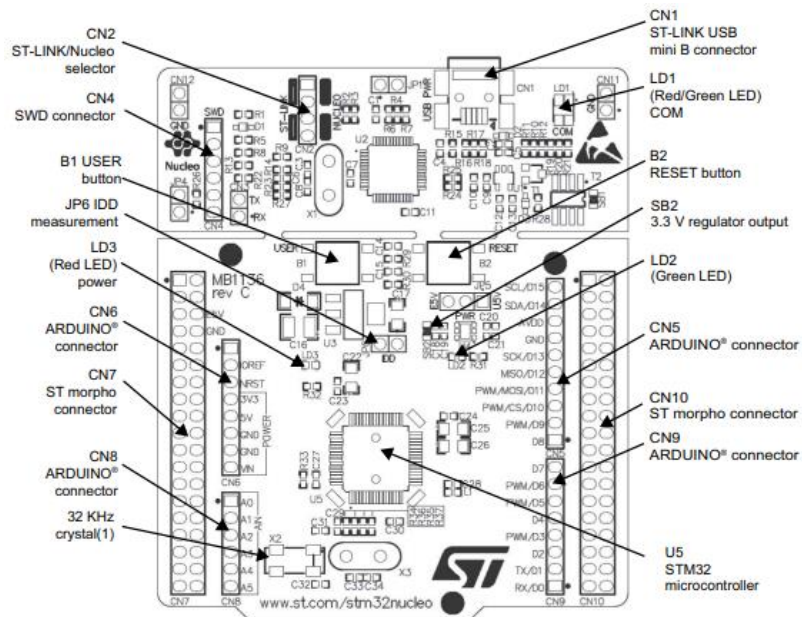


# Анализ периферии и схемы тактирования отладочного модуля на базе STM32

*Анализ библиотек STM32*

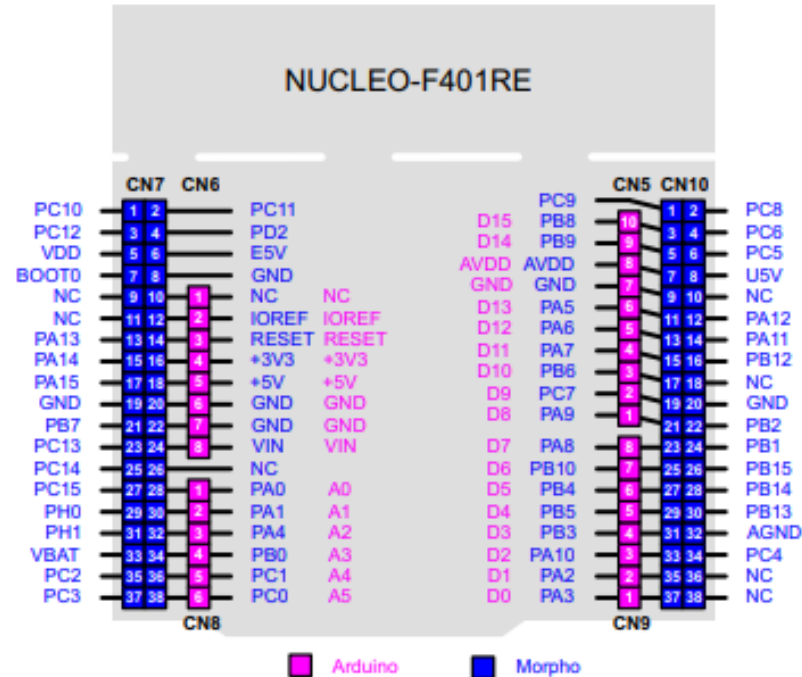
# Отладочный набор Nucleo-F401RE

## Features

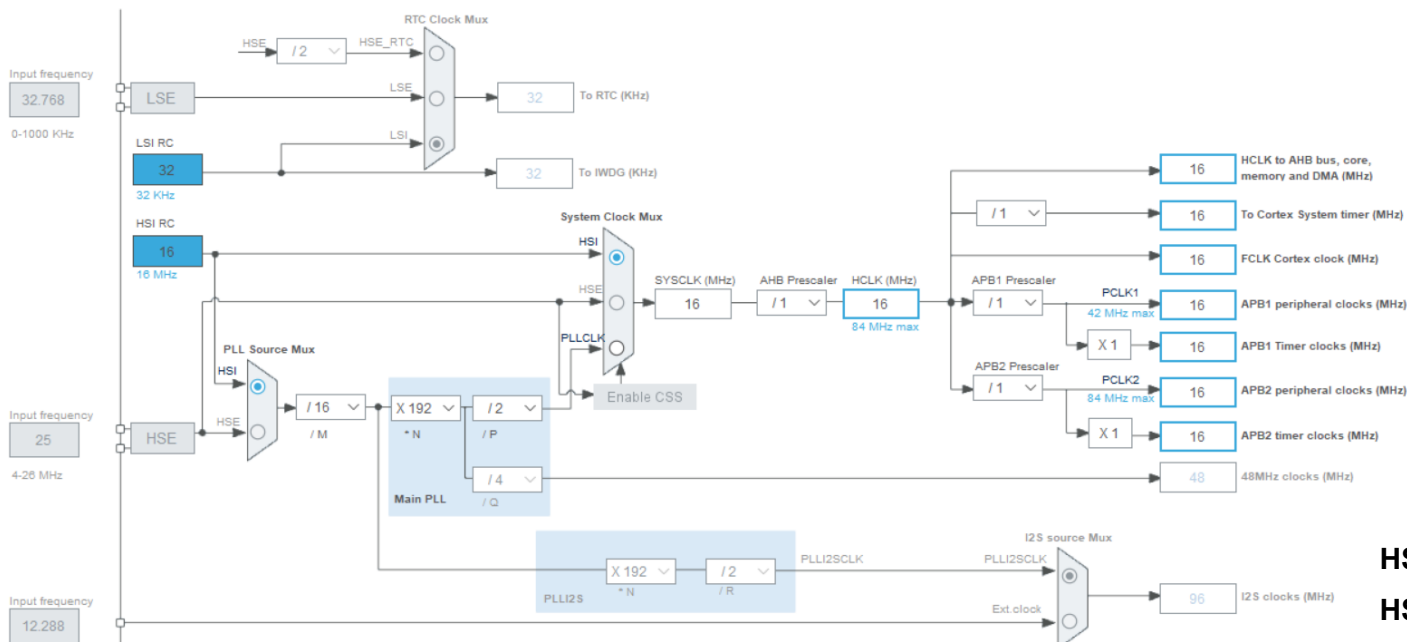


- Ядро: Cortex-M4
- Рабочая частота: 84 МГц
- Разрядность: 32 бита
- Набор машинных инструкций: ARMv7-M
- Производительность: 105 DMIPS
- Объем SRAM: 128 кБ
- Объем Flash: 512 кБ
- АЦП: 12 бит, 16 каналов
- Интерфейсы: 3×I2C, 3×USART, 4×SPI, SDIO, USB 2.0 Host/Device/OTG
- Количество таймеров/счетчиков: 11
- Питание: USB VBUS или внешнее питание (3.3V, 5V, 7 — 12V)
- Возможность подключения плат расширения
- Поддержка через USB виртуального COM-порта, внешнего накопителя и отладочного порта

# Отладочный набор Nucleo-F401RE



# Схема тактирования



**HSI** - high-speed internal RC oscillator

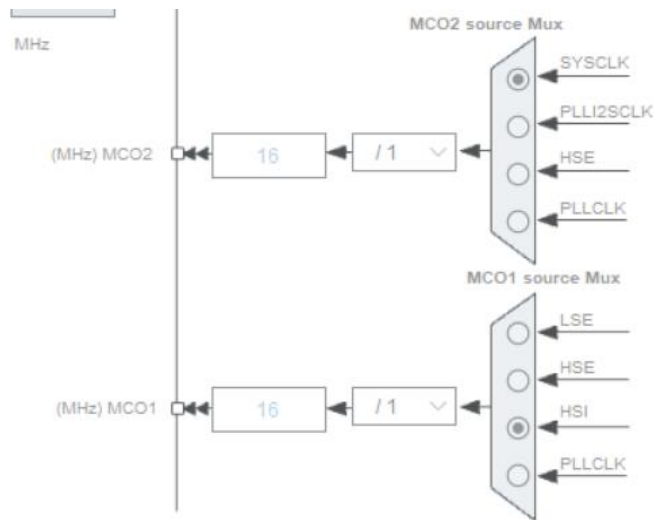
**HSE** - high-speed external oscillator

**LSE** - low-speed external

**LSI** - low-speed internal RC

**PLL** - phase-locked loop

# Схема тактирования



## RCC clock control register (RCC\_CR)

Bits 15:8 **HSICAL[7:0]**: Internal high-speed clock calibration

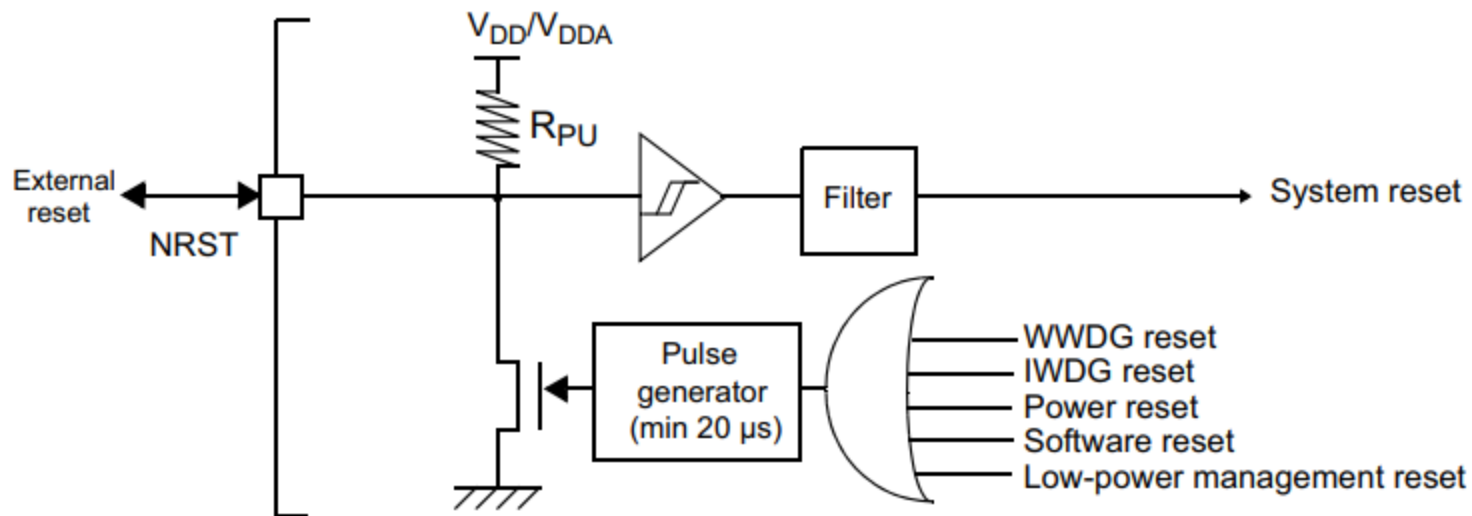
These bits are initialized automatically at startup.

Bits 7:3 **HSITRIM[4:0]**: Internal high-speed clock trimming

These bits provide an additional user-programmable trimming value that is added to the HSICAL[7:0] bits. It can be programmed to adjust to variations in voltage and temperature that influence the frequency of the internal HSI RC.

**MCO** - microcontroller clock output

# Система сброса



# Латентность памяти

Wait states (WS) (LATENCY)	HCLK (MHz)			
	Voltage range 2.7 V - 3.6 V	Voltage range 2.4 V - 2.7 V	Voltage range 2.1 V - 2.4 V	Voltage range 1.71 V - 2.1 V
0 WS (1 CPU cycle)	$0 < \text{HCLK} \leq 30$	$0 < \text{HCLK} \leq 24$	$0 < \text{HCLK} \leq 18$	$0 < \text{HCLK} \leq 16$
1 WS (2 CPU cycles)	$30 < \text{HCLK} \leq 60$	$24 < \text{HCLK} \leq 48$	$18 < \text{HCLK} \leq 36$	$16 < \text{HCLK} \leq 32$
2 WS (3 CPU cycles)	$60 < \text{HCLK} \leq 84$	$48 < \text{HCLK} \leq 72$	$36 < \text{HCLK} \leq 54$	$32 < \text{HCLK} \leq 48$
3 WS (4 CPU cycles)		$72 < \text{HCLK} \leq 84$	$54 < \text{HCLK} \leq 72$	$48 < \text{HCLK} \leq 64$
4 WS (5 CPU cycles)	-	-	$72 < \text{HCLK} \leq 84$	$64 < \text{HCLK} \leq 80$
5 WS (6 CPU cycles)	-	-	-	$80 < \text{HCLK} \leq 84$

## Flash access control register (FLASH\_ACR)

Bits 3:0 **LATENCY:** Latency

These bits represent the ratio of the CPU clock period to the Flash memory access time.

0000: Zero wait state

0001: One wait state

0010: Two wait states

-

-

-

1110: Fourteen wait states

1111: Fifteen wait states

**WS** – wait state

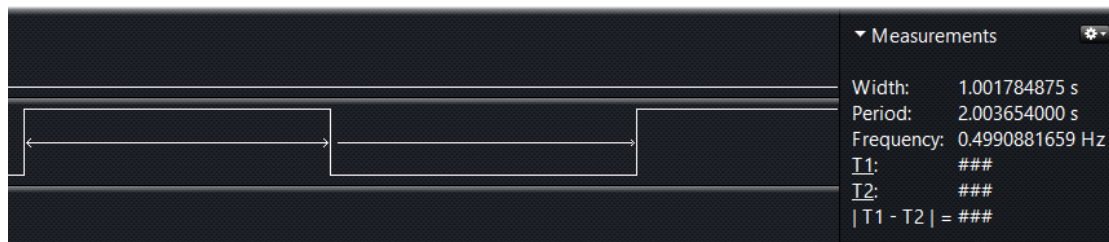
# Библиотеки STM32



- CMSIS
  - SPL
  - LL
  - HAL
- 
- Libopencm3





# Программа Blinking\_LED





# Объем занимаемой памяти



- CMSIS

Region	Start address	End address	Size	Free	Used	Usage (%)
 RAM	0x20000000	0x20017fff	96 KB	94,47 KB	1,53 KB	1.60%
 FLASH	0x08000000	0x0807ffff	512 KB	510,94 KB	1,06 KB	0.21%

- LL

Region	Start address	End address	Size	Free	Used	Usage (%)
 RAM	0x20000000	0x20017fff	96 KB	94,47 KB	1,53 KB	1.60%
 FLASH	0x08000000	0x0807ffff	512 KB	509,06 KB	2,94 KB	0.57%

- HAL

Region	Start address	End address	Size	Free	Used	Usage (%)
 RAM	0x20000000	0x20017fff	96 KB	94,45 KB	1,55 KB	1.61%
 FLASH	0x08000000	0x0807ffff	512 KB	506,26 KB	5,74 KB	1.12%

# После оптимизации

- CMSIS

Region	Usage (%)
RAM	1.60%
FLASH	0.21%

Optimize for size



Usage (%)
1.60%
0.16%

$$k_{сж} = 23\%$$

- LL

Region	Usage (%)
RAM	1.60%
FLASH	0.57%

Optimize for size



Usage (%)
1.60%
0.28%

$$k_{сж} = 51\%$$

- HAL

Region	Usage (%)
RAM	1.61%
FLASH	1.12%

Optimize for size



Usage (%)
1.61%
0.75%

$$k_{сж} = 33\%$$

# Проблемы оптимизации

```
uint32_t SysTick_Count = 0;

void SysTick_Handler(void)
{
    if (SysTick_Count > 0) SysTick_Count--;
}

void delay_ms(int ms)
{
    SysTick->LOAD = 16000-1;
    SysTick_Count = ms;
    while (SysTick_Count){}
}
```

При оптимизации компилятор предполагает, что значение SysTick\_Count не изменяется между двумя его применениями и в SysTick\_Handler переменная уже не отражает свое актуальное состояние

## Решение

```
volatile uint32_t SysTick_Count = 0;

__attribute__((optimize(0))) void delay_ms(int ms)
```

# Скорость выполнения операции

- CMSIS

start	uint32_t	3
end	uint32_t	12
cycles	uint32_t	9

```
GPIOA->BSRR = GPIO_BSRR_BS5;
```

- LL

start	uint32_t	3
end	uint32_t	48
cycles	uint32_t	45

```
LL_GPIO_TogglePin(GPIOA, LL_GPIO_PIN_5);
```

- HAL

start	uint32_t	3
end	uint32_t	49
cycles	uint32_t	46

```
HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
```

# CMSIS

## Преимущества

- Простой интерфейс
- Удобный доступ к регистрам
- Малый объем используемой памяти
- Производительность приложения

## Недостатки

- Трудность переносимости в другой проект
- Требуется более глубокое понимание

# LL

## Преимущества

- Низкоуровневые API на уровне регистров
- Добавление проверок `assert_param`
- Ускорение программирования из-за обертки в виде структур и методов с описанием параметров

## Недостатки

- Занимает больше места

# HAL

## Преимущества

- Выше уровень переносимости
- Выше скорость написания программы
- Компактный код программы
- Читабельность
- Низкий порог вхождения
- Функции Callback

## Недостатки

- Наличие ошибок
- Сложность настройки с нуля
- Перегрузки в коде
- Большой объем занимаемой памяти
- Снижена производительность приложения



# Репозиторий

<https://github.com/Vernicovskiy/STM32-library-analysis.git>