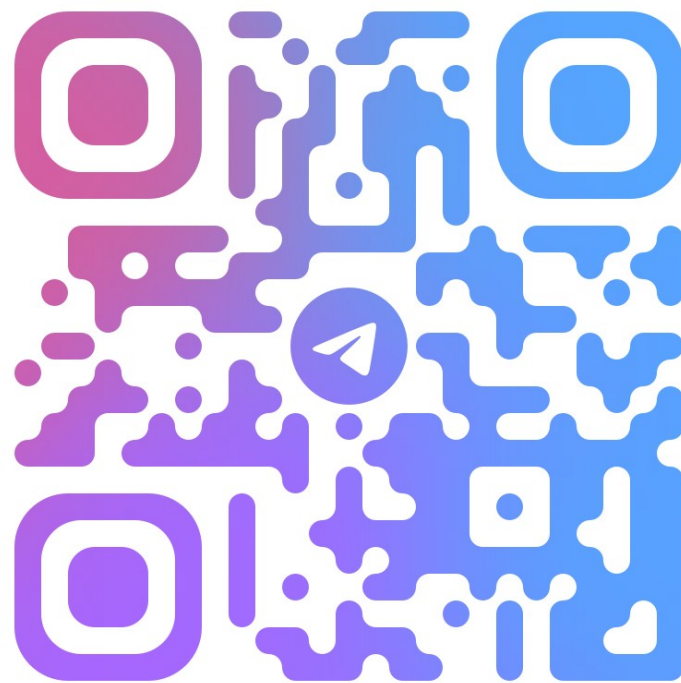


Как начать работать с микроконтроллером?



@MCUMIPTRU

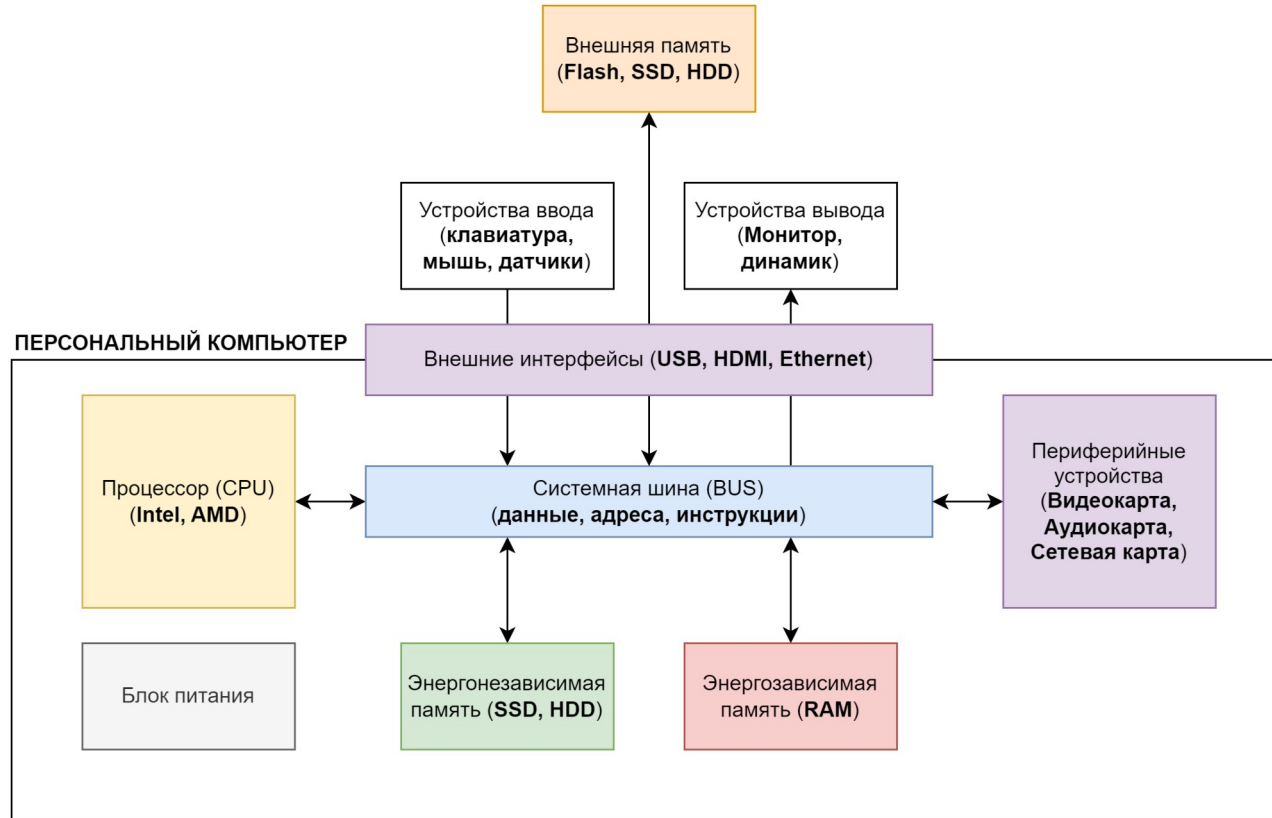


Цель занятия

- Понять что такое микроконтроллер, где они используется
- Установить набор программ для сборки прошивок
- Создать первый проект, собрать его и прошить в микроконтроллер

Путь от компьютера к микроконтроллеру

Что такое компьютер?



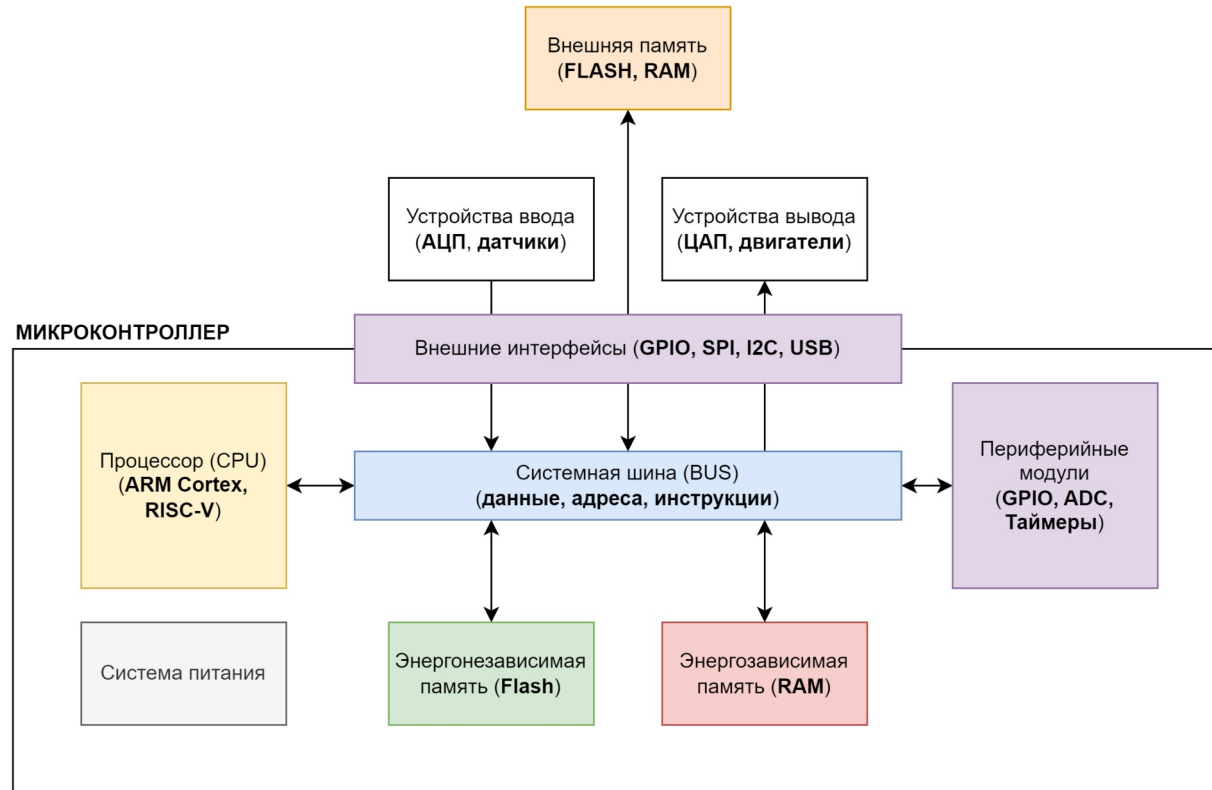
Процессор, микропроцессор, микроконтроллер

- Процессор — устройство с фиксированным набором инструкций, связанное с памятью для получения команд, чтения данных и записи результатов.
- Микропроцессор — это процессор, реализованный в виде одной интегральной схемы, на одном кристалле кремния. Сегодня практически все процессоры создаются именно так, поэтому термины «процессор» и «микропроцессор» часто используются как синонимы.
- Микроконтроллер — интегральная схема, объединяющая процессор, память для команд и данных, а также периферийные устройства.

Микропроцессор vs микроконтроллер

Компонент	Микропроцессор	Микроконтроллер
Вычислительное ядро	встроенная	встроенная
Оперативная память (RAM)	внешняя	встроенная
Постоянная память (Flash)	внешняя	встроенная
Таймеры	внешняя	встроенная
АЦП/ЦАП	внешняя	встроенная
Интерфейсы (UART, SPI, I2C)	внешняя	встроенная

Микроконтроллер



Применение микроконтроллеров

Pros&Cons

Преимущества

- Энергоэффективность
- Работа в сложных условиях
- Отказоустойчивость
- Автономность
- Предсказуемое время отклика
- Низкая стоимость

Недостатки

- Ограниченная вычислительная мощность
- Малый объем памяти
- Сложность разработки
- Ограниченная переносимость кода
- Меньше готовых решений
- Ограниченные возможности пользовательского ввода-вывода

Встраиваемые системы

Встраиваемая система (англ. Embedded System) — специализированное вычислительное устройство, встроенное в оборудование для измерения, управления или обработки данных.

Примеры:

- 1. Датчик угла атаки — измеряет угол между крылом и набегающим потоком воздуха. Если угол слишком большой — самолёт потеряет подъёмную силу. Микроконтроллер считывает сигнал с датчика и передаёт данные в систему предупреждения пилота.
- 2. FADEC (управление двигателем) — микроконтроллер получает команду пилота «увеличить тягу», считывает десятки параметров (температура турбины, давление, обороты) и рассчитывает оптимальную подачу топлива. Реакция должна быть мгновенной и точной.
- 3. Противообледенительная система — микроконтроллер следит за температурой воздуха и наличием влаги. При угрозе обледенения автоматически включает обогрев кромок крыльев и датчиков.

Что выбрать?

Микроконтроллер

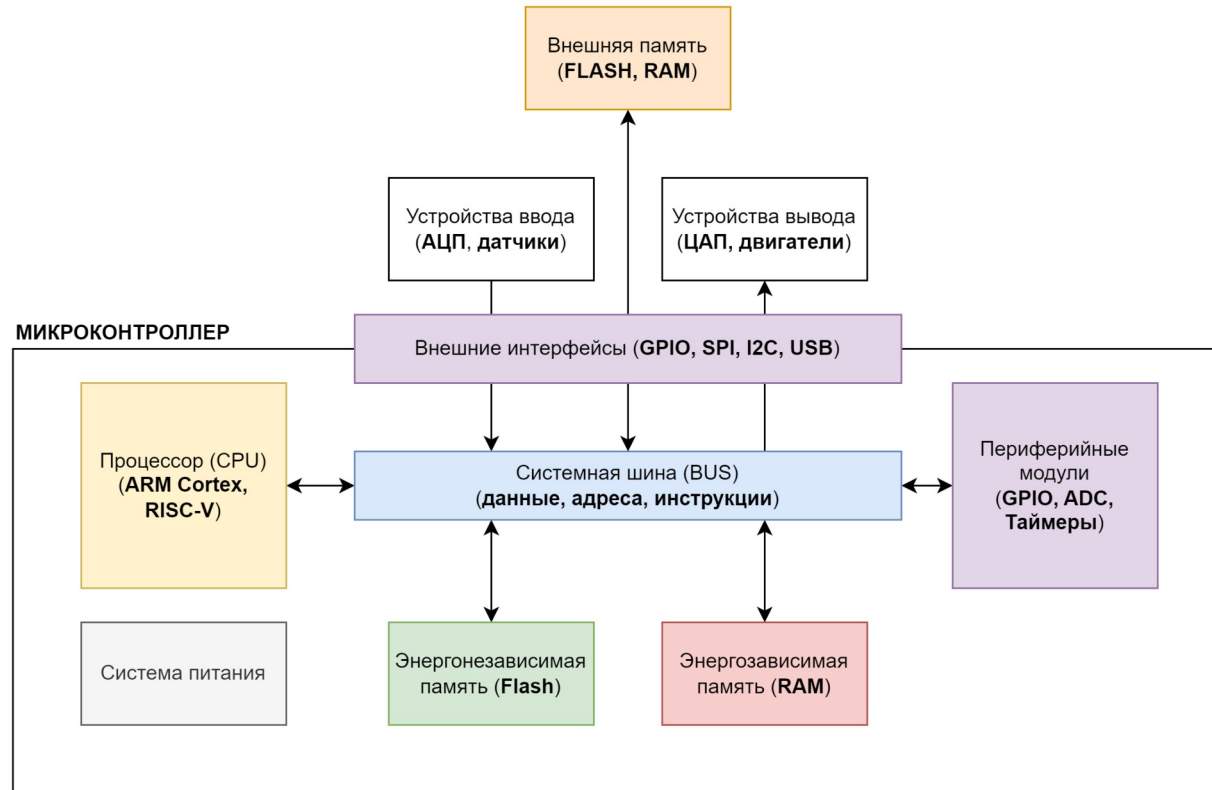
- Нужна работа в реальном времени и быстрая реакция на события
- Нужен полный контроль выполняемых вычислительных операций
- Важно низкое энергопотребление (батарейное питание)
- Нужна низкая стоимость (серийное производство)
- Простая задача (управление, измерение, связь)

Компьютер

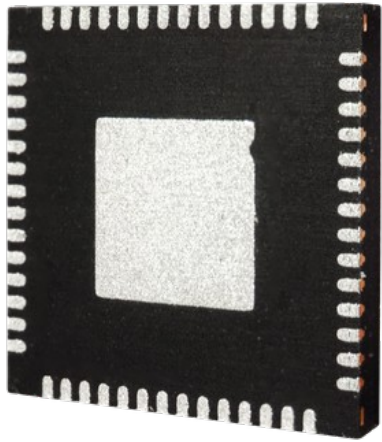
- Нужна операционная система (Linux)
- Нужна большая вычислительная мощность
- Нужна сложная работа с сетью (полноценный веб-сервер с базой данных, множество одновременных подключений)
- Нужен графический интерфейс
- Сложная обработка данных (машинное обучение, компьютерное зрение)

Устройство микроконтроллера

Микроконтроллер

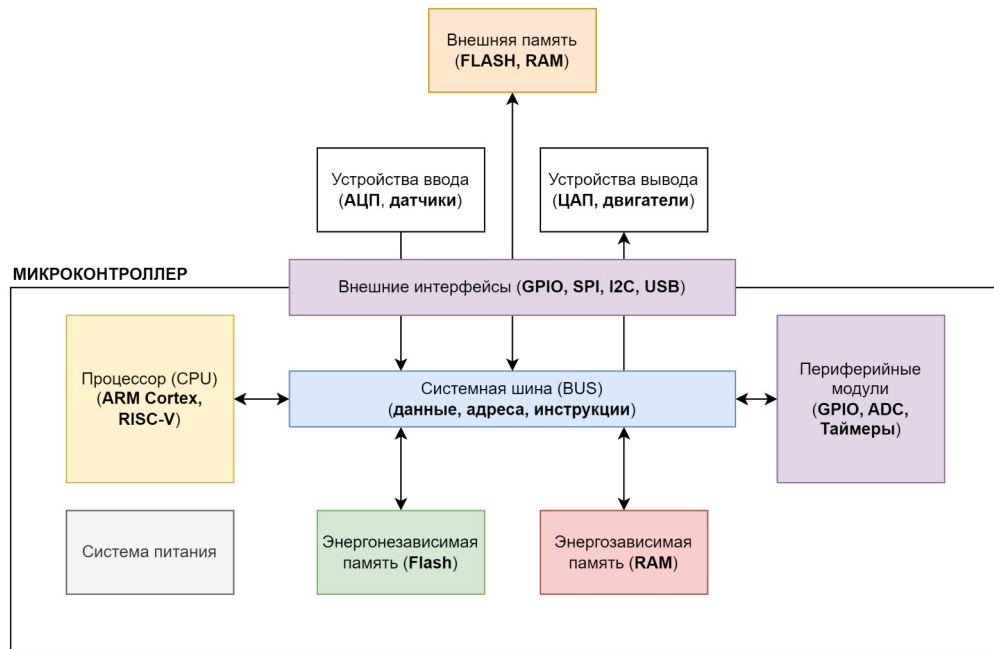
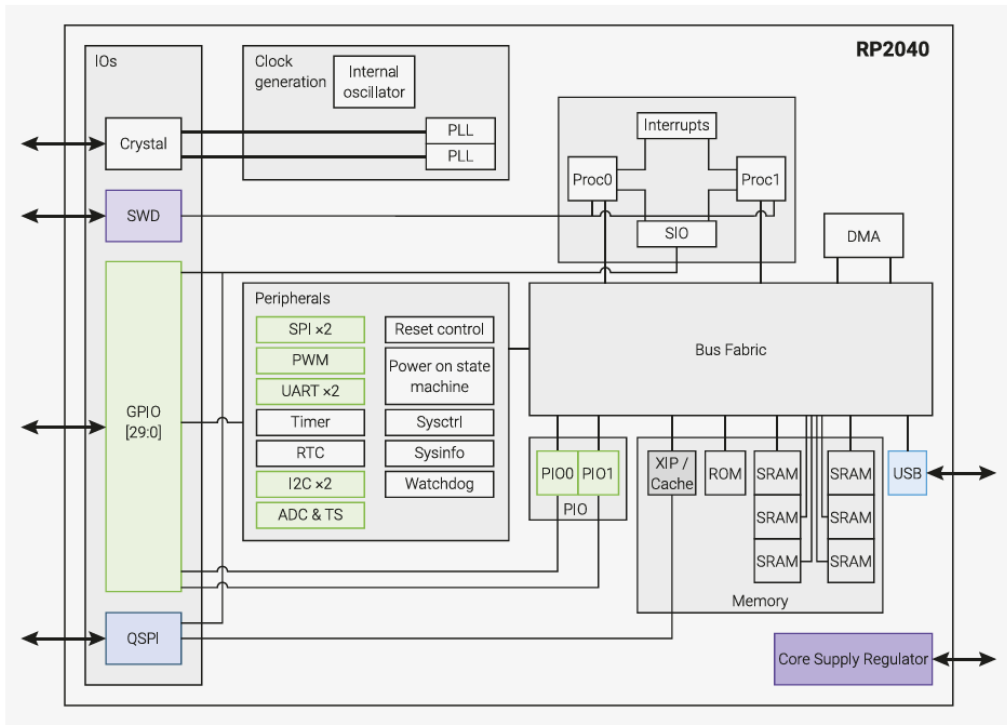


Что снаружи?



- Снаружи микроконтроллер это интегральная схема, чаще всего облачённая в корпус для защиты самого кристалла от физических повреждений, бывают специальные версии микроконтроллеров без корпусов, но это большая редкость.
- По бокам микроконтроллера или снизу выведены множество электрических контактов, которые называются пинами или "ножками" микроконтроллера. Каждый из них имеет собственный функционал и назначение.

Что внутри?



ВПО (прошивка)

Встраиваемое программное обеспечение

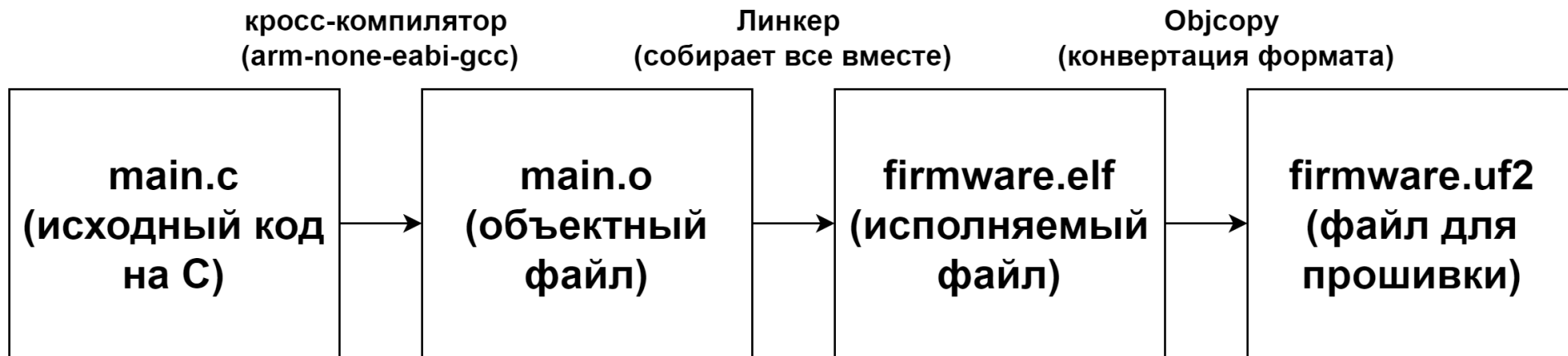
- Встроенное программное обеспечение (ВПО, англ. Emebedded Software) — программа (или набор программ) необходимый встраиваемой системе для выполнения её специализированных функций.

Прошивка vs программа ПК

Прошивка	Программа ПК
Работает без ОС	Запускается под управлением ОС
Хранится во Flash МК	Хранится на диске/SD-карте
Работает постоянно (пока есть питание)	Можно запустить и остановить
Одна программа на МК	Можно запустить несколько программ
Программа сама управляет всем	ОС управляет памятью и устройствами

Сборка прошивки

Путь от исходного кода до прошивки



Кросс-компиляция

Когда вы компилируете программу на самом Raspberry Pi для него же, компилятор создаёт код для процессора ARM в Raspberry Pi. Программа компилируется и запускается на одной и той же платформе.

Когда вы создаёте прошивку для микроконтроллера, ситуация другая:

- **Компиляция** происходит на вашем ПК (процессор x86 или ARM)
- **Выполнение** будет на микроконтроллере (процессор ARM Cortex-M)

Это называется **кросс-компиляция** (cross-compilation) — компиляция на одной платформе для выполнения на другой.

Поэтому нужен специальный **кросс-компилятор**. Для ARM микроконтроллеров обычно используется **arm-none-eabi-gcc**.

SDK — готовые программные компоненты

Варианты создания прошивки

Без SDK

- Прочитать 650 страниц datasheet RP2040
- Найти адреса регистров GPIO
- Написать код для записи в эти регистры
- Разобраться с битовыми масками
- Настроить систему тактирования
- ...

С использованием SDK

- Использовать готовую функцию из SDK: `gpio_put(25, HIGH);`

Что входит в SDK?

- Драйверы периферии
- Стартовый код
- Примеры
- Система сборки
- Документация

Программатор и запись прошивки

Программатор

Программатор — это аппаратное устройство, которое записывает прошивку во Flash микроконтроллера. Для управления программатором используется специальное программное обеспечение (например, OpenOCD, picotool, ST-Link Utility).

Работа с документацией

Datasheet

Что внутри:

- - Характеристики микроконтроллера
- - Распиновка (какой вывод за что отвечает)
- - Электрические параметры (напряжение, ток, температура)
- - Описание периферии

Когда использовать:

- - Нужны характеристики (сколько GPIO? сколько тока выдаёт вывод?)
- - Нужна распиновка (какой пин подключить?)
- - Нужны электрические параметры (можно ли подключить 5V устройство?)

SDK documentation

Что внутри:

- - Описание всех функций SDK
- - Параметры функций
- - Примеры использования

Когда использовать:

- - Нужно понять, как использовать функцию
- - Нужно узнать, какие параметры принимает функция

Examples

Что внутри:

- Готовые рабочие примеры
- Код для GPIO, UART, SPI, I2C и т.д.

Когда использовать:

- Нужен рабочий пример для старта
- Нужно понять, как использовать периферию