

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ЭЛЕКТРОННОЙ КНИГИ IBOOK V8

С.А. Семериков¹, И.А. Теплицкий¹, Е.П. Линник², Г.И. Корнилов³

¹ Украина, г. Кривой Рог, Криворожский государственный педагогический университет

² Украина, г. Кривой Рог, Институт воздушного транспорта Национального авиационного университета

³ Украина, г. Кривой Рог, Криворожский институт Кременчугского университета экономики, информационных технологий и управления

В статье [1] нами был рассмотрен новый класс технических средств обучения – электронные книги. Несмотря на то что, подобные устройства принято сравнивать с КПК, они таковыми не являются: эргономика устройства, органы управления материалы и технологии, используемые при создании устройства полностью рассчитаны на чтение. В то же время рассматривать это устройство исключительно как «читалку» не стоит: то, что IBook V8 начали производить менее двух лет назад, является причиной небольшого выбора готового программного обеспечения, но вместе с тем дает простор для написания собственных программ.

В устройстве используется процессор S1C33L05 фирмы Seiko-Epson – 32-разрядный RISC процессор, работающий на частоте 48 МГц. Процессор имеет 16 регистров (R0-R15), 16 кБ встроенного ОЗУ. Полная документация по S1C33 доступна на <http://www.eea.epson.com>.

Основные особенности программирования процессора:

- S1C33 имеет little-endian организацию данных в памяти, т.е. младший байт слова хранится первым;
- обращение по адресам, не кратным размеру данных, не допускается (в отличие от x86);
- процессор не имеет инструкций для работы с плавающей точкой;
- инструкции сдвига/вращения на переменное число бит компилируются в довольно большой и неэффективный фрагмент кода. Следует по возможности избегать таких конструкций там, где критична скорость выполнения.

Устройство имеет 4 Мб масочное ПЗУ, в котором хранится начальный загрузчик и процедуры тестирования и обновления прошивки, 4 Мб флеш-памяти и 1 Мб ОЗУ.

В IBook V8 установлен eInk-дисплей производства PVI с разрешением 600x800, позволяющий отображать 4 градации серого. Управляется экран с помощью контроллера Apollo, документацию по программированию которого доступна на <http://sourceforge.net/projects/thinspace/>.

Дисплей имеет 3 режима обновления: 1) обновление всего экрана с «зачернением», обеспечивающий наилучшее качество изображения; 2) быстрое обновление всего экрана или его части; 3) вывод монохромного изображения.

Операционная система устройства Wolf OS имеет модульную структуру. Прошивка состоит из отдельных модулей, которые можно заменять, не затрагивая остальные. Каждый модуль располагается по фиксированному адресу в памяти. Прошивка IBook V8 состоит из следующих модулей: Register – отвечает за первоначальную загрузку устройства; WolfSDK – ядро ОС; BookMain – основное меню (список книг); TxtRead – чтение текстовых файлов; HtmRead – чтение HTML-файлов; WolfRead – чтение WOL-файлов; DevDrv – драйвера устройств; LcdApp – приложения вспомогательного экрана; Logo – заставка; Picture – графические ресурсы; Res – шрифты.

Wolf OS не содержит динамического редактора связей, а модули запускаются непосредственно из флеш-памяти. Модуль начинается с заголовка, в котором содержатся указатели на процедуру статической инициализации модуля, обработчик событий, таблицу экспортируемых функций API. Передача управления от одного модуля к другому выполняется с помощью функции StartModule.

Модуль BookMain запускается автоматически при рестарте устройства. При инициализации по событию EVT_INIT модуль должен выполнить следующие задачи: 1) загрузить пользовательские настройки; 2) выполнить калибровку экрана; 3) установить перехватчики событий EVT_CHARGING, EVT_CHARGED, EVT_NOTCHARGED, EVT_POWERLOW (вывод информационных сообщений), EVT_SDIN (обновление списка книг), EVT_SDOUT (закрытие активных экранов и отображение заставки), EVT_KEYDOWN (звук при нажатии клавиши), EVT_KEYPRESS (закрытие открытого приложения при нажатии); 4) выполнить функцию init1 модуля LcdApp.

Модуль Reader запускается при открытии файла с «книжной полки», получая имя запрошенного файла с помощью функции GetFileName, дальнейшее его поведение ничем не регламентируется. Выход по кнопке «Bookshelf» происходит автоматически (при этом всем открытым экранам посылаются события EVT_EXIT). В качестве области статических переменных можно использовать 64 Кб памяти по адресу 0x02000000. Следует иметь в виду, что область статических переменных не инициализируется автоматически, это нужно делать вручную в функции init1.

WolfOS предполагает запуск внешних программ в формате ALM – перемещаемых модулей, адрес загрузки которого задается на этапе компиляции. По умолчанию базовый адрес модуля установлен на

0x2003000 – это статическая область памяти размером около 50 Кб, что определяет максимальный размер модуля.

При запуске модуля для него создается основной экран, обработчик событий которого указан в заголовке модуля. Первое событие, получаемое экраном – EVT_INIT, по которому можно выполнить инициализацию (выделение памяти, чтение конфигурации и т.п.). По событию EVT_SHOW нужно выполнить отрисовку содержимого экрана. Событие EVT_EXIT возникает перед закрытием экрана, по нему нужно освободить память, выделенную в EVT_INIT. EVT_TIMER возникает 40 раз в секунду, когда устройство активно (в течение 2-х секунд после нажатия клавиши, либо постоянно, если подключено зарядное устройство). EVT_KEYDOWN, EVT_KEYPRESS – при нажатии клавиши. EVT_KEYREPEAT – 1 раз в секунду при удержании клавиши. EVT_KEYRELEASE – при отпускании нажатой клавиши. EVT_CHARGING – при подключении зарядки. EVT_CHARGED – при окончании процесса зарядки. EVT_NOTCHARGED – при отключении зарядного устройства (аккумулятор зарядился не до конца). EVT_POWERLOW – при разряде аккумулятора. EVT_SDIN – при установке SD-карты в гнездо (также после загрузки устройства, если SD-карта уже установлена). EVT_SDOUT – при вытаскивании SD-карты.

Кроме реальных событий, генерируемых ОС, экрану можно передавать синтетические события с помощью функции SendEvent.

В первую очередь обработчик должен получить очередное событие из очереди с помощью функции GetNextEvent. Затем производится обработка события; в конце обязательно должна вызываться функция TranslateEvent. Если требуется подавить обработку события, достаточно изменить класс события на 0.

Выбор окна в Wolf OS производится по номеру, который задается вручную при создании окна функцией CreateWindow. Основные экраны имеют фиксированные номера: 1000 – eInk, 3000 – LCD. Окна с номерами 80-82 и 90-92 используются модулем BookMain для вывода сообщений. Координаты и размеры дочерних окон (как горизонтальные, так и вертикальные) должны быть кратны 4 для eInk и 8 для LCD.

Экран eInk может иметь как портретную, так и альбомную ориентацию, это реализовано аппаратно. При старте приложения основное окно имеет портретную ориентацию. Изменить ориентацию окна можно с помощью функций SetPortraitLayout и SwitchLayout.

В устройстве используются растровые шрифты – монохромные и с 4 градациями серого. Шрифты находятся в модуле Res.

WolfOS поддерживает SD-карты объемом до 1 Гб с файловой системой FAT16. Набор функций для работы с файловой системой довольно

ограниченный; кроме того, поддержка длинных имен реализована только для файлов, записанных на ПК, сама же система может создавать файлы только в формате 8.3. SD-карта представлена в системе диском «В». Диск «А» представляет собой временный диск в флеш-памяти.

Для работы с файлами предусмотрены функции CreateFile, OpenFile, ReadFile, WriteFile, CloseFile, DeleteFile, SeekFile, TellFile, GetFileSize, для чтения каталога предусмотрены функции FindFirstFile, FindNextFile.

Для записи данных в энергонезависимую память Wolf OS имеет 2 функции: EraseFlash и WriteFlash. Используемая в устройстве флеш-память имеет размер блока 64 Кб (0x10000). Адрес, указываемый EraseFlash, должен указывать на границу блока. Данные, записываемые с помощью WriteFlash, могут начинаться с любого адреса, но не должны пересекать границу блока. Операция записи в флеш-память изменяет только биты 1 на 0, поэтому при записи блока в неочищенную область будет выполняться операция AND над старыми и новыми данными.

Звуковые возможности внешнего динамика устройства представлены функцией Beep, издающей звук при нажатии клавиш, и функцией Play, проигрывающей звук указанной тональности и длительности.

Для сборки программ под S1C33 нужно задать такие опции кросс-компилятора GNU C 3.3.2: -O1 -nostdinc -fno-builtin -ffreestanding -S

Исходные коды компилируются в ассемблерные файлы (*.s), которые затем собираются с помощью программы alink. Библиотеки хранятся тоже в виде ассемблерных текстов.

alink осуществляет перевод ассемблерных текстов в код S1C33 и сборку модуля/ALM файла. Пример использования alink для сборки ALM, загружаемого по адресу 0x02003000:

```
alink -o output.alm -a 0x02003000 *.s
```

Выводы:

1. Для IBook V8 возможна разработка специализированных программ, ориентированных как на кнопочный, так и сенсорный интерфейсы.

2. Расширив набор программ для IBook V8, его можно превратить в многофункциональное устройство класса «записная книжка».

3. Перспективным является создание программного обеспечения для тестирования знаний на основе комбинированного кнопочно-сенсорного интерфейса как компонента технологии мобильного обучения.

Литература:

1. Теплицький І.О., Семеріков С.О., Шокалюк С.В., Ліннік О.П. Новий технічний засіб навчання – електронна книга // Рідна школа. – 2007. – №7-8. – С. 53–54.