

Программирование микроконтроллеров STM32

Модуль часов реального времени

Зачем нужен RTC?

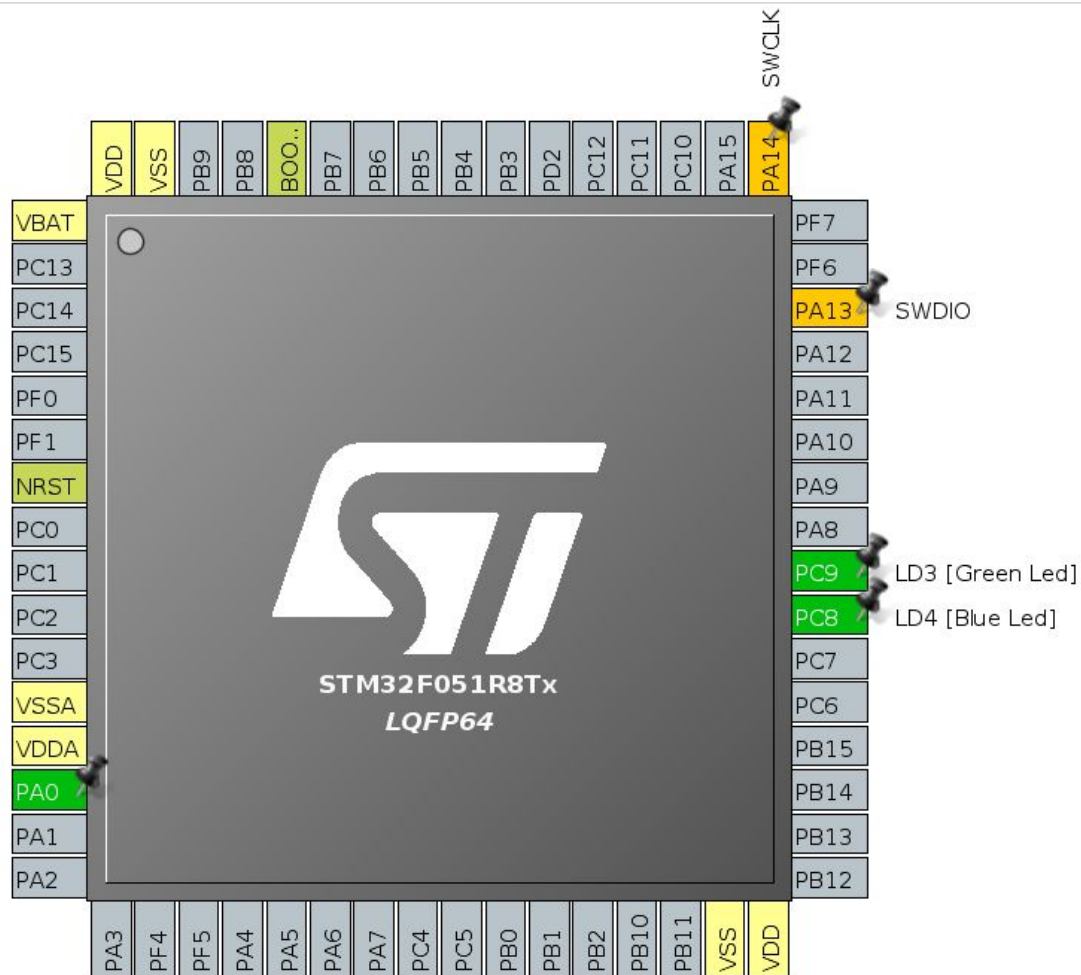
- Задача: отсчитывать каждую секунду
- Решение?

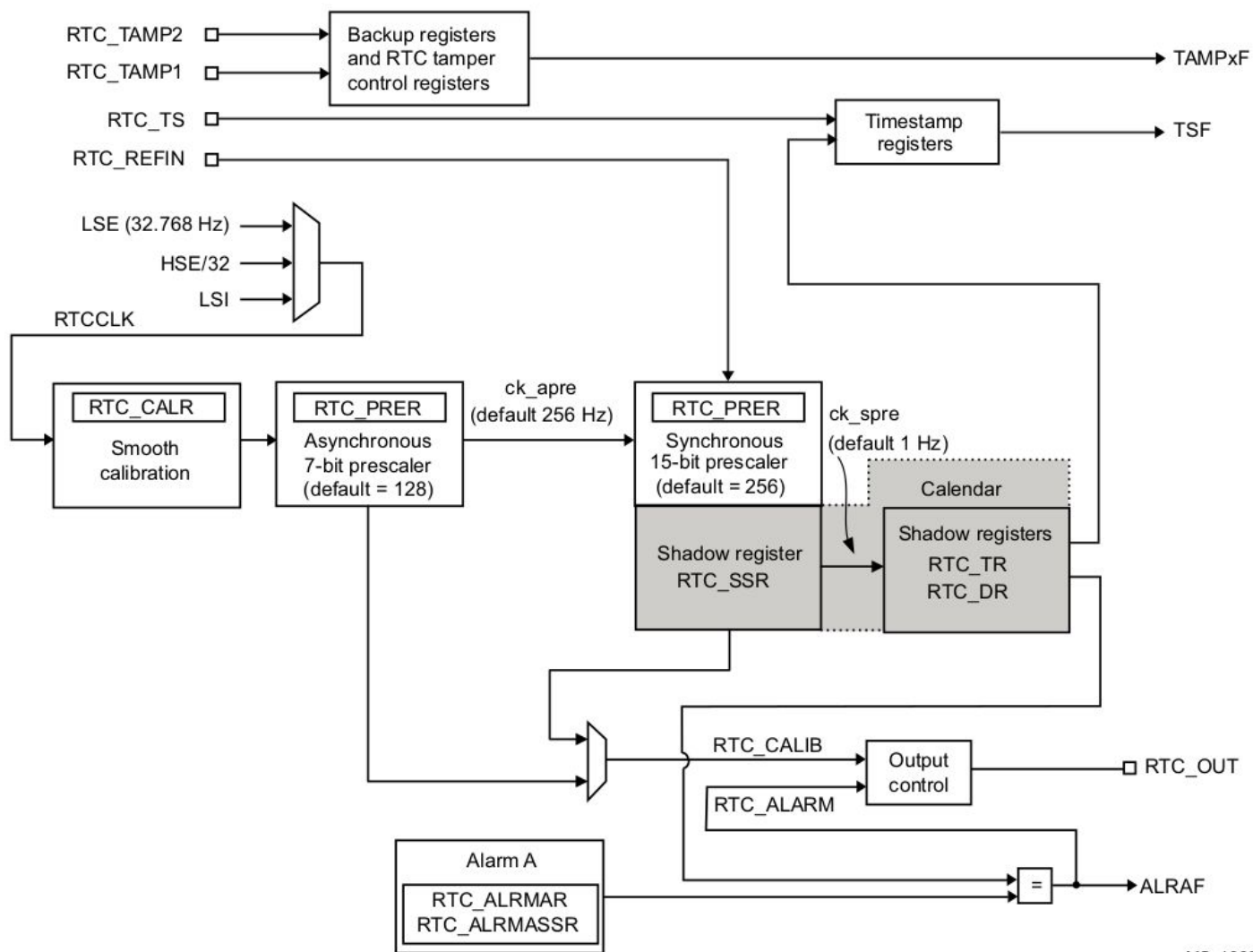
Возможности RTC в STM32

- Поддержка календаря
- Поддержка перевода стрелок в летний период
- Расширенный будильник с поддержкой прерывания
- Таймер включения (wake up timer)
- Корректировка на 60Гц с дополнительного источника тактирования
- Поддержка временных меток по внешнему событию (time stamping)
- 5 бэкап регистров
- Поддержка быстрого удаления содержимого бэкап регистров (tamper detection)

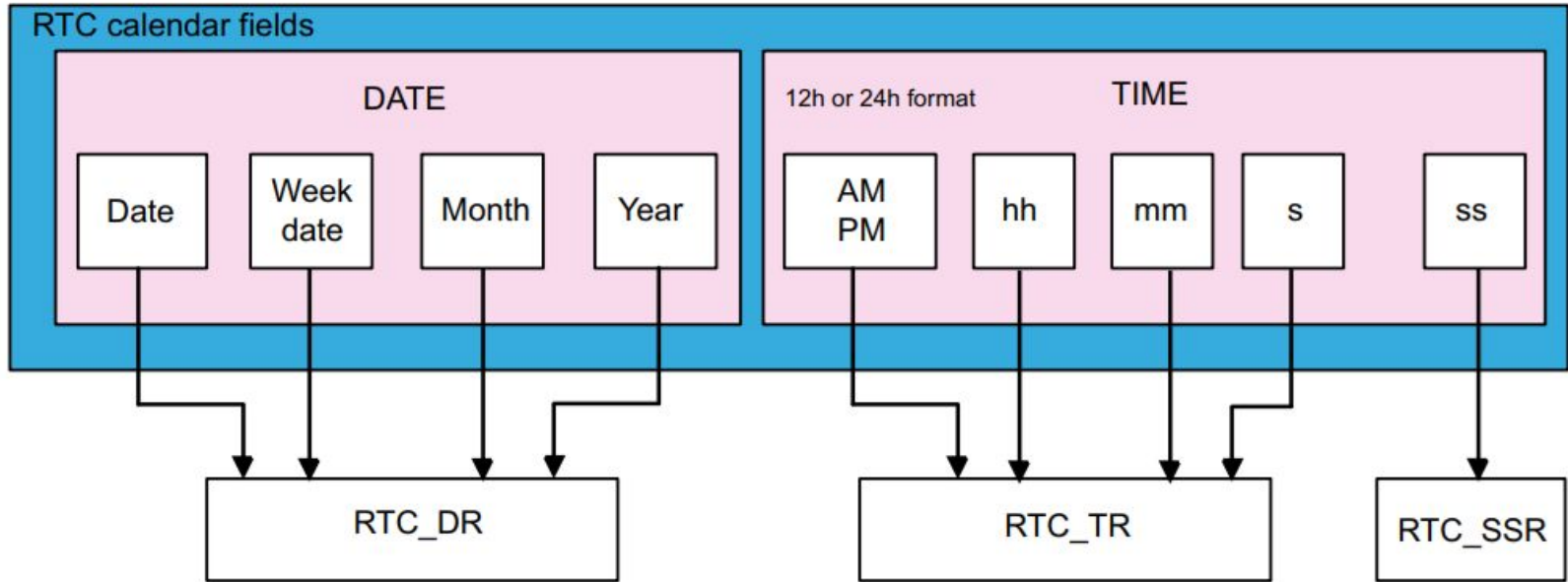
RTC. Батарейное питание

B1 [Blue PushButton]



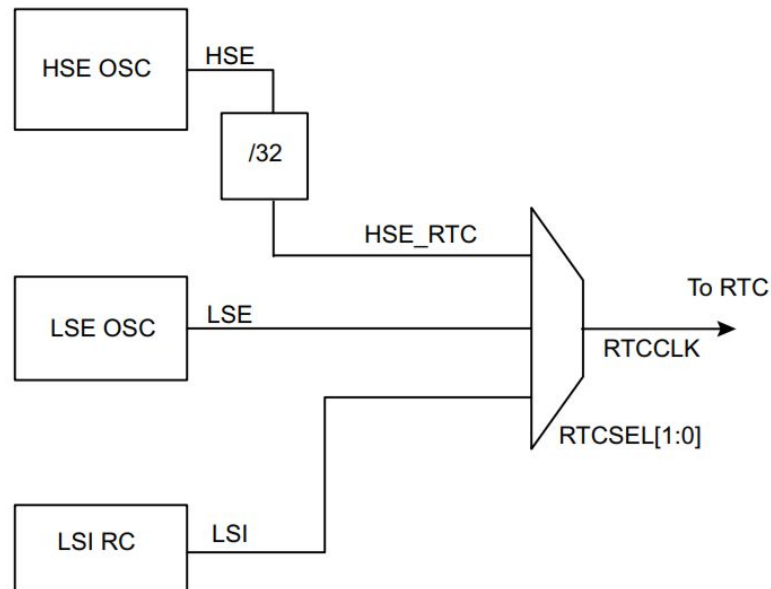


Хранения времени. Регистры. BCD формат



RTC. Включение периферии

1. `LL_RCC_LSI_Enable()`
2. **Включение доступа к периферии RTC!!**
`LL_PWR_EnableBkUpAccess()`
3. `LL_RCC_SetRTCClockSource()`
`LL_RCC_RTC_CLKSOURCE_LSI`
4. `LL_RCC_EnableRTC()`



RTC. Настройка календаря

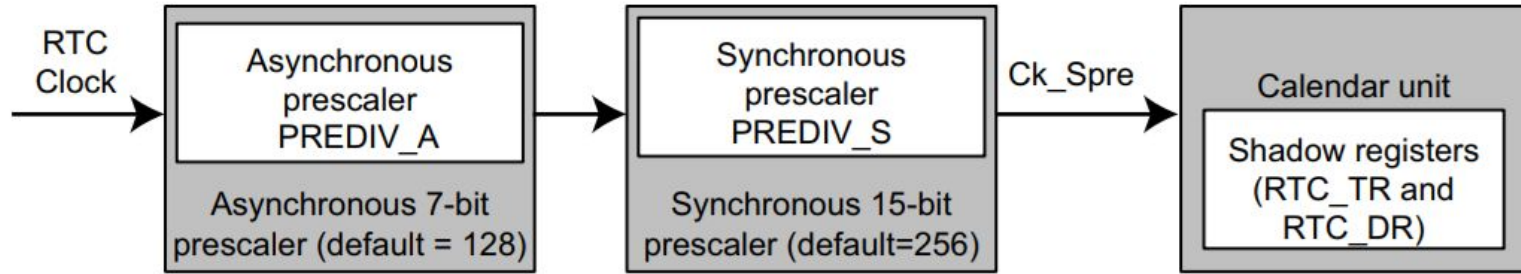
Любая настройка модуля часов реального времени начинается с разблокировки регистров

- *LL_RTC_DisableWriteProtection*(RTC) [0xCA, 0x53 -> RTC_WPR]
- *LL_RTC_EnableWriteProtection*(RTC) [0xFF -> RTC_WPR]

Включение/выключение режима инициализации:

- *LL_RTC_EnableInitMode*() [RTC_ISR, Init and status reg]
- *LL_RTC_DisableInitMode*()
- *LL_RTC_IsActiveFlag_INIT*() [RTC_ISR]

RTC. Настройка календаря



$$\text{ck_Spre} = \text{RTCCLK} / ((\text{PREDIV_A} + 1) \times (\text{PREDIV_S} + 1)) \rightarrow 127, 311$$

- *LL_RTC_SetAsynchPrescaler()* -> [RTC_PRER, prescaler register]
- *LL_RTC_SetSynchPrescaler()* -> [RTC_PRER]

RTC. Настройка календаря

- *LL_RTC_SetHourFormat* [RTC_CR]
 - LL_RTC_HOURFORMAT_24HOUR
 - LL_RTC_HOURFORMAT_AMPM
- *LL_RTC_DATE_Config* [RTC_DR]
 - День недели, LL_RTC_WEEKDAY_FRIDAY
 - День, 1
 - Месяц, LL_RTC_MONTH_MARCH
 - Год, 2019
- *LL_RTC_TIME_Config* [RTC_TR]
 - Формат, LL_RTC_TIME_FORMAT_AM_OR_24
 - Время, (часы: 11, минуты: 22, секунды: 00)

RTC. Чтение текущего времени

LL_RTC_EnableShadowRegBypass() [RTC_CR]

Преимущества теневых регистров:

- Данные когерентны

Недостатки теневых регистров:

- Данные сбрасываются после сброса
- После выхода из сна нужно вручную запрашивать данные
- Приходится ждать загрузки данных

RTC. Чтение текущего времени

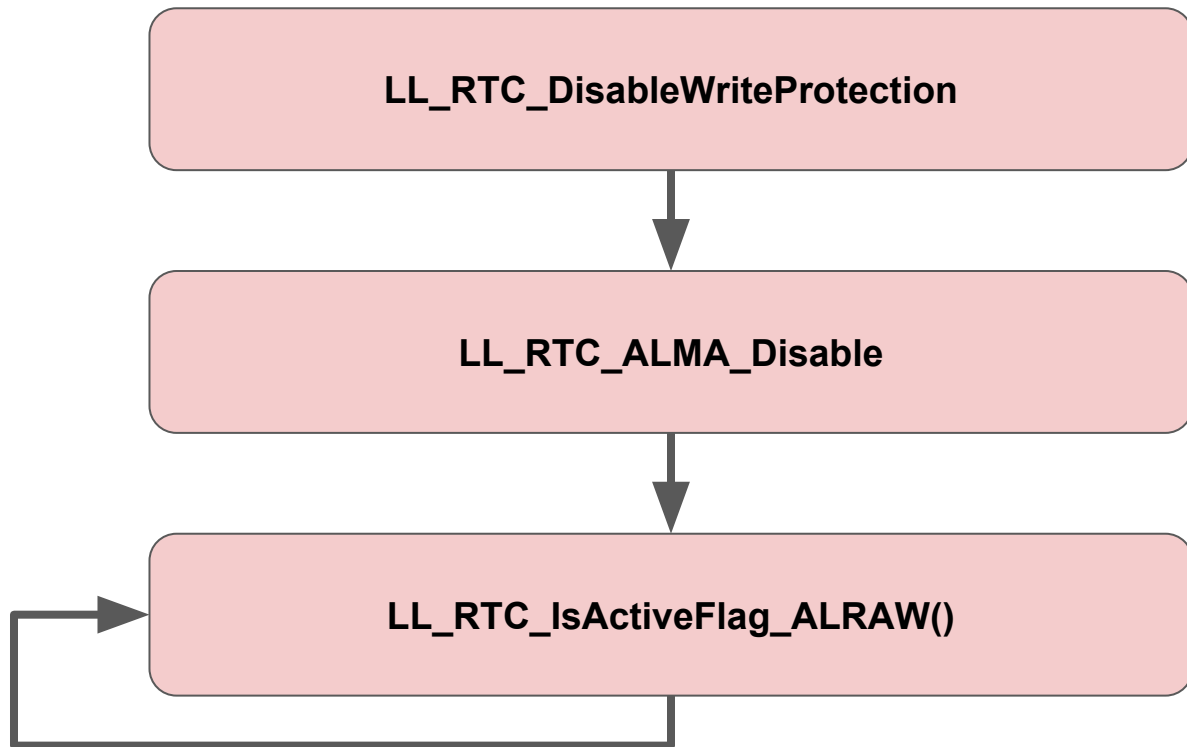
- *LL_RTC_IsActiveFlag_RS()* [RTC_ISR]
- *LL_RTC_TIME_Get()*
- *LL_RTC_DATE_Get()*

Но $f_{APB_CLK} \leq f_{RTC_CLK}$ в 7 раз,
то прочитать дважды или даже трижды!

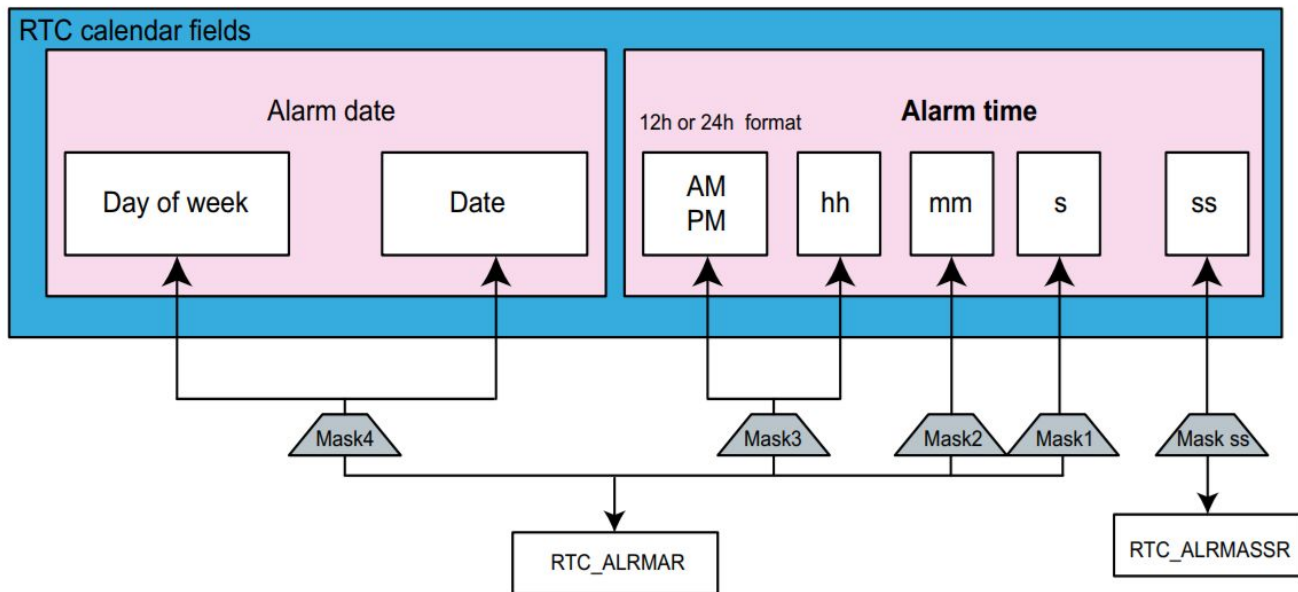
RTC. Будильник

- Большое количество условий для срабатывания
- Возможность пробуждать микроконтроллер
- Возможность конфигурации на внешний вывод портов ввода-вывода
- Возможность генерирования прерывания

RTC. Будильник. Настройка



RTC. Будильник. Настройка



LL_RTC_ALMA_SetMask()

LL_RTC_ALMA_ConfigTime()

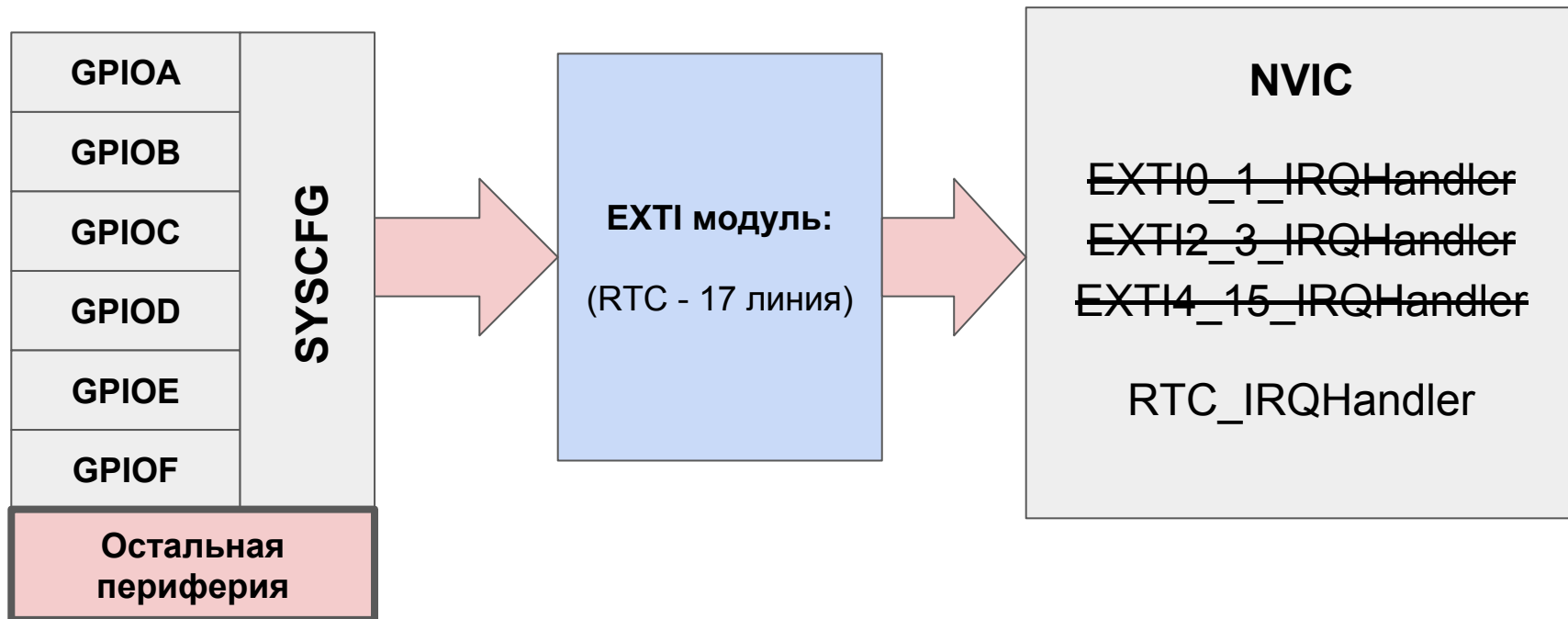
RTC. Будильник. Маскирование

MSK3	MSK2	MSK1	MSK0	Поведение
0	0	0	0	23:15:07, каждый ПН
0	0	0	1	23:15, Каждый ПН
0	0	1	0	23:XX:07 Каждый ПН

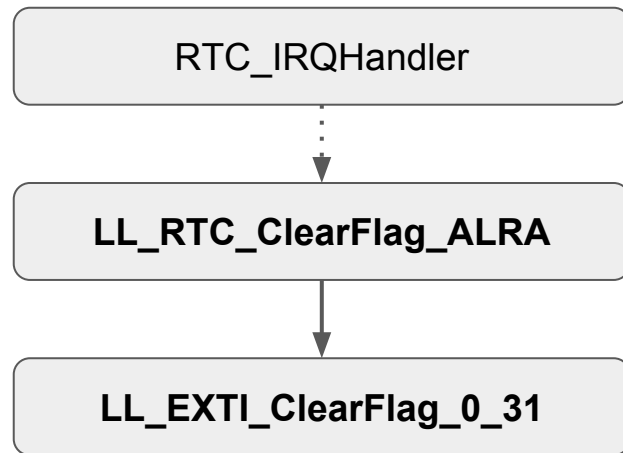
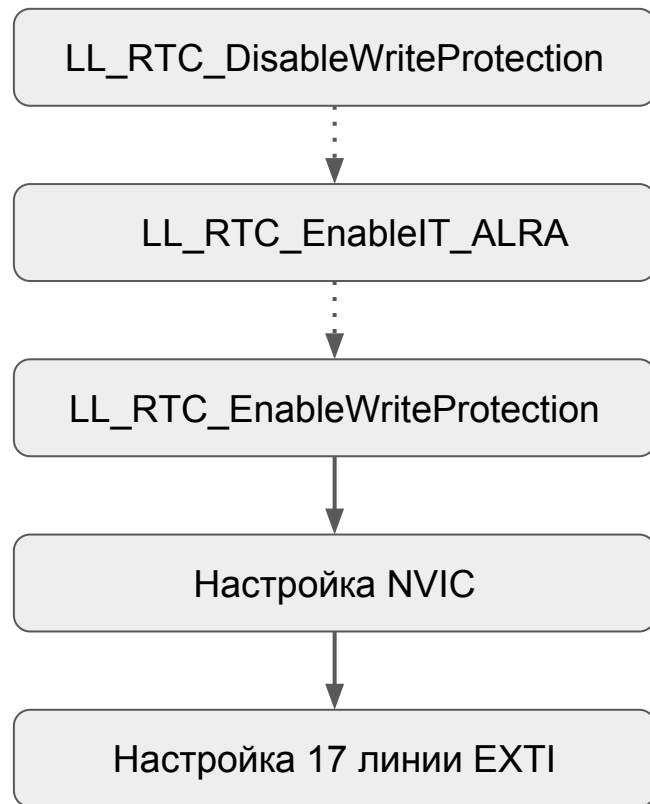


LL_RTC_ALMA_Enable

Связь EXTI, NVIC и RTC



RTC. Будильник. Прерывание



Репозиторий

https://github.com/edosedgar/stm32f0_ARM