

Санкт-Петербургский Государственный Университет

Математико-механический факультет

Кафедра системного программирования

**Графическая подсистема ОСРВ Embox
для роботов LEGO Mindstorms NXT 2.0**

Курсовая работа студента 345 группы
Дзендзик Дарьи Анатольевны

Научный руководитель:

Бондарев А. В.
аспирант кафедры СП СПбГУ

Санкт-Петербург

2011

Оглавление

1. Введение и постановка задачи.....	3
2. Реализация.....	4
a. Описание экрана.....	4
b. Шрифт. Вывод строк.....	5
c. Рисование объектов.....	6
d. Примеры применения и анализ эффективности.....	7
i. Выведение логотипа.....	7
ii. Текстовое меню.....	7
iii. Змейка.....	8
e. Возникшие проблемы.....	8
3. Заключение	9
a. Результат.....	9
b. Дальнейшее развитие.....	9
4. Список используемой литературы.....	10

Введение

В настоящее время по всему миру активно развивается робототехника. Роботов можно встретить уже в повседневной жизни, причем многие из них обладают хорошо развитыми мультимедийными возможностями, такими как обработка аудио- и видео- информации.

Простейшим примером робота является робот-конструктор LEGO Mindstorms. Несмотря на примитивность, он оснащен встроенным экраном для вывода информации и умеет воспроизводить звук.

Embox – модульная операционная система реального времени для встроенных систем, разрабатываемая на математико-механическом факультете СПбГУ. ОС портирована на несколько платформ: SPARC, x86, Microblaze и ARM. Embox успешно запускается на роботах LEGO Mindstorms NXT 2.0.

Моя работа была направлена на создание графической подсистемы для ОСРВ Embox на платформе LEGO.

Постановка задачи

Основной модуль робота оснащён черно-белым экраном, звуковым динамиком и четырьмя кнопками.

Вывод на экран – самый простой и понятный вариант получения обратной связи. Он позволяет определить, что именно происходит в системе в данный момент или узнать о произошедшей ошибке. Это дает возможность, используя экран и кнопки, производить простейшую отладку не подключая робота к компьютеру.

Кроме того экран вместе с кнопками представляет собой эффективный способ управления роботом через простейший графический интерфейс пользователя.

Таким образом, реализация графической подсистемы очень важна. Требуемая система должна уметь выводить на экран как текст, так и графические объекты различной структуры. Следует также отметить, что платформа Lego Mindstorms сильно ограничена по ресурсам, поэтому реализация графической подсистемы не должна быть громоздкой.

Реализация

Описание экрана

Робот имеет дисплей размером 100x64 пикселя. Имеется некий буфер, который затем транслируется на экран. Буфер состоит из восьми вертикальных ячеек в 8 байт (1x8 пикселей). Экран в ячейках имеет размер 100x8. Ячейки располагаются сверху вниз и справа налево (рис 1). В каждой ячейке хранится число от 0 до 255. В двоичном виде – восьмизначное число, «1» соответствует чёрному пикселю, «0» - белому. Числа представляются в шестнадцатеричном виде. Заполнение ячейки идет снизу вверх. Соответственно числу 0x40 (0100 0000 или 64) будет соответствовать одна чёрная точка, расположенная на второй снизу (соответственно 7 сверху) позиции (рис. 2 а), а числу 0xE7 (1110 0111 или 231) будет соответствовать чёрная полоска с двумя белыми точками посередине (рис. 2 б).

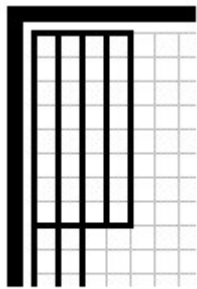
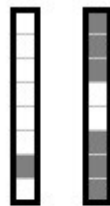


рис. 1



а б
рис. 2

Наличие буфера позволяет накапливать некоторые изменения и потом одновременно обновить дисплей робота. Возможность считывать значения ячеек из буфера позволяет проводить логические операции с записанными числами (например, инверсию: инверсия нулей и единиц соответствует инверсии цвета) и изменять произвольную область экрана независимо от того, скольким ячейкам или в каком месте ячейки находится эта область.

Шрифт. Вывод строк

Пяти ячеек достаточно, чтобы «нарисовать» букву английского алфавита, таким образом, каждая буква имеет размер 5x8 пикселей и представляется в виде массива из пяти чисел. Алфавит из 128-ми значений таблицы ASCII описан двумерным массивом размера 128x5, каждый элемент которого является массивом из пяти чисел (одна буква). Первая координата массива – это номер символа в таблице ASCII, вторая – номер ячейки в одном символе. Например, английская буква «а» имеет вид: {0x20, 0x54, 0x54, 0x54, 0x78} (рис. 3). Имеется метод

void display_char(int c),

который принимает код ASCII символа и выводит его на экран робота, последовательно записывая в буфер соответствующие значения с проверкой границ экрана. После вывода символа выводится пустая ячейка – разделитель. Экран обновляется, когда символ полностью записан в буфер.

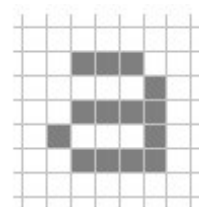


рис. 3

Вывод строки производится как последовательный вывод символов. Эту команду выполняет метод

void display_string(const char *str),

принимающий на вход строку. После вывода происходит перевод строки, т.е. последующий вывод начнётся с новой строки.

Также имеется метод, который выводит на экран первые 13 символов строки с отступом от левого края на 8 пикселей:

void tab_display(const char *str).

Если запустить его в цикле, то на экране появится список из 8 строк. Этот метод используется для вывода названия приложений в текстовом меню.

Рисование объектов

Для вывода на экран изображения используется метод:

int display_draw(uint8_t x, uint8_t y, uint8_t width, uint8_t height, uint8_t *buff).

В качестве параметров он принимает:

- координаты **x** (в пикселях) и **y** (в ячейках – 8 пикселей на единицу) – верхний левый угол (напомним, что координата **y** растёт сверху вниз);
- размер области - ширина **width** (измеряется в ячейках, т.е. по 8 пикселей на единицу), высота **height** (измеряется в пикселях) – оптимальным является соотношение 1x8 – квадратик 8x8 пикселей;
- **buff** - буфер с числовыми значениями, которые будут записаны в эту область.

Необходимо, чтобы размер буфера совпадал с шириной области. Следует отметить, что все параметры – исключительно целые положительные числа, не превышающие соответствующие им ограничения. Данный метод универсален – на экран можно вывести абсолютно любой «чёрно-белый» рисунок, размеры которого не превосходит размер экрана, но он достаточно трудоёмкий.

Для перекрашивания произвольной области в один цвет используется метод:

int display_fill(uint8_t x, uint8_t y, uint8_t width, uint8_t height, uint8_t q)

В качестве параметров она принимает:

- координаты начала области – верхний левый угол;
- ширину и высоту;
- цвет: 0 - белый, 1 - черный.

В этом методе все координаты и размеры измеряются в пикселях и не зависят от ячеек.

Метод

void display_clear_screen(void)

записывает нули во весь буфер экрана, т. о. происходит очистка экрана.

Примеры применения. Анализ и оценки эффективности

Метод вывода строк используется для вывода сообщений и любой текстовой информации.

1. Выведение логотипа

В системе Embox присутствует возможность вывести логотип системы на экран. При этом логотип опускается сверху, после чего буквы, входящие в него, поочерёдно «подпрыгивают» на месте. Для реализации анимации необходимо, чтобы между кадрами экран очищался, иначе происходит наложение одного рисунка на другой и в результате на экране изображено нечто неосмысленное. Выведение логотипа было реализовано тремя способами.

При первом был использован метод рисования букв квадратиками с полным очищением и перерисовыванием экрана в промежутках между кадрами. Это операция, на которую уходит много времени, поэтому даже человеческим глазом визуальны были задержки и мигания.

Второй метод использует метод перекраски области в один цвет. При этом становится возможно очищать (красить в белый цвет) только тот участок экрана, где происходит движение, оставляя неподвижные части без изменений. В этом случае обновление экрана происходит значительно быстрее.

В третьем случае был реализован вывод на экран логотипа только при помощи заливки области. В этом случае ожидаемо смена кадров происходит еще быстрее.

2. Текстовое меню

В системе Embox есть возможность отобразить на экране текстовое меню, содержащее список всех доступных для запуска тестовых примеров. Для отображения меню на экране был применен вывод текста с отступом. В этом случае отступ необходим для положения визуального указателя на текущее приложение. Визуальный указатель имеет вид большой точки и выводится на экран с помощью метода рисования внутри квадрата. Эта же самая функция применяется для изменения положения указателя - удаление с его предыдущего положения.

Когда указатель смещается за пределы экрана, происходит смещение списка отображаемых приложений в меню. Переход реализован двумя способами:

Первый просто перезаписывает строки, затирая название предыдущего теста названием нового. Данная операция проходит последовательно, поэтому заметны неприятные человеческому глазу мильтишения.

Второй способ очищает экран перед новой записью. В этом случае мильтишения не происходит, но процесс занимает некоторое время, и визуально наблюдаем.

3. Змейка

Традиционным для системы Embox тестовым приложением для демонстрации работы на различных архитектурах является запуск классической игры “Змейка”. При появлении поддержки дисплея на Lego Mindstorms мы смогли поиграть в эту игру прямо на работе.

Возникшие проблемы

На данном этапе, метод заливки не работает в том случае, когда внутри одной ячейки есть некоторая небольшая область, и мы хотим внутри этой области перекрасить область ещё меньшего размера. В этом случае перекрашивается полностью ячейка, включая точки, вне той области внутри которой мы красим (рис. 4 Есть а. Хотим б, получаем в). Описанная проблема решается путём считывания из буфера.

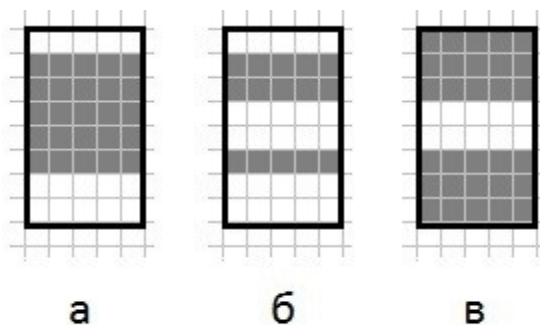


рис. 4

Заключение

Результаты

На сегодняшний день есть следующий результат:

- Рассмотрена текущая реализация графической подсистемы LEGO
- Реализован драйвер монитора для ОС Embox
- Реализовано минимальное API для вывода текстовой и графической информации
- Проведён сравнительный анализ способов вывода
- Выявлены недостатки и способы их устранения
- Реализовано визуальное текстовое меню

Развитие графического интерфейса позволило визуализировать работу системы, упростить процесс отладки и тестирования ОСРВ Embox на платформе LEGO Mindstorms NXT 2.0.

Дальнейшее развитие

В ближайшее время планируется расширять API для вывода на экран: линий по координатам начала и конца и толщине; по толщине, углу наклона и координатам начала; окружность по координатам центра и радиуса. Поддержка кириллицы. А также разбор файла в формате BMP и вывод его на экран.

Ссылки:

URL проекта: <http://code.google.com/p/embox/>

Ник: [DaryaDzen](#)

Ревизия: 2546

Библиография

1. LEGO MINDSTORMS NXT Hardware Developer Kit
2. LEGO MINDSTORMS NXT Software Developer Kit
<http://mindstorms.lego.com/en-us/support/files/default.aspx>
3. Классификация алгоритмов компьютерной графики
<http://www.codenet.ru/progr/alg/alg.php>