

**Embox — ОС RV, позволяющая запускать Linux ПО на
микроконтроллерах**

Антон Бондарев

OSSDEVCONF, 2019

Embox

Embox — свободная операционная система реального времени (RTOS), разрабатываемая для встроенных систем

Основная идея Embox использование ПО
Линукс без Линукс

“Linux” ПО

“Linux” ПО — Программное обеспечение разрабатываемое для ОС общего назначения, в том числе и для мобильных ОС

Микроконтроллер

- Система на кристалле
 - Внутренняя память
 - Надежнее
 - Меньше потребляют
 - Менее производительные

Производительность

- STM32F769i
 - Arm Cortex-M7 + FPU, 462DMIPS
 - 2MB Flash
 - 512+16+4KB RAM
- Pentium MMX
 - 233 МГц 553 MIPS
 - кэш внешний до 2 Мбайт

ПО общего назначения

- Жрущее по памяти
- Активно использует виртуальную память
- Исполняется в своем пространстве
- Грузится в оперативную память
- Использует современные языки и фреймворки

Производительность

**Калькулятор в Windows 10
занимает в состоянии простоя 16,2
МБ оперативной памяти**

Микроконтроллеры

- Используют MPU (не MMU)
- Используют гарвардскую архитектуру (отдельная шина для данных и команд)
- XIP (*execute-in-place*) исполнение напрямую из flash памяти

Производительность

- `fork()`
 - real 0m0.135s
 - user 0m0.000s
 - sys 0m0.052s
- `vfork()`
 - real 0m0.028s
 - user 0m0.000s
 - sys 0m0.016s

ПО для микроконтроллеров

- Маленькие RTOS
 - FreeRTOS, Zephyr OS, eCos, RTEMS,...
 - Собственное API
 - Не требовательны к ресурсам
- bare-metal
 - Готовые конфигурации
 - C++
 - Требуют внешних служб
- ucLinux (Linux NoMMU)

ПО для микроконтроллеров

- Мало потребляет ресурсов
- Требуется разработки почти всего пользовательского кода

Зачем Linux ПО на микроконтроллерах?

- Функциональность растет
- Переиспользование кода снижает
 - Затраты на разработку
 - Время выхода на рынок
 - Количество ошибок

Linux ПО на Embbox

- C
 - Python
 - PJSIP
 - ...
- C++
 - Qt
 - OpenCV
 - ...

C++

- Runtime
 - Конструкторы и деструкторы
 - RTTI - run-time type identification
 - Исключения (stack unwind)
- STL — шаблоны
- Boost

C++

- Можно взять библиотеки из g++
- Конструкторы располагаются в специальной секции
- Есть трудности в вызове глобальных конструкторов

Виртуальная память

- Избегаем использования `fork()`.
 - Заменяем на `vfork()` или `posix_spawn()`
- Единое адресное пространство
 - Статическая линковка все символы размещаются на этапе сборки

Типы памяти

- Характеризуются
 - временем доступа
 - энергонезависимостью
 - ценой
 - объемом
 - количеством перезаписей
 - методом доступа

Типы памяти микроконтроллера

- Внутренние - Внешние
- Доступные на системной шине
- Перезаписываемые (с произвольным методом доступа) - только на чтение (блочный доступ)

Типы памяти микроконтроллера

- 2 mb flash (Внутренняя ROM)
- 512+16+4kb RAM (внутренняя ОЗУ)
- 8 Мб SDRAM
- 16 Мб QSPI

Типы памяти микроконтроллера

- Embox + OpenCV
 - в SDRAM 40 сек
 - во внутренней памяти 2 сек

Типы памяти микроконтроллера

- Разделение на шины данных и команд
- flash - быстрая на чтение
- внутренняя память быстрая на чтение и запись

Embox

- Статически конфигурируемый
- Включаются только требуемые части
- POSIX совместимый. Файловая, сетевая и другие подсистемы.
- XIP - выполнение сразу из flash памяти
- Собственная система сборки с возможностью собирать сторонние проекты

Embox

- PJSIP (C)- 1mb flash 512kb sram
- OpenCV (C++) -2mb flash 380kb sram
- Qt (C++) 3.5 mb cam Qt

Embox PJSIP

\$(CONFIGURE) :

```
export EMBOX_GCC_LINK=full; \  
cd $(BUILD_ROOT) && ( \  
    ./aconfigure \  
        CC=$(EMBOX_GCC) \  
        CXX=$(EMBOX_GXX) \  
        CPPFLAGS="$(PJSIP_CPPFLAGS)" \  
        --host=$(AUTOCONF_TARGET_TRIPLET) \  
        --target=$(AUTOCONF_TARGET_TRIPLET) \  
        --prefix=/ \  
        $(DISABLE_FEATURES:%=--disable-%) \  
        --with-external-pa; \  
    echo export CFLAGS+="$(PJSIP_CPPFLAGS)" > user.mak; \  
    echo export CXXFLAGS+="$(EMBOX_CXXFLAGS) $(PJSIP_CPPFLAGS)" >> user.mak; \  
)  
cp ./config_site.h $(BUILD_ROOT)/pjl原因/include/pj/config_site.h  
touch $@
```


Embox PJSIP

- Добавили Port Audio интерфейс
- Ограничили количество приемных буферов до 16
- На плате с 320 kb RAM, пришлось тщательно размещать RAM

Embox PJSIP

```
include embox.kernel.task.resource.idesc_table(idesc_table_size=16)
```

```
include embox.net.util.protoent(max_aliases_num=4)
```

```
include embox.net.util.servent(max_aliases_num=4)
```

```
include embox.net.util.hostent(max_aliases_num=4,max_addrs_num=4)
```

```
include embox.compat.posix.proc.exec_stub
```

```
include embox.kernel.thread.thread_cancel_disable
```

```
include embox.kernel.timer.sys_timer(timer_quantity=4)
```

```
include embox.kernel.sched.sched
```

```
include embox.kernel.sched.idle_light
```

```
@Runlevel(2) include embox.kernel.sched.strategy.priority_based
```

```
include embox.net.skbuff(amount_skb=16)
```

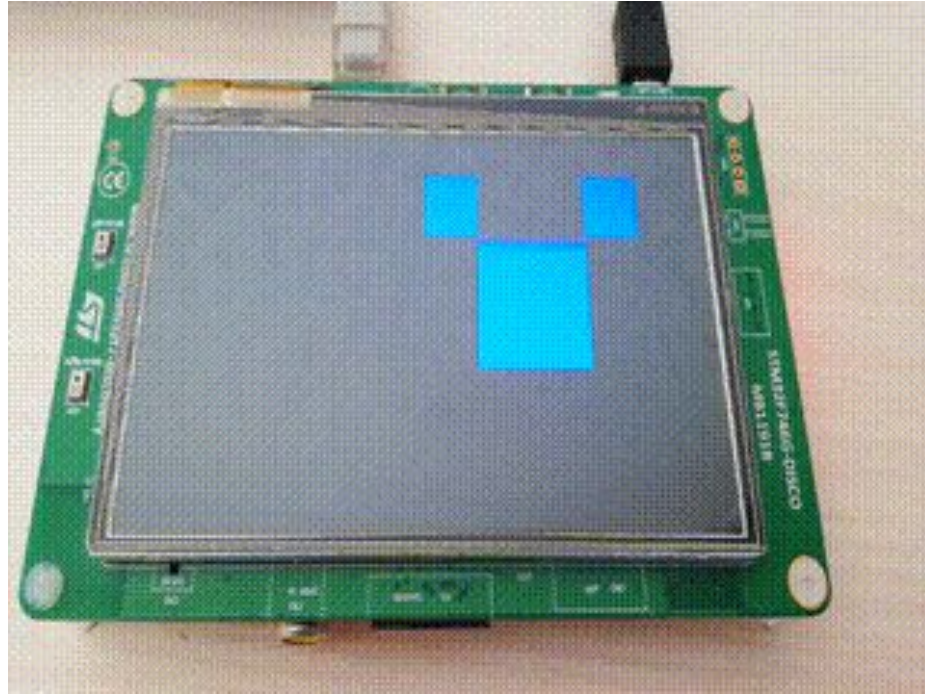
```
include embox.net.skbuff_data(amount_skb_data=16,data_size=1300)
```

...

Embox Qt

- C++ (без исключений и rtti)
- Минимальная конфигурация
- Добавили свой QPA плагин
- Освоили QSPI

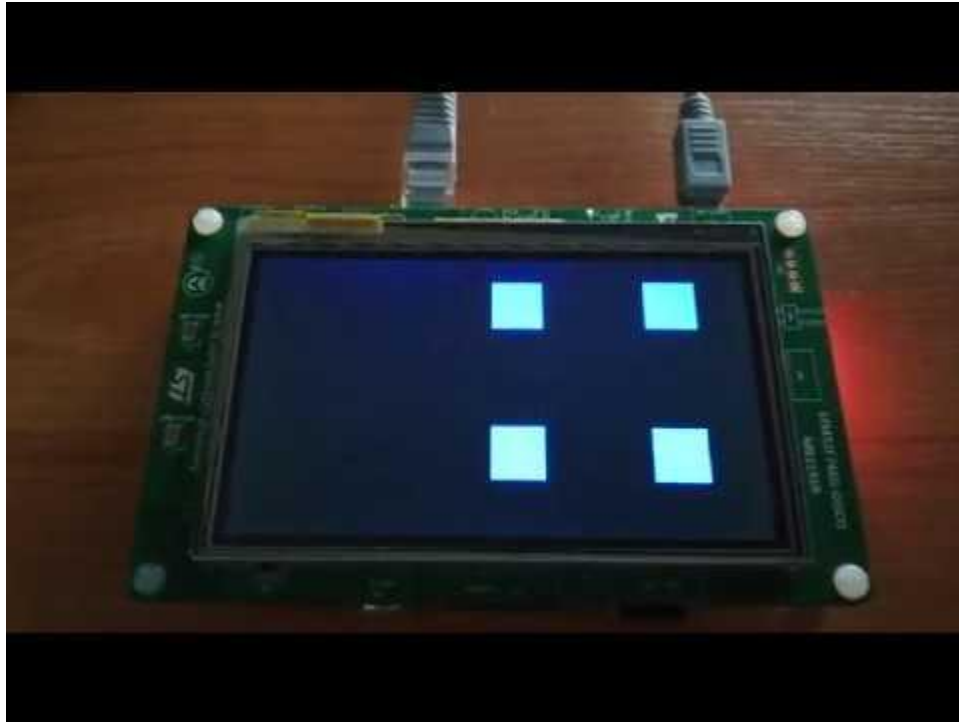
Embox Qt



Embox Qt

- Выполнение из SDRAM
 - Небольшое ускорение
- Включили плавающую точку FPU
 - Видимого ускорения не получили
- Отделили embox для запуска из чипа
 - Заметное ускорение
- Перенесли видеобуфер внутрь чипа
 - Очень заметное ускорение

Embox Qt



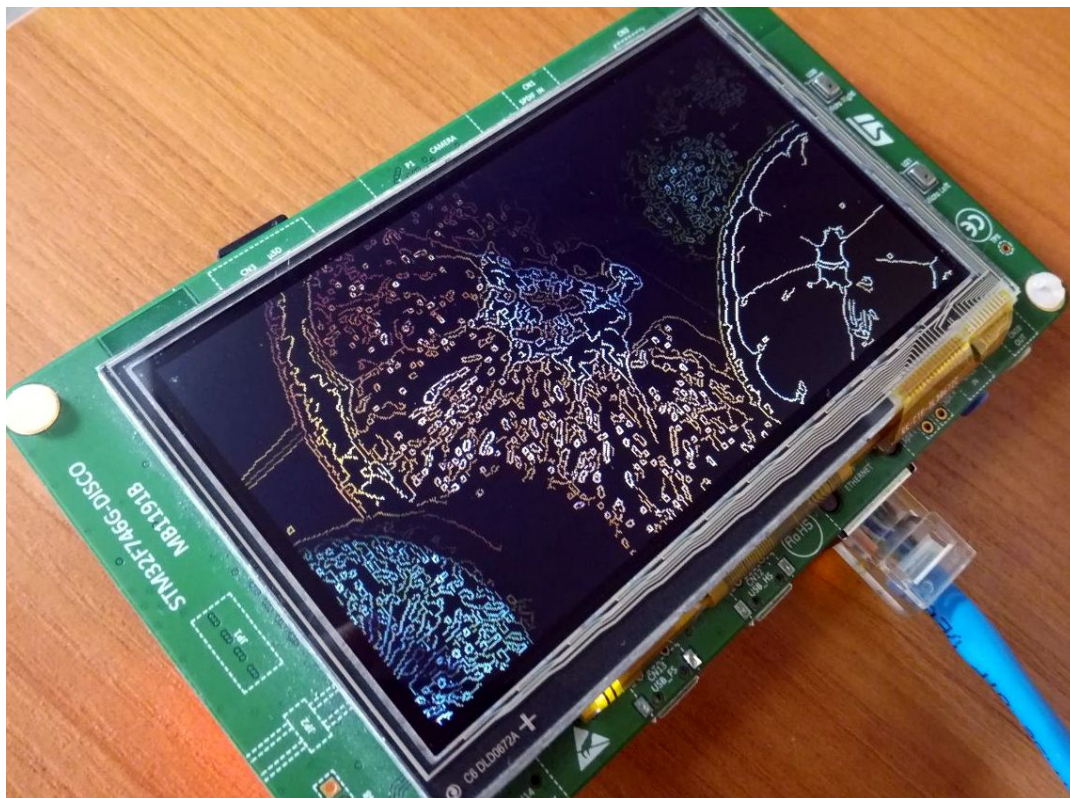
Embox OpenCV

- C++
 - Требуется полный c++ (STL, rtti, exceptions)
- Относительно плохо конфигурируется.
- Заметно влияет FPU
- Ужасно влияет размещение во внешней памяти

C++

- STLPort - 2008 год
 - Требуется полный c++ (STL, rtti, exceptions)
- Все можно взять из кросс-компилятора
- Не забыть секции
 - .eh_frame
 - .ctors
 - .dtors
- Не забыть вызов функций инициализации

Embox OpenCV



Qt на микроконтроллерах

- Команда самого Qt анонсировала qt5 на подобных платформах без ОС
- Embox — много приложений
- Направление - виджеты для автопрома и умных домов

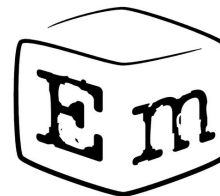
Итоги

- Универсальное ПО на микроконтроллере возможно
- Использование существующего ПО сильно уменьшает трудозатраты
- Нужно учитывать аппаратные особенности платформы

Ссылки

- `fork()` vs `vfork()`
 - <https://habr.com/ru/company/embox/blog/232605/>
- PJSIP
 - <https://habr.com/ru/company/embox/blog/431134/>
- OpenCV
 - <https://habr.com/ru/company/embox/blog/457724/>
- Qt
 - <https://habr.com/ru/company/embox/blog/459730/>

Контакты



Страница проекта



<http://embox.github.io/>

Репозиторий проекта

<https://github.com/embox/>



Антон Бондарев

anton.bondarev2310@gmail.com