

Embox

*свободная ОСРВ позволяющая запускать сложные
C++ приложения на микроконтроллерах*

Антон Бондарев

СПО: от обучения до разработки

Переславль-Залесский 2021

C++

- Объектно ориентированный язык
- Мощная стандартная библиотека включающая STL
- Небольшие накладные расходы сопоставимые с C

Микроконтроллеры

- Относительно небольшие ресурсы
- Дешевые, надежные и низкопотребляющие платформы
- Специализированные средства разработки

Embox

- Открытая и свободная ОС для встроенных систем
- Позволяет запускать ПО Линукс без изменений
- Требуется минимально необходимое количество ресурсов

C++ на микроконтроллерах

- mbed - полностью ориентирован на C++
 - <https://os.mbed.com/>
- Zephyr project
 - <https://www.zephyrproject.org/>
- NuttX
 - <https://nuttx.apache.org/>
- Keil, AIR, gcc (bare-metal)
- ...

Ограничения на C++ для MCU

- Невозможность перезапуска
- Невозможность использования некоторой функциональности из стандартной библиотеки

Базовый синтаксис

- Конструкторы и деструкторы
- Библиотека типов
- Исключения

Конструкторы

- Глобальные
 - Требуют вызова кода до входа программы в `main()`
- Локальные
 - Оператор `new()`

Глобальные конструкторы

```
void cxx_invoke_constructors(void) {  
    extern const char _ctors_start, _ctors_end;  
    typedef void (*ctor_func_t)(void);  
    ctor_func_t *func = (ctor_func_t *) &_ctors_start;  
  
    ....  
  
    for ( ; func != (ctor_func_t *) &_ctors_end; func++) {  
        (*func)();  
    }  
}
```

Глобальные деструкторы

- Два варианта
 - Используют `__сха_atexit()`
 - Сохраняют указатели в специальных секциях `.fini_array` и `.fini`.
 - `-fno-use-сха-atexi`

Глобальные деструкторы (Embox)

```
int __cxa_atexit(void (*f)(void *), void *objptr, void *dso) {  
    if (atexit_func_count >= TABLE_SIZE) {  
        printf("__cxa_atexit: static destruction table overflow.\n");  
        return -1;  
    }  
}
```

```
    atexit_funcs[atexit_func_count].destructor_func = f;  
    atexit_funcs[atexit_func_count].obj_ptr = objptr;  
    atexit_funcs[atexit_func_count].dso_handle = dso;  
    atexit_func_count++;
```

```
    return 0;
```

```
};
```

Глобальные деструкторы (Zephyr)

```
int __cxa_atexit(void (*destructor)(void *), void *objptr, void *dso)
{
    ARG_UNUSED(destructor);
    ARG_UNUSED(objptr);
    ARG_UNUSED(dso);
    return 0;
}
```

Сравнение подходов

- ОС для микроконтроллеров
 - Единственное приложение
 - Перезагружается вся система
 - Необходимо учитывать эти факты при разработке
- Embox
 - Множественные приложения
 - Можно перезапустить приложение
 - Разработка логики системы

Операторы new/delete

```
void* operator new(std::size_t size) throw() {  
    void *ptr = NULL;  
  
    if ((ptr = std::malloc(size)) == 0) {  
        if (alloc_failure_handler) {  
            alloc_failure_handler();  
        }  
    }  
  
    return ptr;  
}  
void operator delete(void* ptr) throw() {  
    std::free(ptr);  
}
```

RTTI & Exceptions

- Более сложная функциональность
- Лучше использовать `libsupc++` из кросс-компилятора
 - Совместимость с линкер скриптом
 - Регистрация обработчика

RTTI & Exceptions (регистрация)

```
void register_eh_frame(void) {  
    extern const char _eh_frame_begin;  
    __register_frame((void *)&_eh_frame_begin);  
}
```


RTTI & Exceptions (линкер скрипт)

```
.ARM.exidx : {  
    __exidx_start = .;  
    KEEP(*(.ARM.exidx*));  
    __exidx_end = .;  
} SECTION_REGION(text)
```

```
.ARM.exstab : {  
    KEEP(*(.ARM.exstab*));  
} SECTION_REGION(text)
```

Стандартная библиотека языка

```
arm-none-eabi-gcc -v
```

```
Using built-in specs.
```

```
COLLECT_GCC=arm-none-eabi-gcc
```

```
COLLECT_LTO_WRAPPER=/home/alexander/apt/gcc-arm-none-eabi-9-2020-q2-update/bin/./lib/gcc/arm-none-eabi/9.3.1/lto-wrapper
```

```
Target: arm-none-eabi
```

```
Configured with: *** --with-gnu-as --with-gnu-ld --with-newlib ***
```

```
Thread model: single
```

```
gcc version 9.3.1 20200408 (release) (GNU Arm Embedded Toolchain 9-2020-q2-update)
```

Стандартная библиотека (Embox)

- Собственная стандартная библиотека
- Конфигурируемая
- Позволяет добиваться существенной экономии ресурсов
- Позволяет создавать уникальные (более безопасные) образы
- Позволяет упрощать сертификацию и верификацию
- Необходим слой совместимости с newlib

Стандартная библиотека языка

```
arm-none-eabi-gcc -v
```

```
Using built-in specs.
```

```
COLLECT_GCC=arm-none-eabi-gcc
```

```
COLLECT_LTO_WRAPPER=/home/alexander/apt/gcc-arm-none-eabi-9-2020-q2-update/bin/../../lib/gcc/arm-none-eabi/9.3.1/lto-wrapper
```

```
Target: arm-none-eabi
```

```
Configured with: *** --with-gnu-as --with-gnu-ld --with-newlib ***
```

```
Thread model: single
```

```
gcc version 9.3.1 20200408 (release) (GNU Arm Embedded Toolchain 9-2020-q2-update)
```

Стандартная библиотека

- Программа содержит
 - `std::mutex`
- Возникает ошибка
 - namespace "std" has no member "mutex"

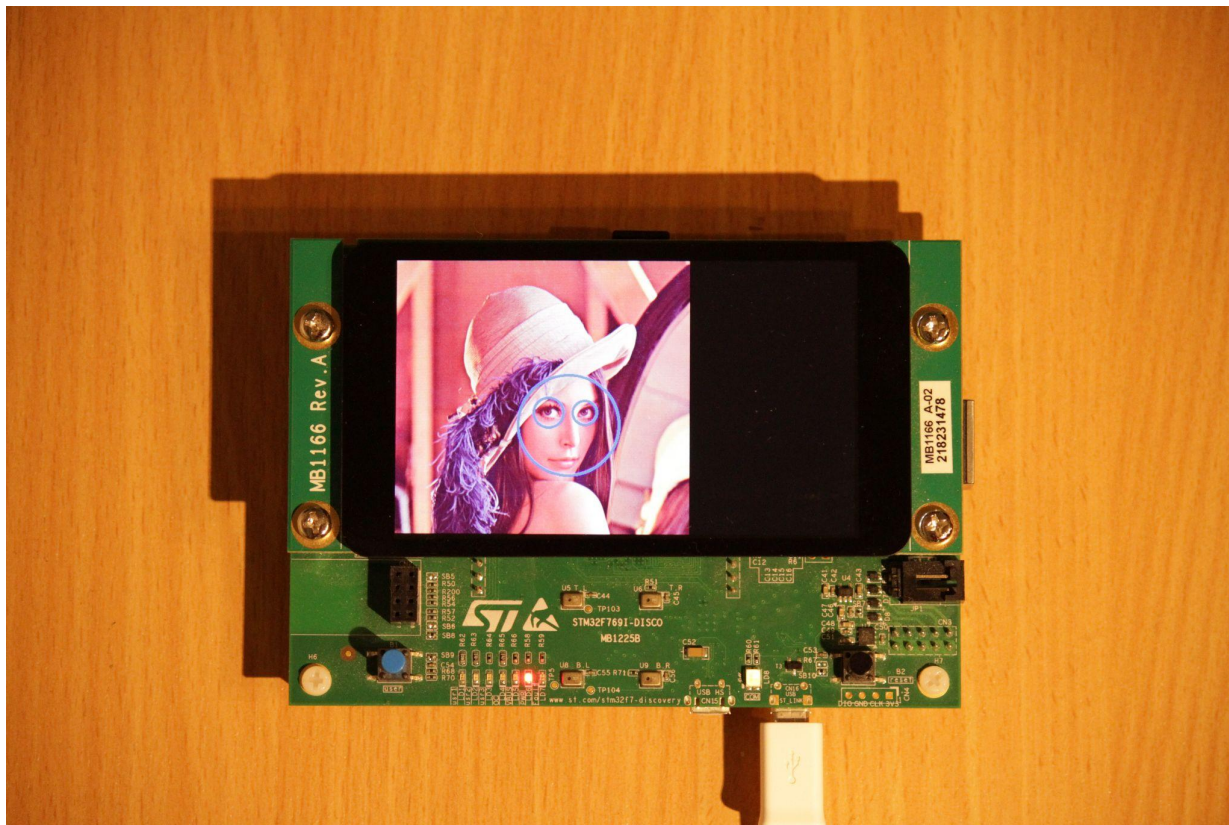
Стандартная библиотека (Embox)

- Собираем из исходников стандартную библиотеку
- Thread model: Posix

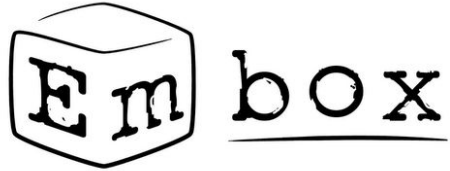
Заключение

- Embox позволяет обеспечивать на микроконтроллерах полноценную поддержку C++.
- Позволяет запускать и перезапускать множественные приложения на C++
- Позволяет использовать на микроконтроллерах такое сложное C++ ПО как OpenCV

Заключение



Contacts



Essential toolbox for embedded development



Embox Project Homepage

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



Anton Bondarev

- anton.bondarev2310@gmail.com