

Часы-календарь реального времени на основе МК LPC2129

Павел Редькин (г. Ульяновск)

Предлагаемая вниманию читателей статья завершает ранее опубликованный цикл по МК семейства LPC2000 (СЭ, 2006, №№ 7 – 9; 2007, №№ 1 – 5). В статье описывается разработанный на базе МК LPC2000 простой пользовательский проект – автономно работающие часы-календарь реального времени с кварцевой стабилизацией частоты. Кроме того, статья содержит информацию об организации интерфейса ввода-вывода целевой пользовательской системы, которая может быть полезна специалистам, желающим освоить МК семейства LPC2000.

Платформа для создания приложений

Как и для многих других популярных универсальных МК, на базе уст-

ройств LPC2000 ряд сторонних производителей выпускает различные отладочные комплекты, стартовые наборы разработчика, контроллеры-

конструкторы. Их удобно использовать как для первоначального освоения, так и для создания макетов целевых пользовательских систем.

В качестве базового устройства для создания своих приложений автор статьи выбрал МК LPC2129, а в качестве аппаратной платформы использовал недорогую и доступную отладочную плату LPC-H2129 производства фирмы Olimex [1], принципиальная схема которой показана на рисунке 1. Для отладки приложений использовался JTAG-адаптер Wiggler также производства Olimex [2]. Как можно видеть из рисунка,

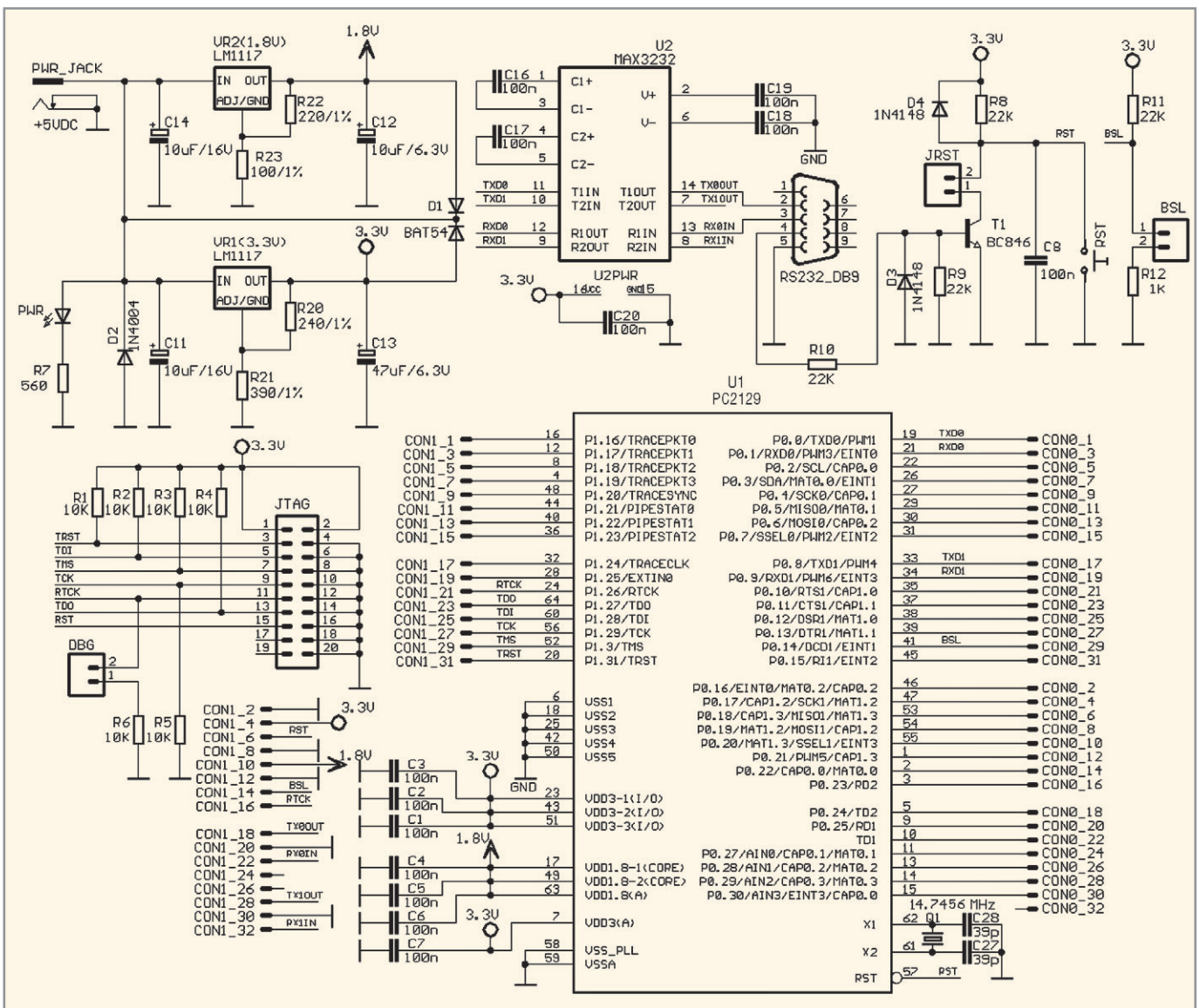


Рис. 1. Принципиальная схема отладочной платы LPC-H2129 Olimex

плата содержит МК U1 LPC2129, все выводы которого «оконечены» разъёмами, два стабилизатора питающих напряжений VR1 и VR2 (на 3,3 и 1,8 В соответственно), микросхему-адаптер уровней интерфейса RS-232 U2 MAX3232, подключенную к порту UART0 МК, разъём и цепи подключения адаптера JTAG, фильтры питания, элементы и цепи синхронизации, сброса и прочие элементы схемной «обвязки» LPC2129. Колебания встроенного тактового генератора МК стабилизируются внешним кварцевым резонатором на частоту 14,7456 МГц. Кнопка RST предназначена для «ручного» сброса LPC2129.

Внешний низкий уровень, необходимый для запуска обработчика команд ISP, подаётся на линию P0.14 LPC2129 с помощью перемычки BSL.

Перемычка JRST используется для подключения линии 4 (DTR) интерфейса RS-232 через цепь согласования уровней (D3, D4, T1, R8, R9) к входу /RESET (RST) МК, что даёт возможность с помощью программы-загрузчика LPC2000 Flash Utility производить «ручной» сброс LPC2129.

Перемычкой DBG линия RTCK интерфейса JTAG отладочной платы может быть подключена к низкому уровню через резистор сопротивлением 10 кОм, что необходимо для отладки пользовательских приложений в целевой системе.

Питание отладочной платы осуществляется от внешнего источника постоянного напряжения +5 В, подаваемого через разъём PWR_JACK.

Программная часть проекта включает в себя файл главной функции (основного цикла управляющей программы) main.c, а также файлы специальных функций и специальных заголовков и определений hardware.c и hardware.h соответственно. К специальным функциям относятся функция инициализации системы *CPUinit(void)*, инициализирующая модули МК PLL, MAM, GPIO, функция инициализации прерываний *INTERRUPTSinit(void)*, функции – обработчики прерываний, а также ещё несколько функций, о которых будет рассказано ниже. К специальным заголовкам и определениям относятся определения типов величин, макросы битовых констант, макросы сброса/установки/тестирования битов, коды символов (фонтов) ЖКИ и прочие определения. Настройки проек-

та предусматривают загрузку управляющей программы в Flash-память МК, однако, каталог проекта содержит дополнительный подкаталог SRAM, где находятся файлы такого же проекта, но с настройками, предусматривающими выполнение управляющей программы в SRAM-памяти МК. Исходные тесты программ этих двух вариантов различаются только в одном: в файле hardware.c проекта, предназначенного для Flash-памяти, закомментирована строка:

```
//MEMMAP = 2;
//Переотобразить векторы прерываний в SRAM в случае выполнения кода в SRAM
```

Функция инициализации блока управления системы *CPUinit()* размещена в файле hardware.c. Функция инициализации прерываний *INTERRUPTSinit()*, размещённая в этом же файле, содержит команды инициализации регистра управления памятью MEMMAP и контроллера векторов прерываний VIC.

В файле hardware.c также находится функция – обработчик прерываний от Таймера 0 – *TimerInterrupt()*. Она содержит команды вызова всех событий, которые должны происходить в системе регулярно. К таким событиям относятся вызовы функций опроса кнопок управления *opros_kn00* – *opros_kn30*, а также генерация импульсной последовательности, используемой для формирования дополнительного отрицательного напряжения для ЖКИ – «-U ЖКИ». Прерывание от Таймера 0 обрабатывается в системе как прерывание IRQ. Таймер 0 конфигурируется и настраивается с помощью функции *TIMER0initstart()*. Прерывания от Таймера 0 генерируются при совпадении его содержимого со специальным регистром T0MR0, а частота генерации этих прерываний выбрана в программе равной 1000 Гц. Поскольку состояние выхода P1.16 генерации импульсной последовательности для формирования «-U ЖКИ» инвертируется при каждом вызове функции *TimerInterrupt()*, частота этой последовательности составляет $1000/2 = 500$ Гц.

Помимо штатных элементов отладочной платы, установленных производителем, макет целевой пользовательской системы, схема которого приведена на рисунке 2, содержит дополнительные внешние устройства:

линейку кнопок управления SB1 – SB4 («0» – «3»), светодиоды HL1, HL2 («1», «2») и символьный жидкокристаллический индикатор со встроенным контроллером HG1. Соединительные разъёмы, установленные на плате LPC-H2129, на схеме (рис. 2) условно не показаны.

ИНТЕРФЕЙС ВВОДА-ВЫВОДА В ЦЕЛЕВОЙ ПОЛЬЗОВАТЕЛЬСКОЙ СИСТЕМЕ

Интерфейс кнопок управления

В качестве внешних устройств ввода в проекте используются кнопки. Предлагаемый интерфейс кнопок обеспечивает программное подавление «дребезга». Никаких дополнительных внешних элементов, подключаемых к микроконтроллеру, при этом не требуется.

Блок-схема алгоритма интерфейса обслуживания одной кнопки приведена на рисунке 3. Программная реализация этого алгоритма на «С» представляет собой отдельную функцию, которая отслеживает моменты нажатия и отпускания опрашиваемой кнопки, производя при этом фильтрацию импульсов «дребезга» её механических контактов.

Для построения функции программного интерфейса одной кнопки разработчику требуется выделить следующие ресурсы МК: статическую переменную, служащую для подсчёта тайм-аута подавления «дребезга», ещё две статические переменные, служащие для индикации момента нажатия на кнопку (флаг «нажато») и длительности удержания кнопки (флаг «удерживается»). При этом переменная флага «нажато» должна быть описана в программе как доступная для других модулей, поскольку она в них также модифицируется. Кроме того, функция использует некоторую константу, задающую значение порога тайм-аута подавления «дребезга».

Для корректной работы кнопочного интерфейса при первоначальной инициализации флаги «нажато» и «удерживается» всех опрашиваемых кнопок должны быть сброшены. Как можно видеть из блок-схемы, при каждом вызове функции при условии, что опрашиваемая кнопка не нажата, переменная подсчёта тайм-аута и флаг «удерживается» будут обнуляться, т.е. будет происходить постоянная установка цифрового программного фильтра «дребезга» в началь-

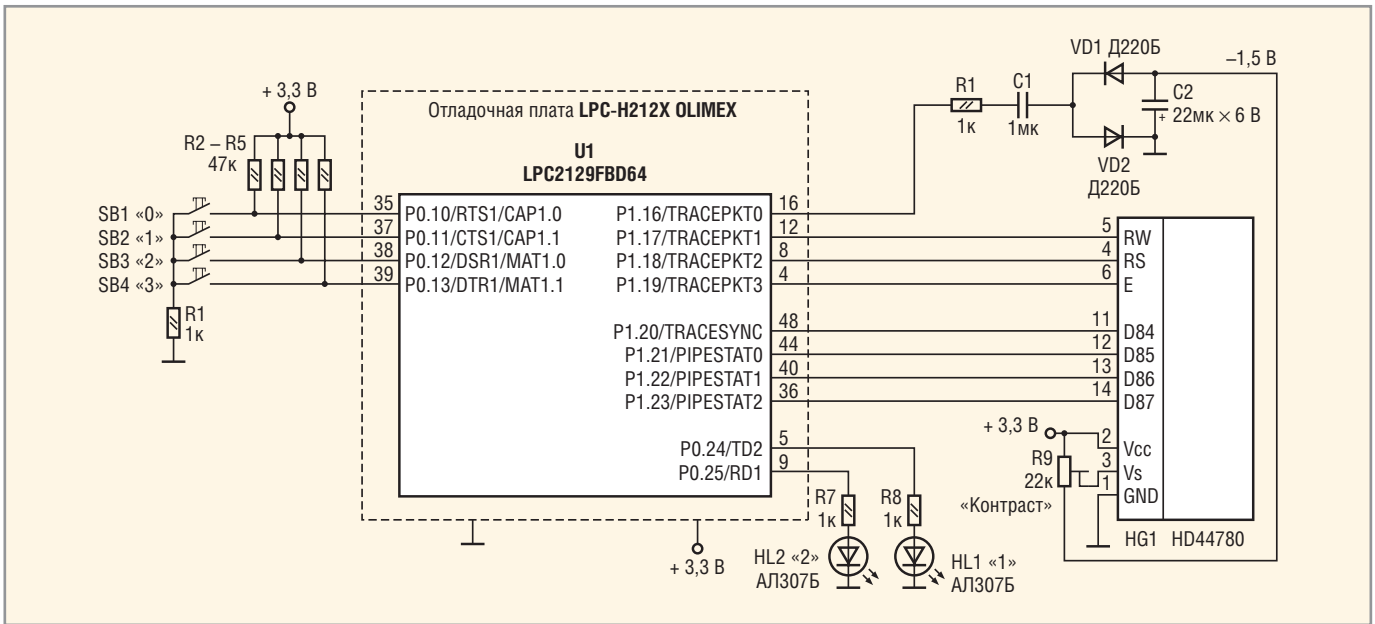


Рис. 2. Принципиальная схема базовой части макета целевой пользовательской системы

ное состояние. Вызовы функции в программе должны производиться с некоторой периодичностью. Когда кнопка нажимается, значение переменной подсчёта тайм-аута начинает инкрементироваться при каждом вызове функции, причём реально увеличиваться оно станет только после окончания «дребезга», когда переменная перестанет обнуляться из-за пауз между «дребезговыми» импульсами. Только спустя некоторое время после окончания импульсов «дребезга» наступит момент, когда значение переменной подсчёта тайм-аута превысит значение константы порога тайм-аута и, следовательно, произойдёт установка фла-

гов «нажато» и «удерживается» опрашиваемой кнопки.

Состояние флага «удерживается» инверсно отображает текущее состояние входа подключения соответствующей ему кнопки с учётом подавления импульсов «дребезга», т.е. он будет оставаться в единичном состоянии столько времени, сколько удерживается кнопка после окончания «дребезга», и сбросится при первом же импульсе «дребезга» в момент отпускания кнопки. Этот флаг можно использовать в программе для реализации каких-то дополнительных функций данной кнопки в интерфейсе ввода разрабатываемого устройства, например, для задания какого-

либо режима, необходимым (но не достаточным) условием которого служит текущее нажатое состояние этой кнопки.

Флаг «нажато» в предлагаемом алгоритме обладает свойством «защёлкивания», т.е. при отпускании кнопки он в подпрограмме не сбрасывается. Опрос состояния с целью обнаружения установки и последующего сброса флага «нажато» возлагается на ту часть основной программы, которая производит обработку обнаружения нажатия на данную кнопку.

Конкретное значение константы порога тайм-аута «дребезга» может задаваться разработчиком для обеспечения требуемого времени реакции устройства на нажатую кнопку в зависимости от значения периода вызова функции кнопочного интерфейса и конкретного типа (линейных размеров) кнопки.

Исходный текст набора функций, реализующих описанный выше алгоритм обслуживания четырёх кнопок, содержится в файле knop.c. Этот файл используется в проекте как подключаемый. Вызовы функций опроса кнопок производятся в программах проектов с частотой 1 кГц из функции – обработчика прерываний от Таймера 0 `TimerInterrupt()`. Определения входов подключения кнопок к МК содержатся в файле hardware.h. В файле knop.c содержатся функции `opros_kn0()` – `opros_kn3()` обслуживания четырёх кнопок, каждая из которых подключена к отдельной линии ввода-вывода. Эти линии в функциях

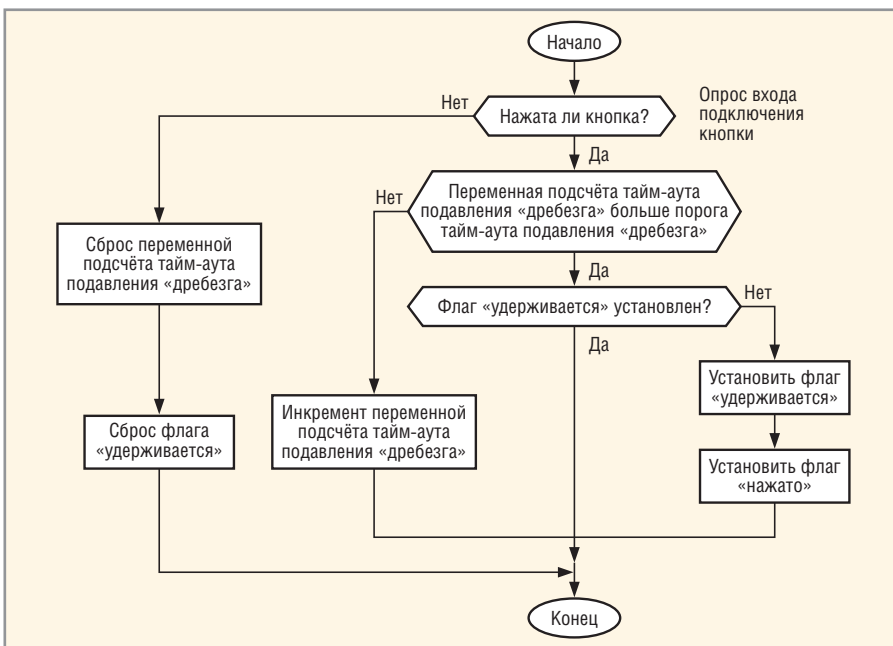


Рис. 3. Блок-схема алгоритма опроса одной кнопки управления

имеют формальные имена IN_KNOPO – IN_KNOP3. При этом предполагается, что кнопки являются нормально разомкнутыми и включены между входами МК и общим проводом устройства (последовательно с токоограничительными резисторами 1 кОм), а линии подключения кнопок сконфигурированы в МК как входы линий ввода-вывода общего назначения (GPIO) и подтянуты к «плюсу» источника питания внешними резисторами, как показано на рисунке 2. Константы значения порога тайм-аута «дребезга» в функциях имеют имена DREB_KN0_K – DREB_KN3_K, а их значения содержатся в файле hardware.h.

Интерфейс ЖКИ

Интерфейс вывода в устройстве на базе МК в большинстве случаев должен включать в себя какие-то аппаратные средства индикации с соответствующей программной поддержкой. В настоящее время весьма широко распространены символьные (буквенно-цифровые) жидкокристаллические модули со встроен-

ном контроллером управления, совместимым с контроллером HD44780 фирмы Hitachi. Автор использовал именно такой индикатор – WH1602 фирмы Winstar (русифицированный, 16 символов в строке, 2 строки).

Передача информации из МК в HD44780-совместимый индикатор возможна по параллельной восьми- или четырёхпроводной шине данных. Предлагаемое далее пользовательское программное обеспечение реализует передачу данных из LPC2000 в модуль ЖКИ только по четырёхпроводной шине данных. При взаимодействии с индикатором МК может только передавать в него данные без опроса его состояния или производить его опрос на предмет готовности к приёму данных и передавать данные по готовности. В первом случае готовность ЖКИ к приёму данных МК не проверяет, а вместо этого программно генерирует временные задержки, заведомо большие интервалов времени, необходимых ЖКИ на подготовку к приёму.

Исходный текст набора функций, реализующих интерфейс обмена с

ЖКИ по четырёхпроводной шине данных с опросом его состояния или без такового, содержится в файле lcd.c. Этот файл используется как подключаемый. В файле содержатся следующие функции: функция начальной инициализации ЖКИ после сброса *lcd_init()*, функция очистки экрана ЖКИ *lcd_clear()*, функция передачи в ЖКИ одной команды *lcd_com(command)*, функция записи в ОЗУ ЖКИ одного байта данных (индикации на экране ЖКИ одного символа) по текущему адресу *lcd_tek_data(date)* или по произвольному адресу – *lcd_pro_data(date, addr)*, функция опроса состояния ЖКИ *lcd_opros()*, возвращающая управление по готовности последнего, а также несколько вспомогательных функций следующего уровня вложенности. Если пользователем выбран вариант интерфейса с опросом текущего состояния ЖКИ, то в файле необходимо «вручную» раскомментировать (убрать стоящие перед ними символы «//») команды вызова функции *lcd_opros()* во всех функциях, где они встреча-

Полупроводники НА ОСНОВЕ КАРБИДА КРЕМНИЯ



Практическое применение

ПРИМЕНЕНИЕ SiC-ДИОДОВ ШОТКИ ПОЗВОЛЯЕТ

Характеристики высоковольтных диодов Шоттки фирмы Cree

- Снизить потери в диоде и ключевом транзисторе в 2 раза
- Уменьшить количество силовых электронных компонентов в 3 раза
- Увеличить надёжность
- Повысить частоту преобразования, уменьшить массу и габариты
- Получить выигрыш в стоимости и эффективности одновременно

Наименование	CSD04060	CSD06060	CSD10060	CSD20060	CSD05120	CSD10120	CSD20120
U _{макс} , В	600	600	600	600	1200	1200	1200
I _{пост} , А	4	6	10	20	5	10	20
Типы корпусов	T0252, T0220-2, T0220-3	T0263, T0220-2, T0220-3	T0263, T0220-2, T0220-3	T0247-3	T0220-2	T0220-2, T0247-3	T0247-3

ОБЛАСТИ ПРИМЕНЕНИЯ:

- Активные корректоры коэффициента мощности — снижение динамических потерь в ключевом транзисторе и диоде до 60%
- Антипараллельные диоды MOSFET- и IGBT-транзисторов и модулей для жёсткого переключения — снижение динамических потерь на 20...30%
- Мощные высоковольтные выпрямители для частот до единиц мегагерц



ПРОСОФТ – официальный дистрибьютор компании CREE в России и странах СНГ



ПРОСОФТ – АКТИВНЫЙ КОМПОНЕНТ ВАШЕГО БИЗНЕСА

Телефон: (495) 232-2522 • E-mail: info@prochip.ru • Web: www.prochip.ru

ются, а команды вызова функций временных задержек `delay(3000)` и `delay(108000)`, наоборот, везде закомментировать (поставить перед ними символы `///
»). В случае выбора варианта с отсутствием опроса (генерации задержек ожидания) необходимо, наоборот, раскомментировать вызовы этих задержек и закомментировать вызовы функции опроса lcd_opros(). Следует заметить, что для такого «быстрого» МК, каким является LPC2129, очевидно, рациональнее будет использовать вариант интерфейса ЖКИ с опросом его состояния: в этом случае программа тратит значительно меньшее общее время на индикацию. Определения и макросы задания состояний линий подключения ЖКИ к МК содержатся в файле hardware.h. Там же находятся определения констант-фонов русского и латинского алфавитов HD44780-совместимого ЖКИ.`

Поскольку питание ЖКИ осуществляется от системного питающего напряжения +3,3 В, контрастность его изображения может оказаться недостаточной. По этой причине в системе производится генерация дополнительного отрицательного напряжения «-U ЖКИ» со значением около -1,5 В для питания ЖКИ, как показано на рисунке 2. Для его получения используется импульсная последовательность частотой 500 Гц, генерируемая на линии P1.16 по прерыванию от Таймера 0 в функции `TimerInterrupt()`.

Интерфейс обслуживания RTC – часы-календарь реального времени

Проект интерфейса обслуживания модуля часов-календаря реального времени RTC LPC2000 имеет очевидное прикладное значение. Он представляет собой встраиваемый программно-аппаратный модуль часов-календаря, который может быть использован как самостоятельное устройство «часы-календарь», а также в качестве программно-аппаратного дополнения к какому-либо устройству.

Интерфейс обслуживания модуля RTC LPC2000 [3] реализован в проекте, все файлы которого находятся в каталоге RTC, который доступен на сайте журнала. Функции обслуживания RTC, их заголовки и определения находятся в файлах `rtc.c` и `rtc.h` соответственно, а основная программа

интерфейса модуля RTC – в файле `main.c` этого проекта. Программа производит циклическое чтение содержимого специальных регистров консолидированного времени модуля RTC и вывод на индикацию значений текущего времени и даты, а также реализует возможность пользовательской установки текущего времени и даты с помощью кнопок управления.

В основном цикле программа производит периодическое чтение и индикацию на ЖКИ значений текущего времени и даты с помощью функции `RTCread_ind_date_time()`.

Кнопка «0» задаёт режим установки показаний. Текущий режим индицируется светодиодами «1» и «2». При наличии двух светодиодов может быть задано четыре их состояния (режима установки). Текущий режим определяется в программе значением переменной `led1_old_state`. При выключенных светодиодах «1» и «2» (`led1_old_state=0`) кнопки «1» – «3» имеют следующие функции: кнопка «1» – инкремент текущего значения часов (от 0 до 23 по кольцу), кнопка «2» – инкремент текущего значения минут (от 0 до 59 по кольцу), кнопка «3» – обнуление текущего значения секунд (для обеспечения точной установки времени). При `led1_old_state=1` (светодиод «1» горит, «2» – не горит) кнопки «1» – «3» имеют следующие функции: кнопка «1» – инкремент текущего значения дней месяца (от 1 до 31 по кольцу), кнопка «2» – инкремент текущего значения месяцев (от 1 до 12 по кольцу), кнопка «3» – инкремент текущего значения лет (от 1990 до 2030 по кольцу). Границы установки текущего значения лет заданы в функции `RTCset_year()` (файл `rtc.c`) и при желании могут быть изменены. При `led1_old_state=2` (светодиод «1» не горит, «2» – горит) кнопка «3» имеет функцию инкремента текущего значения дней недели (от 0 до 6 по кольцу).

Функция `RTCread_ind_date_time()` производит чтение регистров консолидированного времени модуля RTC с помощью соответствующих битовых масок, преобразование полученных значений в десятичное представление и их вывод на ЖКИ в удобном для восприятия виде с разделительными точками и двоеточиями. Дни недели индицируются в виде буквенных аббревиатур: ПН, ВТ, СР, ЧТ, ПТ, СБ, ВС. При пользовательской

установке значений времени и даты соответствующие функции производят сначала остановку счёта времени в модуле RTC с помощью вызова функции `RTCstop()` (файл `rtc.c`), потом запись нового значения в регистр счётчика времени RTC, а затем запуск счёта времени в модуле RTC с помощью вызова функции `RTCstart()` (файл `rtc.c`). При установке значений времени и даты следует учитывать, что программа не следит за корректностью задания начальных даты и дня недели, а лишь корректно (с учётом количества дней в текущем месяце и високосных годов) производит дальнейший счёт от установленной пользователем даты. Проверка корректности установки даты возлагается на самого пользователя. Например, при включении питания функция начальной инициализации модуля RTC `RTCinit()` записывает в счётчики времени корректные значения, которым соответствуют показания индикатора «00:00:01 BC 01.01.2006».

Функция начальной инициализации модуля RTC `RTCinit()` помимо констант инициализации содержит команды настройки узла тактирования RTC (предделителя) на частоту 32 768 Гц, которая генерируется путём деления частоты `plck`.

Прерывания от RTC в программе не используются, поскольку счёт времени в устройстве производится аппаратно и программе нет необходимости реагировать на изменение содержимого счётчиков времени, а индикация содержимого этих счётчиков может производиться программой в произвольные моменты.

Для получения информации о других демонстрационных и прикладных проектах на базе МК LPC2000 рекомендуется обратиться к [4].

ЛИТЕРАТУРА

1. LPC-H2129 Header Board for LPC2129 ARM7TDMI-S Microcontroller. Copyright(c) 2004, Olimex Ltd., www.olimex.com/dev.
2. ARM-JTAG Wiggler Compatible Dongle for Programming and Debugging. Copyright(c) 2004, Olimex Ltd., www.olimex.com/dev.
3. AN10382. Using the RTC efficiently in the LPC213x. Rev. 01 – 6 June 2005 Application note. Philips Semiconductors.
4. *Редькин П.П.* Микроконтроллеры ARM7. Семейство LPC2000 компании Philips: полное руководство пользователя. М.: Додэка-XXI. 2007.



NEC научилась распознавать лица в видео

Японская компания NEC разработала технологию распознавания лиц людей, появляющихся в видео. По мнению разработчиков, такое приложение помогает облегчить поиск необходимого материала в архивах. Применять данное решение можно в мобильных телефонах, на телевидении и в Интернете (например, на YouTube). NEC также собирается предлагать свой продукт корпоративным клиентам. Правда, на данном этапе авторы ещё не готовы к релизу. Необходимо дополнительное время, но они надеются, что релиз состоится уже в ближайшем будущем.



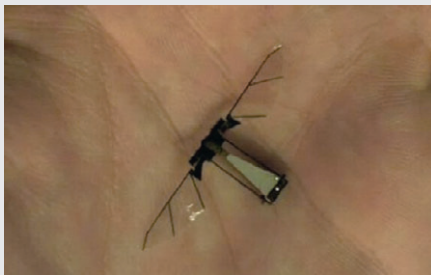
Программа от NEC выуживает из любого видео лица людей, затем группирует их в специальные списки, исходя из уровня сходства (для этого программа проделывает сложный процесс группировки изображений человека под отличными углами и при разном освещении). Благодаря такому «генератору галерей» пользователь может сразу увидеть, кто появится в видео. Точность работы технологии пока составляет около 80%.

techon.nikkeibp.co.jp

Робот-муха готовится стать разведчиком

Идеальный разведчик – это не только тот, который ничего не расскажет врагу, но и которого чрезвычайно сложно обнаружить. Именно такого шпиона создал профессор Гарвардского университета Роберт Вуд (Robert Wood) – летающего робота-муху, вес которого составляет всего 60 мг, а размах крыльев не превышает 3 см. При этом специалисту потребовалось десять лет, чтобы научить столь миниатюрное создание уверенно чувствовать себя в воздухе. Одной из основных проблем стала разработка крыльев робота и движителей, в результате робот практически полностью повторяет технику полёта насекомых.

Но пока назвать робота-муху полностью автономным нельзя – источником питания для устройства выступает высо-



ковольтный усилитель, а передача энергии осуществляется посредством тончайших проводников. И именно оснащение робота «наборным» источником питания является следующей целью Вуда.

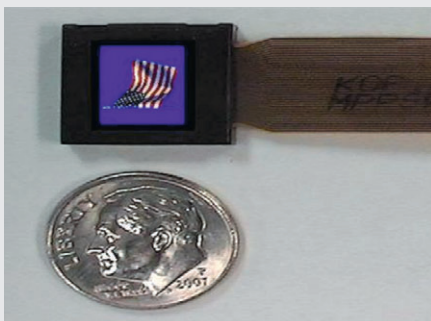
На данный момент основным предназначением столь миниатюрного робота, способного летать, является именно работа в качестве разведчика. И не случайно проектом заинтересовалось агентство передовых оборонных исследовательских проектов DARPA (Defense Advanced Research Projects Agency) и выделило средства на финансирование разработок. Но до завершения проекта ещё далеко – помимо задачи оснащения робота автономным источником питания, необходимо установить миниатюрную камеру, а также организовать систему беспроводной передачи данных. Даже сам Вуд пока не говорит о возможных сроках завершения работ.

www.3dnews.ru

Самый маленький SVGA-экран от Kopin

Американская компания Kopin на крупнейшей ежегодной выставке потребительской электроники CES 2008 явила общественности свою новую разработку: ЖК-экран с диагональю 0,44 дюйма и разрешением 800 × 600 пикселей. Новинка, получившая наименование CyberDisplay SVGA LVS, объявлена компанией самым маленьким цветным экраном с разрешением SVGA.

Как отметил генеральный директор компании Джон Фан (John Fan), модель может похвастать на 30% меньшим энергопотреблением и на 45% меньшим раз-



мером по сравнению со стандартными 0,59-дюймовыми аналогами. Площадь пикселя в 11,25 квадратных микрон является самой маленькой среди доступных в данный момент коммерческих решений.

Во время презентации особое внимание было обращено и на преимущество решений компании. На самом деле CyberDisplay SVGA LVS обладает сходными габаритами с его предшественником, CyberDisplay VGA (640 × 480 пикселей), что позволит легко модернизировать все устройства прошлого поколения. Ожидается, что главной областью применения новинки станут мультимедийные очки для просмотра видеofilмов.

Компания начала отгрузку тестовых образцов CyberDisplay SVGA LVS для производителей.

kopin.com

Тайваньские инженеры улучшили систему LED-подсветки

Современные жидкокристаллические панели уже успели обзавестись светодиодной системой подсветки, однако перед повсеместным использованием LED-подсветки в крупноформатных ЖК-телевизорах и информационных табло стоят некоторые преграды. Одной из них является проблема «деградации» цвета при непрерывной работе устройства в течение 1000 ч, что влечёт за собой не только снижение яркости и контрастности изображения, но и искажения его цветовой гаммы.

Своё решение представленной проблемы предложили сотрудники тайваньского института Industrial Technology Research Institute (ITRI), разработавшие систему автоматической настройки баланса белого цвета для светодиодной подсветки ЖК-панелей. Технология должна найти своё применение именно в крупноформатных устройствах и high-end-системах, где изменение яркости наиболее заметно для человеческого глаза, – сообщают представители ITRI.

На данный момент разработчики уже получили несколько патентов на свои «ноухау», которые используются в системе настройки баланса белого. По всей видимости, технология уже полностью готова к коммерческому использованию – ITRI ведёт переговоры с производителями ЖК-панелей для дальнейшего совместного сотрудничества.

www.3dnews.ru