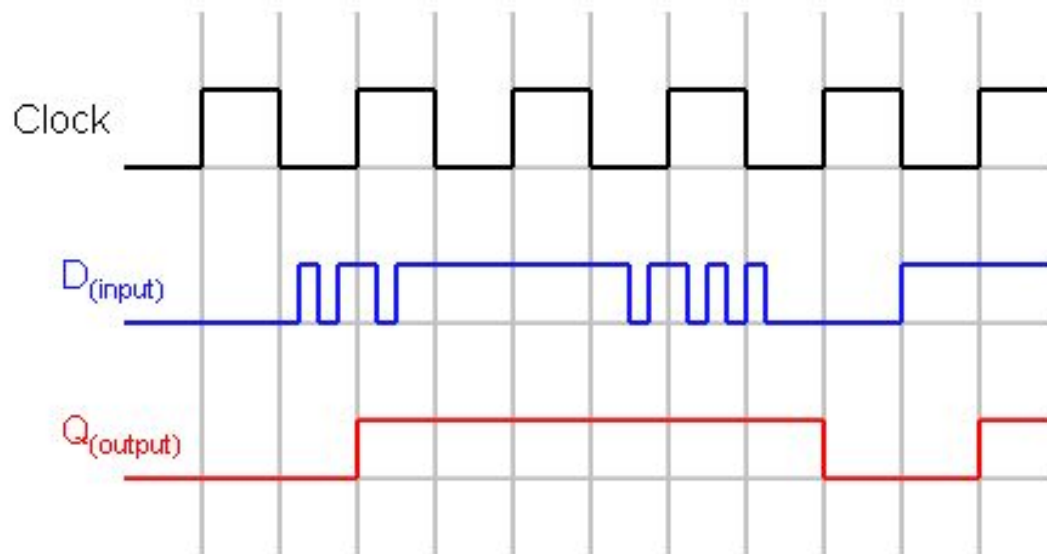


Программирование микроконтроллеров STM32

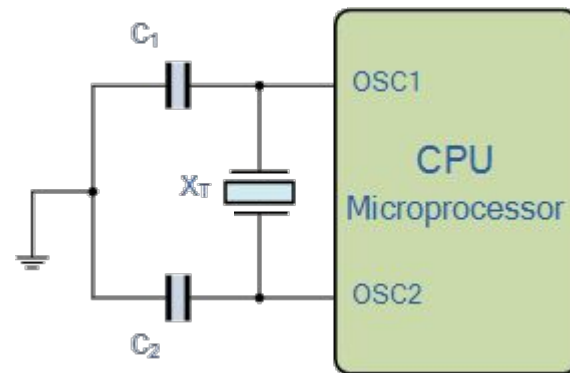
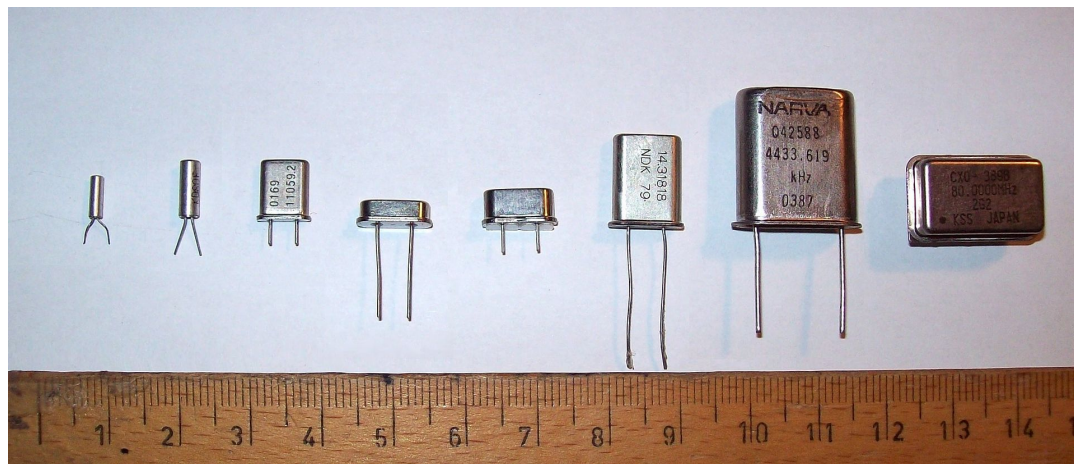
Система тактирования и сброса

Тактовый сигнал

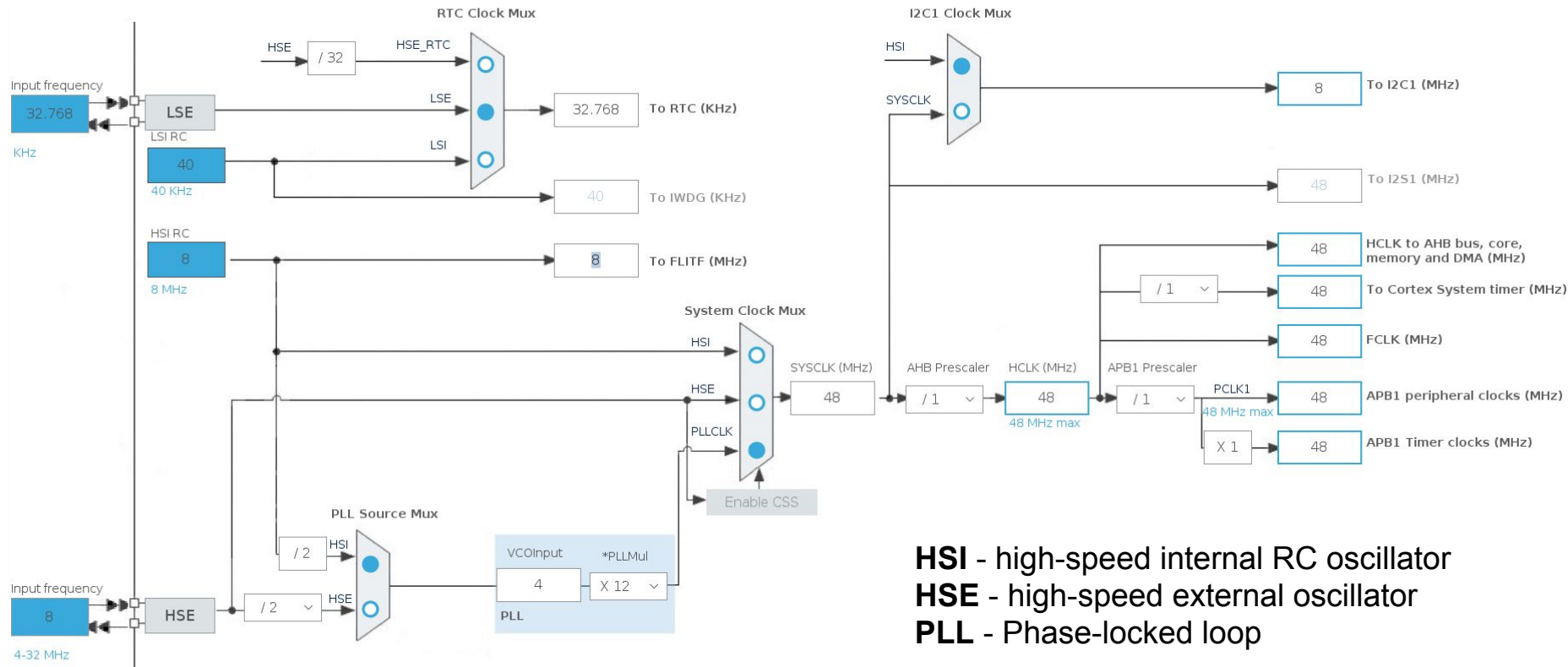
- Сигнал для согласования логических операций одной и более логических схем



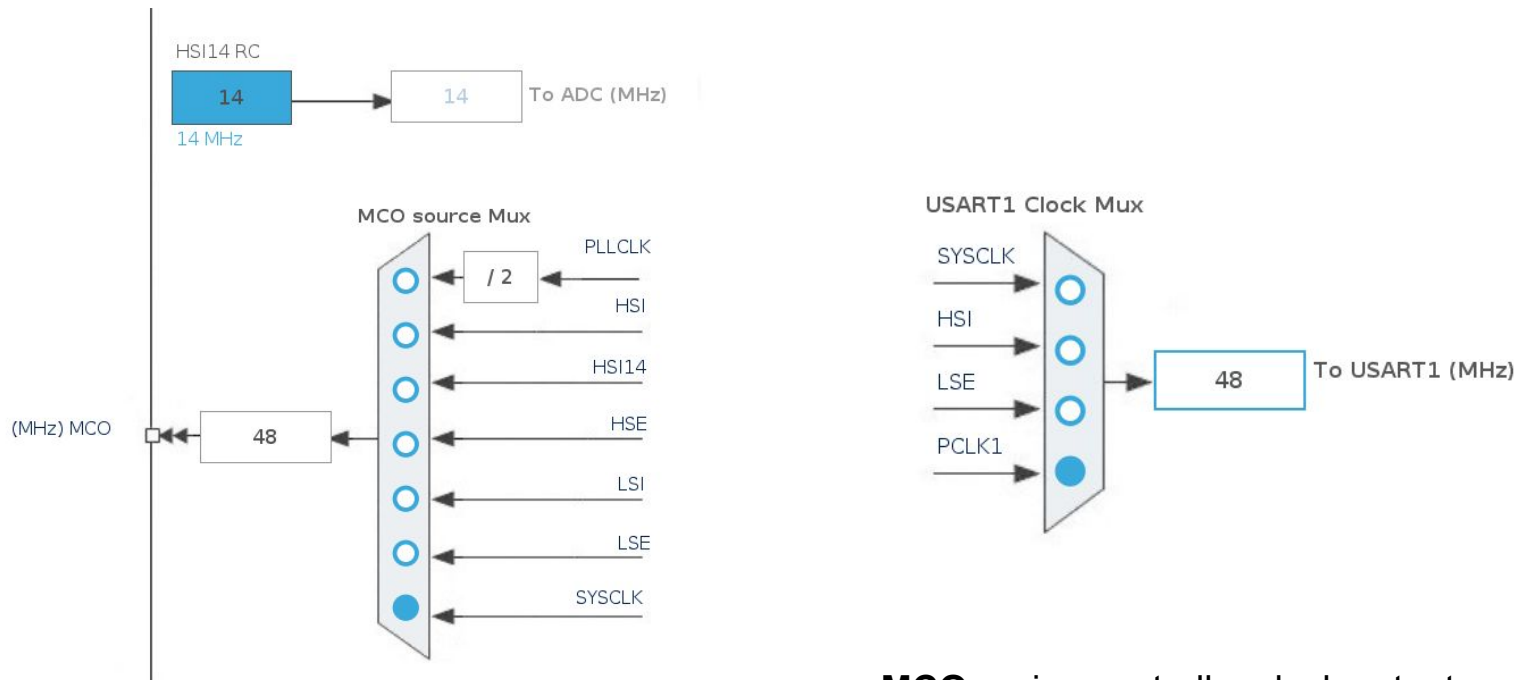
Источники тактирования



Основная схема тактирования в STM32



Основная схема тактирования в STM32



Пример инициализации. 03_system_clock

```
LL_FLASH_SetLatency(LL_FLASH_LATENCY_1);
LL_RCC_HSI_Enable();
while (LL_RCC_HSI_IsReady() != 1);

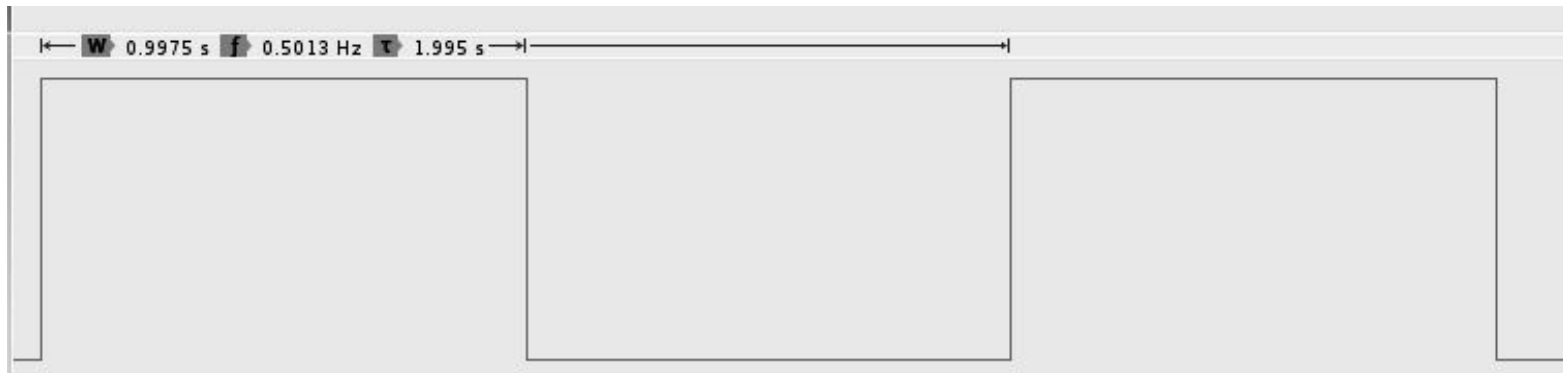
LL_RCC_PLL_ConfigDomain_SYS(LL_RCC_PLLSOURCE_HSI_DIV_2,
                             LL_RCC_PLL_MUL_12);

LL_RCC_PLL_Enable();
while (LL_RCC_PLL_IsReady() != 1);
LL_RCC_SetAHBPrescaler(LL_RCC_SYSCLK_DIV_1);
LL_RCC_SetSysClkSource(LL_RCC_SYS_CLKSOURCE_PLL);
while (LL_RCC_GetSysClkSource() != LL_RCC_SYS_CLKSOURCE_STATUS_PLL);
LL_RCC_SetAPB1Prescaler(LL_RCC_APB1_DIV_1);
```

Латентность памяти

```
#define FLASH_0LAT_DELAY0LAT
```

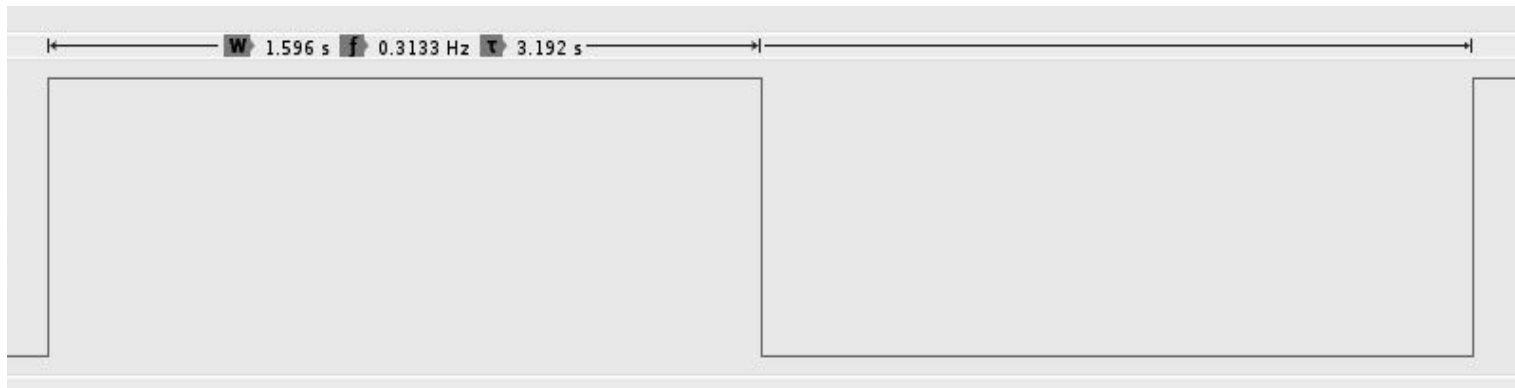
PC8 (Blue led)



Латентность памяти

```
#define FLASH_1LAT_DELAY0LAT
```

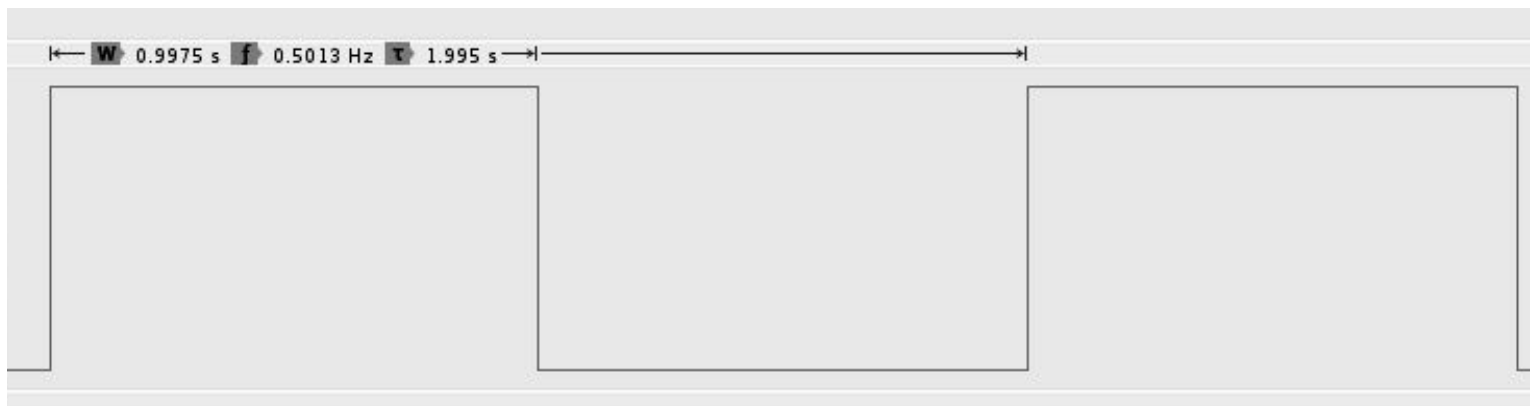
PC8 (Blue led)



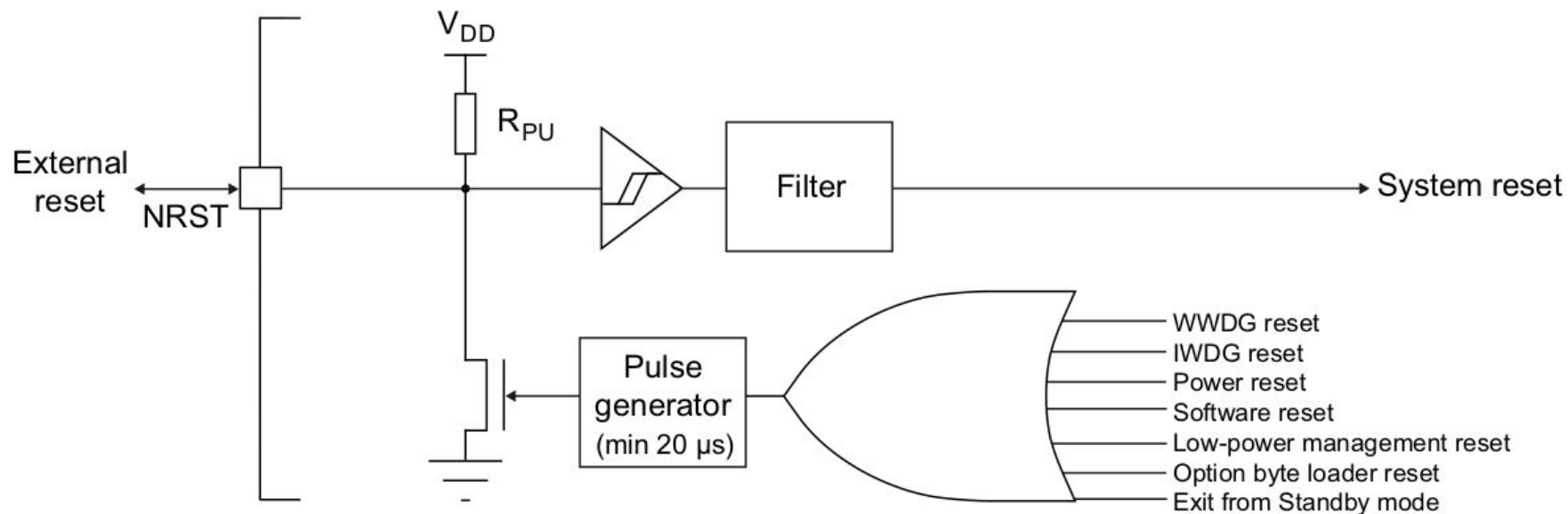
Латентность памяти

```
#define FLASH_1LAT_DELAY1LAT
```

PC8 (Blue led)



Система сброса



Система сброса

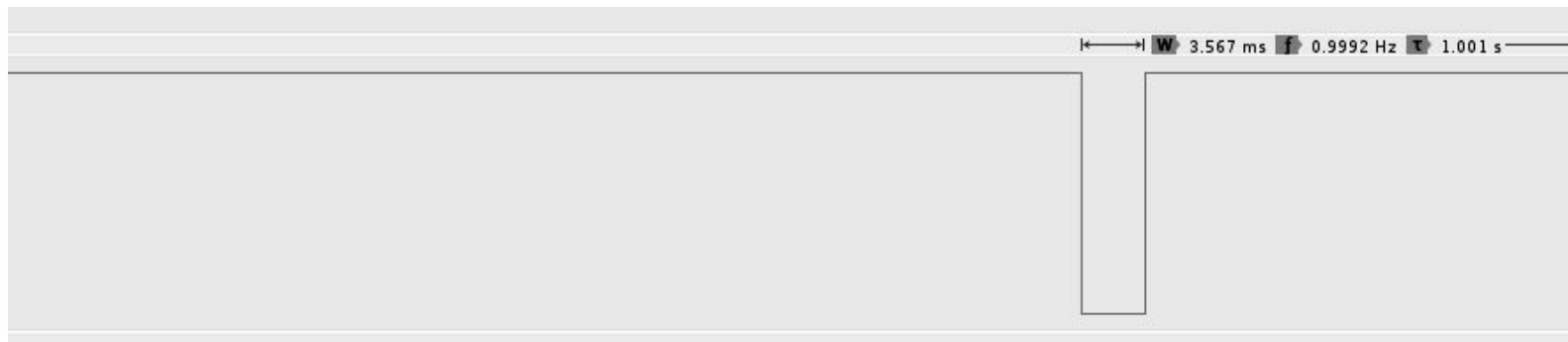
- Сброс всех регистров кроме RCC_CSR and регистров модуля реального времени
- Очистка оперативной памяти (стандартный обработчик сброса в startup_stm32f051x8.s)

Система сброса. 03_reset

```
if (LL_RCC_IsActiveFlag_SFTRST()) {  
    LL_GPIO_SetOutputPin(GPIOC, LL_GPIO_PIN_9);  
    LL_RCC_ClearResetFlags();  
}  
if (LL_RCC_IsActiveFlag_PINRST())  
    LL_GPIO_TogglePin(GPIOC, LL_GPIO_PIN_8);  
  
while (1) {  
    delay();  
    NVIC_SystemReset();  
}
```

Система сброса

PC9 (Green led)



Документация и CubeMX

- cortex_m0_ll.pdf - описание библиотеки LL
- STM32CubeMX - удобное средство конфигурации с визуализацией



Репозиторий

https://github.com/edosedgar/stm32f0_ARM