



ITMO UNIVERSITY

Встроенные системы

Мастер-класс 2

«Программирование микроконтроллеров без операционной системы
(bare-metal программирование)»

Быковский Сергей Вячеславович,
доцент ф-та ПИиКТ

Гончаров Алексей Андреевич,
ассистент ф-та ПИиКТ



План занятия

Теоретическая часть

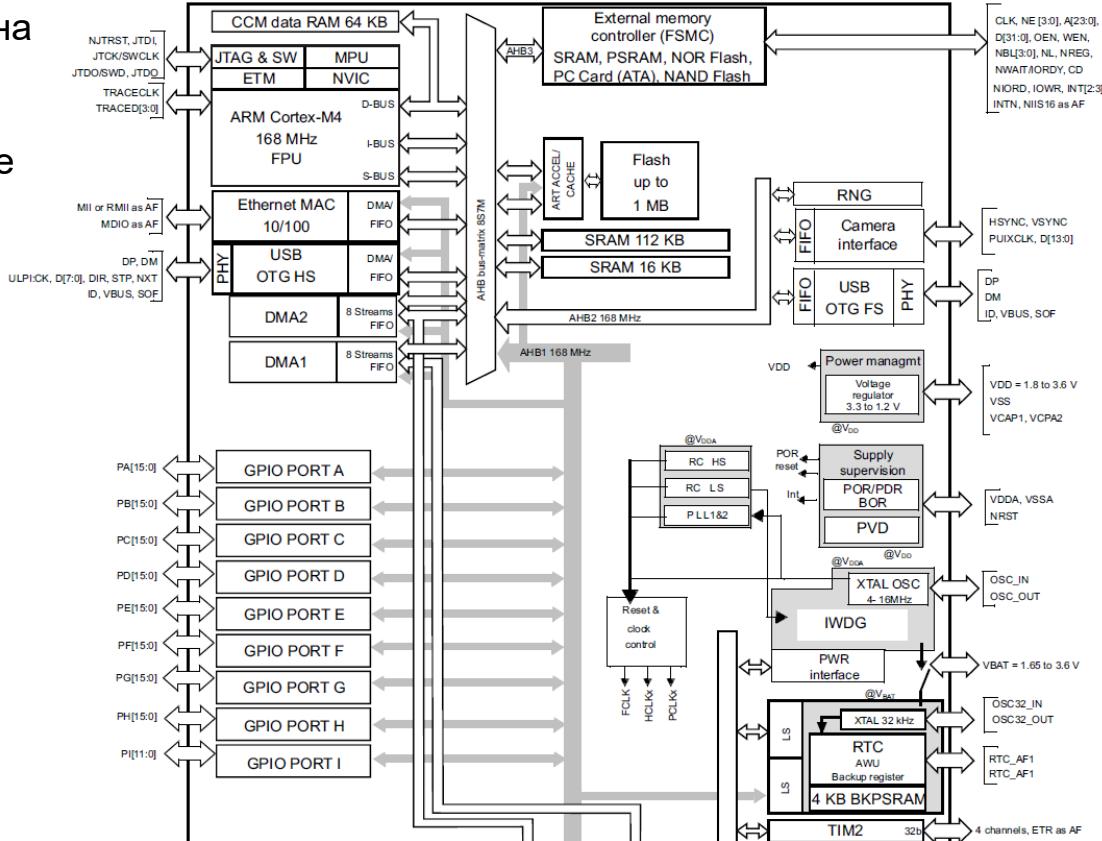
- Понятие микроконтроллера
- Роль операционной системы
- Необходимые навыки и типовые ошибки при программировании МК
- Учебный стенд SDK1.1M
- Типы обмена с внешними устройствами
- Формирование и использование цифровых сигналов

Практическая часть:

- Создание и загрузка проекта в стенд SDK1.1M
- Обсуждение задания

Понятие микроконтроллера

Микроконтроллер – это система на кристалле, включающая вычислительное ядро и систему ввода/вывода, выполненная в виде интегральной микросхемы и ориентированная на задачи управления.



Роль операционной системы

Чем хорошо иметь операционную систему?

Чем плохо иметь операционную систему?

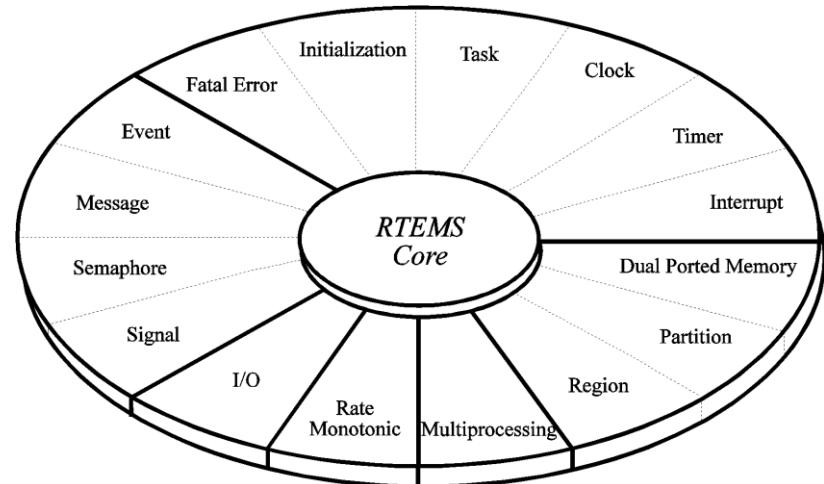
Операционные системы реального времени

Примеры операционных систем реального времени:

- RTEMS
- FreeRTOS
- QNX

Особенности:

- Имеют простые алгоритмы планирования процессов (Round Robin и т.п.)
- Обычно компилируются вместе с кодом приложений
- Ядро включает только самые необходимые компоненты: планировщик, менеджер памяти, средства межпроцессорного взаимодействия



Навыки, необходимые для программирования микроконтроллеров

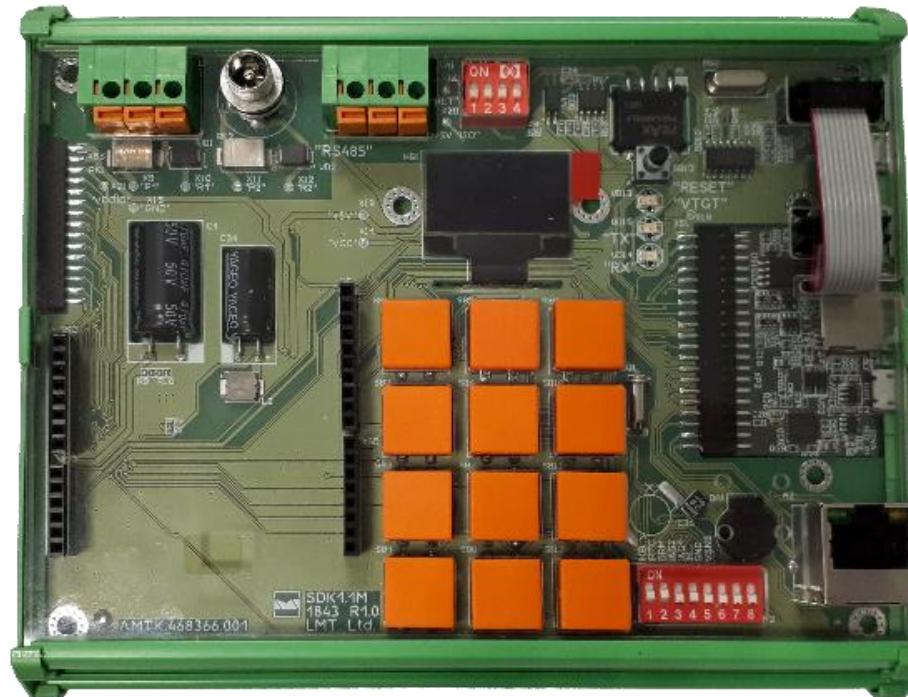
- Знание языка программирования (обычно C/C++)
- Умение программировать
- Умение читать документацию и находить нужное в сотнях и тысячах страниц
- Базовые знания схемотехники и умение читать принципиальные схемы

Типовые ошибки при программировании микроконтроллеров

- Лень читать документацию и попытка методом перебора добиться нужной функциональности
- Процессы происходят с очень большой скоростью! Многое можно просто не заметить и сделать вывод о неработоспособности!
- Компиляторы не совершенны! Вы пишете один код, а исполняется другой!
- При программировании не учитывается состояние аппаратного обеспечения и наличие ошибок в нем

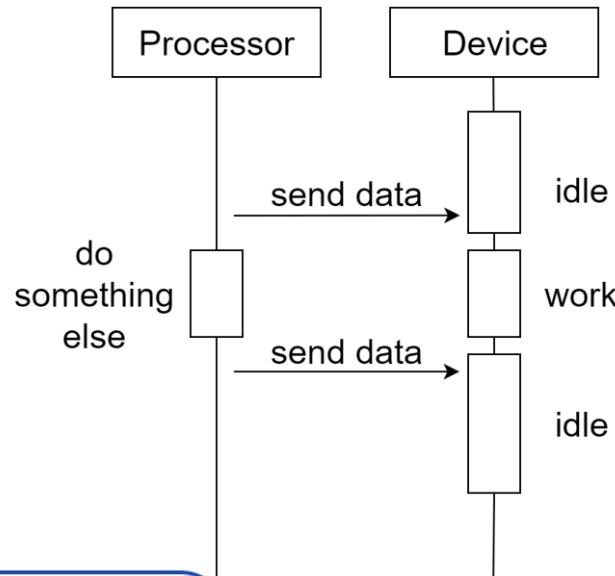
Стенд SDK1.1М

- Модульная организация
 - Совместим с модулями расширения для Arduino
 - Периферия: клавиатура, дисплей, переключатели, светодиоды, инерциальный модуль, излучатель звука, интерфейсы USB, Ethernet.

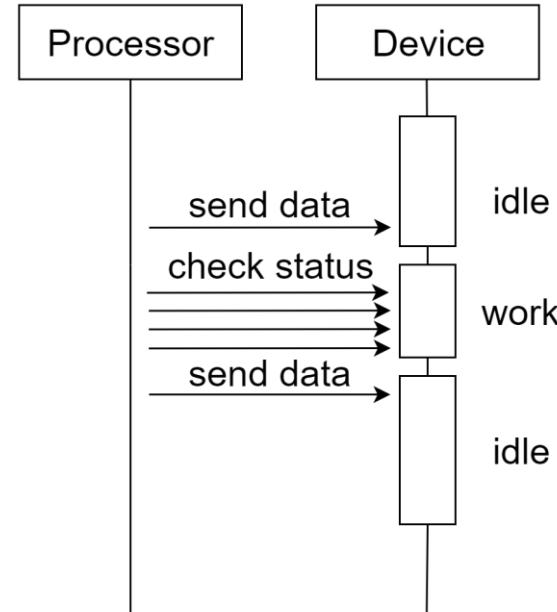


Типы обмена

Синхронный



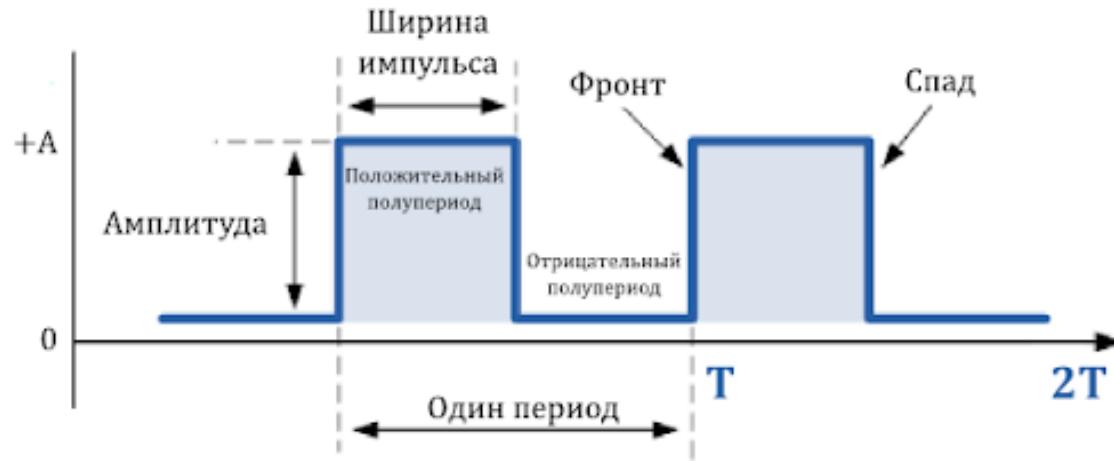
Асинхронный (по опросу, по прерыванию)



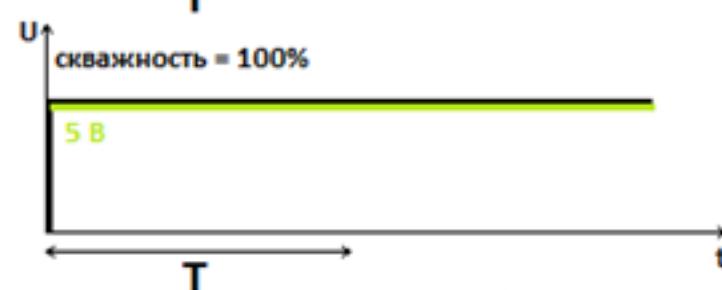
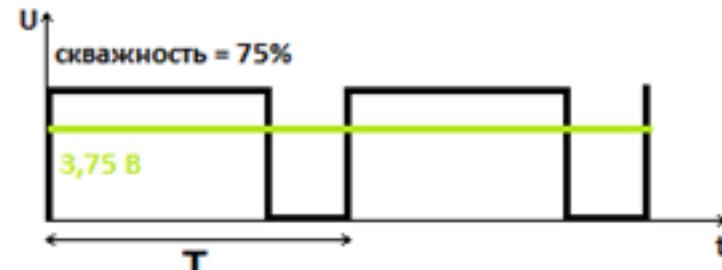
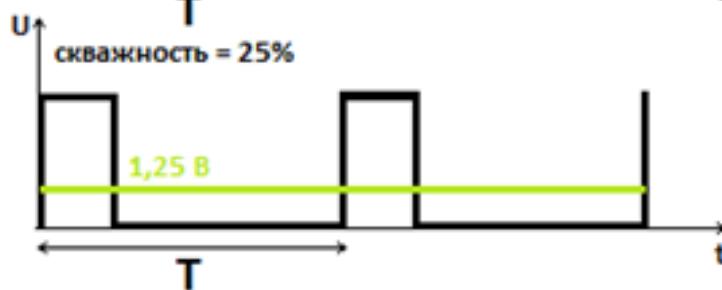
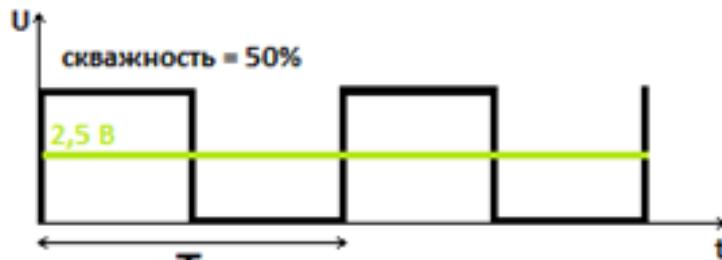
ШИМ и скважность сигналов

ШИМ – широтно-импульсная модуляция

Скважность - отношение периода сигнала к длительности импульса

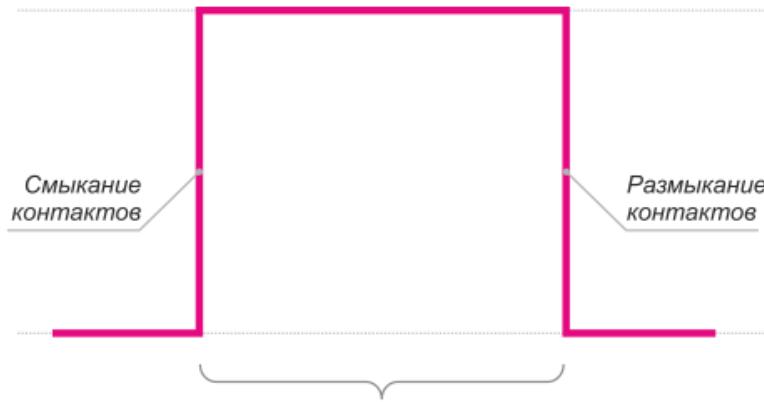


ЦАП на ШИМ

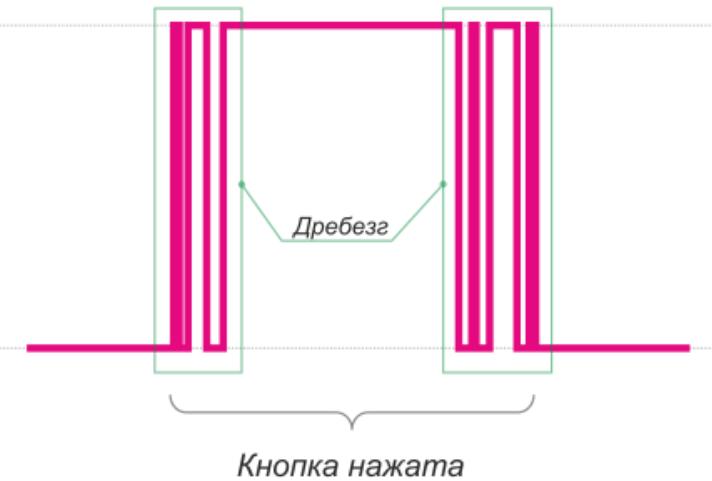


Дребезг контактов

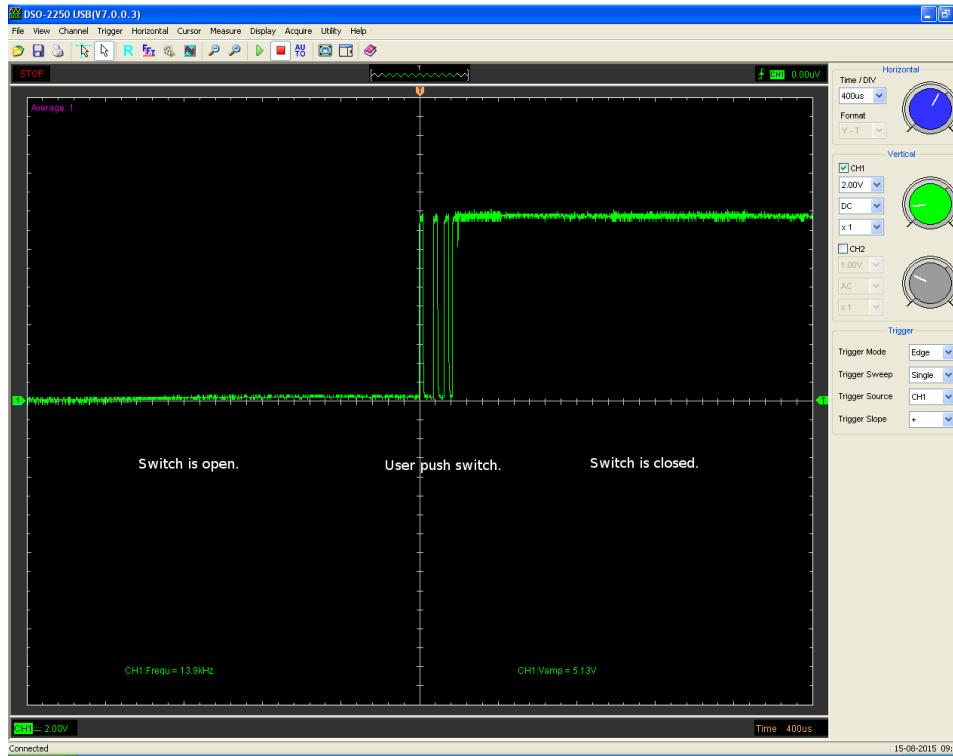
Идеальное нажатие кнопки



Реальное нажатие кнопки



Дребезг контактов



Пример проекта в STM32CubeIDE

1. Выбор микроконтроллера в мастере создания проекта
2. Включение отладочного интерфейса JTAG (5 pins)
3. Настройка дерева синхронизации/тактирования
4. Настройка порта GPIO
5. Файлы конфигурации отладчика
6. Настройка программирования и отладки
7. Установка драйверов встроенного программатора-отладчика
8. Проверка работы/возможности отладки на стенде

Задание (группа до 2-ех человек)

- Вариант 1. Написать программу, реализующую функции калькулятора. Должны поддерживаться операции сложения, вычитания и умножения целых чисел. Данные должны вводиться с клавиатуры стенда и отображаться на дисплее стенда.
- Вариант 2. Написать программу реализующую функцию таймера. Должна быть обеспечена возможность ввода начального значения с помощью клавиатуры стенда. Текущее значение таймера должно отображаться на дисплее стенда. По истечению времени, на которое установлен таймер, стенд должен подавать звуковой сигнал.

Пособие и примеры программ

- Учебное пособие: <https://books.ifmo.ru/file/pdf/3031.pdf>
- Примеры программ: <https://github.com/lmtspbru/SDK-1.1M>
(рекомендуется взять за основу проект SDK_Keyboard)