

Курс по STM32

Лекция #2:

- Минимальная программа под МК.
- Карта памяти и регистры микроконтроллера.
- Сборка минимальной программы.
- Система тактирования и сброса.
- Выдача ДЗ-1.

11.02.2023 / 13.02.2023

Минимальная программа под МК



Мигание диодом PC8

Локализуем на плате диод:



Из документации на отладочную плату ([docs/stm32f0discovery.pdf](https://www.st.com/en/development-tools/docs/stm32f0discovery.pdf)):

- User LD3: Green user LED connected to the I/O PC9 of the STM32F051R8T6.
- User LD4: Blue user LED connected to the I/O PC8 of the STM32F051R8T6.

Нужен пин PC8: GPIO порт C, пин 8.

Адреса регистров МК

Запись в регистр GPIOC_ODR – вывод напряжений на пины порта C.

8.4.6

GPIO port output data register (GPIOx_ODR) (x = A..F)

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **ODRy**: Port output data bit (y = 0..15)

These bits can be read and written by software.

Адреса регистров МК

Расположение регистров – в даташите на МК ([docs/stm32f0xx_rm.pdf](https://www.st.com/en/microcontrollers/stm32f0xx-rm.pdf))

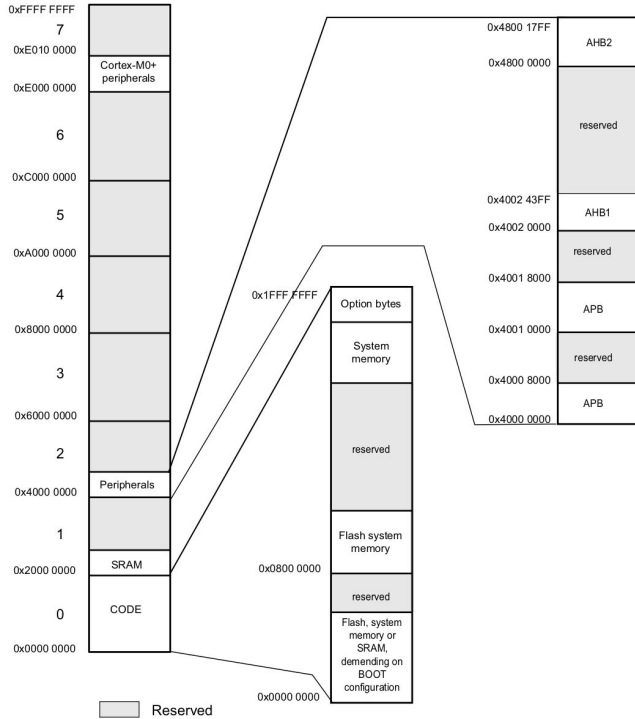
Table 1. STM32F0xx peripheral register boundary addresses

Bus	Boundary address	Size	Peripheral	Peripheral register map
	0xE000 0000 - 0xE00F FFFF	1MB	Cortex [®] -M0 internal peripherals	
	0x4800 1800 - 0x5FFF FFFF	~384 MB	Reserved	
AHB2	0x4800 1400 - 0x4800 17FF	1KB	GPIOF	<i>Section 8.4.12 on page 163</i>
	0x4800 1000 - 0x4800 13FF	1KB	GPIOE	<i>Section 8.4.12 on page 163</i>
	0x4800 0C00 - 0x4800 0FFF	1KB	GPIOD	<i>Section 8.4.12 on page 163</i>
	0x4800 0800 - 0x4800 0BFF	1KB	GPIOC	<i>Section 8.4.12 on page 163</i>

Теперь известен адрес регистра GPIOC_ODR: 0x48000800 + 0x14

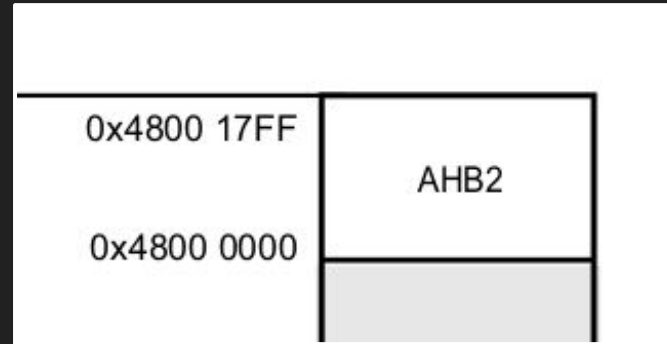
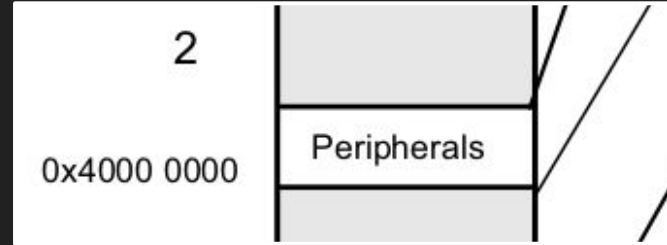
Карта памяти МК

Figure 2. Memory map



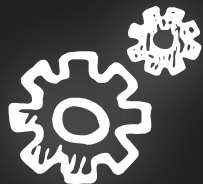
MS33181V4

Искомый адрес: 0x48000814



Тип “памяти”: периферия шины ANB2

Сборка минимальной программы



Сборка минимальной программы (01_blinkled)

Сборка программы и прошивка – через Makefile:

```
~/path/to/stm32f051_rewind/labs/01_blinkled>
> make
arm-none-eabi-gcc -march=armv6-m -mcpu=cortex-m0 -o build/entry.o -c entry.S
arm-none-eabi-gcc -march=armv6-m -mcpu=cortex-m0 -o build/blinkled.o -c blinkled.c
arm-none-eabi-gcc -nostdlib -march=armv6-m -mcpu=cortex-m0 -Wl,-T,entry.lds build/entry.o build/blinkled.o -o build/blinkled.elf
arm-none-eabi-objcopy -O binary build/blinkled.elf build/blinkled.bin
> make flash
st-flash write build/blinkled.bin 0x08000000
st-flash 1.6.0
2023-02-11T12:07:32 INFO common.c: Loading device parameters....
2023-02-11T12:07:32 INFO common.c: Device connected is: F0 device, id 0x20006440
2023-02-11T12:07:32 INFO common.c: SRAM size: 0x2000 bytes (8 KiB), Flash: 0x10000 bytes (64 KiB) in pages of 1024 bytes
2023-02-11T12:07:32 INFO common.c: Attempting to write 348 (0x15c) bytes to stm32 address: 134217728 (0x8000000)
Flash page at addr: 0x08000000 erased
2023-02-11T12:07:32 INFO common.c: Finished erasing 1 pages of 1024 (0x400) bytes
2023-02-11T12:07:32 INFO common.c: Starting Flash write for VL/F0/F3/F1_XL core id
2023-02-11T12:07:32 INFO flash_loader.c: Successfully loaded flash loader in sram
1/1 pages written
2023-02-11T12:07:32 INFO common.c: Starting verification of write complete
2023-02-11T12:07:32 INFO common.c: Flash written and verified! jolly good!
```

Но что под капотом?

Сборка минимальной программы (01_blinkled)

```
~/path/to/stm32f051_rewind/labs/01_blinkled>
```

```
> tree
```

```
.
├── blinkled.c          <<<< Код программы (main и всё остальное)
├── build
│   ├── blinkled.bin    <<<< Бинарный образ прошивки (ложится во Flash)
│   ├── blinkled.elf    <<<< Прошивка + метаинформация (символы, секции, ...)
│   ├── blinkled.o      <<<< Объектный файл blinkled.c
│   └── entry.o         <<<< Объектный файл entry.S
├── entry.lds           <<<< Linker-скрипт с настройкой линковки
├── entry.S             <<<< Стартовый код МК (asm)
├── Makefile            <<<< Makefile с командами сборки
└── README.md
```

Сборка минимальной программы (01_blinkled)

Заглянем внутрь бинарных файлов:

```
~/path/to/stm32f051_rewind/labs/01_blinkled>  
> arm-none-eabi-objdump -x build/blinkled.o  
<output>  
> arm-none-eabi-objdump -x build/entry.o  
<output>  
> arm-none-eabi-objdump -x build/blinkled.elf  
<output>  
> <your disassembler> build/blinkled.elf  
<output>  
> arm-none-eabi-objdump -D build/blinkled.elf # Без дизассемблера  
<output>
```

Сборка минимальной программы

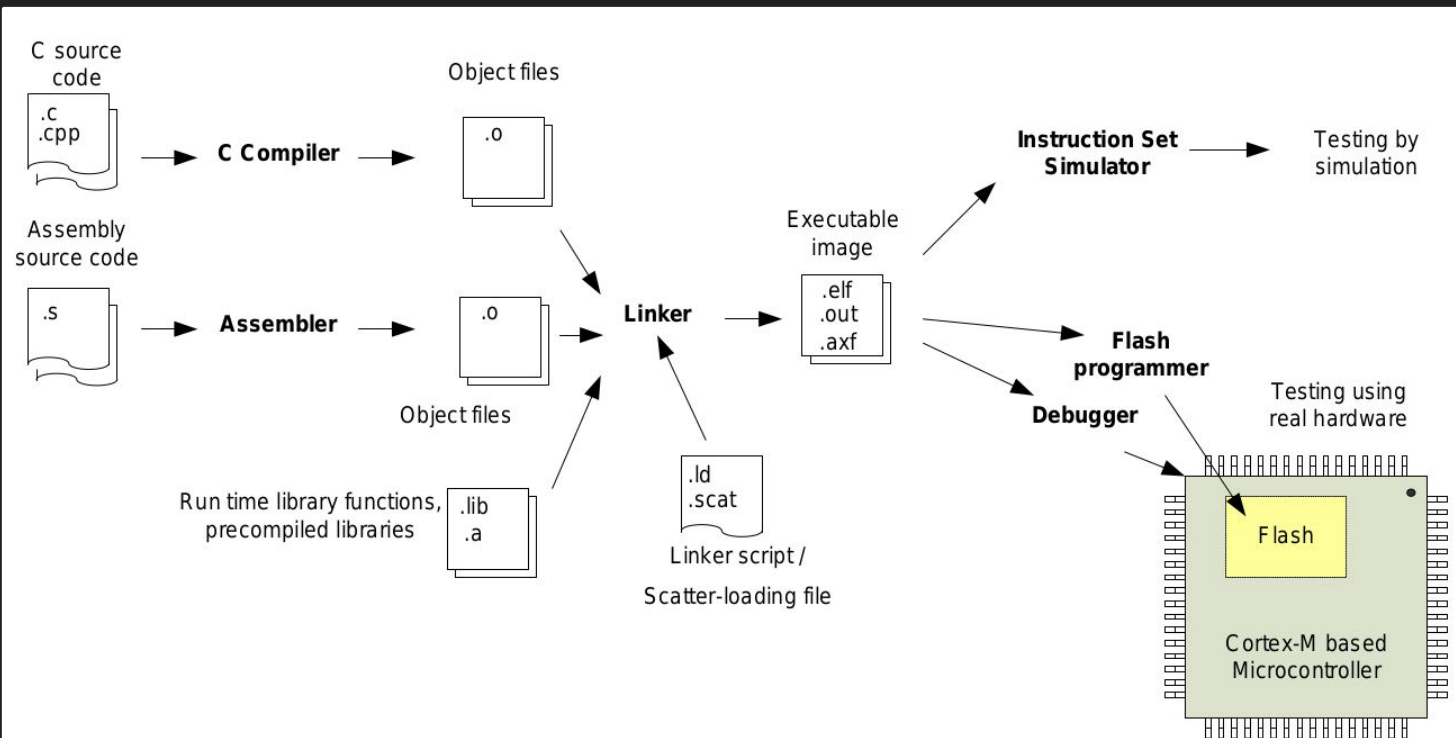


Figure 3.13
Typical program generation flow.

Прошивка платы

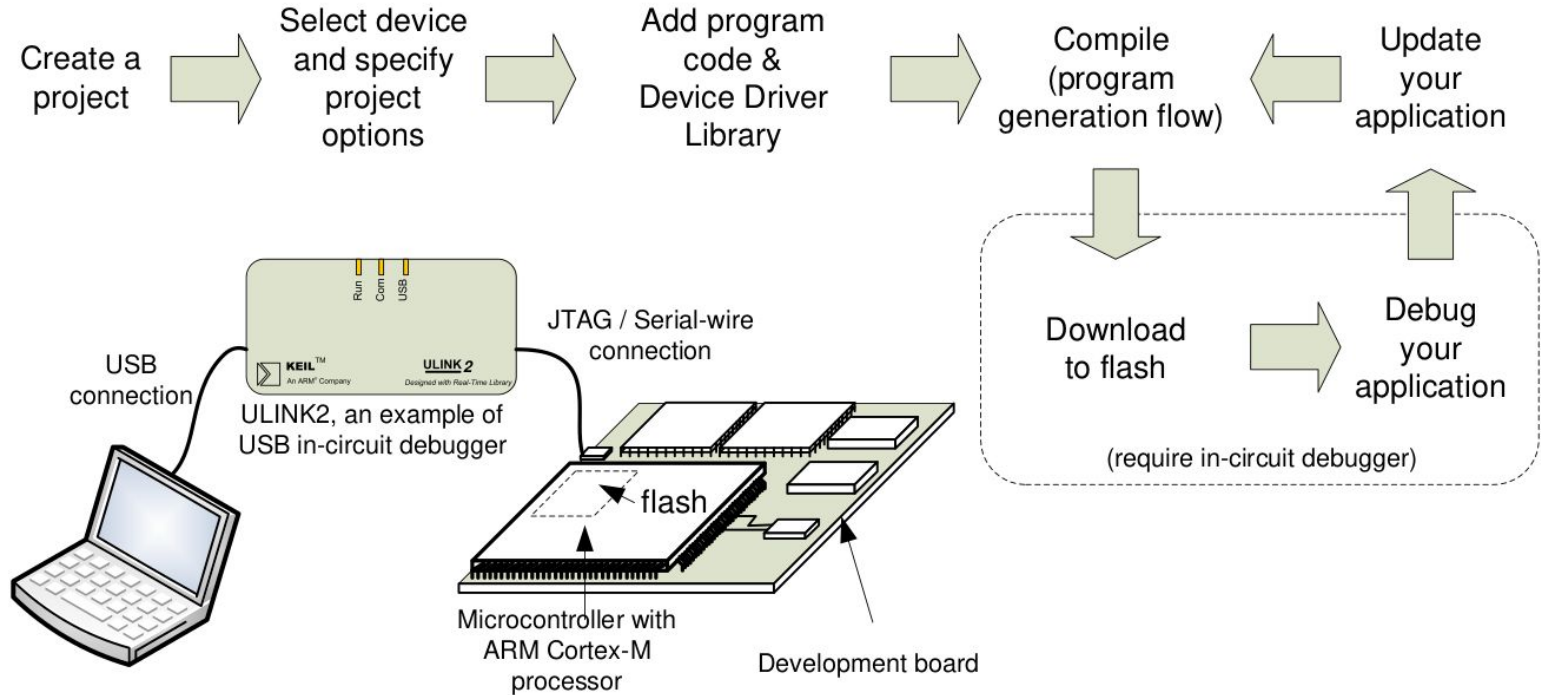


Figure 3.14
An example of development flow.

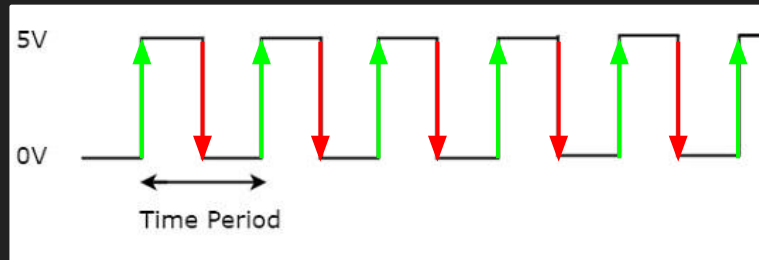
Система тактирования микроконтроллера



Тактовый сигнал

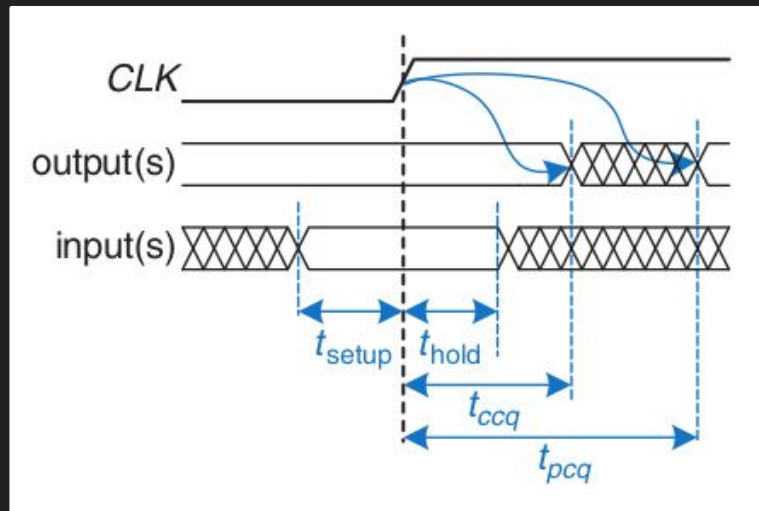
Фронты тактового сигнала:

- Передний фронт (rising edge)
- Задний фронт (falling edge)



Функции:

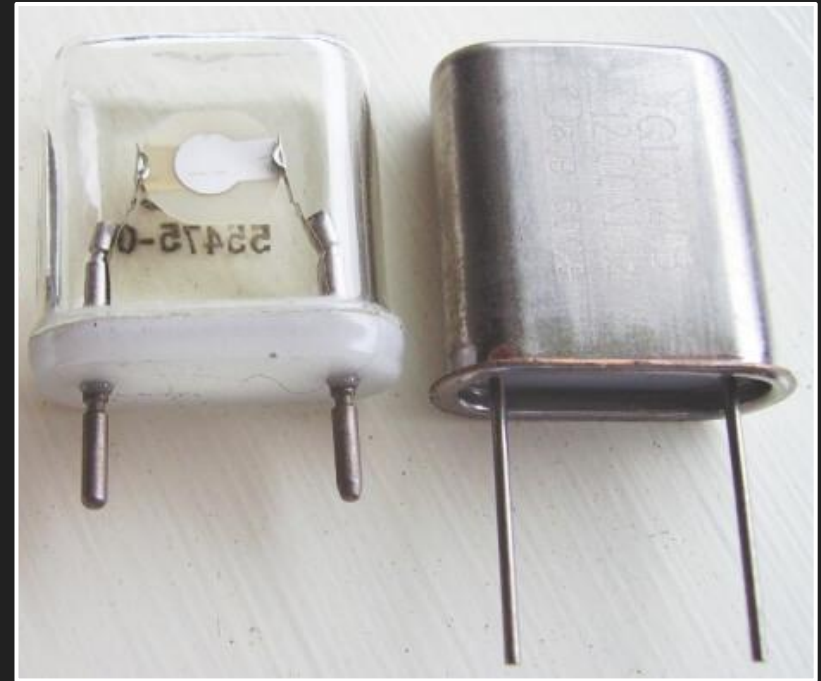
- Синхронизация элементов логики
- Период тактового сигнала –
– единица времени



Источники тактирования

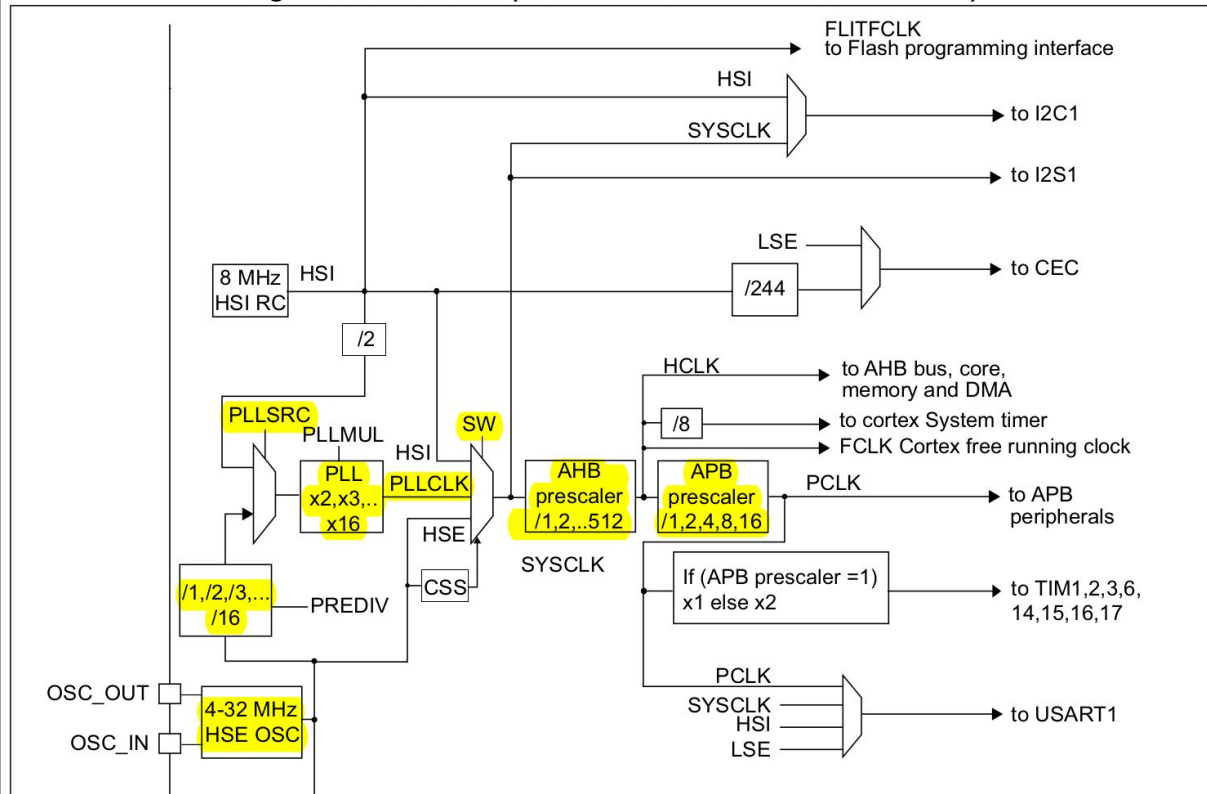
Источники тактового сигнала
на STM32F051:

- Кварцевый резонатор
(на STM32F051: HSE, 8 МГц)
- Генератор на RC-цепочке
(на STM32F051: HSI, 8 МГц)
- Синтезатор частоты
на основе ФАПЧ
(на STM32F051: PLL)



Дерево тактирования МК

Figure 10. Clock tree (STM32F03x and STM32F05x devices)



Настройка тактирования МК

Порядок подачи тактирования:

- 1) Включение осциллятора HSE
- 2) Предделитель PLL
- 3) Выбор входа PLL
- 4) Коэффициент умножения PLL
- 5) Включение PLL
- 6) Предделитель шины AHB
- 7) Переключение входа шины AHB
- 8) Предделитель шины APB
- 9*) Подключение тактирования к GPIOC

```
void board_clocking_init()
{
    // (1) Clock HSE and wait for oscillations to setup.
    *REG_RCC_CR = 0x00010000U;
    while ((*REG_RCC_CR & 0x00020000U) != 0x00020000U);

    // (2) Configure PLL:
    // PREDIV output: HSE/2 = 4 MHz
    *REG_RCC_CFGR2 |= 1U;

    // (3) Select PREDIV output as PLL input (4 MHz):
    *REG_RCC_CFGR |= 0x00010000U;

    // (4) Set PLLMUL to 12:
    // SYSCLK frequency = 48 MHz
    *REG_RCC_CFGR |= (12U-1U) << 18U;

    // (5) Enable PLL:
    *REG_RCC_CR |= 0x01000000U;
    while ((*REG_RCC_CR & 0x02000000U) != 0x02000000U);

    // (6) Configure AHB frequency to 48 MHz:
    *REG_RCC_CFGR |= 0b000U << 4U;

    // (7) Select PLL as SYSCLK source:
    *REG_RCC_CFGR |= 0b10U;
    while ((*REG_RCC_CFGR & 0xcU) != 0x8U);

    // (8) Set APB frequency to 24 MHz
    *REG_RCC_CFGR |= 0b001U << 8U;
}
```

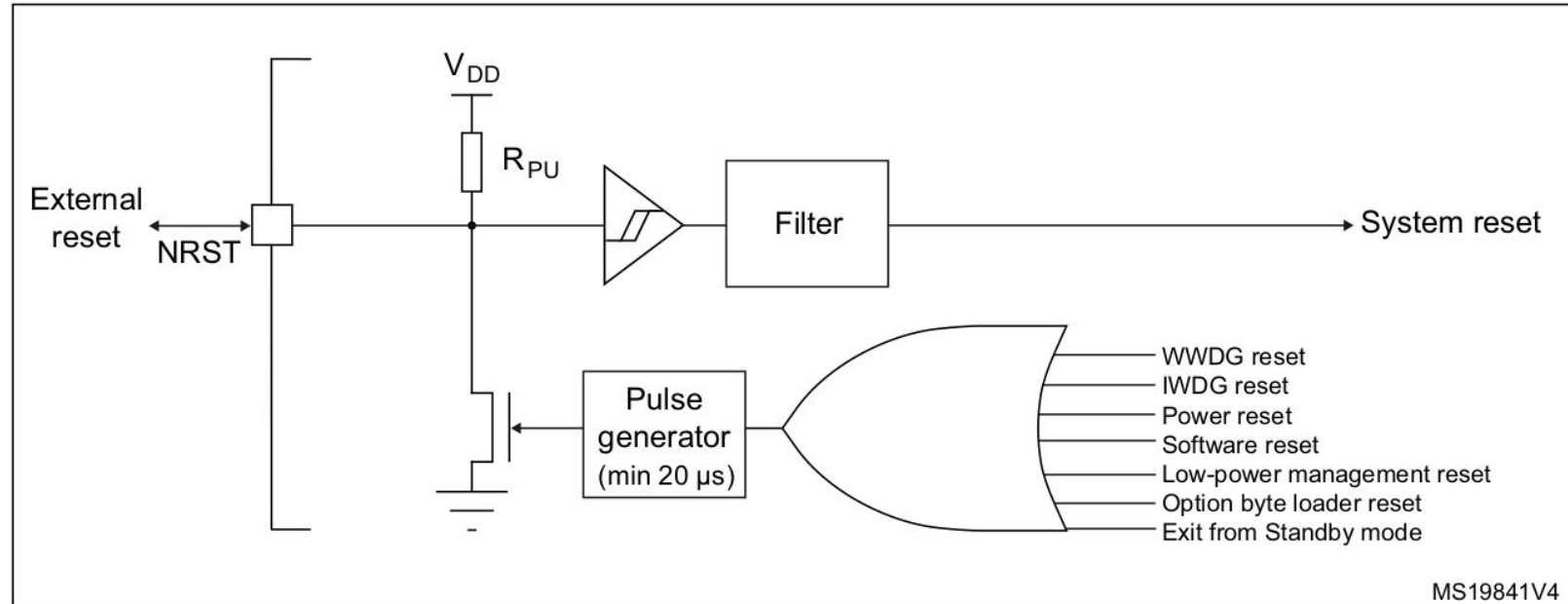
Система сброса микроконтроллера



Сброс микроконтроллера

Сброс МК – переход в начальное строго определённое состояние.

Figure 9. Simplified diagram of the reset circuit



MS19841V4

Лаба #1: 01_ledblink



Требования к ДЗ#1

- [] Код запускается в эмуляторе.
- [] Компиляция минимального примера проделана вручную.
- [] Код примера -- отрефакторен:
 - [] Используются регистры только по их именам.
 - [] Используются биты регистров только по их именам.
- [] Синий и зелёный диоды мигают попеременно.

```
GPIO_ODR_BitSet(*GPIOC_ODR, 8U);
```

```
// GPIO_ODR bit:  
#define GPIO_ODR_BitSet(reg, bit) (reg) = (reg) | (1U << (bit))
```

Спасибо за внимание!