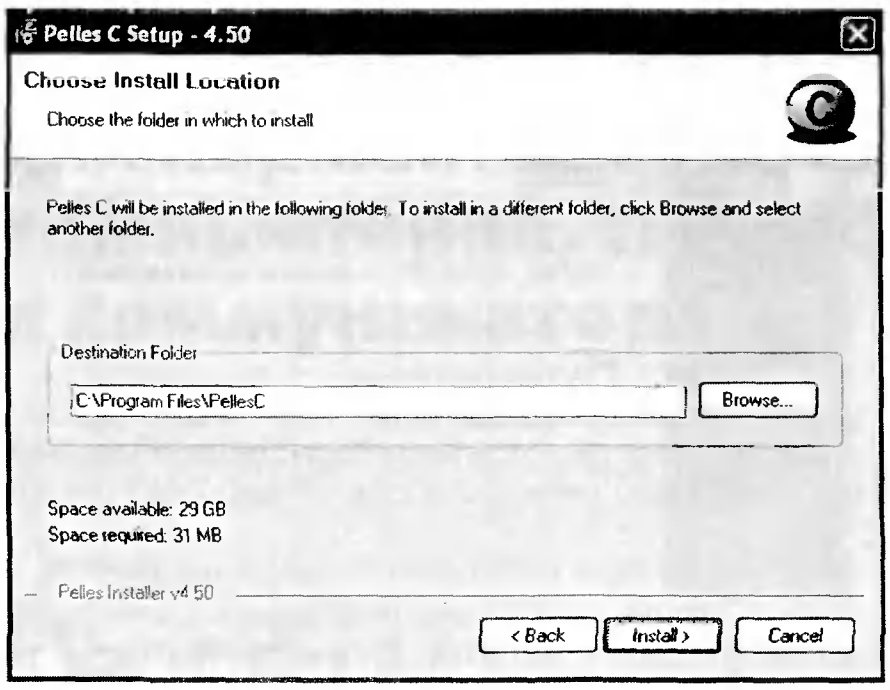


Рис. 5

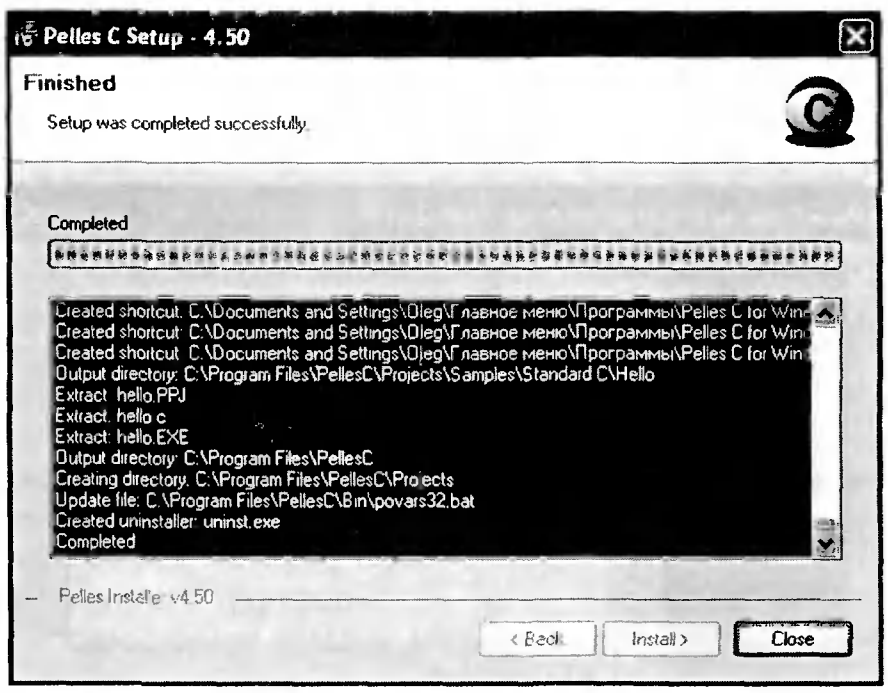


Рис. 6

разив соответствующее окно сообщения (рис. 7). Здесь можно согласиться с запуском программы, щелкнув по программной кнопке «Да» (Yes) или отказаться от немедленного запуска, щелкнув по кнопке «Нет» (No).

После установки, программа среды разработки Pelles C IDE будет находиться в каталоге «C:\Program Files\PellesC\Bin\poide.exe». Ее можно запустить на выполнение в любое время, с помощью команды Пуск->Программы->Pelles C for Windows->Pelles C IDE. Помимо среды разработки можно вызвать справочные службы среды разработки или консольную оболочку программы Pelles C Command Promt (рис. 8).

После запуска среды разработки Pelles C IDE на экране монитора появится окно заставки программы (рис. 9). Это окно исчезает через несколько секунд самостоятельно, а также принудительно после щелчка по нему левой кнопкой мыши.

Под этим окном располагается среда разработки, имеющая привычный интерфейс программ для Windows (рис. 10).

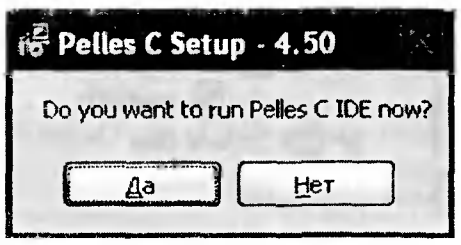


Рис. 7

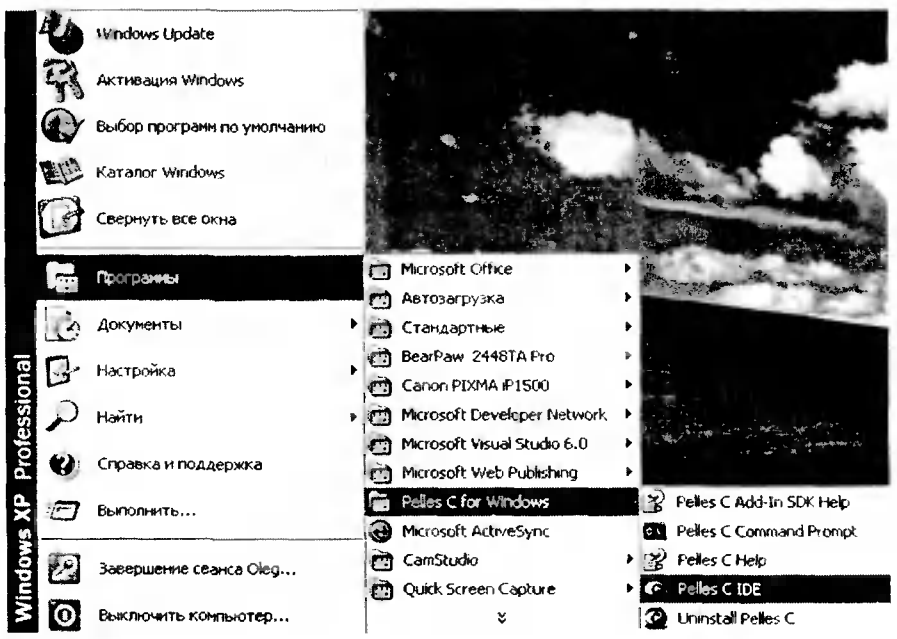


Рис. 8

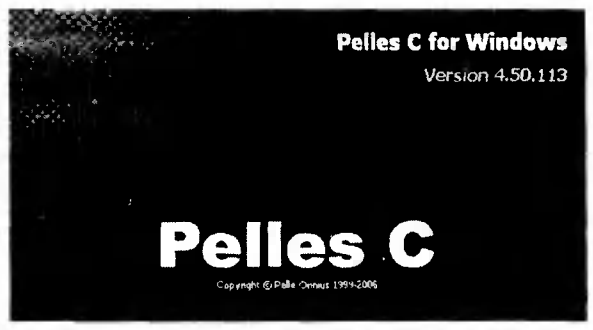


Рис. 9

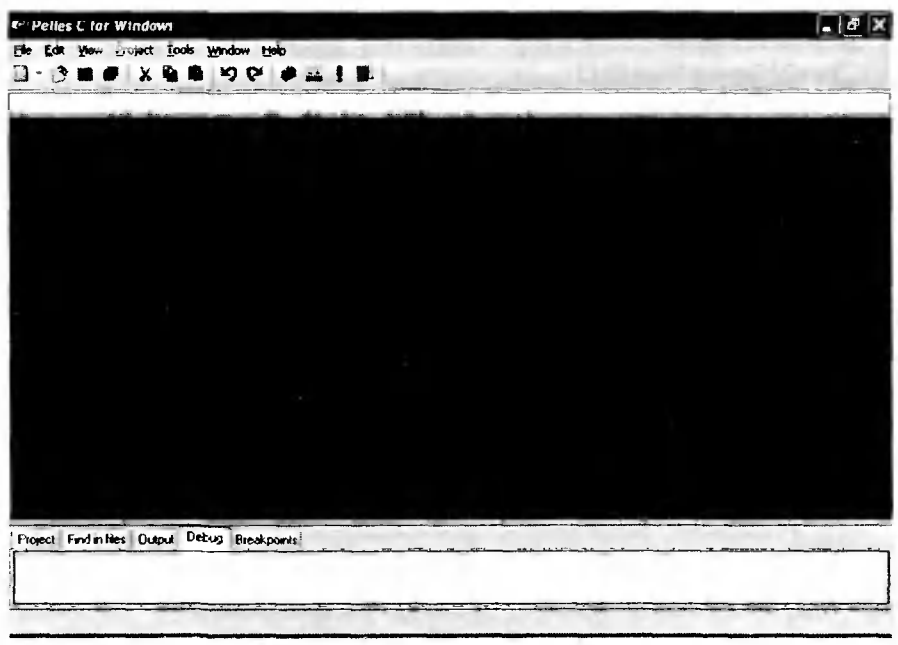


Рис. 10

Рассмотрим пример создания программы, демонстрирующий работу в данной среде разработки и позволяющий быстро научиться разрабатывать собственные программы для КПК, смартфонов и коммуникаторов. В качестве такого примера создадим программу, которая позволит производить вычисления по конвертированию рублей в валюту других стран, а именно в доллары США и в евро. Данный пример позволит продемонстрировать создание диалоговых программ для КПК с операциями ввода чисел, вычислений над ними и отображения результатов.

Итак, после запуска среды разработки Pelles C IDE последовательно выберите в главном меню интерфейса среды команды File->New->Project, как это показано на рис. 11. После чего на экране компьютера отобразится окно мастера новых проектов (рис. 12).

С помощью данного мастера можно создавать заготовки программ различного назначения и для разных операционных систем. Вид отображения иконок программных приложений данного окна можно изменять с помощью

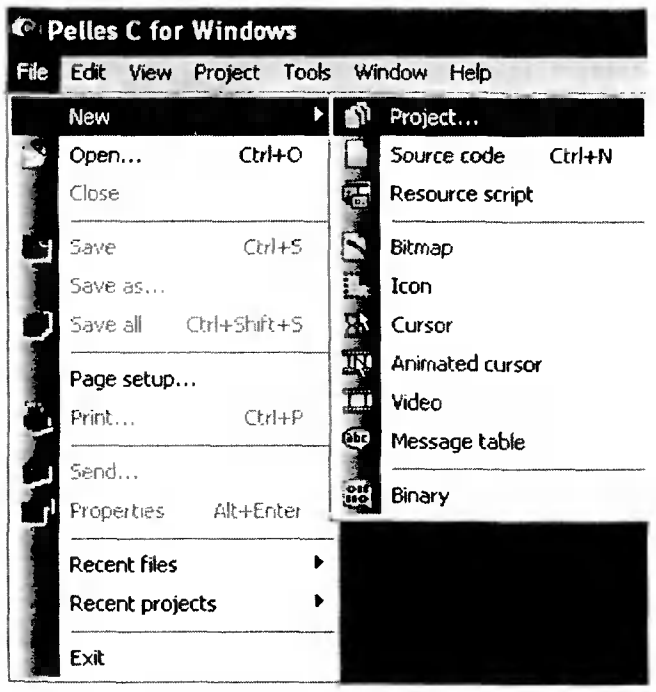


Рис. 11

Рис. 12

двух программных кнопок, расположенных в верхней правой части окна.

Выделите в этом окне программное приложение мастера с названием Pocket PC application wizard. В строке с заголовком Name введите имя нового проекта с названием exchange и щелкните по программной кнопке Ok.

В следующем открывшемся окне мастера приложений (рис. 13) отредактируйте название будущей программы, удалив в предложенном названии «exchange Program» слово «Program» и оставив только «exchange», как показано на рисунке.

Ниже предлагается выбрать тип проекта. Первый из этих типов позволит создать заготовку программы простейшего приложения. Второй тип проекта создает знакомую многим программу, которая при запуске на КПК отображает сообщение «Hello, world!» (Здравствуй, мир!). В будущем можно будет создать любой из предлагаемых типов для самостоятельного изучения создаваемых проектов.

Для нашей программы среди предлагаемых типов проекта следует выбрать тип «A dialog based program», после чего следует нажать программную кнопку «Далее».

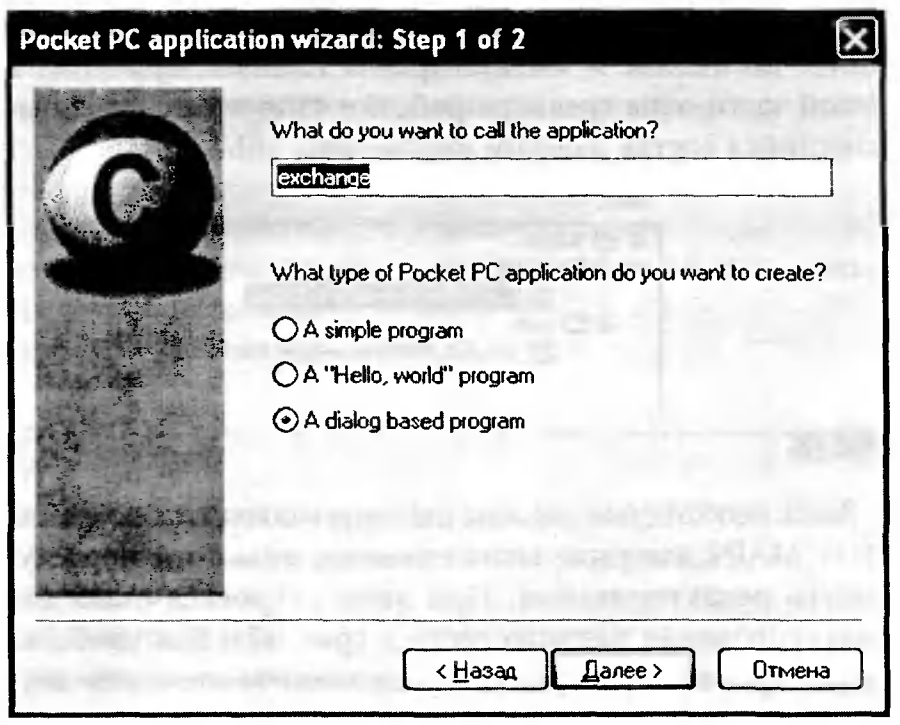


Рис. 13

В последнем окне мастера приложений (рис. 14) выводится сообщение, о том какие файлы были созданы в процессе его работы и в каком каталоге они находятся.

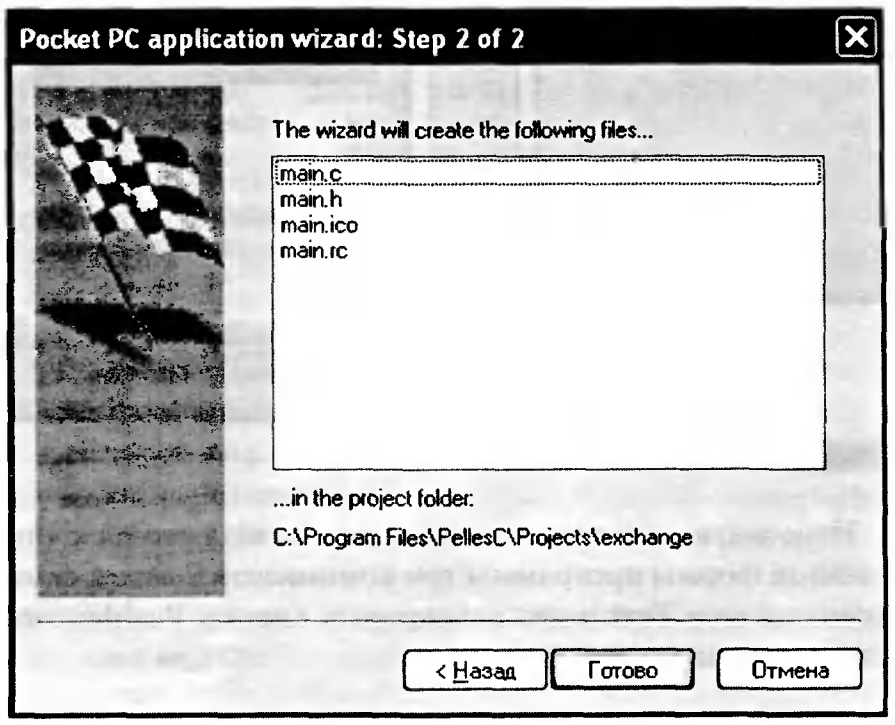


Рис. 14

Для завершения работы мастера приложений в этом окне следует нажать программную кнопку «Готово». После этого в правой части окна среды разработки отобразится состав созданного проекта в виде структуры каталога и перечня файлов с иконками (рис. 15).

Познакомиться с основными понятиями проектов для Windows можно в [4].

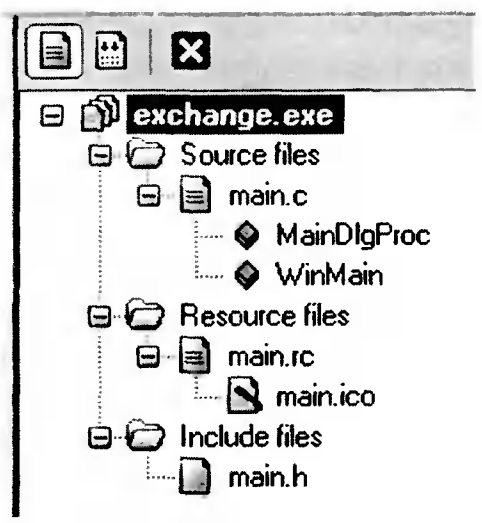


Рис. 15

Для редактирования ресурсов программы дважды щелкните по иконке и именем файла main.rc. При этом в левой части окна среды разработки отобразится раскрывающийся состав данного файла (рис. 16).

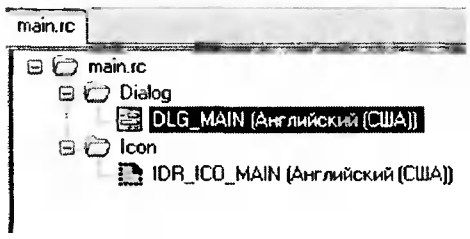


Рис. 16

Здесь необходимо дважды щелкнуть по названию файла DLG_MAIN для раскрытия его содержимого и последующего редактирования. При этом откроется окно для редактирования данного ресурса (рис. 17). Для удобства редактирования раскройте и расположите элементы данного окна по своему вкусу.

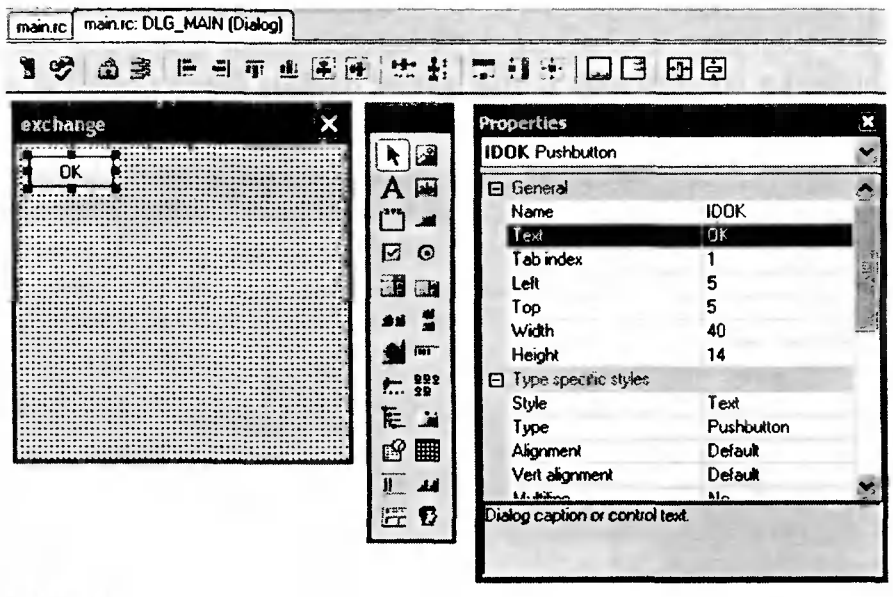


Рис. 17

Используя палитру компонентов, поместите на окно главной формы программы три компонента Editbox, семь компонентов Text и два компонента кнопок Pushbutton, как показано на рис. 18.

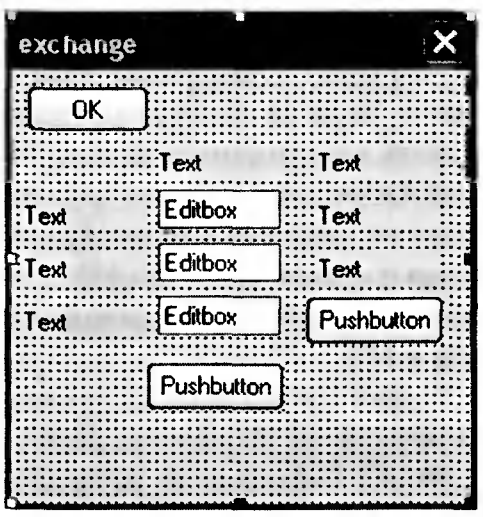


Рис. 18

Теперь необходимо изменить свойства Name и Text помещенных на форму программы компонентов. Для этого необходимо щелкнуть по компоненту подлежащему редактированию свойств, а затем дважды щелкнуть в окне свойств по строке, подлежащей редактированию, и ввести новые значения. Например, для кнопки OK следует заменить в свойстве Text слово «OK» на слово «Выход». Для остальных компонентов необходимо произвести изменения свойств в соответствии с табл. 2.

Таблица 2. Новые свойства компонентов программы

Компонент	Свойство	
	Text	Name
Text	Курс	Не изменять
Text	Сумма	Не изменять
Text	Доллар США	Не изменять
Text	Евро	Не изменять
Text	Рубли	Не изменять
Text	0	IDTEXT1
Text	0	IDTEXT2
Editbox	26,5	IDEDIT1
Editbox	34,5	IDEDIT2
Editbox	0	IDEDIT3
Pushbutton	Обмен	IDRUN
Pushbutton	Сброс	IDINIT

Это позволит инициализировать начальные значения, сформировать осмысленное содержание формы программы и использовать в программе заданные имена компонентов для обращения к ним. В результате такого редактирования должно получиться окно формы, показанное на рис. 19 вместе с окном редактирования свойств компонента кнопки.

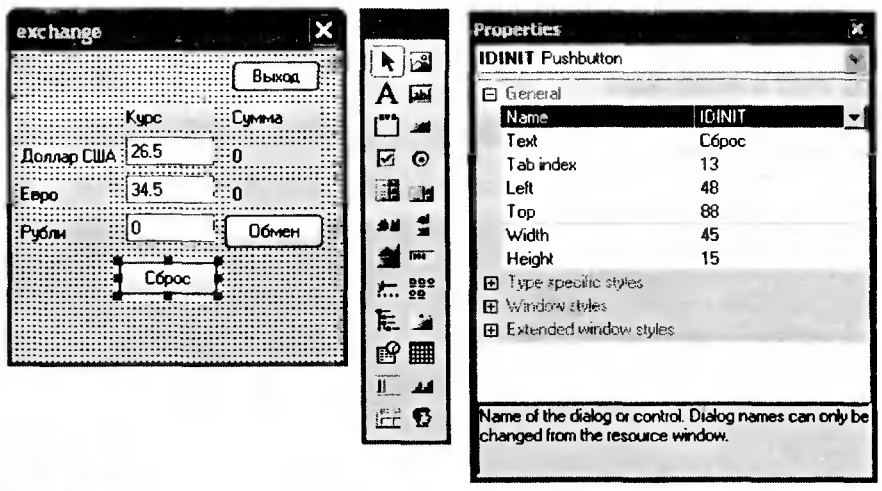


Рис. 19

Теперь можно выполнить трансляцию программы с помощью команд меню Project->Build или нажав на клавиатуре компьютера клавиши Ctrl и B. При успешной трансляции в окне сообщений, расположенном внизу интерфейса разработки, должны последовательно появиться строки сообщений:

```
Building main.obj.
Building main.res.
Building exchange.exe.
Done.
```

При этом средой разработки будет создан исполняемый код программы exchange.exe и код программы main.c. Код программы можно просмотреть и отредактировать, щелкнув дважды по имени файла main.c в правом окне среды разработки. При этом откроется окно редактирования кода, текст которого приведен в листинге 1.

Вставьте в текст этой программы после строки #include “main.h” строку комментария и две строки макроопределений, которые будут использоваться в программе (листинг 2).

После строки комментария /* TODO: Add more control ID’s, when needed.*/, предлагающей ввести код программы для обработки дополнительных кнопок программы, введите текстовые строки программы из листинга 3.

Листинг 1

```
//*************************************************************
// File   : main.c
//
// Purpose : Generic dialog based Pocket PC application.
//
// History : Date    Reason
// 00/00/00 Created
//
//*************************************************************

#define UNICODE
#include <windows.h>
#include <windowsx.h>
#include <aygshell.h>
#include "main.h"


/** Prototypes ************************************************/
static LRESULT CALLBACK MainDlgProc(HWND, UINT, WPARAM,
LPARAM);

/** Global variables ******************************************/
static HANDLE ghInstance;

//*************************************************************
// Function: WinMain
// Purpose : Initialize the application. Register a window
class,
//       create and display the main window and enter the
//       message loop.
// History : Date    Reason
// 00/00/00 Created
//*************************************************************
int PASCAL WinMain(HINSTANCE hInstance, HINSTANCE
hPrevInstance, LPWSTR lpszCmdLine, int nCmdShow)
{
  WNDCLASS wc;
  HWND hwnd;

  /* Always check first if program is already running */
  hwnd = FindWindow(L"exchangeClass", NULL);
  if (hwnd)
  {
    /*
     * Set focus to the foremost child window. The "|0x01"
is used to
     * bring any owned windows to the foreground and
activate them.
     */
    SetForegroundWindow((HWND)((ULONG)hwnd|0x00000001));
    return 0;
  }

  ghInstance = hInstance;

  // Get system dialog information.
  if (!GetClassInfo(NULL, L"Dialog", &wc))
    return 0;

  // Register our own dialog.
  wc.hInstance = hInstance;
  wc.hIcon = LoadIcon(hInstance, MAKEINTRESOURCE(IDR_ICO_
MAIN));
  wc.lpszClassName = L"exchangeClass";
  if (!RegisterClass(&wc))
    return 0;

  // The user interface is a modal dialog box.
  return DialogBox(hInstance, MAKEINTRESOURCE(DLG_MAIN), NULL,
(DLGPROC)MainDlgProc);
}

//*************************************************************
//
// Function: MainDlgProc
// Purpose : Process messages for the Main dialog.
// History : Date    Reason
// 00/00/00 Created
//*************************************************************

static LRESULT CALLBACK MainDlgProc(HWND hwndDlg, UINT uMsg,
WPARAM wParam, LPARAM lParam)
{
  switch (uMsg)
  {
    case WM_INITDIALOG:
    {
      SHINITDLGINFO shidi;

      /*
       * Create a Done button and size the dialog.
       */
      shidi.dwMask = SHIDIM_FLAGS;
      shidi.dwFlags = SHIDIF_DONEBUTTON|SHIDIF_
SIPDOWN|SHIDIF_SIZEDLGFULLSCREEN;
      shidi.hDlg = hwndDlg;
      SHInitDialog(&shidi);
      return TRUE;
    }

    case WM_SIZE:
      /*
       * TODO: Add code to process resizing, when needed.
       */
    return TRUE;

    case WM_COMMAND:
      switch (GET_WM_COMMAND_ID(wParam, lParam))
      {
        /*
         * TODO: Add more control ID's, when needed.
         */

       case IDOK:
            EndDialog(hwndDlg, TRUE);
            return TRUE;
       }
       break;

    case WM_CLOSE:
      EndDialog(hwndDlg, 0);
      return TRUE;
  }

  return FALSE;
}
```

Листинг 2

```
/* Пользовательские макроопределения */
#define NELEMS(a) (sizeof(a) / sizeof((a)[0]))
#define SetDlgItemFocus(hwnd,idCtl) FORWARD_WM_
NEXTDLGCTL((hwnd), GetDlgItem((hwnd),(idCtl)), TRUE, PostMes-
sage)
```

цифровых клавиш сенсорной виртуальной клавиатуры произвольную сумму и щелкните стилусом по кнопке «Обмен». При этом программа произведет вычисления и отобразит под текстом «Сумма» количество долларов и евро, которые можно обменять на введенную сумму рублей. На рис. 20 приведено окно с результатом работы программы.

Листинг 3

```
// Начало дополнительного кода программы
case IDRUN: // Если нажали кнопку "Обмен"
{
WCHAR wszBuffer[40], *pwszEnd; // Организовать буфер ввода и
указатель конца буфера
double Dollar; // переменная для долларов
double Evro; // переменная для евро
double Rubl; // переменная для рублей

// Считать доллары как строку из диалогового компонента
GetDlgItemText(hwndDlg, IDEDIT1, wszBuffer, NELEMS(wszBuffer));

// Конвертировать в переменную плавающего типа
Dollar = wcstod(wszBuffer, &pwszEnd);

// Проверить правильность ввода цифр
if (*pwszEnd != L'\0')
{
// Если значение неправильно. Отобразите сообщение об ошибке,
// и выйти устанавить фокус на ошибку ввода
// Сообщения пишем на английском языке
MessageBox(hwndDlg,
L"Sorry, can't understand this value. Try again.",
L"Input error",
MB_OK|MB_SETFOREGROUND);
SetDlgItemFocus(hwndDlg, IDEDIT1);
break;
}

// Считать evro как строку из диалогового компонента
GetDlgItemText(hwndDlg, IDEDIT2, wszBuffer, NELEMS(wszBuffer));

// Конвертировать в переменную плавающего типа
Evro = wcstod(wszBuffer, &pwszEnd);

// Проверить правильность ввода цифр
if (*pwszEnd != L'\0')
{
// Если значение неправильно. Отобразите сообщение об ошибке,
// и выйти устанавить фокус на ошибку ввода
MessageBox(hwndDlg,
L"Sorry, can't understand this value. Try again.",
L"Input error",
MB_OK|MB_SETFOREGROUND);
SetDlgItemFocus(hwndDlg, IDEDIT2);
break;
}

// Считать рубли как строку из диалогового компонента
GetDlgItemText(hwndDlg, IDEDIT3, wszBuffer, NELEMS(wszBuffer));

// Конвертировать в переменную плавающего типа
Rubl = wcstod(wszBuffer, &pwszEnd);

// Проверить правильность ввода цифр
if (*pwszEnd != L'\0')
{
// Если значение неправильно. Отобразите сообщение об ошибке,
// и выйти устанавить фокус на ошибку ввода
MessageBox(hwndDlg,
L"Sorry, can't understand this value. Try again.",
L"Input error",
MB_OK|MB_SETFOREGROUND);
SetDlgItemFocus(hwndDlg, IDEDIT3);
break;
}

// Произвести вычисления
Dollar = Rubl/Dollar; // сколько долларов можно купить за
рубли
Evro = Rubl/Evro; // сколько евро можно купить за рубли

// Преобразовайть результат в строку
swprintf(wszBuffer, L"%G", Dollar);
// Поместить строку в элемент текта диалога для долларов
SetDlgItemText(hwndDlg, IDTEXT1, wszBuffer);
// Аналогичные действия выполнить для евро
swprintf(wszBuffer, L"%G", Evro);
SetDlgItemText(hwndDlg, IDTEXT2, wszBuffer);
break;
}

case IDINIT: // Если нажата кнопка "Сброс"
{
SetDlgItemText(hwndDlg, IDTEXT1, L" ");
SetDlgItemText(hwndDlg, IDTEXT2, L" ");
break;
}
// Конец дополнительного кода программы
```

Данный листинг снабжен комментариями, поясняющими действия, выполняемые программой, которая обрабатывает ввод данных, нажатие программных кнопок и отображение результата вычислений.

На этом редактирование программы заканчивается и можно выполнить ее окончательную трансляцию, нажав клавиши Ctrl и B.

После успешной трансляции программы загрузите файл с ее исполняемым кодом exchange.exe в КПК и запустите программу. Для загрузки программы в КПК используется специальный загрузочный кабель связи КПК с персональным компьютером и программа ActiveSync, поставляемые в комплекте с КПК. После запуска программы на КПК введите в поле «Рубли» с помощью

Подобное окно работы программы КПК можно получить с помощью самой среды разработки. Для этого необходимо выполнить команды главного меню интерфейса разработки Tools->Capture Screen from Pocket PC и в открывшемся окне программы выполнить команды File->New bitmap.

Как было сказано в начале, среда разработки позволяет создавать программы не только для КПК, но также для смартфонов и коммуникаторов с операционной системой Windows CE или Windows Mobile. Принципиальных отличий в разработке программ для этих устройств нет. Необходимо лишь при создании проекта использовать соответствующий мастер проектов для смартфона.

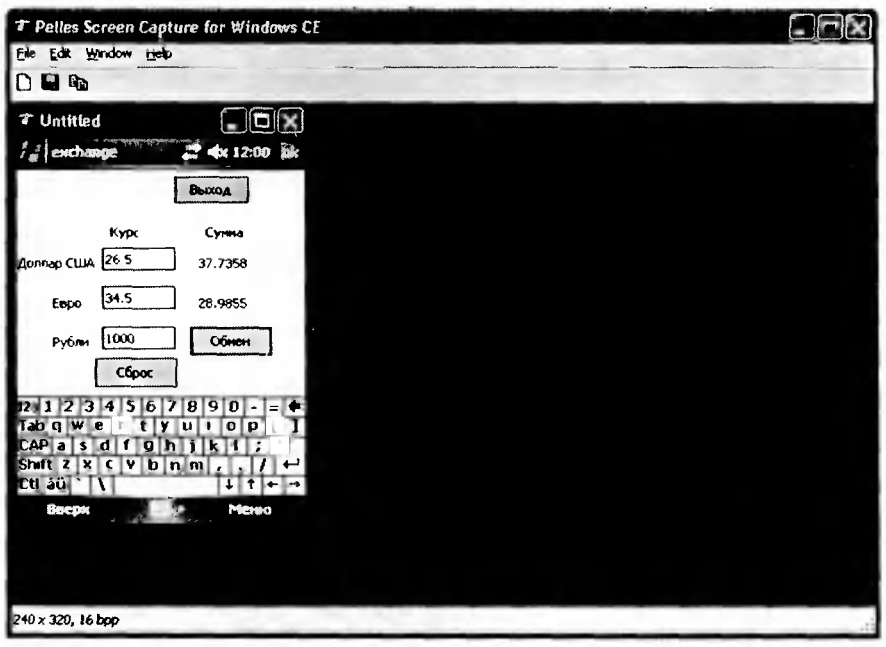


Рис. 20

Помимо основных средств для программирования, среда разработки предоставляет разработчику встроенные редакторы ресурсов программы, позволяющие создавать иконки и другие рисунки, файлы разных форматов и прочие объекты, используемые разрабатываемой программой.

Аналогично рассмотренному примеру можно создавать различные программы специального назначения для решения возникающих в жизни задач с помощью КПК, смартфона или коммуникатора.

ЛИТЕРАТУРА:

1. *http://www.microsoft.com/*
2. *http://www.smorgasbordet.com/pellesc/*
3. *http://www.smorgasbordet.com/pellesc/index.htm*
4. *Вальпа О.Д. Borland C++ Builder. Экспресс-курс — СПб.: BHV, 2006. —224с.: ил.*

Олег Вальпа,
г. Миасс Челябинской обл.

Математическая батарейка

Сегодня в распоряжении радиолюбителей, для которых копаться в сложных технических проблемах — предмет хобби, имеется компьютер и разного типа САПР. Лет 20….30 тому назад такой оснащенности позавидовал бы любой ученый. Это позволяет радиолюбителям решать весьма нетривиальные задачи. В этой статье, представлены результаты математического анализа эквивалентной схемы батарейки с помощью математического САПР Maple-11.

Многие давно знакомы с математическими САПР (Mathematica, MathCAD, Maple, MatLAB) — это программы поразительной мощности. С их помощью можно решить практически любую задачу, если вы понимаете ее сущность и можете корректно ее описать. Творческий акт человека здесь сводится к поиску задачи и вводу ее в программу. Дальнейшие труды, как правило, сводятся к нажатию кнопки «ИСПОЛНИТЬ» и ожиданию результатов.

Недавно я приобрел диск с новейшей программой Maple-11. Возможности нового математического САПР я решил опробовать на батарейках. Тривиальная на первый взгляд задачка позволила несколько расширить свое представление о возможностях таких программ. Всем известно, какая полезная вещь батарейка. Однако за маленькой батарейкой кроется весьма нетривиальная математическая задачка, которая заставила меня покопаться во всех учебниках и Help Maple в поисках способа ее решения. Для многих читателей могут быть интересны полученные результаты.

Рассмотрим эквивалентную электрическую схему батарейки (рис. 1). Она содержит источник ЭДС и последовательный резистор внутреннего сопротивления батарейки, Rw. Известно, что величина ЭДС довольно постоянна и зависит в основном от материалов, использованных для изготовления анода и катода, а Rw увеличивается по мере разряда батарейки или аккумулятора. Из этого следует, что внутреннее сопротивление батарейки должно быть пропорционально израсходованной в нагрузке мощности.

Также известно, что при длительном хранении батарейка или аккумулятор подвержены саморазряду, что эквивалентно некоторому большому сопротивлению Rc, подключенному параллельно. Батарейка характеризуется также параметром C — емкостью батарейки в амперчасах. Очевидно, что при разряде батарейки в нагрузке будет рассеяна энергия не больше, чем емкость.

Исходя из этого, были составлены соответствующие уравнения. В результате было получено интегральное

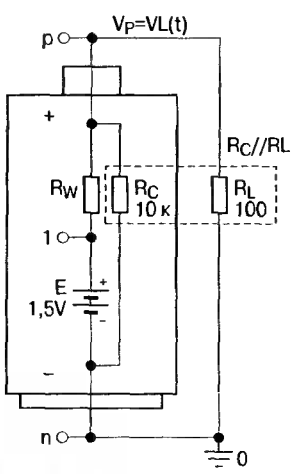


Рис. 1. Эквивалентная схема батарейки с саморазрядом

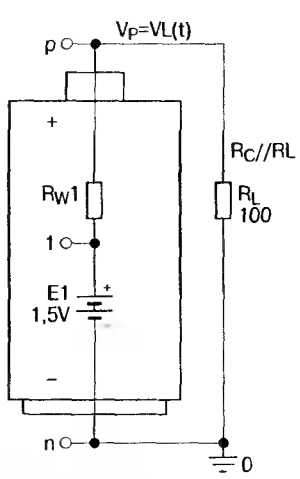


Рис. 2. Эквивалентная схема батарейки без саморазряда

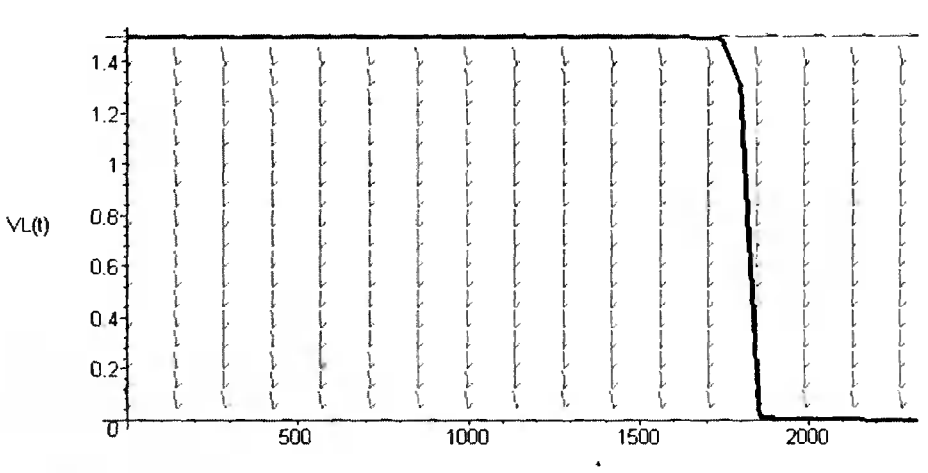


Рис. 3. Точное решение уравнения нагруженной батарейки

уравнение (1) характеризующее работу батарейки под нагрузкой. Слева стоит член, выражающий внутреннее сопротивление батарейки исходя из законов сохранения энергии, справа — исходя из законов Кирхгофа, действующих в схемах на рис. 1 и 2. Из этого уравнения надо найти VL(t). Это было сделано с помощью математического САПР Maple-11. Заметим, что величина C может характеризовать любой энергоноситель, в том числе и $m \cdot C^2$.

$$\frac{1}{C-\int_0^t \frac{VL(t)^2(R_c+RL)}{RL\cdot R_c}dt} = -\frac{RL\cdot R_c}{R_c+RL} + \frac{E\cdot R_c\cdot RL}{(R_c+RL)VL(t)} \quad (1)$$

Было получено точное аналитическое решение (2) и построен его график (рис. 3).

$$VL(t) = \left[VL(t) = -e^{\text{RootOf}(2_Ze^{(2_Z)}-2_Ze^{-Z}E+tE^3e^{-Z}-tE^2e^{(2_Z)}+E^2-2e^{-Z}} \times \right.$$
$$\left. \times 1^{E+2\ln(-e^{-Z}+E)Ee^{-Z}-2\ln(-e^{-Z}+E)e^{(2_Z)}+_ClE^3e^{-Z}-_ClE^3e^{(2_Z)})+E} \right] \quad (2)$$

Однако оказалось, что точное решение настолько сложное (здесь приведена сокращенная форма записи уравнения в терминах Maple), что использовать его невозможно. Это выражение может посчитать только Maple. Для практических целей интереснее иметь упрощенное приближенное решение, которое пригодно для расчетов.

ПРИБЛИЖЕННОЕ РЕШЕНИЕ УРАВНЕНИЯ БАТАРЕЙКИ

Если внимательно посмотреть на график точного решения и на уравнение (1), то можно заметить, что напряжение на нагрузке практически равно величине ЭДС почти до полного разряда. Тогда при расчете интеграла приближенно можно принять, что VL(t)=E. Выполнив эту подстановку, получим упрощенное уравнение (3).

$$\frac{1}{C-\frac{E^2(R_c+RL)t}{RL\cdot R_c}} = -\frac{RL\cdot R_c}{R_c+RL} + \frac{E\cdot R_c\cdot RL}{(R_c+RL)VL(t)}. \quad (3)$$

Решив это уравнение относительно VL(t), получим упрощенную формулу для напряжения нагруженной батарейки.

$$VL(t) := \frac{E(C\cdot RL\cdot R_c - E^2t\cdot R_c - E^2\cdot t\cdot RL)}{R_c+RL+C\cdot RL\cdot R_c - E^2\cdot t\cdot R_c - E^2\cdot t\cdot RL}. \quad (4)$$

Подставим типовые исходные данные и построим график упрощенного решения (рис. 4).

Как видно из графика приближенное решение справедливо не во всем временном диапазоне. Однако эта формула достаточно точна в диапазоне времен работоспособности батареи и позволяет решать важные практические задачи с помощью калькулятора, а не персонального компьютера.

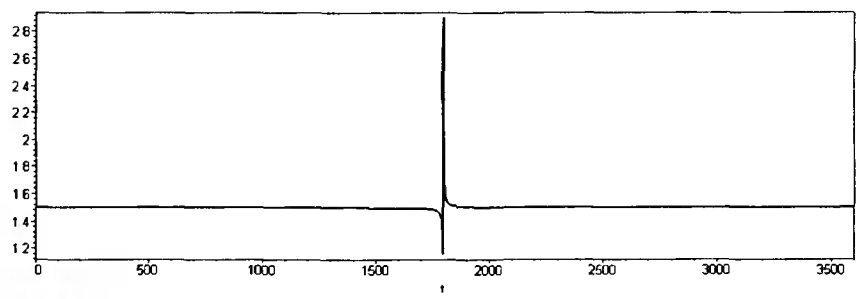


Рис. 4. Упрощенное решение уравнения нагруженной батарейки.

РЕШЕНИЕ ПРАКТИЧЕСКИХ ЗАДАЧ

Задача 1. Прогнозирование

Какой емкости должна быть новая батарейка в амперчасах, чтобы на нагрузке 1,5 Ом в течение часа было напряжение не менее 0,8 В? Для этого решим упрощенное уравнение (3) относительно C и подставим исходные данные:

$$C := = \frac{-E^2\cdot t\cdot R_c\cdot VL(t)+E^3\cdot t\cdot RL\cdot VL(t)+E^3}{RL\cdot R_c(-VL(t)+E)} \times \frac{t\cdot RL+R_c\cdot VL(t)+RL\cdot VL(t)}{1}. \quad (5)$$

В результате получим, что C=1,5 А/ч.

О таком результате можно было бы догадаться, хотя вряд ли это для всех очевидно. По крайней мере, мы проверили, как работает приближенная формула. Таким образом, батарейка емкостью 1,5 А/ч будет давать ток 1 А в течение часа при напряжении 1,5 В.

Задача 2. Фотоаппарат

Сколько снимков можно сделать на фотоаппарате Canon A710s с комплектом двух батареек типа AA емкостью по 2700 мА/ч, если один снимок эквивалентен работе батареек на нагрузку 0,5 Ом в течение 1 с.

Ток саморазряда учитывать не будем, это важно только для электронных часов. Тогда уравнение (3) упростится до уравнения (6).

$$\frac{C\cdot RL-E^2t}{RL} = \frac{VL}{RL(-VL+E)}. \quad (6)$$

Решив уравнение (6), найдем время в секундах, численно равное количеству снимков.

$$t := \frac{-C\cdot RL\cdot VL+C\cdot RL\cdot E-VL}{(-VL+E)E^2}. \quad (7)$$

Подставим исходные данные:

E = 1,2 В — напряжение комплекта свежих батарей;

C = 2700 × 2 мА·ч — емкость батареек;

RL = 0,5 Ом — сопротивление нагрузки;

VL: = 0,8·E — обычно аккумуляторы допускают разряд до 0.8E.

Получим ответ: t = 658 снимков.

Формулу (7) можно использовать при закупке аккумуляторов, когда вы отправляетесь в отпуск и интенсивно снимаете.

Задача 3. Радиоприемник

Емкость новых батареек типа AA ссотавляет 2000 мА/ч. Сколько проработает радиоприемник с током потребления 25 мА от четырех свежих батареек (1,5 В), соединенных последовательно, если минимальное напряжение, при котором он работоспособен — 3 В? Ток саморазряда учитывать не будем, это важно только для электронных часов, таким образом, будем использовать уравнение (6).

Решим уравнение (6) относительно времени:

$$t := \frac{-C\cdot RL\cdot VL+C\cdot RL\cdot E-VL}{(-VL+E)E^2}. \quad (8)$$

Подставим исходные данные:

C = 2000×4 мА·ч — емкость четырех батареек;

E = 1,5×4 В — напряжение комплекта свежих батарей;

IL = 0,025 мА — ток нагрузки;

RL = E/IL — сопротивление нагрузки;

VL: = E/2 — минимальное напряжение, при котором приемник работает.

Получим время работы радиоприемника в секундах: t = 53333,3 с.

Время работы радиоприемника в часах получится, если поделить это число на 3600, t=14,82 ч. Таким образом, приемник проработает почти 15 часов. Формулу (8) удобно использовать для расчета запаса батареек, если вы отправляетесь в туристический поход по местам, где нет магазинов.

Задача 4. Зенитная ракета

Известно, что в ракетной технике избыточный вес недопустим, так как на это тратится много топлива. Тогда решим такую задачку. Электрооборудование зенитной ракеты потребляет 500 Вт мощности и сохраняет работоспособность при снижении напряжения питания до 24 В. Какой емкости (а значит и веса) должна быть батарея напряжением 36 В, чтобы обеспечить полет ракеты в течение 1 мин. Будем использовать приближенную формулу для расчета оценки нижней границы требуемой емкости, а затем уточним его по точному решению. Очевидно, что при таком малом времени работы саморазряд учитывать не следует и можно использовать уравнение (6).

Из уравнения 6 найдем формулу для емкости (9)

$$C := \frac{-E^2 \cdot t \cdot VL + E^3 \cdot t + VL}{RL(-VL + E)}. \qquad (9)$$

Исходные данные: PL = 500 Ом, VL_min = 24 В, E = = 36 В, t = 60.

Сопротивление нагрузки определим из необходимости потреблять мощность 500 Вт даже при минимально допустимом напряжении, т. е RL = 24/500 = 1,152 Ом. Тогда C = 67501,73611 мА·ч.

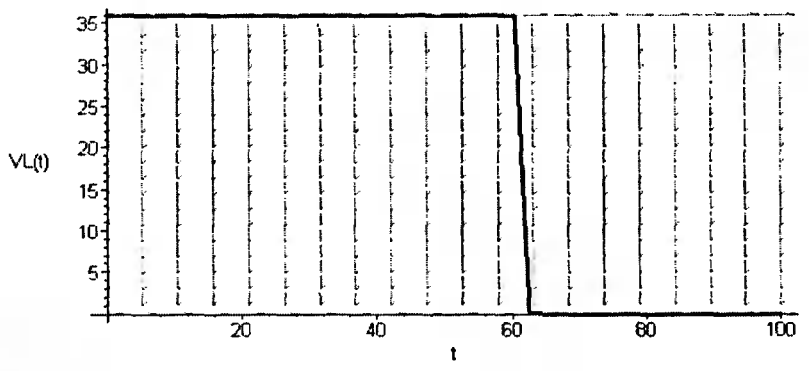


Рис. 5. Точное решение разряда батареи зенитной ракеты

Таким образом, нам нужна батарея емкостью не менее 67,5 А/ч. Однако проектирование технических систем без определенного запаса прочности практикуется исключительно редко. Округлив до большего целого, выберем емкость 70 А/ч.

В виду особой важности задачи была выполнена проверка качества решения по точной формуле (2). При этом программой Maple был построен график (рис. 5).

ЛИТЕРАТУРА:

1. В. П. Дьяконов. Maple-9 в математике, физике, образовании. — М.: СОЛОН-Пресс, 2004.

2. С. И. Баскаков. Лекции по теории цепей. — М.: Изд-во МЭИ, 1991.

3. А. Ф. Белецкий. Теория линейных электрических цепей. — М.; Радио и связь, 1986.

4. Аналитические расчеты в электронике. — Схемотехника, 2006. №7—12. 2007, №1—3.

5. http://pspicelib.narod.ru

6. http://pspice.comtv.ru

Олег Петраков,
г. Москва

Новые книги

А. В. Фрунзе

МИКРОКОНТРОЛЛЕРЫ ФИРМЫ PHILIPS СЕМЕЙСТВА x51

Издательский дом «Скимен» выпустил первый том книги Александра Фрунзе «Микроконтроллеры фирмы Philips семейства x51». В этой книге приведен подробный обзор всех выпускаемых фирмой Philips микроконтроллеров, дана информация по доступным отечественному пользователю микроконтроллерам семейства x51, изготавливаемым другими ведущими мировыми производителями. В ней приведена самая свежая на момент выпуска сводная таблица по всем выпускаемым Philips микроконтроллерам, а также терминологический словарь. Далее подробно рассматриваются особенности классических микроконтроллеров семейства x51 — система команд (подробно описаны все команды микроконтроллеров с кодами операций, операндами, примерами дей-ствия команд, временем их исполнения) и аппаратные особенности (структура памяти и регистров, арифметическо-логическое устройство, таймеры-счетчики и режимы их работы, система прерываний, стандартный последовательный порт со всеми режимами работы и система управления питанием). Подробно описаны микроконтроллеры с разработанным фирмой Philips оригинальным ядром 80C51+, в том числе микроконтроллеры с флэш-памятью программ, с программируемой матрицей счетчиков, с расширенной памятью программ и данных. Рассмотрены также микроконтроллеры с дополнительными портами ввода/вывода, с аппаратно реализованной шиной I^2C и встроенным аналого-цифровым преобразователем. Вся приведенная информация основана на справочных материалах фирмы Philips.

Книга рассчитана на широкий круг инженерно-технических работников, интересующихся последними достижениями микроконтроллерной техники.

Книга формата А4 имеет мягкую обложку и содержит 336 с.

Вы можете подписаться на эту книгу через редакцию. Стоимость подписки — 260 руб., в эту сумму включена пересылка по России.

Звездное небо на микроконтроллере PIC18F2525

Данная разработка универсальна — она может быть использована в качестве индикатора или светового табло, т. к. позволяет задавать яркость индивидуально для каждого из 64 светодиодов матрицы 8×8.
Автору понадобилось реализовать «звездное небо» — множество огоньков, расположенных на театральной декорации, которые плавно загораются и гаснут в случайном порядке, как, например, в заставке файлового менеджера Norton Commander.

В качестве источника света выбраны светодиоды, т. к. ими проще всего управлять, они долговечны. В продаже можно найти множество достаточно ярких белых или синих светодиодов, марка светодиода в данном случае не имеет особого значения. Светодиоды собраны в матрицу 8×8. Для реализации «звездного неба» матрица набирается проводами за декорациями, а если необходимо устройство индикации, можно использовать готовые матрицы светодиодов.

Основная проблема, с которой пришлось столкнуться в процессе разработки устройства — нелинейная зависимость субъективной яркости светодиода от ширины подаваемого на него импульса.

Был проведен эксперимент, в ходе которого на светодиод подавались импульсы со скважностью 1:240, 2:240, ..., X:240. Ширина импульса X задается программно, от нее зависит яркость светодиода. В ходе эксперимента оказалось, что увеличение яркости заметно при малых числах, а по мере увеличения числа X изменения яркости становятся незаметны. Это можно объяснить тем, что ток через светодиод зависит от X линейно, но относительное приращение X уменьшается с ростом значения числа X, кроме того, человеческий глаз воспринимает яркость объекта также нелинейно — чем больше яркость, тем меньше видны ее приращения. Для разрабатываемой схемы более-менее заметными ступенями получились числа до 10, далее яркость почти не меняется, а начиная с тридцати изменения и вовсе ничтожны. Исходя из этих соображений выбрано ограничение числа X на уровне 30, что позволило при общем числе ступеней 240 реализовать работу восьми горизонталей светодиодов по очереди.

В качестве основного контроллера выбран PIC18F2525. Это достаточно мощный микроконтроллер, в данном случае используется только малая часть его возможностей, но выбор оправдан наличием встроенного RC-генератора и узла умножения частоты, что позволило реализовать программный ШИМ, поскольку аппаратные схемы в данном случае неэффективны из-за большого числа каналов. Использование встроенного генератора с умножением частоты задает почти максимальную частоту тактирования для этого микроконтроллера — 36 МГц. Возможно применение и других микроконтроллеров, однако при уменьшении тактовой частоты возможно «мерцание» светодиодов.

Схема устройства приведена на рис. 1.

Устройство содержит контроллер, матрицу светодиодов, транзисторы для управления строками и стабилизатор напряжения питания. Контроллер не нуждается ни в кварцевом резонаторе, ни в формирователе сигнала сброса, т. к. имеет встроенные генератор и узел сброса.

Необходимость использования транзисторных ключей вызвана тем, что нагрузка на один порт контроллера не должна превышать 25 мА. Стабилизатор напряжения питания нужен как для устойчивой работы

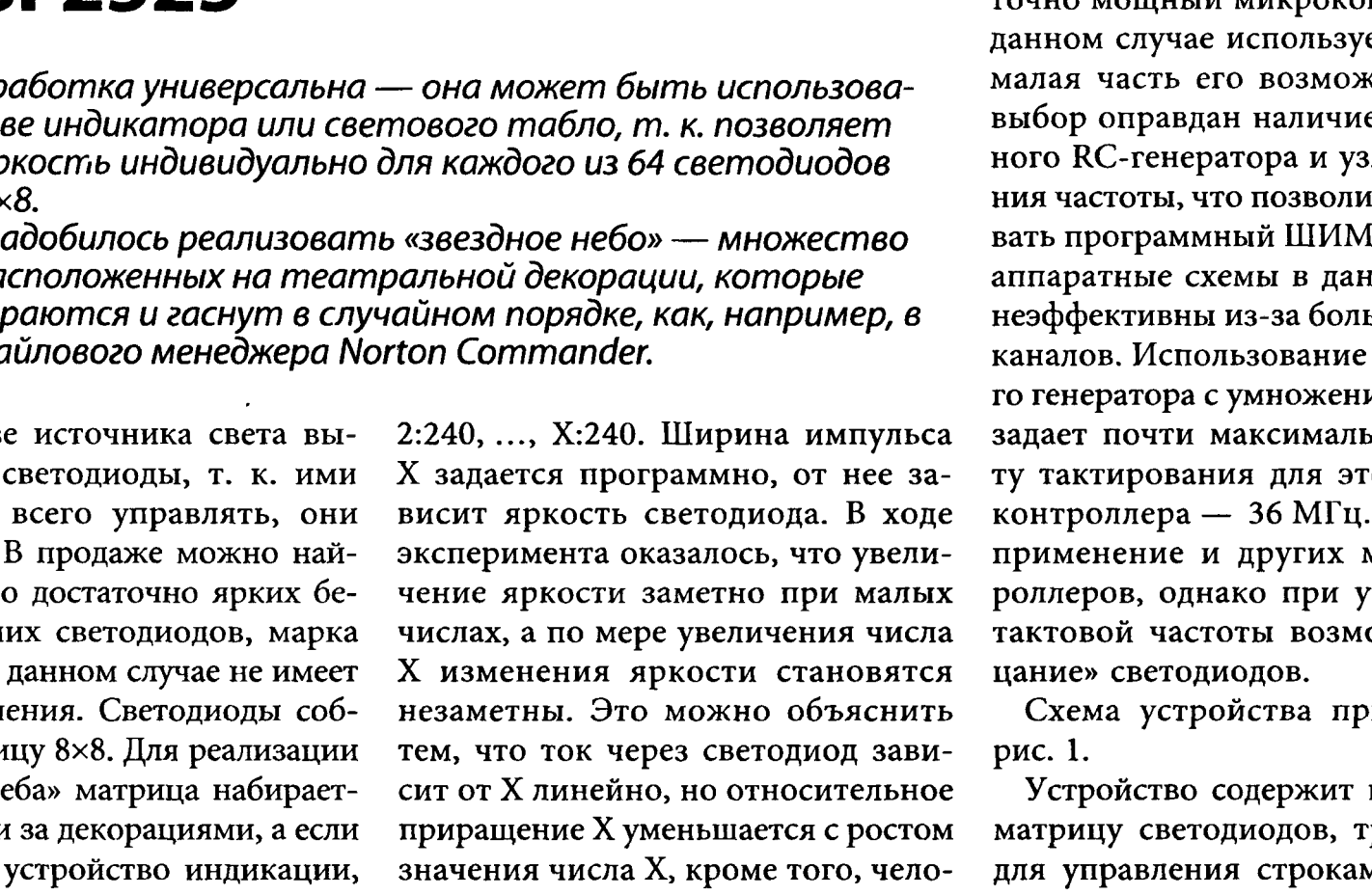


Рис. 1

Листинг 1

```
#include <pic18.h>
#include <stdlib.h>

__CONFIG(1, IESODIS & FCMDIS & RCIO);
__CONFIG(2, BOREN & BORV28 & PWRTEN & WDTPS32K);
__CONFIG(3, MCLREN & PBDIGITAL & LPT1DIS & CCP2RC1);
__CONFIG(4, XINSTDIS & DEBUGDIS & LVPDIS & STVREN);
__CONFIG(5, CPALL);
__CONFIG(6, WRTEN);
__CONFIG(7, 0xFFFF);

#define LEDNUM 64

char cpr;          // Текущий период
char bt;           // Выводимая вертикаль
char adbt;         // Добавляемое значение
char flg;

char cur[LEDNUM]; // Текущая яркость
int st[LEDNUM]; // Текущий делитель
// Бит 0 - идет нарастание
// Бит 2 - идет спад
// Бит 3 - пауза во включенном состоянии
// Бит 4 - пауза в выключенном состоянии
char f[LEDNUM]; // Текущее состояние

void interrupt IOinterrupt(void) {
  char * p;
  if (TMROIF) {
    TMROIF=0;
    if (cpr>30) {
      cpr=0;
      LATA|=0b00101110;
      LATB|=0b01111111;
      bt++;
      if (bt>8) bt=0;
      LATC=255;
      switch (bt) {
        case (0):LATC0=0;adbt=0;break;
        case (1):LATC1=0;adbt=8;break;
        case (2):LATC2=0;adbt=16;break;
        case (3):LATC3=0;adbt=24;break;
        case (4):LATC4=0;adbt=32;break;
        case (5):LATC5=0;adbt=40;break;
        case (6):LATC6=0;adbt=48;break;
        case (7):LATC7=0;adbt=56;break;
      }
      flg++;
    }
    p=cur+adbt;
    if (*(p++)>cpr) LATA1=0; else LATA1=1;
    if (*(p++)>cpr) LATA2=0; else LATA2=1;
    if (*(p++)>cpr) LATA3=0; else LATA3=1;
    if (*(p++)>cpr) LATA5=0; else LATA5=1;
    if (*(p++)>cpr) LATB6=0; else LATB6=1;
    if (*(p++)>cpr) LATB5=0; else LATB5=1;
    if (*(p++)>cpr) LATB4=0; else LATB4=1;
    if (*(p++)>cpr) LATB3=0; else LATB3=1;
    cpr++;
  }
}

void main() {
  char c;
  char d;
  char p;
  int * pst;
  char * pcur;
  char * pf;
// Настраиваем генератор
  OSCCON=0b01110000;
  OSCTUNE=0b11000000;
// Отключаем АЦП для 18F2525
  ADCON0=0b00000000;
  ADCON1=0B00001111;
  TRISA=0B00000000; // SCL,SDA - выходы
  TRISB=0B00000111;
  TRISC=0b00000000;
  LATC=1;
  PORTA=0;
  PORTB=0;
  PORTC=0xff;
// Датчик батареи
  LVDCON=0B00000101; //Отключаем
////////////////////////////////////////////
// ОБЯЗАТЕЛЬНО ДЛЯ ФУНКЦИОНИРОВАНИЯ ПРЕРЫВАНИЙ
// Таймер 0 - переполняться каждые 512 команд
  TOCON=0b11000000;
  TMROIP=1;
  TMROIE=1;
// Разрешаем прерывания и поехали!
  PEIE=1;
  GIE=1;
////////////////////////////////////////////
  for (c=0;c<LEDNUM;c++) {
    f[c]=1;st[c]=0;cur[c]=rand()&31;
  }
  for (;;) {
    CLRWDT();
    if (flg) {
      flg--;
      pcur=cur;
      pst=st;
      pf=f;
      for (c=0;c<LEDNUM;c++) {
// Выясняем состояние
        if (*pst) {
          (*pst)--;
        } else {
          if ((*pf)&1) {
// Увеличение яркости
            if (*pcur<20) p=40-*pcur; else p=5;
            if (*pcur>=30) {
// Яркость достигнута
              (*pf)=4;
              (*pst)=((rand()&0x03ff)*3); // Время «горе-
ния» светодиода
              (*pst)>>=1;
            } else {
              (*pcur)++;
              (*pst)=p;
            }
          } else if ((*pf)&2) {
// Уменьшение яркости
            if (*pcur<20) p=40-*pcur; else p=5;
            if (*pcur==0) {
// Яркость достигнута
              (*pf)=8;
              (*pst)=((rand()&0x07ff)*3); // Время «не го-
рения» светодиода
              (*pst)>>=1;
            } else {
              (*pcur)--;
              (*pst)=p;
            }
          } else if ((*pf)&4) {
            (*pst)=40;
            (*pf)=2;
          } else if ((*pf)&8) {
            (*pst)=40;
            (*pf)=1;
          }
        }
        pf++;pst++;pcur++;
      }
    }
  }
};
```

микроконтроллера, так и для стабилизации яркости свечения светодиодов. Следует помнить, что потребляемый ток может доходить до 200 мА, поэтому применять миниатюрные стабилизаторы нежелательно.

Резисторы R1—R8 задают ток светодиодов, несколько смягчая зависимость яркости светодиода от ширины импульса. Практика показывает, что устройство работает и без них — в этом случае ограничителем тока служат выходные узлы портов микроконтроллера. Кроме того, при использовании светодиодов белого свечения сопротивление резисторов может понадобиться уменьшить или вовсе заменить резисторы перемычками из-за большого падения напряжения на светодиодах. Изменение сопротивления резисторов в пределах 0...200 Ом не вызывает существенного изменения яркости свечения светодиодов.

Программа написана на языке С и компилировалась с использованием компилятора HiTech C18 версии 9.50. Текст программы содержится в листинге 1.

Прошивка микроконтроллера, программа и другие необходимые файлы находятся на сайте редакции журнала по адресу www.dian.ru.

Процедура вывода регенерации индикатора реализована отдельно, для ее работы необходимы четыре глобальные переменные — cpr, bt, adbt, flg и массив значений яркости cur[]. Сама процедура полностью реализована внутри обработчика прерывания IOinterrupt. Как было указано выше, ШИМ реализуется программно. Для этого прерывание вызывается с помощью таймера 0, всегда с одной и той же частотой — 512 командных циклов, либо каждые 56,8 мкс. Каждый вызов прерывания задает один квант ШИМ. Регенерация проводится следующим образом. Выбирается вертикаль, для нее отсчитываются 30 ступеней, каждый раз светодиоды с яркостью меньшей, чем текущая ступень, гасятся, а с большей — зажигаются. Затем происходит переход к следующей вертикали.

Поскольку в матрице используется всего восемь вертикалей, общее число квантов автоматически получается равным 240. Эти 240 квантов длительностью по 56,8 мкс каждый задают общий период регенерации 13,65 мс, что дает частоту 73 Гц — этого вполне достаточно, чтобы глаз не замечал мерцания.

Основная программа может быть заменена любой другой за исключением разрешения прерываний и инициализации таймера, отмеченных в комментариях. Правда, прерывания занимают примерно половину процессорного времени, но оставшейся мощности микроконтроллера вполне должно хватить, например, для генерации каких-либо узоров. В данном случае основная программа, используя два дополнительных массива — st[] и f[] — реализует циклическое изменение яркости светодиода, где паузы во включенном и выключенном состоянии задаются, а скорость нарастания и спада фиксирована и зависит от текущей яркости, что позволяет сгладить нелинейность задания яркости. Массив st задает текущее состояние счетчика, а массив f — состояние светодиода. Светодиод может находиться в четырех состояниях — гореть, не гореть, гаснуть, зажигаться. Программа меняет яркость светодиода в зависимости от этих состояний, при этом время нахождения в состояниях «горит» и «не горит» задается случайным образом.

Алексей Брякин,
г. Пенза

Предлагаем организациям и физическим лицам оформить подписку на книгу

«X51-СОВМЕСТИМЫЕ МИКРОКОНТРОЛЛЕРЫ ФИРМЫ SILICON LABORATIRES (CYGNAL)»

Автор О. И. Николайчук, обложка мягкая, 640 стр.

Второе расширенное издание настоящей книги предназначено для ознакомления читателей с полным спектром x51-совместимых микроконтроллеров, выпускаемых в настоящее время фирмой SiLabs, которая объединилась с фирмой Cygnal.

В книге подробно рассмотрены архитектура, состав периферийных подсистем, структура регистров и особенности применения высокоинтегрированных систем обработки данных x51-совместимых микроконтроллеров со смешанной аналоговой и цифровой архитектурой, включающих высокопроизводительное микроконтроллерное ядро CIP-51, построенное по конвейерному принципу. Благодаря этому многие семейства микроконтроллеров фирмы SiLabs развивают пиковую производительность 25 MIPS, а одно из семейств — 50 и даже 100 MIPS, что является наивысшим результатом по производительности среди микроконтроллеров в мире. Это уникальное микроконтроллерное ядро оснащено не менее мощными аналого-цифровыми и цифро-аналоговыми узлами, расширенной цифровой периферией, включающей интерфейсы UART, SMBus (I^2C), SPI, CAN, USB, мощными подсистемами памяти и прерываний, а также усовершенствованными средствами внутрисистемного программирования и отладки. Таким образом, современные микроконтроллеры фирмы SiLabs (Cygnal) являются не только самыми высокопроизводительными, но и самыми мощными по составу аналоговой и цифровой периферии.

Во второе издание дополнительно введена информация о четырех новых семействах, которые появились с момента первого издания книги. Кроме того, исправлены некоторые неточности, допущенные в первом издании.

Книга рассчитана на широкий круг инженерно-технических работников, а также студентов и аспирантов, интересующихся последними достижениями современной микроконтроллерной техники.

Стоимость подписки через редакцию на книгу — 250 руб., в эту стоимость включена стоимость пересылки по России.

Сводное содержание журнала «Схемотехника» за 2007 год

СОФТ

Баранов В. Применение моста USB-UART CP2102 №32
Баранов В. Применение моста USB-UART CP2102 №4
Баранов В. Использование USART микроконтроллера для обеспечения связи с компьютером №6
Баранов В. Использование USART микроконтроллера для обеспечения связи с компьютером №7
Баранов В. Использование USART микроконтроллера для обеспечения связи с компьютером №8
Баранов В. Использование USART микроконтроллера для обеспечения связи с компьютером №9
Вальпа О. Flash-память №1
Вальпа О. Имитатор шины ST-Bus №1
Вальпа О. Flash-память №2
Вальпа О. Микросхемы Flash-памяти типа NAND №3
Вальпа О. Карты flash-памяти xD-Picture №4
Вальпа О. Карты flash-памяти Smart Media №5
Вальпа О. Карты flash-памяти PC Card №6
Вальпа О. Карты памяти Multi Media Card №7
Вальпа О. Карты flash-памяти Compact Flash №8
Вальпа О. Карты flash-памяти Memory Stick №9
Вальпа О. Методика быстрого создания программ для карманных персональных компьютеров, смартфонов и коммуникаторов №12
Дьяконов В. Работа цифровых осциллографов TDS 1000 B/2000 B с системой компьютеров математики MatLab №7
Заиграев Н. Микроконтроллеры. Протокол MODBUS №5
Лозицкий С. Схемотехнические САПР: возможности и проблемы эффективного использования (продолжение полемики) №2
Лозицкий С. Схемотехнические САПР: возможности и проблемы эффективного использования (продолжение полемики) №3
Лозицкий С. Схемотехнические САПР: возможности и проблемы эффективного использования (продолжение полемики) №4
Лозицкий С. Схемотехнические САПР: возможности и проблемы эффективного использования (продолжение полемики) №5
Лозицкий С. Схемотехнические САПР: возможности и проблемы эффективного использования (продолжение полемики) №6
Лозицкий С. Схемотехнические САПР: возможности и проблемы эффективного использования (продолжение полемики) №9
Лозицкий С. Схемотехнические САПР: возможности и проблемы эффективного использования (продолжение полемики) №10
Лозицкий С. Схемотехнические САПР: возможности и проблемы эффективного использования (продолжение полемики) №11
Николаев А. Компиляторы, которые мы выбираем №2
Николаев А. Паскаль для PIC №5
Пахомов А. Согласование антенных усилителей с фидером снижения №6
Петраков О. Аналитические расчеты в электронике №1
Петраков О. Аналитические расчеты в электронике №2
Петраков О. Аналитические расчеты в электронике №3
Петраков О. Расчет электромагнитных компонентов в Transformer Designer Orcad V10.5 №7
Петраков О. Расчет электромагнитных компонентов в Transformer Designer Orcad V10.5 №8
Петраков О. Расчет электромагнитных компонентов в Transformer Designer Orcad V10.5 №9
Петраков О. Математическая батарейка №12
Прищепов Г. Полупроводниковый диод в среде Microcap №60
Прищепов Г. Полупроводниковый диод в среде Microcap №8
Родичкин С. Простая реализация протокола USB для ПК №5
Суров С. Контроллер с классическим ЖКИ и часами реального времени №7
Шевкопляс Б. Использование кода CRC для мультиплексирования основного и дополнительного каналов связи №2
Шевкопляс Б. Использование кода CRC №3
Шевкопляс Б. Повышение качества передачи голосовых сообщенийпо компьютерным сетям №9
Шишкин С. Таймеры на микроконтроллерах AT89C4051-24PI и AT902313-10PI №8

ИСКУССТВО СХЕМОТЕХНИКИ

Борисевич К. Помехоустойчивость КМОП-элементов №5
Борисевич К. Влияние длинных линий передачи на быстродействующие микросхемы КМОП №7
Борисевич К. Влияние длинных линий передачи на быстродействующие микросхемы КМОП №8
Борисевич К. Статические электрические характеристики быстродействующей КМОП-логики семейства MM74HC №9
Вальпа О. Устройство и восстановление электронных музыкальных синтезаторов №2
Вострухин А. Двухпозиционный терморегулятор с аналоговым входом на базе микроконтроллера ATtiny2313 №11
Дьяконов В. Импульсные устройства на обычных и программируемых однопереходных транзисторах №3
Дьяконов В. Импульсные устройства на обычных и программируемых однопереходных транзисторах №4
Ершов М. Реализация измерителя импенданса на основе дискретного преобразователя Фурье №4
Задорожный С. Линеаризованный активный смеситель №8
Задорожный С. Линеаризованный активный смеситель №9
Задорожный С. Использование не имеющей затуханий цепи отрицательной обратной связи для расширения динамического диапазона активного двойного балансного смесителя №10
Задорожный С. Использование не имеющей затуханий цепи отрицательной обратной связи для расширения динамического диапазона активного двойного балансного смесителя №11
Зобнин И. Новые способы вычислений на микроконтроллерах применительно к топливным расходомерам №10
Литаврин А. многоканальные усилительные структуры в усилителях мощности звуковой частоты №8
Литаврин А. многоканальные усилительные структуры в усилителях мощности звуковой частоты №9
Межлумян А. Влияние входного синфазного напряжения на работу компараторов и линейных усилителей на ОУ №6
Морозов В. Расчет и моделирование повышающе-понижающего преобразователя напряжения №4
Морозов В. Расчет и моделирование повышающе-понижающего преобразователя напряжения №5
Морозов В. Расчет и моделирование повышающе-понижающего преобразователя напряжения №6
Морозов В. Расчет и моделирование повышающе-понижающего преобразователя напряжения №7
Морозов В. Расчет и моделирование повышающе-понижающего преобразователя напряжения №8
Николайчук О. Использование цветного графического LCD от телефонов Nokia в микроконтроллерных системах №12
Олейник В. ЦАП или цифровой потенциометр? Выбор за вами №10
Пахомов А. Согласование антенных усилителей с фидером снижения №6
Петраков О. Увеличение выходного тока нагрузки высокоскоростных ОУ №6
Поляков В. Рамочно-лучевая или настоящая ЕН антенна №5
Прищепов Г. Квазирезонансные усилители и УПЧ №1
Прищепов Г. Миниатюрные УРУ №2
Прищепов Г. Квазирезонансное усиление низких часто в многокаскадных усилителях №4
Прокофьев В. Контроль достоверности в цифровых системах передачи №1
Реутов Ю. Синхронный детектор с удвоением напряжения №9
Фрунзе А. Вольтметр постоянного тока на ADuC824 №1
Фрунзе А. Вольтметр постоянного тока на ADuC824 №2

Шевкопляс Б. Кодонезависимый и кодозависимый джиттер (вандер) сигнала в линии связи №1
Шевкопляс Б. Высокоскоростные преобразователи кодов в телекоммуникационных устройствах №3
Шевкопляс Б. Восстановление искаженных пакетов данных без их повторной передачи по линии связи №4
Шевкопляс Б. Повышение надежности передачи пакетов данных в режиме CBR №5
Шевкопляс Б. Передача данных по фантомным цепям телекоммуникационных систем №6
Шевкопляс Б. Обнаружение узких мест трасс передачи данных в компьютерной сети и оценка их пропускной способности №7
Шевкопляс Б. Способ уменьшения объема буферной памяти узла компьютерной сети №8
Шевкопляс Б. Повышение качества передачи голосовых сообщений по компьютерным сетям №9
Шевкопляс Б. Повышение надежности асинхронной передачи по радиоканалу №10
Шуклин А. Беспроводные столкновения №11
Ушаков А., Пименов А., Землянов М. и др. Повышение надежности бортового специализированного комплекса телеметрии №4

ПРАКТИКА

Алексеев А. Электронное реле времени №9
Баранов В. Подключение программаторов к плате компьютера №10
Белоусов А. Устройство регулирования температуры воды в баке №4
Бондарь М. Улучшение основных параметров устройств выборки и хранения №2
Борисевич К. Генераторы на элементах КМОП №6
Брякин А. «Звездное небо» на микроконтроллере PIC18F2525 №6
Брякин А. Сувенир "Живое сердце" №12
Вальпа О. Простая система безопасности с каналом сотовой связи №11
Заец Н. Автомобильные часы, вольтметр, таймер на микроконтроллере PIC16F676 №8
Заец Н. Изготовление макетных плат №9
Зобнин И. Как изготовить неповреждаемый силовой трансформатор №9
Ершов М. Реализация измерителя импеданса на основе дискретного преобразования Фурье №3
Ефремов А. Простой УМЗЧ с выходным каскадом на транзисторах MOSFET №1
Ильин М., Круг А. Пассивные компоненты. Эффективная защита от перенапряжения №12
Колокольцев А, Дахнович А., Бабаев А. Функциональный генератор звукового диапазона №1
Квашин А. Freescale — применение технологий компании в рамках проекта «Мастер КИТ» №1
Кривецкий А. Переговорное устройство №1
Кривецкий А. Нормализатор напряжения сети №4
Кремлев М. Кодовый замок на микроконтроллере PIC16f84 №5
Литаврин А. МКУС в УМЗЧ с крайне глубокой стопроцентной ООС №8
Лютько А. Автомат управления освещением №5
Межлумян А. Напряжение на корпусе компьютера №9
Межлумян А. Не слишком ли дорого обходятся дешевые компьютерные блоки питания? №12
Мельник В. Четыре устройства на КР1533ИР23 №4
Морохин Л. Малогабаритная сирена №5
Мусиенко А. Таймер садовода-огородника №2
Нургалиев И., Шарков В. Терморегулятор с ПИД алгоритмом управления для охлаждения полупроводниковых лазеров №7
Олейник В. Управление световым потоком ламп дневного света №5
Олейник В. Регулируемый повышающий преобразователь напряжения без индуктивных элементов №6
Олейник В. Снижение мощности, потребляемой реле №7
Олейник В. Генератор многофазных синусоидальных колебаний №12
Пахоменков Ю. Преобразователь сигналов частотных датчиков №1
Пахоменков Ю. Преобразователь унифицированных токовых сигналов №3
Пахоменков Ю. Преобразователь сигналов шунтов в цепях постоянного тока №7
Петин г. ШИМ-регулятор на основе таймера и его применение в импульсных источниках питания №2
Петраков О. Микрофонный микшер №5
Поляков В. О ближнем поле случайных и суррогатных антенн №4
Поляков В. Простой узкополосный фильтр №8
Рудаков А., Рудаков М. Датчик уровня топлива на AVR-микроконтроллере №1
Садиков Ю. Автоматический включатель освещения на базе датчика движения №7
Садиков Ю. Источник бесперебойного питания с переключаемым выходным напряжением №9
Садиков Ю. Экономичный светильник на мощных светодиодных лампах для радиолюбительского рабочего места №12
Суров А. Автомобильный цифровой спидометр №1
Телеш Д. Простой измеритель ESR №8
Тречекас Й. автоматическое поддержание температуры в помещениях при помощи многоканального микропроцессорного таймера «Мастер КИТ» №4
Харитонов С. Шифратор и дешифратор команд числоимпульсного кода №3
Ходаковский И. Передача телеметрических данных по радиоканалу №2
Феколкин К. Микросборки бестрансформаторных AC/DC преобразователей Rohm №2
Феколкин К. Стабилизированные источники питания постоянного напряжения на базе микросборок ROHM №3
Феколкин К. Бестрансформаторные импульсные стабилизированные источники питания на основе микросборок ROHM №5
Феколкин К. Автомат для освещения дома и подъезда №6
Феколкин К. Электронный замок с бесконтактными ключами RFID №8
Фрунзе А. Преобразователь фототока для кремниевого фотодиода №3
Фрунзе А. Преобразователь фототока для кремниевого фотодиода №4
Шатохин Р. Драйвер управления униполярным шаговым двигателем от LPT порта с автоматическим переходом на ток удержания №6

КОМПОНЕНТЫ

Богданов С. Новая GPS технология u-blox5 №10
Дьяконов В. Современная лаборатория разработчика электронных схем №6
Дьяконов В. Современная лаборатория разработчика электронных схем №7
Дьяконов В. Современная лаборатория разработчика электронных схем №8
Ефимов Г. Системы бесконтактной идентификации для складских приложений и логистики №11
Зыбайло А. Датчики температуры Heraeus №10
Зыбайло А. Система измерения уровня давления в шинах №10
Зыбайло А. Магнетометры и компасы фирмы Honeywell №12
Бирюков С. Микросхемы для активных корректоров коэффициента мощности NCP1653 и NCP1653A №1
Бирюков С. Микросхемы для активных корректоров коэффициента мощности NCP1653 и NCP1653A №2
Бирюков С. Микросхемы для активных корректоров коэффициента мощности NCP1653 и NCP1653A №3
Бирюков С. Микросхемы для активных корректоров коэффициента мощности MC34261 и MC33261 №6
Бирюков С. Микросхемы для активных корректоров коэффициента мощности MC34261 и MC33261 №7
Корнев А. Новые силовые компоненты Mitsubishi Electric №10
Крюков М. Потенциометры для регулировки контраста №11
Николайчук О. Новое семейство микроконтроллеров C8051F36x фирмы Silicon Laboratories №11
Танигучи Н., Штумпф Т., Скачков А., Корнев А. L-серия IPM: низкие потери, низкие помехи, новое управление №10
Троицкий Ю., Нестеров А. Основные принципы построения светодиодных индикаторных панелей №2
Шишкин С. Система передачи данных на базе вторичных преобразователейи весовых терминалов с аналоговыми выходами №1

MAXIM DALLAS SEMICONDUCTOR

Infineon technologies

EPCOS

Honeywell

SICK

VISHAY

BOURNS Reliable Electronic Solutions

IOR

Tyco Electronics Our commitment Your advantage

Panasonic ideas for life

CRYDOM

DATA VISION

MITSUBISHI ELECTRIC

Kingbright

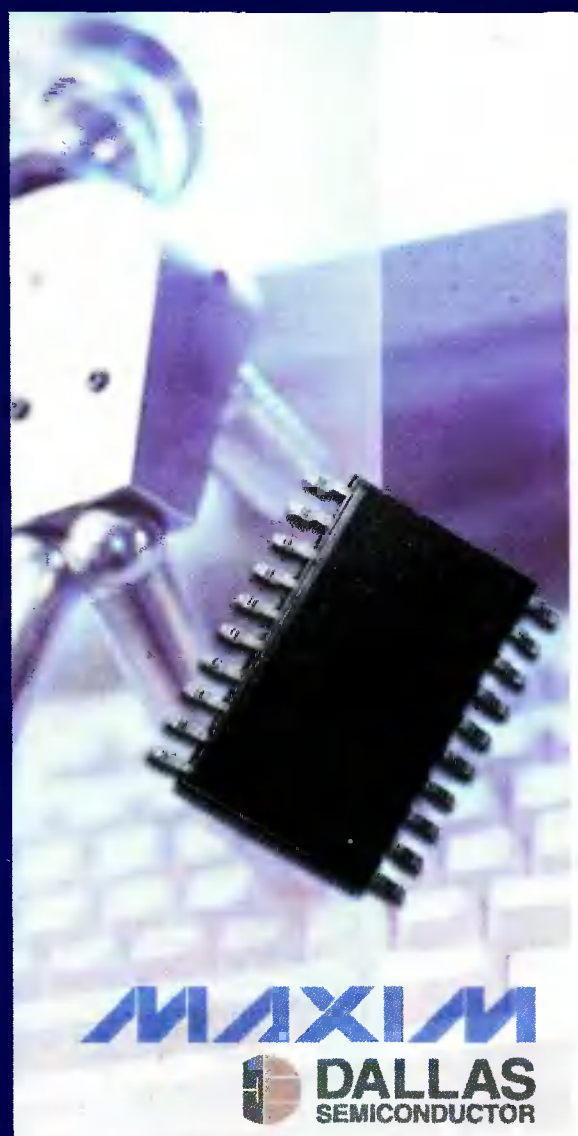
muRata Innovator in Electronics

velleman

ЭЛЕКТРОННЫЕ КОМПОНЕНТЫ ОТ ВЕДУЩИХ ПРОИЗВОДИТЕЛЕЙ

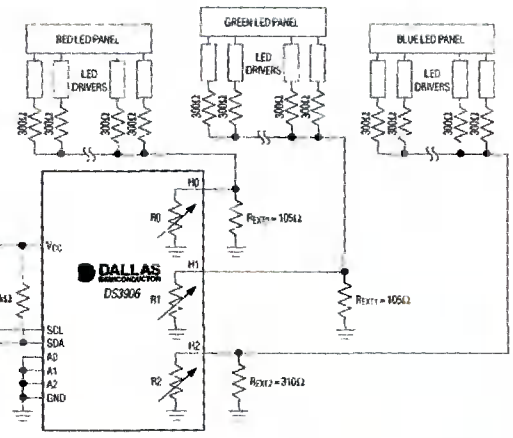
ЦИФРОВЫЕ ПОТЕНЦИОМЕТРЫ

MAXIM DALLAS SEMICONDUCTOR



цифровое интерфейсное управление

- контроль яркости, контрастности, громкости, усиления
- объединение до шести цифровых потенциометров в одном корпусе
- модели с энергонезависимой памятью, позволяющей сохранять установки движка при отключении питания
- модели с индивидуально программируемой зависимостью сопротивления от положения движка
- линейная и логарифмическая зависимость сопротивления от положения движка
- минимальные габаритные размеры со сверхнизким энергопотреблением



Наименование	Хар-ка	Энергонезависимая память	Число каналов	Интерфейс управления	Кол-во положений	Номинальное сопротивление, кОм	Допуск на сопротивление, %	Напряжение питания, В	Тип корпуса
DS1808	логариф.	нет	2	2-проводной	32	45	20	5, ±12	SOIC-16
MAX5410	логариф.	нет	2	3-проводной SPI	32	10	30	4.5...5.5	QSOP-16 QFN-16
DS1844	линейн.	нет	4	2-проводной или 5-проводной послед.	64	10 50 100	20	2.7...5.5	PDIP-20 SOIC-20 TSSOP-20
DS1803	линейная	нет	2	2-проводной	256	10 50 100	20	2.7...5.5	TSSOP-14 SOIC-16 PDIP-16
DS1666	логариф.	нет	1	инкремент/ декремент	128	10 50 100	20	±5	PDIP-14 SOIC-16
DS1267	линейная	нет	2	3-проводной послед.	256	10 50 100	20	4.5...5.5	PDIP-14 SOIC-16 TSSOP-20
MAX5414	линейная	нет	2	3-проводной SPI	256	50	25	2.7...5.5	TSSOP-14
DS1869	линейная	есть	1	переключ., процессор	64	10 50 100	20	2.7...8.0	PDIP-8 SOIC-8
DS1867	линейная	есть	2	3-проводной послед	256	10 50 100	20	4.5...5.5	PDIP-14 SOIC-16 TSSOP-20

www.platan.ru

ПЛАТАН

Офисы в Москве: **м. Молодежная**: ул.Ивана Франко, 40, стр.2, (495) 97-000-99, 121351, Москва, а/я 100, e-mail: platan@aha.ru
м. Новослободская: 1-й Щемиловский пер., 16, стр.2 (495) 744-70-70, platan@platan.ru

Офис в Санкт-Петербурге: ул. Зверинская, 44 (812) 232 88 36, 232 23 73, baltika@platan.spb.ru
Представительства: Воронеж: (4732) 59 75 57 **Казань:** (843) 292 18 06 **Киев:** (38044) 494 37 92 **Новосибирск:** (3832) 16 57 73 **Омск:** (3812) 24 69 03 **Томск:** (3822) 55 65 30 **Ульяновск:** (8422) 37 65 67 **Уфа:** (3472) 32 10 79
Региональные дилеры:Белгород: (4722) 31-30-84 **Екатеринбург:** (343) 353-75-16 **Ижевск:** (3412) 20-71-85 **Йошкар-Ола:** (8362) 45-17-45 **Красноярск:** (3912) 41-80-61 **Минск:** (375-17) 287-28-60 **Нижний Новгород:** (831) 430-32-33 **Новосибирск:** (383) 229-75-28 **Новосибирск:** (383) 290-90-40, 217-38-38 **Омск:** (3812) 24-10-90, 25-15-71, факс: (3812) 24-68-65 **Пермь:** (342) 237-17-46, 212-54-00 (многокан.) **Ростов-на-Дону:** (8632) 443-448 **Самара:** (846) 267-31-39
Санкт-Петербург: (812) 327-9692 **Саратов:** (8452) 606-888 **Тольятти:** (8482) 70-91-03 **Томск:** (3822) 41-78-16, 511-225 **Тюмень:** (3452) 75-11-17 **Уфа:** (347) 235-63-73 **Чебоксары:** (8352) 28-81-57 **Якутск:** (4112) 35-07-19, 43-32-86

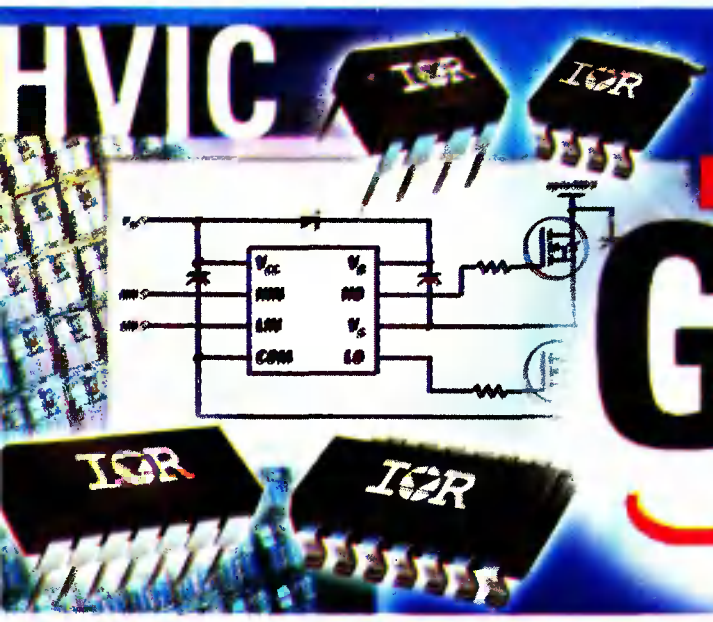
www.platan.ru

ПЛАТАН

International IOR Rectifier

ЭЛЕКТРОННЫЕ КОМПОНЕНТЫ ОТ ВЕДУЩИХ ПРОИЗВОДИТЕЛЕЙ

MAXIM DALLAS · Infineon technologies · Honeywell · EPCOS · VISHAY · BOURNS · ANALOG DEVICES · AMP · CRYDOM · Kingbright · SICK · DATA VISION



ВЫСОКОВОЛЬТНЫЕ ИС ДРАЙВЕРОВ МОП-затворов 5-го поколения

G5 HVIC

ПРИМЕНЕНИЕ

- электроприводы
- плазменные и ЖК панели
- светотехника
- аудио приложения
- импульсные источники питания

ПРЕИМУЩЕСТВА высокая плотность структуры, более компактный кристалл и более интегрированная ИС при равной площади кристалла · один или два входа · совместимость с логикой 3.3 В · ток накачки затвора до 2.5 А · более жесткие допуски на выходные параметры · раздельные выводы для сигнальной и силовой «земли» · блокировка по низкому напряжению сети · фиксированная или программируемая пауза на переключение для драйверов полумоста · интегральный бутстрепный транзистор для реализации бутстрепного источника питания · компактные DIP и SOIC-корпуса

МИКРОСХЕМЫ ДРАЙВЕРОВ G5 HVIC

В новых высоковольтных ИС использован усовершенствованный технологический процесс производства кристаллов G5HVIC, который реализует сдвиг высоковольтного уровня, обеспечивающий превосходные электрические характеристики, защиту от перегрузок и более высокий уровень надежности. Микросхемы подключаются к низковольтному источнику напряжения и обеспечивают управление затвором МОП и IGBT транзисторов с одновременной защитой высоковольтного оборудования. Помимо этого, монолитные ИС драйверов G5 HVIC обладают интегрированными функциями, позволяющими упростить схемотехническое решение и снизить конечную стоимость продукта. Выпускаются модели с интегрированным бутстрепным транзистором для реализации источника питания. Это позволит разработчикам отказаться от достаточно габаритного и дорогого внешнего источника питания, которого обычно требуют базовые схемы на дискретных компонентах (оптопарах или трансформаторах).

МАКСИМАЛЬНАЯ НАДЕЖНОСТЬ

Микросхемы 5-го поколения G5 HVIC прошли испытания на повышенные требования к надежности и аварийной работе. Микросхемы нормированы на срок эксплуатации, превышающий стандартный срок службы оборудования. Компоненты выдержали несколько испытаний на работу на отказ, включая высокотемпературный 2000 часовой тест, при котором температура p-n перехода превышала максимально допустимую по спецификации.

ДРАЙВЕРЫ ПОЛУМОСТА

Наименование	Кол-во выводов	Ток сток/исток, мА	Описание
IRS2103(S)PbF	8	290/600	блокировка по низкому напряжению питания
IRS2104(S)PbF	8	290/600	входная логика для выключения; блокировка по низкому напряжению питания
IRS2109(S)PbF	8	290/600	входная логика для выключения; блокировка по низкому напряжению питания и напряжению зарядного насоса
IRS2308(S)PbF	8	290/600	блокировка по низкому напряжению питания и напряжению зарядного насоса
IRS2111(S)PbF	8	290/600	блокировка по низкому напряжению питания и напряжению зарядного насоса
IRS21844(S)PbF	14	1900/2300	входная логика для выключения, программируемая пауза, блокировка по низкому напряжению питания и напряжению зарядного насоса

НЕЗАВИСИМЫЕ ДРАЙВЕРЫ ВЕРХНЕГО/НИЖНЕГО КЛЮЧЕЙ

Наименование	Кол-во выводов	Ток сток/исток, мА	Описание
IRS2101 (S)PbF	8	290/600	блокировка по низкому напряжению питания
IRS2106(S)PbF	8	290/600	блокировка по низкому напряжению питания и напряжению зарядного насоса
IRS2181(S)PbF	8	1900/2300	блокировка по низкому напряжению питания и напряжению зарядного насоса
IRS2110(S)PbF	16/14	2500/2500	входная логика для выключения; блокировка по низкому напряжению питания и напряжению зарядного насоса
IRS2113(S)PbF	16/14	2500/2500	входная логика для выключения; блокировка по низкому напряжению питания и напряжению зарядного насоса

www.platan.ru ПЛАТАН

Офисы в Москве: **м. Молодежная**: ул.Ивана Франко, 40, стр.2, (495) 97-000-99, 121351, Москва, а/я 100, e-mail: platan@aha.ru
м. Новослободская: 1-й Щемиловский пер., 16, стр.2 (495) 744-70-70, platan@platan.ru

Офис в Санкт-Петербурге: ул. Зверинская, 44 (812) 232 88 36, 232 23 73, baltika@platan.spb.ru
Представительства: Воронеж: (4732) 59 75 57 **Казань:** (843) 292 18 06 **Киев:** (38044) 494 37 92 **Новосибирск:** (3832) 16 57 73
Омск: (3812) 24 69 03 **Томск:** (3822) 55 65 30 **Ульяновск:** (8422) 37 65 67 **Уфа:** (3472) 32 10 79
Региональные дилеры:Белгород: (4722) 31-30-84 **Екатеринбург:** (343) 353-75-16 **Ижевск:** (3412) 20-71-85 **Йошкар-Ола:** (8362) 45-17-45 **Красноярск:** (3912) 41-80-61 **Минск:** (375-17) 287-28-60 **Нижний Новгород:** (831) 430-32-33 **Новосибирск:** (383) 229-75-28 **Новосибирск:** (383) 290-90-40, 217-38-38 **Омск:** (3812) 24-10-90, 25-15-71, факс: (3812) 24-68-65 **Пермь:** (342) 237-17-46, 212-54-00 (многокан.) **Ростов-на-Дону:** (8632) 443-448 **Самара:** (846) 267-31-39
Санкт-Петербург: (812) 327-9692 **Саратов:** (8452) 606-888 **Тольятти:** (8482) 70-91-03 **Томск:** (3822) 41-78-16, 511-225 **Тюмень:** (3452) 75-11-17 **Уфа:** (347) 235-63-73 **Чебоксары:** (8352) 28-81-57 **Якутск:** (4112) 35-07-19, 43-32-86