uMAPI version 1 SDK

MicroMachine Application Program Interface Software Development Kit

© Romanich

Vladivostok 2006

1) Что это?

uMAPI version 1 SDK - комплект разработки программного обеспечения для МикроМашины (мМ)
Данный пакет содержит тот минимум информации, который необходим для создания uMAPI-приложений

uMAPI-приложения БИ-платформенны: они выполняются на следующих двух аппаратных платформах:
- IBM PC
- uM

Первая платформа позволяет вести начальный процесс разработки uMAPI-приложения
Очень удобна в плане многоразовой перекомпиляции приложения и его отладки
Данная платформа программно эмулирует вторую платформу с точностью до интерфейса uMAPI

Вторая платформа позволяет портировать uMAPI-приложение с IBM PC на uM
Служит для окончательного получения исходного кода приложения и дальнейшей его компиляции
Данная платформа позволяет получить окончательную прошивку на мМ с целью её использования

2) Смысл

Смысл БИ-платформенности uMAPI заключается в том, что МикроМашина имеет встроенную в MCU флеш-память с ограниченным числом стирания/записи (1000..10000)
Естественно, при написании и отладки приложения на мМ, приходится многократно перекомпилировать и перезапускать (а значит - стирать/записывать флеш-память) это приложение
Не желая лишний раз перепрошивать флеш-память, было решено написать программную эмуляцию интерфейса uMAPI на IBM PC, который может загружать и запускать приложение безлимитно

Кроме того, платформа IBM PC открывает возможность программистам, не имеющим МикроМашину, писать под неё uMAPI-приложения (подобно тому, как любители Dandy и SEGA, не имея реальных приставок, пишут для них программы под эмуляторы)
Это позволяет почувствовать мощь 2D-аркадной машины (т.е. МикроМашины) и окончательно принять решение - стоит ли её приобретать/собирать или нет

Мало того, чем больше софта народ напишет, тем популярнее станет МикроМашина
Не зря говорят, что успех любой железки определяется количеством написанного под неё софта :)

Смысл uMAPI вообще, состоит в том, чтобы облегчить программисту создание приложений для мМ
Тоесть uMAPI по сути, представляет собой HAL - Hardware Abstraction Layer - "слой аппаратных абстракций" - не нужно детально знать архитектуру МикроМашины, достаточно знать интерфейс uMAPI, который напрямую (и с максимальной скоростью) работает с периферией МикроМашины

Видео под-система мМ требует огромных вычислительных затрат - функции, отвечающие за 2D-рендеринг, написаны ПОЛНОСТЬЮ на Ассемблере, их алгоритм реализован с различными ухищрениями, которые повышают быстродействие данной под-системы

Следует отметить, что такие алгоритмы (с ухищрениями) может реализовать только тот человек, который очень и очень ясно представляет как работает МикроМашина и её периферия
По крайней мере, один такой человек уже есть (её Создатель), далее идут те, кто её сможет её собрать самостоятельно, а далее те, кто разберётся в архитектуре мМ с программной и аппаратной точки зрения

3) Сравнение

Считаю что на @24MHz МикроМашина полностью переплёвывает i386 @33MHz по быстродействию
Связано это с тем, что RISC MCU мМ (ATmega128) выполняет инструкцию в среднем за 1-2 такта, в то время как CISC-процессоры те же действия делают за большее число тактов
В итоге, производительность MCU мМ больше, чем у CPU IBM PC

Причём все регистры общего назначения в MCU мМ подключены к Арифметико-Логическому Устройству (ALU) напрямую, в отличие от i386 (в их АЛУ использовались регистры eax/ecx/edx)

Кроме того, МикроМашина оснащена высокопродуктивной периферией:
- Графическим контроллером LCD, который автономно(!) поддерживает регенерацию изображения и выполнение LCD-команд
- Музыкальным сопроцессором Yamaha, реализующим полноценный FM-синтез (прототип MIDI)
- Параллельной клавиатурой, обеспечивающей одновременный анализ нескольких клавиш

А IBM PC, имеющий CPU i386 в то время имел скромную видеосистему 640x480 16 цветов
В противовес - МикроМашина имеет LCD с одновременно(!) отображающимися 4096 цветами
Правда, в uMAPI version 1 реализован только режим 256 цветов, в дальнейшем, появится расширения uMAPI - будет поддержан программно режим 4096 цветов :)

4) Что необходимо?

Прежде всего необходимо скачать две системы программирования:
- TMT Pascal MS DOS Edition version 3.90
- CodeVisionAVR version 1.24.8b Professional

Первая из них - это полноценная система программирования (компилятор/редактор/...)
Почти на все 100% совместима с Turbo (Borland) Pascal, но только 32-х битная
Позволяет писать 32-х битные программы под DOS (да-да!), используя DOS Extender

Ссылка: http://www.tmt.com или ключевая фраза в поисковиках "TMT Pascal по-русски"

Под DOS данная система распространяется свободно и без ограничений

Вторая - компилятор языка C, включает редактор, программный генератор, ассемблер, мэнеджер проекта, программатор и.т.д.

Ссылка: http://www.hpinfotech.com

Скажу, что лучше ставить version 1.24.8b Professional, так как остальные либо не поддерживают ATmega128 или имеют ограничения на размер программы, либо вообще без регистрационного ключа не работают

5) Этапы создания

- Пишем uMAPI-приложение на TMT Pascal (платформа IBM PC)
- Отлаживаем данное приложение (пишем/компилируем/запускаем/ далее исправляем и снова по кругу)
- После того, как uMAPI-приложение будет полностью написано и отлажено на платформе IBM PC, начинаем его портировать на платформу uM - Переводим программу из Pascal на C
- Компилируем, исправляем синтаксические ошибки в программе
- Получаем конечный файл прошивки и заливаем полученный код в МикроМашину
- Запускаем на мМ портированную программу, испытываем, если что не так - доделываем
- Наслаждаемся мощью МикроМашины и uMAPI-приложением :)

6) Пояснения к материалу

Директории:

uM - целый проект 2D-приложения, демонстрирующий работу всех под-систем МикроМашины (видео/звук/клавиши/светодиод) с вызовом всех(!) функций uMAPI Base Edition
Платформа uM

PC - тот же проект, но для платформы IBM PC

Util - здесь находятся утилиты, которые понадобятся для создания данных uMAPI-приложения

Подробнее:

uM\uMTest.c - исходник тестового uMAPI-приложения
uM\uMAPI.c - исходник интерфейса uMAPI (вот в нём-то и содержится ядро uMAPI для мМ!)
uM\uMTest.prj - файл проекта uMAPI-приложения (его надо открывать в CodeVision C, чтобы данный проект открылся)
uM\uMTest.hex - собственно сама прошивка для MCU МикроМашины

PC\uMTest.pas - исходник тестового приложения
PC\uMAPI.pas - исходник ядра uMAPI для IBM PC платформы
PC\uMTest.exe - откомпилированное uMAPI-приложение для IBM PC
PC\Sprite.bmp - файл спрайтов (со служебными цветами EOS и Ignore)

Util\Sprite_1.exe - конвертит BMP-файл в четыре файла, необходимые для работы далее
Util\Sprite_2.exe - из файлов, получненных Sprite_1.exe создаёт файлы данных о спрайтах, которые легко легко откопировать и вставить в программы
Util\OPN.exe - программа, позволяющая подобрать патчи для звуковой системы
Util\Palette.bmp - палитра из 256 цветов с двумя дополнительными цветами (EOS, Ignore)

uMAPIv1SDK.txt - текущий файл
uMAPIv1.pdf - даташит на uMAPI Base Edition (очень полезная инфа по uMAPI-функциям)
YM2612Device.pdf - изготовление Внешней Звуковой Карты для IBM PC, которая необходима для программы OPN.exe и uMAPI-приложений, написанных на платформе IBM PC

Коротко о главном:

Для разработки uMAPI-приложений необходимо ядро uMAPI (для uM - uMAPI.c, для PC - uMAPI.pas)

BMP-файл со спрайтами, которые нарисованы только цветами из Palette.bmp!
После каждого спрайта должен стоять терминальный пиксел с цветом EOS (см. uMAPIv1.pdf)
Спрайты в BMP-файле должны идти только сверху вниз по одному!
Неиспользуемое пространство в BMP-файле заливается цветом Ignore
Всё это необходимо для правильной работы конвертеров Sprite_1.exe и Sprite_2.exe

Sprite_1.exe вырезает неиспользуемое пространство BMP-файла, располагает спрайты один за другим, конвертирует 24 бита в 8 бит (абсолютно без потерь - используются ПРЕДОПРЕДЕЛЁННЫЕ цвета, описанные в Palette.bmp)
А также она создаёт файлы параметров спрайтов: относительное смещение, ширина, высота спрайтов

Sprite_2.exe - приводит файлы (которые получены программой Sprite_1.exe) к виду, удобному для представления на Pascal и C (преобразует в структуры данных)

Для ясности можно проделать следущее: взять Sprite.bmp и положить его со Sprite_1.exe и Sprite_2.exe
Далее запустить вначале Sprite_1.exe
Появятся 4 файла - S_DATA(видеообразы), S_OFFSET(смещения), S_WIDTH(ширины), S_HEIGHT(высоты)
Эти файлы - чистые бинарники
Потом запустить Sprite_2.exe
Появятся 2 файла - C.txt и Pascal.txt - структуры данные на C/Pascal для их дальнейшей вставки в исходник uMAPI-приложения

OPN.exe - позволяет прослушать патчи, встроенные в uMAPI с помощью Внешней Звуковой Карты
С помощью неё можно подобрать нужные звуковые эффекты, которые необходимы для приложений

YM2612Device.pdf - без этого описалова невозможно будет на IBM PC платформе услышать и заценить звук МикроМашины, тоесть все uMAPI-функции, работающие с Sound SubSystem останутся незамечены

Небольшое отличие, касающееся платформы IBM PC: вместо светодиода используется бипер
Когда в МикроМашине светодиод горит - на IBM PC соответственно пищит бипер :)

Управление на IBM PC:

uM	PC
UP	- UP
DOWN	- DOWN
LEFT	- LEFT
RIGHT	- RIGHT
FIGHT	- CTRL
JUMP	- ALT
SHIFT	- Пробел
START/PAUSE	- ENTER

Возможно одновременное нажатие нескольких клавиш (эмуляция Key SubSystem МикроМашины)

7) Права автора

- Заказ/проектирование/создание/испытание МикроМашины

- Создание ядра uMAPI для платформ PC и uM

- Формирование SDK для МикроМашины

- Автор имеет право без уведомления пользователей вносить изменения в интерфейс uMAPI на низком уровне, без ущерба эффективности функций, если это понадобится

- Автор имеет право расширять интерфейс uMAPI - создавать расширения (новые функции под текущий стандарт) и/или создавать новые (более совершенные) версии uMAPI

- Автор не несёт никакой ответственности за любые воздействия содержания SDK на людей

8) Права пользователей

- Пользователи имеют право использовать/безвозмездно распространять/изменять SDK

- Пользователи имеют право писать uMAPI-приложения, используя ядро uMAPI, присваивая себе авторство в написании своих приложений

- Пользователи НЕ имеют право, используя uMAPI-ядро, присваивать на него авторские права

9) Благодарности

1) Родителям за моральную и материальную поддержку в развитии проекта "МикроМашина"
2) Заводу ОАО "Дальприбор" за предоставление материально-технических ресурсов
3) Сотрудникам "Дальприбра" (конкретно "КО") за полезые советы матёрых бывалых электронщиков
4) СисАдминам "Дальприбора" за бесплатный и необъятный интернет, в целях поиска нужной инфы
5) Родиону Сенину за помощь в подготовке видео-материалов по теме "МикроМашина" и моральную поддержку
6) Роману Чунину за поиск и рассылку звуковых чипов
7) Руслану Захарову за огромнейшую моральную поддержку и будущую поддержку uMAPI
8) Своему брату Денису за полезные советы и будущую поддержку uMAPI
9) Всем тем людям, кто поддержит данный проект

© Romanich 06.06.06