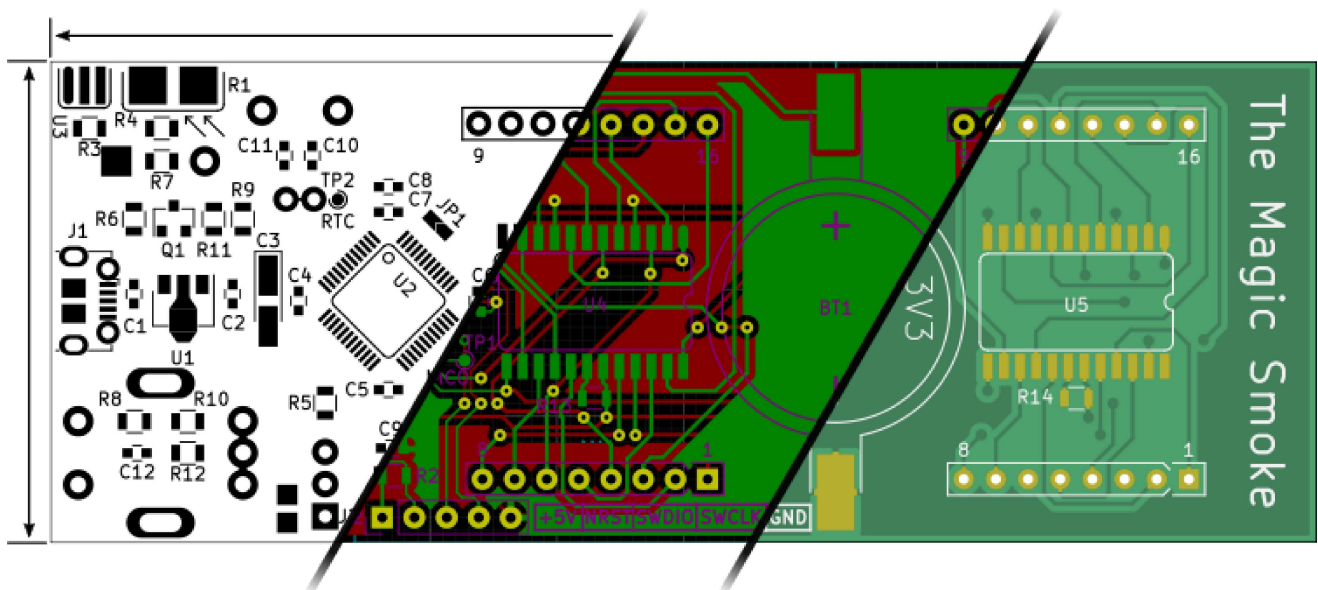


Курс «Штурмуем STM32»



Данный курс является рефлексией и логическим завершением двухлетнего руководства кружком электроники на ИРИТ-РтФ, УрФУ (НЕС – Hardware Engineering Club, позже [IMEN](#) – I Am an Engineer). Курс в был прочитан студентам 3-го курса Радиотехнического Факультета УрФУ в весеннем семестре 2016 года ([отчёт](#)), а также в осеннем семестре 2016 года школьникам СУНЦ УрФУ ([отчёт](#)). После курс был значительно переделан, добавлены новые темы, видоизменено устройство и опробован на десятиклассниках СУНЦ УрФУ осенью-зимой 2018 года. Заказать набор (**3000 ₽ + доставка**) можно связавшись с автором по [электронной почте](#). В дальнейшем появится магазин.

Цель и задачи курса

Главной целью курса является **попытка продемонстрировать весь жизненный цикл цифрового устройства**: от первоначальной идеи до реализации и управления продуктом. Основными задачами являются: научить работать с технической документацией; показать основы программирования микроконтроллерной техники; объяснить, как проектировалось устройство (от идеи и схмотехнической реализации до конечной реализации и эксплуатации устройства).

Требования

Минимальный уровень компетенций, при котором можно начинать прохождение курса, подразумевает умение писать несложные программы на [языке Си](#) с использованием арифметических, логических и побитовых операций. Умение пользоваться циклами и условиями. Умение писать функции. Понимание процедурного и модульного программирования. Понимание законов Ома, Кирхгофа. Умение анализировать несложные электрические схемы. Полноценное прохождение курса требует наличие набора.

Поддержать

Знания должны быть свободными[©]. По этой причине все материалы находятся в открытом доступе. Однако, была проделана не малая работа и потрачено очень много времени, чтобы изложить все в сжатой, но информативной форме. Моржа с набора не такая большая (меньше стоимости среднестатистического сертификата на coursera), а область весьма специфична и не популярна. Я буду рад любой поддержки данного сайта: вы можете купить набор, приобретя текст курса в виде книги (появится чуть позже), докупив книжку «[Си для встраиваемых систем](#)» или [пожертвовать средства напрямую](#). Все эти средства пойдут на еду, мотивацию создавать больше интересного материала и поддержания сайта (хостинг и домен стоят денег).

План курса

1. [Введение](#)

- [Мотивация](#). Почему стоит заниматься электроникой и встраиваемыми системами в частности?
- [Обзор набора](#). На чём вам придётся учиться?
- [Рекомендации по сборке устройства](#). Пайка — это отдельный навык. Нужно не просто научиться прикладывать жало к нужным местам, нужно понимать что ты делаешь. К тому же в некоторых случаях стоит соблюдать порядок пайки.
- [Стоковая прошивка](#). Описание возможностей стоковой прошивки, которая идёт в комплекте с набором.

2. [Теория](#)

- [Что такое микроконтроллер?](#) Раскрывается понятие микроконтроллер, какими они бывают.
- [Что понимает микроконтроллер?](#) Мы привыкли использовать десятичную систему счисления, но, увы, МК её не понимает...
- [Логические уровни и операции](#). Коротко о том, какой язык понятен микроконтроллеру.
- [Микроконтроллер под микроскопом](#). Что у микроконтроллера под капотом? Раскрываются понятия: регистр, система тактирования, архитектура и устройство памяти.
- [Библиотеки](#). Уровни абстракции над железом ускоряют разработку.
 - [Библиотека CMSIS](#). Общая библиотека для всех ARM Cortex-M ядер — CMSIS. Как она устроена и как ей пользоваться.
 - [Стандартная библиотека периферии](#). Старая библиотека от ST, которая до сих пор находит применение в реальных проектах.
 - [Низкоуровневая библиотека](#). Реинкарнация стандартной библиотеки.
 - [Слой аппаратной абстракции HAL](#). Абстракции мало не бывает. Как устроена библиотека HAL, в чём её преимущества и недостатки.

3. [Инструменты](#)

- [Инструменты разработчика](#). Какие интегрированные среды разработки поддерживают stm32? Выбор среды разработки.
- [Atollic TrueStudio](#). Официальная среда от компании ST.
- [IAR Embedded Workbench](#). Проприетарная среда разработки IAR.
- [Прошивка устройства](#). Как подключить программатор? Какой программой пользоваться?

- Генератор кода STM32CubeMX. Как быстро сконфигурировать проект и не копировать файлы библиотек самостоятельно?

4. Программирование

- Простейшая программа. С чего начать программирование? Рассмотрим простейшую программу.
- Оформление кода. Код не стоит писать, как попало. Вводим некоторые соглашения.
- Структура проекта. В устройстве много функций, проводим декомпозицию, разбиваем программу на модули.
- Система тактирования МК. Заводим микроконтроллер от внутреннего генератора (HSI) и пропускаем частоту через умножитель PLL. Поверхностно рассматриваем блок RCC.
- Порты ввода/вывода общего назначения. Для управления внешними цепочками нужно использовать порты ввода/вывода. Как они устроены и как их настроить на нужную функциональность.
- Мигаем светодиодом. Применяем знания, полученные о модуле GPIO, мигаем светодиодом.
- Прерывания, события и NVIC. Некоторые события происходят асинхронно к выполнению программы. Знакомимся с понятием прерывание.
- Обработка нажатия кнопки. Настраиваем ножку на вход, настраиваем блок EXTI, отвечающий за внешние прерывания.
- Таймеры. Обзор. Что такое таймеры и что они умеют?
- Системный таймер — SysTick. В ядро Cortex-M3 входит системный таймер. Настроим прерывание по переполнению, организуем задержку.
- Работа с SysTick без прерывания. Для более коротких задержек, можно реагировать не на прерывание, а на флаг события. Переписываем задержку под нужды датчика температуры.
- Интерфейсы передачи данных. Что такое интерфейсы передачи данных? Какие они бывают?
- Датчик температуры DS18B20. Пишем драйвер для датчика температуры, работающего по протоколу 1-Wire.
- Таймеры общего назначения. Переполнение. Кроме системного таймера в stm32 предусмотрены и другие. Настраиваем базовый таймер.
- Длительное удержание кнопки. Усложняем поведение кнопки, вешаем несколько действий на одну кнопку.
- Работа с энкодером. Простое использование. Настраиваем таймер для работы с инкрементальным энкодером.
- Работа с энкодером. Указатель на функцию. Улучшаем работу с энкодером, добавляем возможность работать с разными переменными, через указатель на функцию.
- Широтно-импульсная модуляция. Таймер может формировать широтно-импульсную модуляцию. Как это сделать и для чего это нужно? Диммирование светодиода.
- Пьезоэлектрический излучатель. Тестирование. Как работает пьезоэлектрический излучатель? Какие частоты он может воспроизвести?
- Пьезоэлектрический излучатель. Мелодия. Мелодия это не только звуки определённой частоты, а ещё и паузы между ними. Как организовать мелодию?
- Пьезоэлектрический излучатель. Синхронизация таймеров. Улучшаем работу динамика, используем два таймера и синхронизируем их между собой.
- Часы реального времени. Рассмотрим ещё один специализированный таймер. Делаем время более или менее точным.
- Аналого-цифровой преобразователь. Как преобразовать аналоговый сигнал в цифровой вид?

- **Датчик освещённости.** Если светодиодная матрица будет светить максимально ярко в ночное время, то она будет мешать спать. Используем АЦП для работы с фоторезистором.
- **Интерфейс SPI.** Как устроен интерфейс SPI, какие параметры важны в нашем устройстве? Рассматриваем возможные топологии.
- **Драйвер микросхемы MAX7219.** Настало время попробовать что-то вывести на светодиодные матрицы. Напишем драйвер для микросхемы MAX7219.
- **Драйвер дисплея.** Драйвера микросхемы недостаточно для вывода нужной нам информации на дисплей. Повышаем уровень абстракции.
- **Машина состояний.** Для организации прошивки в единую программу прибегнем к так называемой машине состояний.

5. Дополнительные главы

Данные главы не приоритетные, и будут писаться постепенно.

- **Прямой доступ к памяти.** Копировать значения из одного адреса в памяти в другой занимает процессорное время. Данную операцию можно переложить на специальный модуль.
- **Виды памяти.** Какие виды памяти бывают, в чём их особенности?
- **Внутренняя flash-память.** Что делать, когда нет EEPROM? Где можно хранить настройки устройства?
- **Идентификатор устройства.** Откуда взять идентификатор устройства?
- **Защита прошивки от копирования.** Что нужно сделать, чтобы ваше устройство не скопировали?
- **Загрузчик.** Когда устройство уже у потребителя, а в прошивке обнаружилась ошибка, единственный правильный способ сохранить лицо — это дать пользователю возможность прошить устройство без программатора.
- **Контрольная сумма CRC32.** Об аппаратном модуле подсчёта контрольной суммы.
- **Сторожевой таймер.** Что если ваша программа зависнет? Устройство превратится в кирпич... что же делать, как быть?
- **Внутренние каналы АЦП.** Во всех МК stm32 есть встроенный датчик температуры. Какие у него особенности и как им пользоваться?
- **Калибровка часов реального времени.** Погрешность в любом случае будет, но как же её нивелировать?
- **Интерфейс UART.** Что из себя представляет интерфейс UART, и как с ним лучше работать?
- **Режим низкого энергопотребления.** Конкретно в нашем устройстве вопроса автономности не стоит, однако рассмотреть режим низкого энергопотребления МК и даже применить его нам ничего не мешает.
- **Ошибки в железе.** Как и в ваших программах, в микроконтроллере могут быть ошибки на уровне железа.
- **Игра змейка.** Подсказки, как реализовать игру.

6. Создание

- **Как проектировалось устройство?** Пора разобраться в том, как именно проектировалось устройство!
- **Идея.** Почему в качестве устройства в курсе были выбраны часы?
- **Проектирование.** Настало время разобраться как было спроектировано устройство. Какие критерии были поставлены, как выбирались компоненты и т.д.
- **Разводка печатной платы.** Мало того, что устройство спроектировано на бумаге, его нужно оформить в виде печатной платы. Делать это нужно с умом, поэтому давайте разберёмся с этим вопросом?

- Автоматизация проектирования. Раньше печатные платы рисовали руками (печатные дорожки, контактные площадки) в прямом смысле этого слова. К счастью XX век нам подарил компьютер, и задача немного упростилась. Какое программное обеспечение можно использовать?
-

При написании данного курса использовались следующие программные продукты: Sublime Text 3, Typora, KiCAD, InkScape.