Санкт-Петербургский государственный электротехнический
университет “ЛЭТИ” им. В.И. Ульянова (Ленина)

МИКРОПРОЦЕССОРНЫЕ УСТРОЙСТВА СИСТЕМ УПРАВЛЕНИЯ

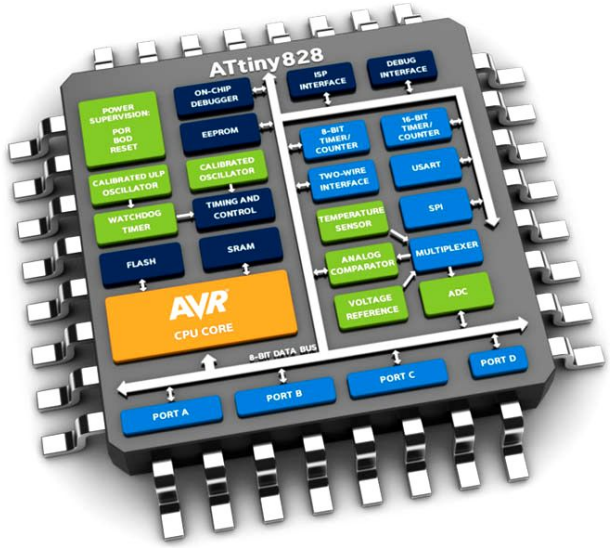
ЦИФРОВАЯ И МИКРОПРОЦЕССОРНАЯ ТЕХНИКА В УПРАВЛЕНИИ

Татаринцев Н.И.

Санкт-Петербург
2019 г.

МИКРОКОНТРОЛЛЕРЫ

Микроконтроллер - микросхема, предназначенная для управления различными электронными устройствами. Отличается от ***микропроцессора*** тем, что объединяет на одном кристалле (в одном корпусе) процессор, память (ОЗУ и/или ПЗУ) и периферийные устройства (порты ввода/вывода, таймеры/счетчики, аналого-цифровые и цифроаналоговые преобразователи, широтно-импульсные модуляторы и другие).



Работой микроконтроллера управляет программа — последовательность команд, заранее загруженная в память.

Первые контроллеры появились в 1971 году (патент фирмы Texas Instruments, предложившей использовать на одном кристалле процессор, память и устройства ввода/вывода).

По существу представляет собой однокристальную микро-ЭВМ (сейчас термин «однокристальная микро-ЭВМ» практически не используется), рассчитанную для решения задач управления.
Термин контроллер (controller) используется в технике для обозначения управляющих устройств.

МИКРОКОНТРОЛЛЕРЫ ARM

ARM *Advanced RISC Machines*

RISC *Reduced Instruction Set Computer*

архитектура процессоров с “сокращённым” набором команд. RISC архитектура получила широкое распространение за счет сочетания высокой производительности и низкого энергопотребления. К числу широко используемых в управлении можно отнести ARM микроконтроллеры семейства **CORTEX**.

CORTEX Семейство процессоров ARM

ПАМЯТЬ МИКРОКОНТРОЛЛЕРА

ROM

Для хранения исполняемого кода программ используется **flash** память (**ROM**), называемая также **памятью программ**. Эта память является энергонезависимой.

RAM

Различные изменяющиеся данные, используемые программой во время ее выполнения, хранятся в **оперативной памяти (RAM)**, она же **память данных (ОЗУ)**. Эта память не является энергонезависимой. Данные в ней сохраняются только при включенном питании микроконтроллера.

EEPROM

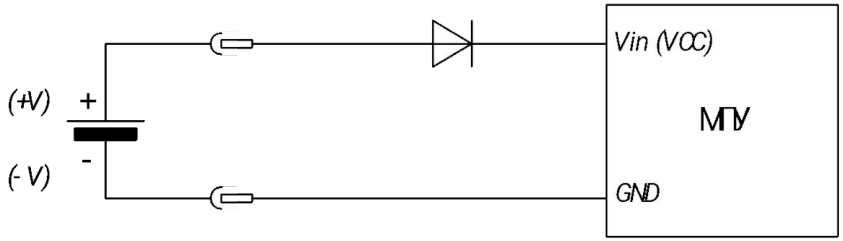
Для хранения постоянных данных (редко изменяющихся) используется **память EEPROM**. Эта память энергонезависимая. В некоторых микроконтроллерах (в частности ARM Cortex) она не имеет своего физического носителя. В этом случае она эмулируется программно, а в качестве физического носителя используется **flash** память.

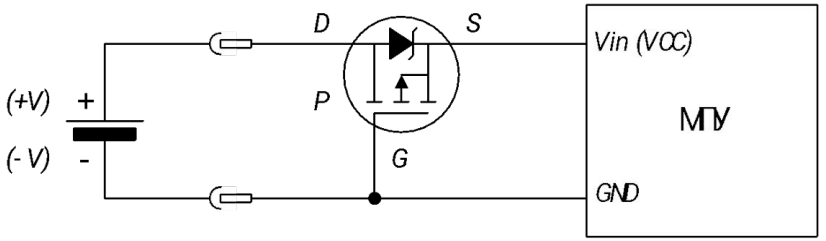
ИНТЕРФЕЙСЫ МИКРОКОНТРОЛЛЕРА

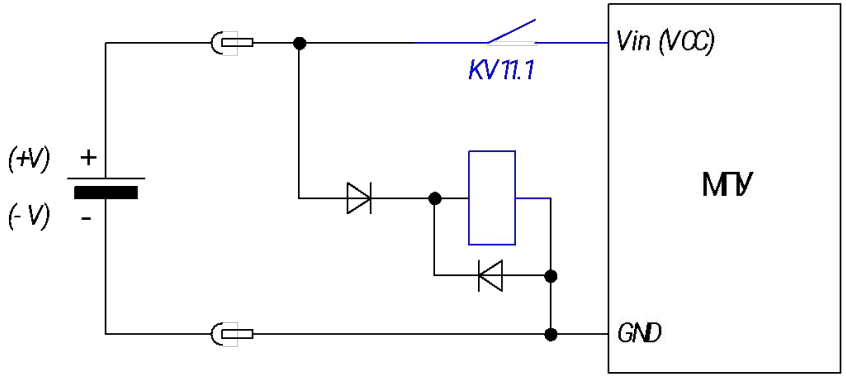
Интерфейс	Описание
USB	Интерфейс передачи данных, предназначен для подключения к микроконтроллеру USB клавиатур, flash накопителей, а также для подключения микроконтроллера к ПК
CAN	Промышленный интерфейс передачи данных, используется в промышленных системах автоматизации, преобразовательной технике, системах числового программного управления, автомобилестроении и т.п.
USART **UART**	Универсальный синхронный (асинхронный) приемо-передающий интерфейс, используемый для обмена данными с устройствами по последовательному каналу связи через RS-232, RS-485 и т.п..
I2C	Двухпроводная шина передачи данных, служит для подключения различных микросхем и других модулей
SPI/(I2S)	Последовательный интерфейс передачи данных, служит для подключения различных микросхем, карт памяти, LCD дисплеев с последовательной шиной и т.п.
FSMC	Интерфейс подключения внешней SRAM, PSRAM, NOR, NANDROM, RAM, NandFlash, Compact Flash памяти, а также для подключения LCD дисплеев с параллельной шиной
SDIO	Интерфейс для высокоскоростного подключения SD карт памяти.

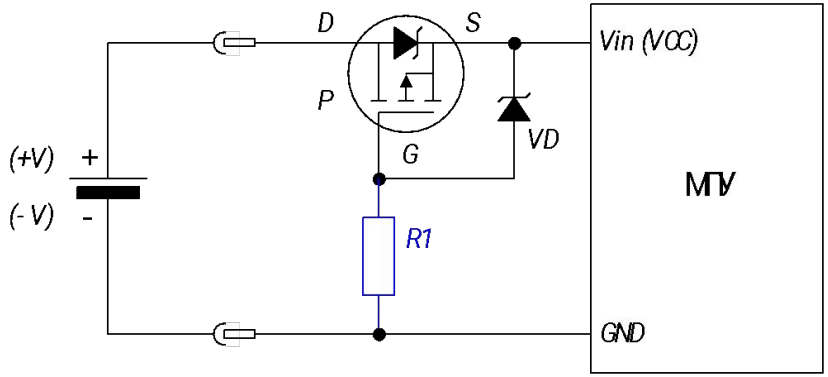
МИКРОКОНТРОЛЛЕРЫ. ЦЕПИ ПИТАНИЯ

ЗАЩИТА ОТ ПЕРЕПОЛЮСОВКИ









Защита от переполюсовки

МИКРОКОНТРОЛЛЕРЫ. ЦЕПИ ПИТАНИЯ

ЗАЩИТА ОТ ПЕРЕПОЛЮСОВКИ

Защита, выполненная на диоде, характеризуется простотой исполнения и невысокой стоимостью. К недостаткам следует отнести падение напряжения на диоде и большие потери энергии на тепло. Это делает ее практически не пригодной к использованию в мобильных системах, требовательных к энергоресурсу.

Защита, выполненная на полевых транзисторах, не имеет указанных выше недостатков. Получила наибольшее распространение в различных электронных системах, в том числе и микропроцессорных. По отношению к первому варианту является более дорогостоящей как по элементной базе, так и по разработке.

Защита, выполненная на реле, не имеет недостатков первого варианта. Главным недостатком по отношению к первым двум защитам является ограниченный срок жизни реле (количество циклов замыкания/размыкания). Может оказаться не приемлемой к использованию в слаботочных малогабаритных устройствах.

МИКРОКОНТРОЛЛЕРЫ. ВНЕШНИЕ ЦЕПИ

ЦИФРОВЫЕ ВХОДЫ/ВЫХОДЫ

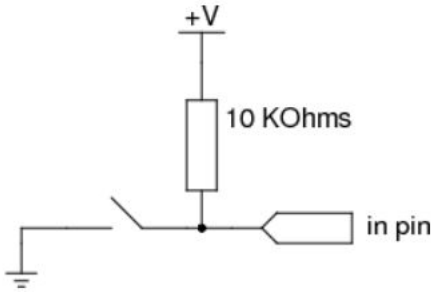
Цифровые контакты (пины, порты, входы/выходы, выводы) могут работать как цифровые **вводы**, так и как цифровые **выводы**. По умолчанию они сконфигурированы как **вводы**. Перевод в режим **вывода** осуществляется программно с помощью команды **pinMode()** . Например pinMode (INA1, OUTPUT);.

Если пин сконфигурирован как **ввод**, то он находится в **высокоимпедансном** состоянии.
Если пин сконфигурирован как **вывод**, то он находится в **низокоимпедансном** состоянии.

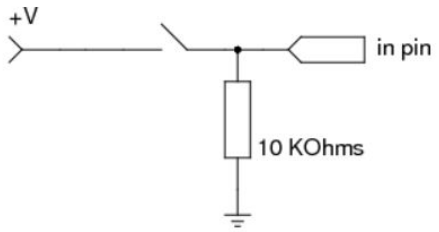
Если к входу **не подключены внешние цепи** (не поступают нужные сигналы), то он окажется в неопределенном состоянии и **будет воспринимать случайные значения** (электрические помехи). Чтобы исключить это, входу нужно задать определенное состояние – подключить либо к питанию (+5В/3.3В) либо к земле через подтягивающие (pull-up)/стягивающие (pull-down) резисторы (рекомендуется 10 кОм).

МИКРОКОНТРОЛЛЕРЫ. ВНЕШНИЕ ЦЕПИ

Аппаратная реализация:



Подтягивающий резистор



Стягивающий резистор

Подтягивающий резистор удерживает вход в состоянии логической единицы, пока внешняя цепь разомкнута, стягивающий - в состоянии нуля.

Программная реализация:

В микроконтроллерах Atmega имеются встроенные программируемые подтягивающие (подтягивают к питанию) резисторы. Программное подключение этих резисторов производится `digitalWrite()`:

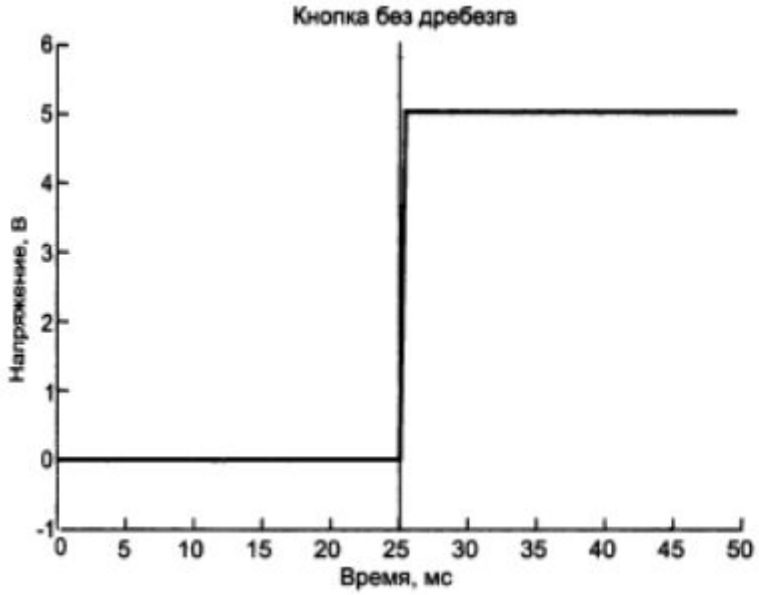
```
digitalWrite(pin, HIGH);
```

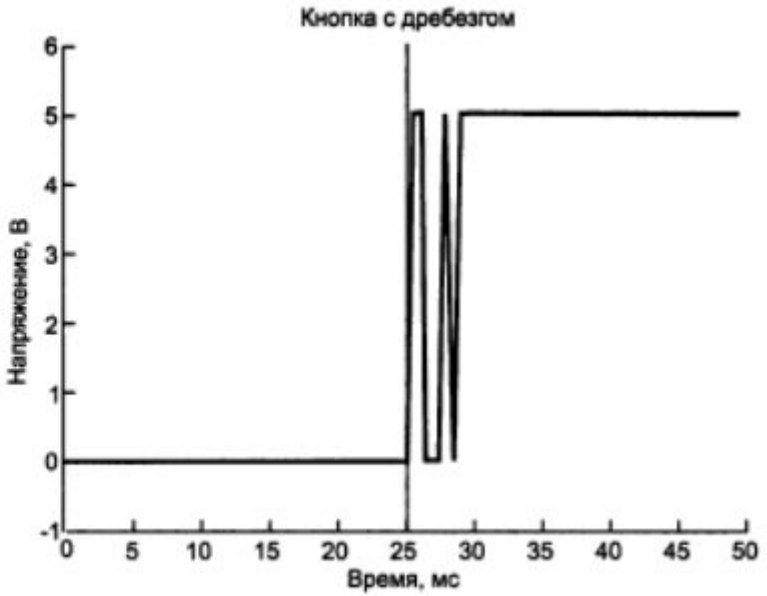
Либо в явном виде (в более современных средах IDE):

```
pinMode (pin, INPUT_PULLUP);
```

МИКРОКОНТРОЛЛЕРЫ. ВНЕШНИЕ ЦЕПИ

ДРЕБЕЗГ КОНТАКТОВ

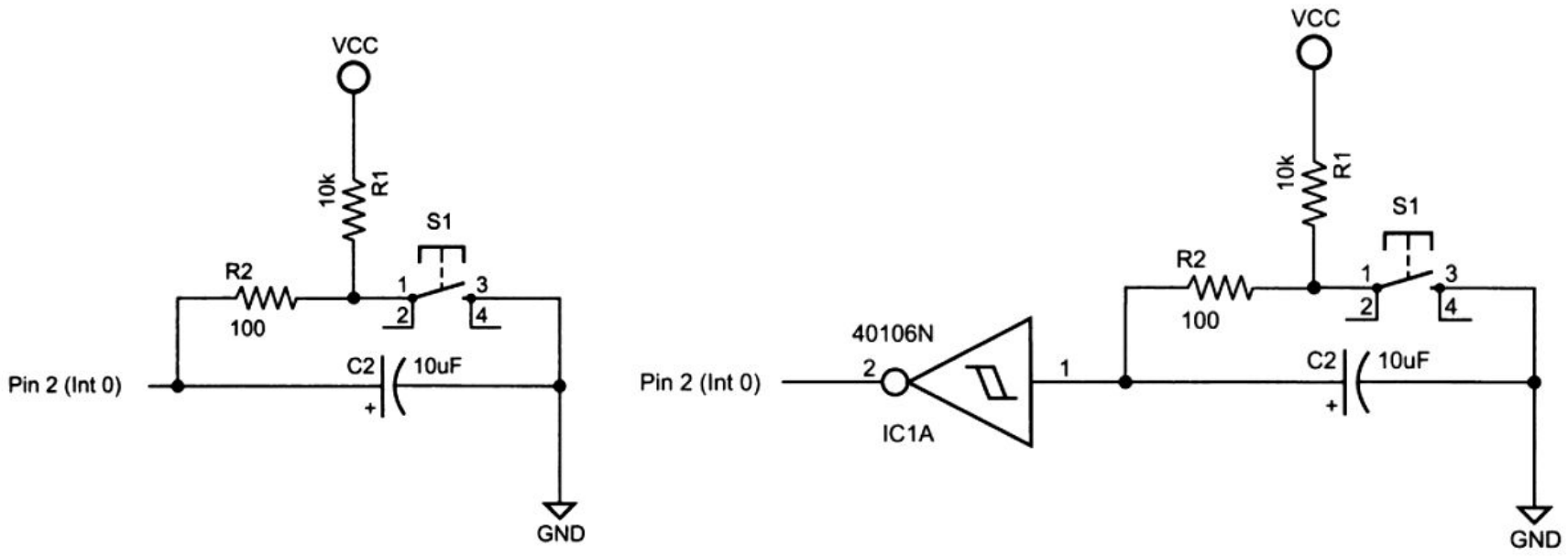




Существуют программные и программно-аппаратные способы устранения дребезга контактов (Л1:55, 257)

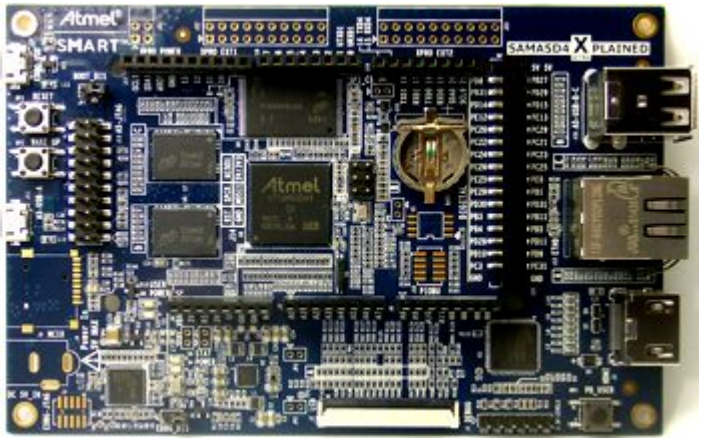
МИКРОКОНТРОЛЛЕРЫ. ВНЕШНИЕ ЦЕПИ

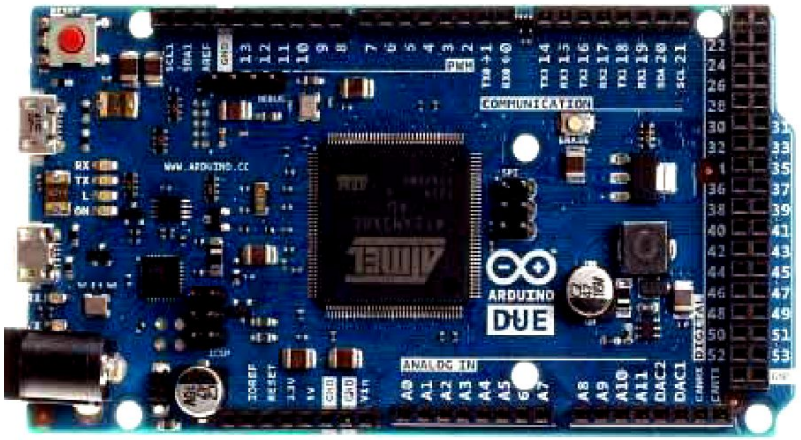
ДРЕБЕЗГ КОНТАКТОВ

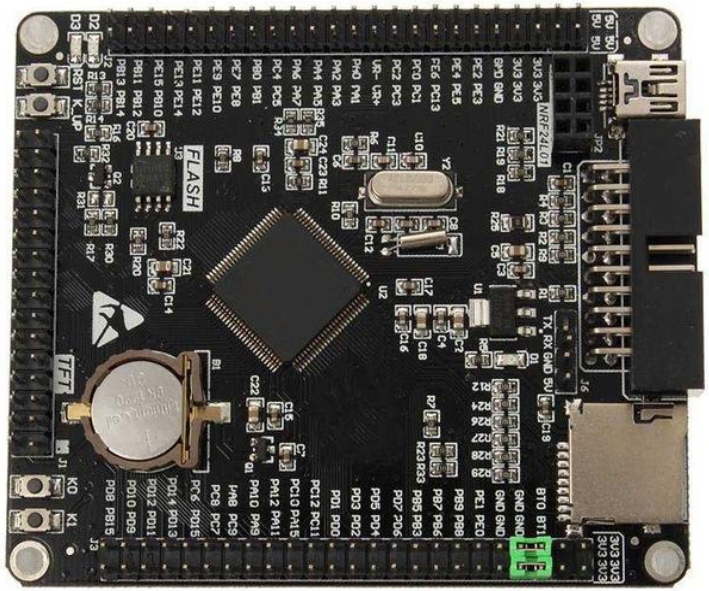


Схемы устранения дребезга контактов (Л1:258, 260)

МИКРОКОНТРОЛЛЕРЫ. ЗАГРУЗКА ПРОГРАММ





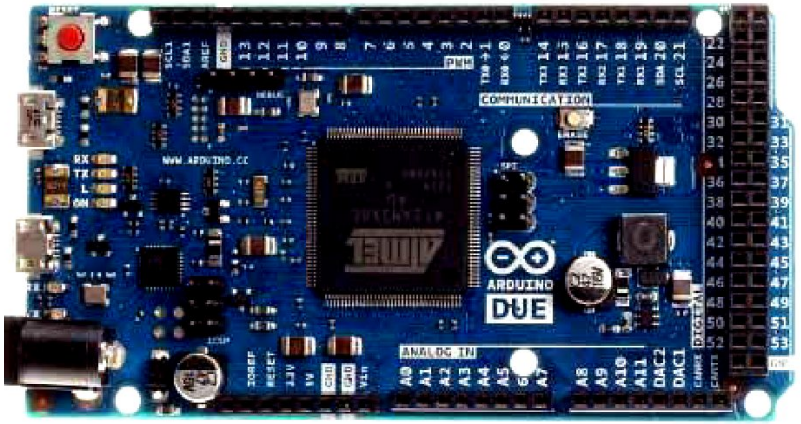


МИКРОКОНТРОЛЛЕРЫ. ЗАГРУЗКА ПРОГРАММ

В общем случае, программы загружаются в микроконтроллеры через с помощью **дополнительного устройства** (программатора).

Особенностью некоторых платформ (например Arduino) является возможность непосредственного программирования через USB-порт (без использования программатора). Эту возможность обеспечивает ***загрузчик***. Загрузчик представляет собой программу (фрагмент программного кода) записанный заводом-изготовителем в зарезервированное пространство памяти микроконтроллера. Загрузчик загружает программу пользователя в микроконтроллер по последовательному порту USART.

На некоторых отладочных платформах, например Arduino, для загрузки программ используется дополнительный контроллер Atmega 8U2/16U2, служащий интерфейсом между портом USB и портом USART основного контроллера. Некоторые основные контроллеры имеют встроенный контроллер USB. В качестве программатора можно использовать другую платформу Arduino запрограммированную на работу в режиме программатора.



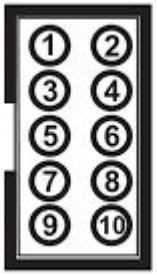
МИКРОКОНТРОЛЛЕРЫ. ЗАГРУЗКА ПРОГРАММ

Загрузчик запускается сразу, после включения питания и работает в течении некоторого времени (секунды/доли секунд). В это время загрузчик по команде из среды программирования IDE загружает программу пользователя в свободную область памяти микроконтроллера, используя последовательный интерфейс USART.

Если из среды IDE **команда не поступила**, то запускается последняя загруженная программа, находящаяся в памяти микроконтроллера.
Если из среды IDE **команда поступила**, то вспомогательный контроллер произведет сброс основного контроллера. После этого среда программирования IDE (находящаяся на компьютере) произведет загрузку программы в основной контроллер через его порт USART.

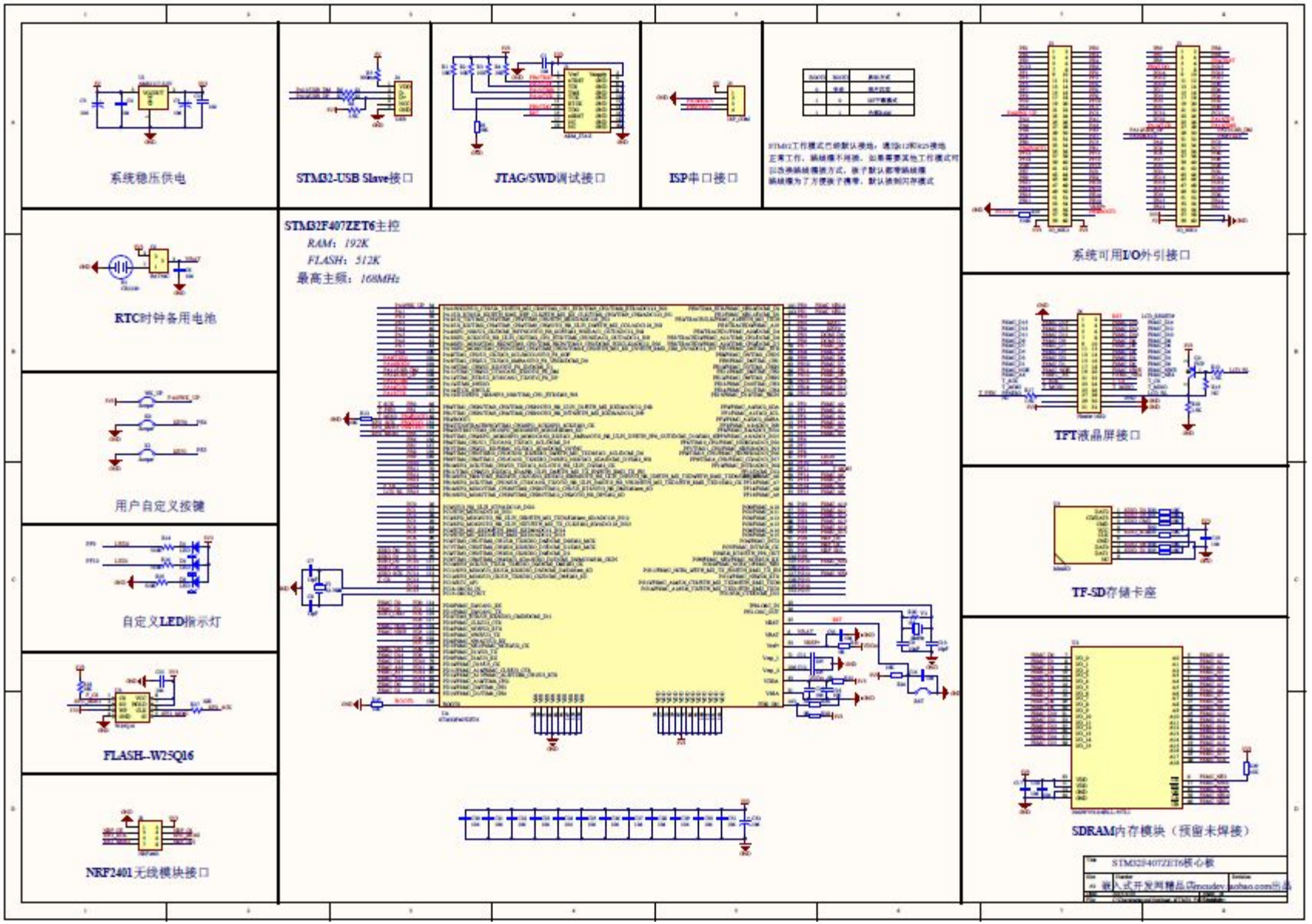
Таким образом, загрузчик обеспечивает удобство программирования микроконтроллера без использования внешнего программатора. Однако он занимает часть памяти микроконтроллера. Кроме того, начало выполнения программы пользователя будет произведено с некоторой задержкой (время на обработку запроса на программирования, сброс контроллера). Пользователь может удалить загрузчик и использовать только внешний программатор.

МИКРОКОНТРОЛЛЕРЫ.
ЗАГРУЗКА ПРОГРАММЧЕРЕЗ ST_LINK32 V2



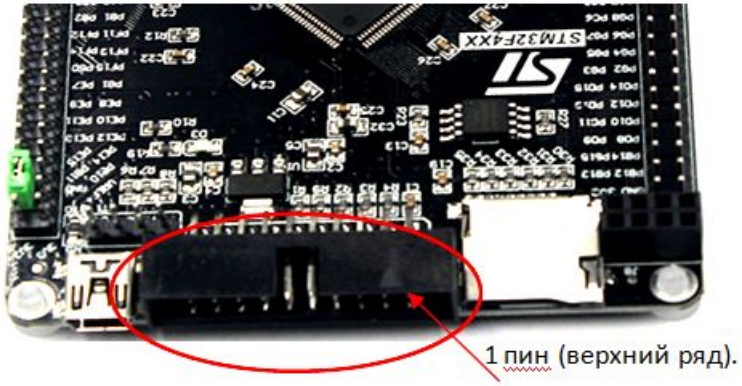
На модуле **ST-LINK V2** используются четыре пина:
2-SWDIO, 4- GND, 6 – SWCLK, 8 – 3.3V.

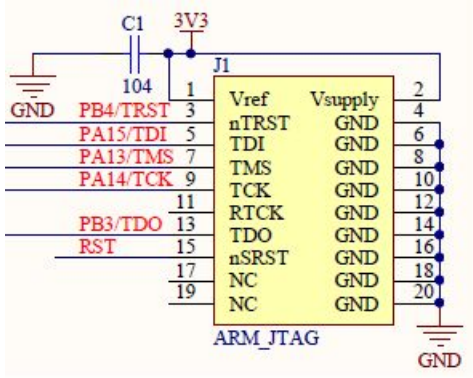
Для 32 битных МК используется используется **SWD** (правый/нижний ряд), а для 8 битных **SWIM** (левый/верхний ряд).

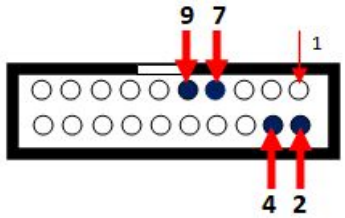

系统稳压供电
STM32-USB Slave接口
JTAG/SWD调试接口
ISP串口接口
系统可用I/O外引接口
RTC时钟备用电池
STM32F407ZET6主控
RAM：192K
FLASH：512K
最高主频：168MHz
TFT液晶屏接口
用户自定义按键
TF-SD存储卡座
自定义LED指示灯
FLASH--W25Q16
SDRAM内存模块（预留未焊接）
NRF2401无线模块接口

МИКРОКОНТРОЛЛЕРЫ.
ЗАГРУЗКА ПРОГРАММЧЕРЕЗ ST_LINK32 V2

На плате **STM32F407ZET6** используется 20-пиновый разъем **JTAG/SWD.**
Пины **2 - 3.3V, 4 –GND, 7 – TMS/SWDIO, 9 – TSK/ SWCLK.**





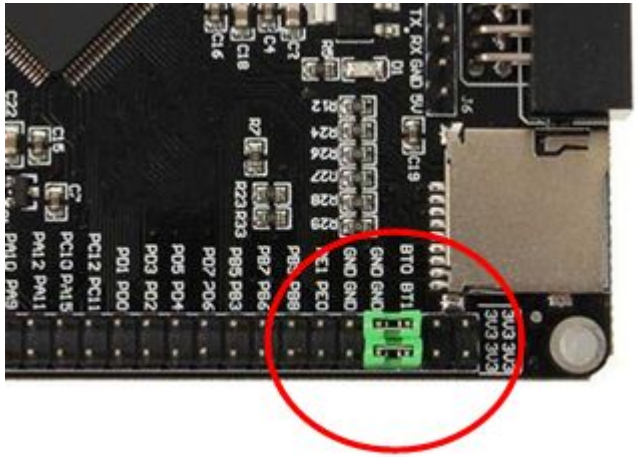


Зеленая перемычка должна стоять как на рисунке.

МИКРОКОНТРОЛЛЕРЫ.
ЗАГРУЗКА ПРОГРАММЧЕРЕЗ ST_LINK32 V2

Перемычки на МК.

На платах микроконтроллеров, как правило, всегда есть ножки **BOOT0** и **BOOT1**. Эти ножки можно замкнуть либо на землю (подключить к нулю), либо подключить к плюсу (подключить к 1) питания МК (3.3В или 5В в зависимости от конкретного МК) с помощью перемычек.



Комбинации перемычек и их установка дается в схемах к МК в «оригинальной» документации.

В «некотором/общем/частом» случае:
- **если BOOT0 подключить к земле,** то МК будет использовать первый способ загрузки, т.е через **внешний загрузчик,** например ST-Link V2**;**
- **если BOOT0 подключить к плюсу питания**, то МК будет использовать второй способ загрузки, т.е через **встроенный программный загрузчик - бутлоадер**.