Программирование микроконтроллеров STM32

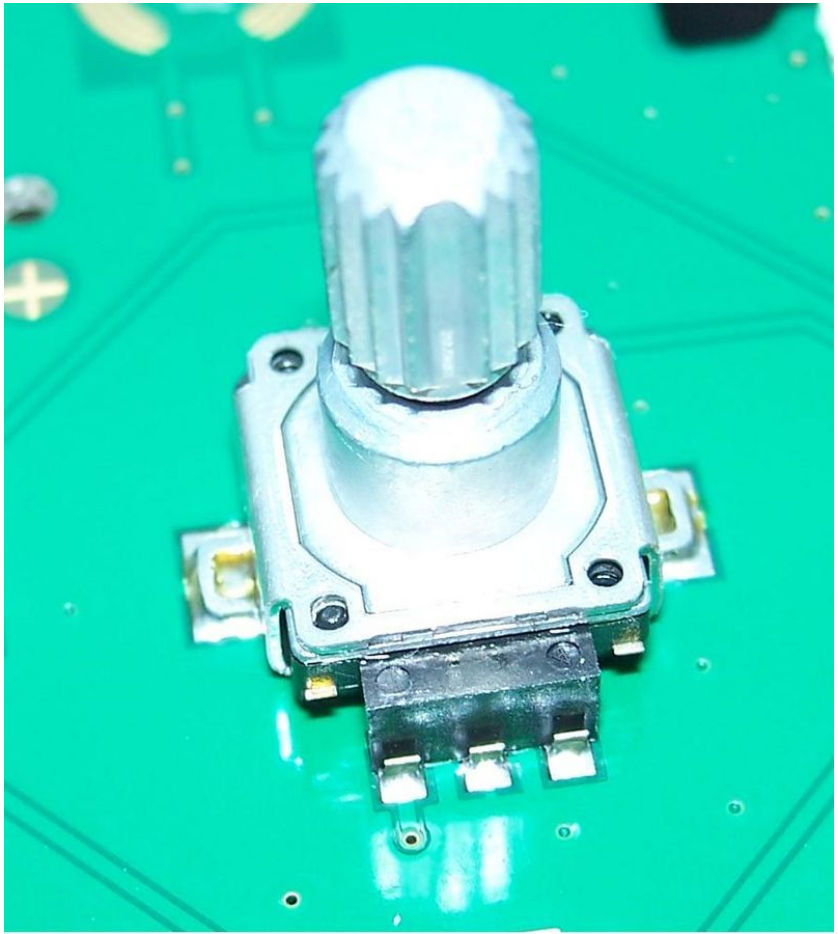
Таймеры общего назначения.
Часть 2

МФТИ, 2019

Таймер общего назначения (TIM2)

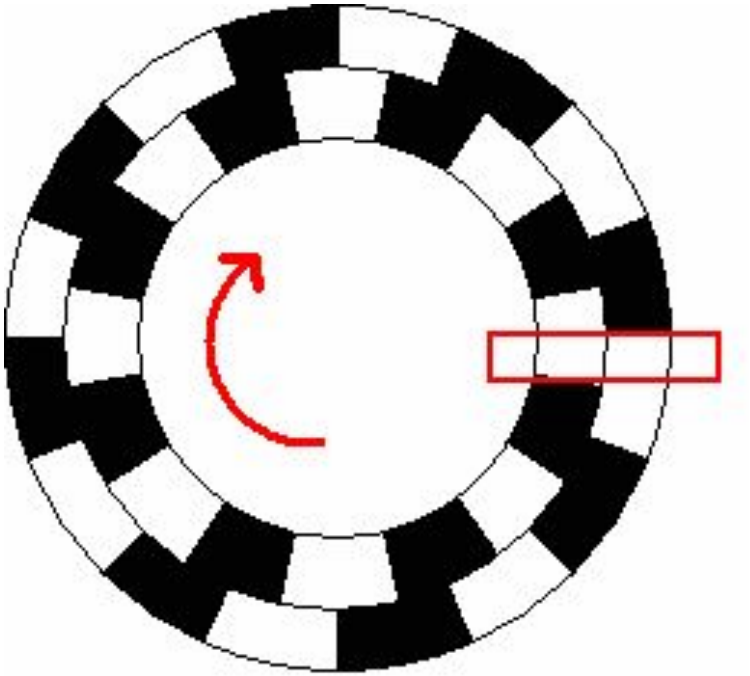
- 32 битный счетчик
- 4 входных канала для захвата сигнала
- **4 выходных канала для сигнала по сравнению**
- **Генерация ШИМ сигнала**
- Поддержка каскадного соединения нескольких таймеров
- Генерирование прерывания
- **Поддержка энкодера**

Инкрементальный энкодер



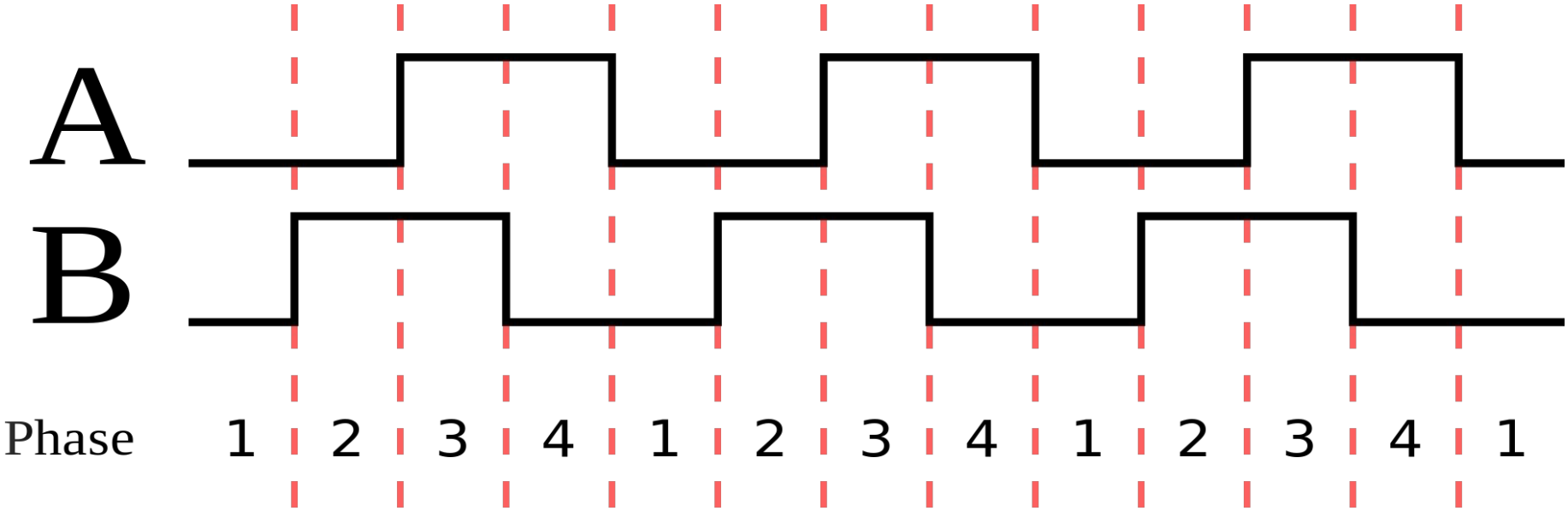
credits: wikipedia

Инкрементальный энкодер



credits: wikipedia

Инкрементальный энкодер

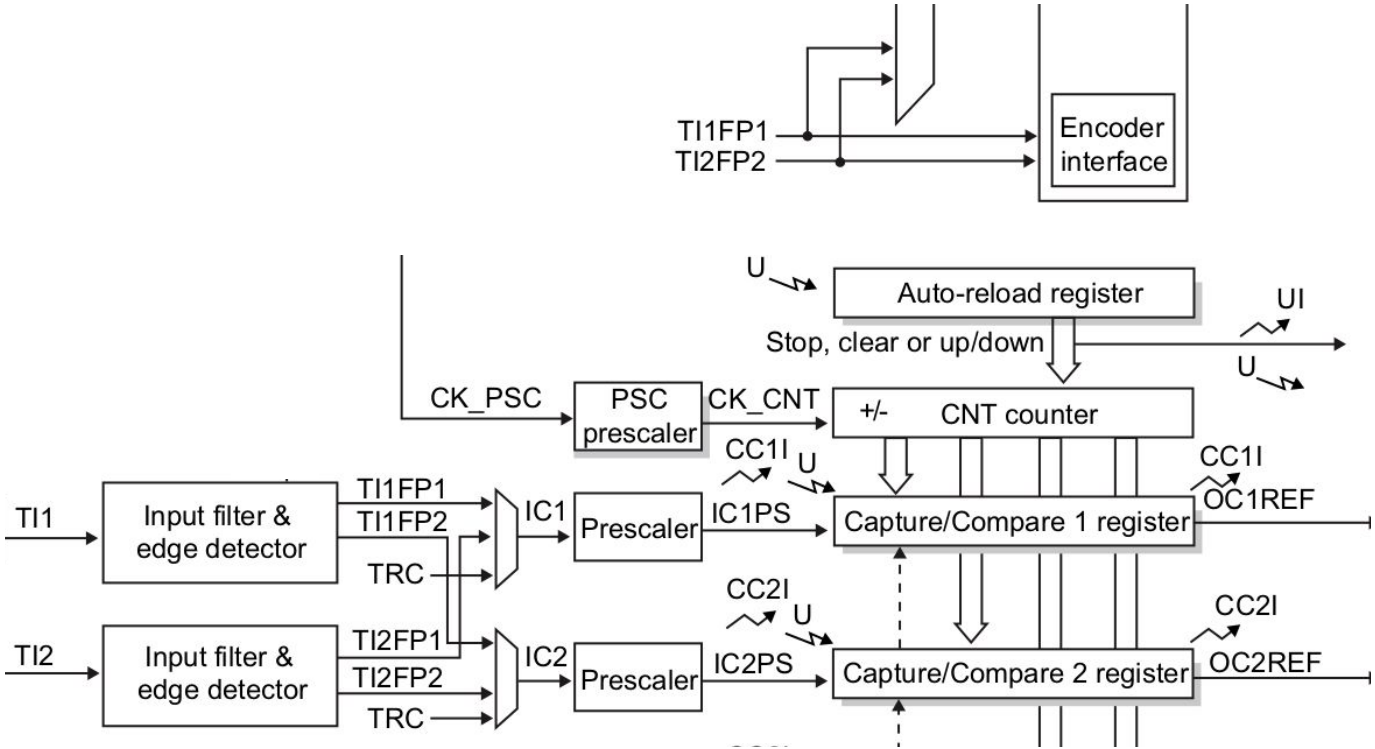


credits: wikipedia

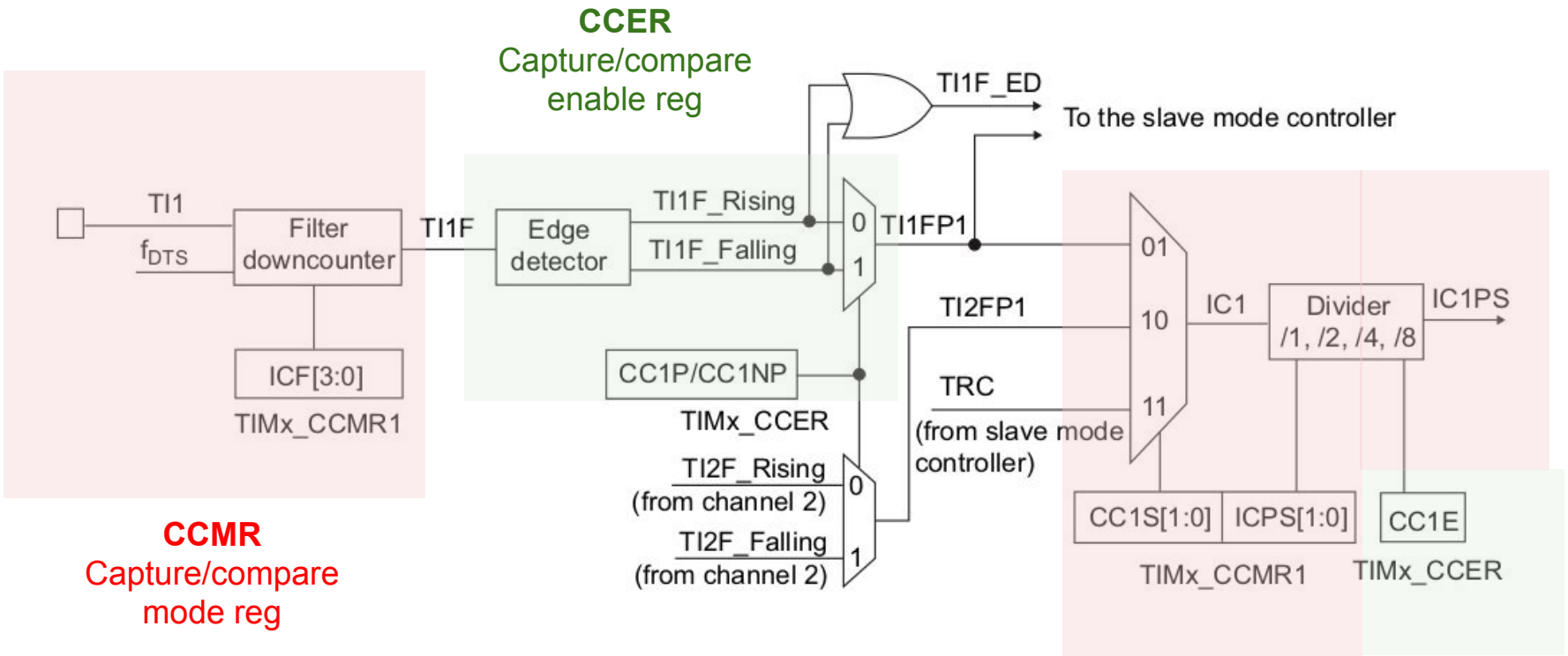
Энкодер и таймер

TIM2,
GPIOA 5

TIM2,
GPIOA 1



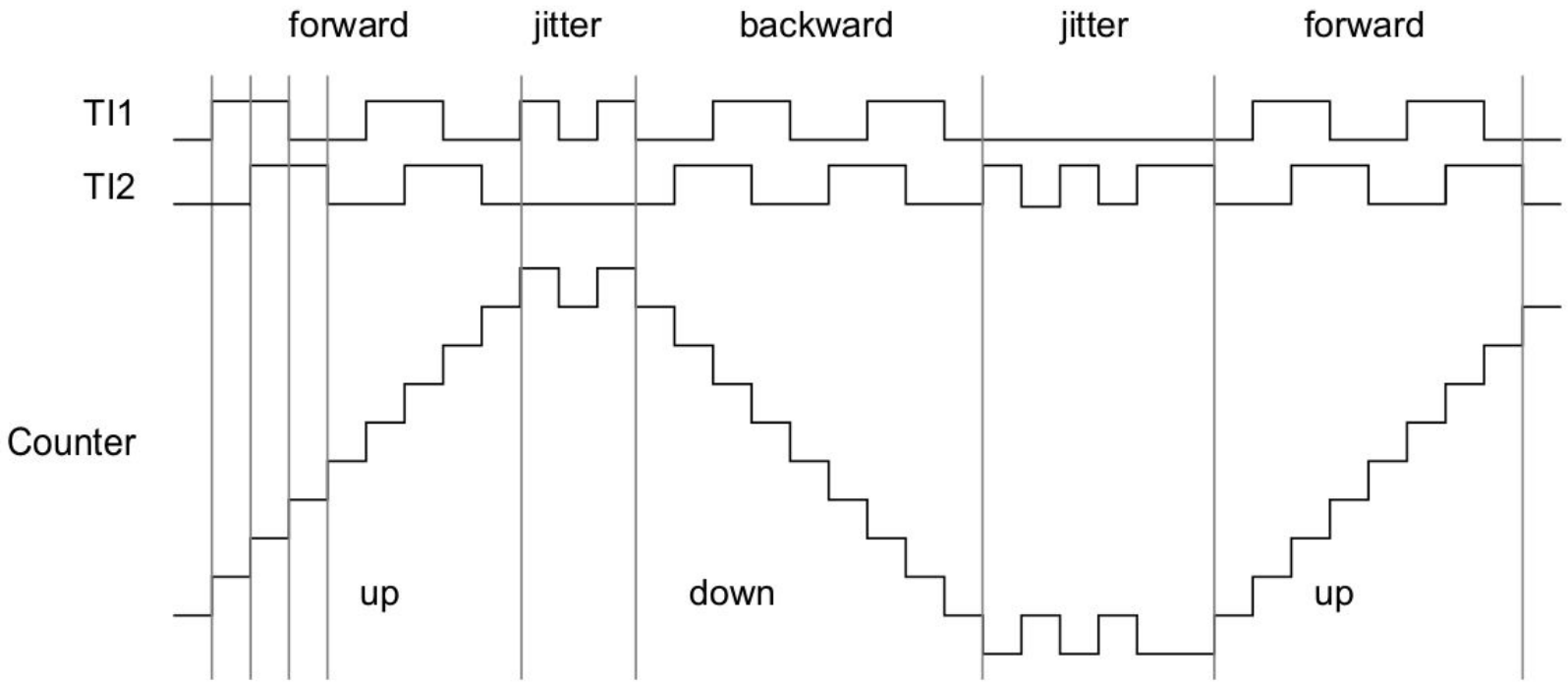
Энкодер и таймер



Энкодер и таймер

- TIMx_CH1 <- Выход A энкодера
- TIMx_CH2 <- Выход B энкодера
- DIR бит регистра TIMx_CR1 -> направление вращения энкодера
- TIMx_CNT тактируется от энкодера
- Пороговое значение для счетчика хранится в TIMx_ARR

Энкодер и таймер



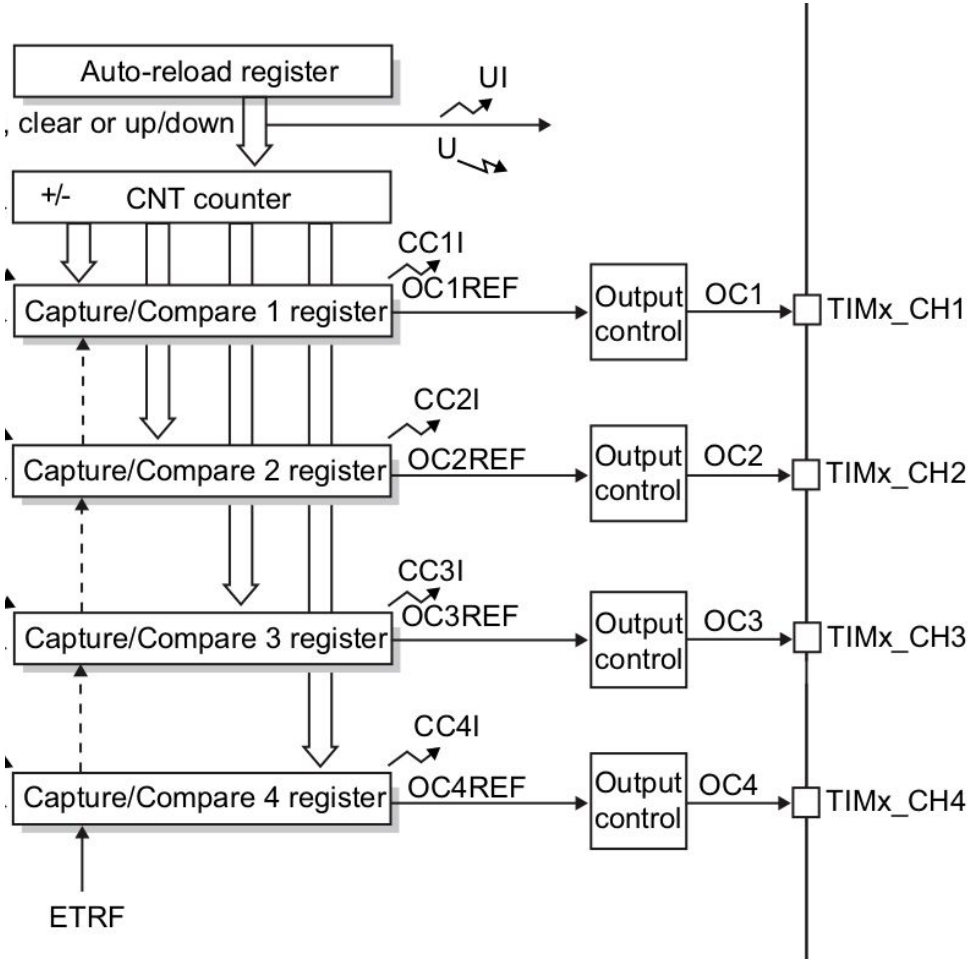
Энкодер и таймер. Инициализация

- Инициализация входов в GPIO (альт. функция, подтяжка на землю)
- Включение тактирования таймера
- Включение режима энкодера [SMCR]
  - *LL_TIM_SetEncoderMode*(TIMx, LL_TIM_ENCODERMODE_X4_TI12)
- Настройка полярности выхода с мультиплексора TIxFPx [CCER]
  - *LL_TIM_IC_SetPolarity*(TIMx, Channel, LL_TIM_IC_POLARITY_FALLING)
- Настройка регистра предзагрузки
  - *LL_TIM_SetAutoReload*
- Включение таймера

Энкодер и таймер. Опрос

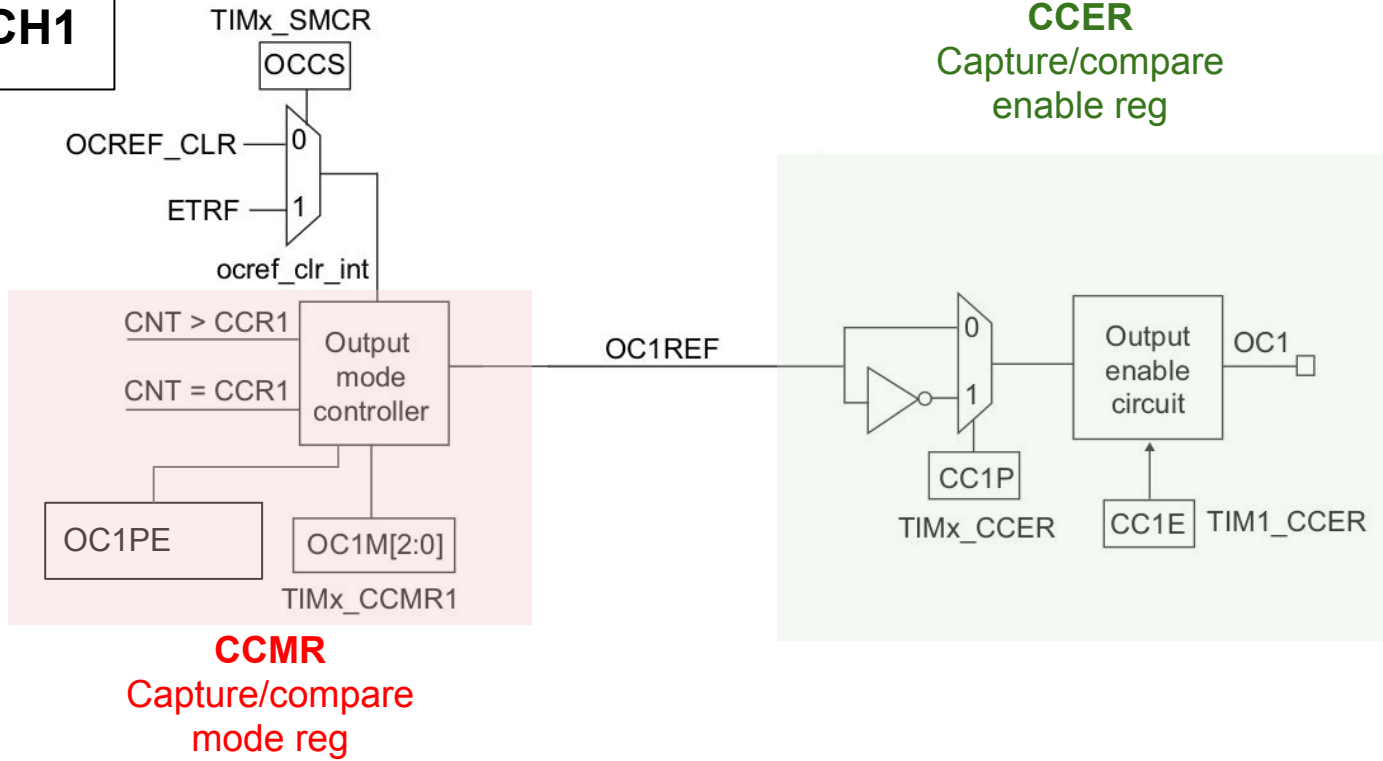
- *LL_TIM_GetCounterMode*(TIMx) [CR1]
- *LL_TIM_GetCounter*(TIMx) [CNT]

Режим по сравнению, ШИМ и таймеры



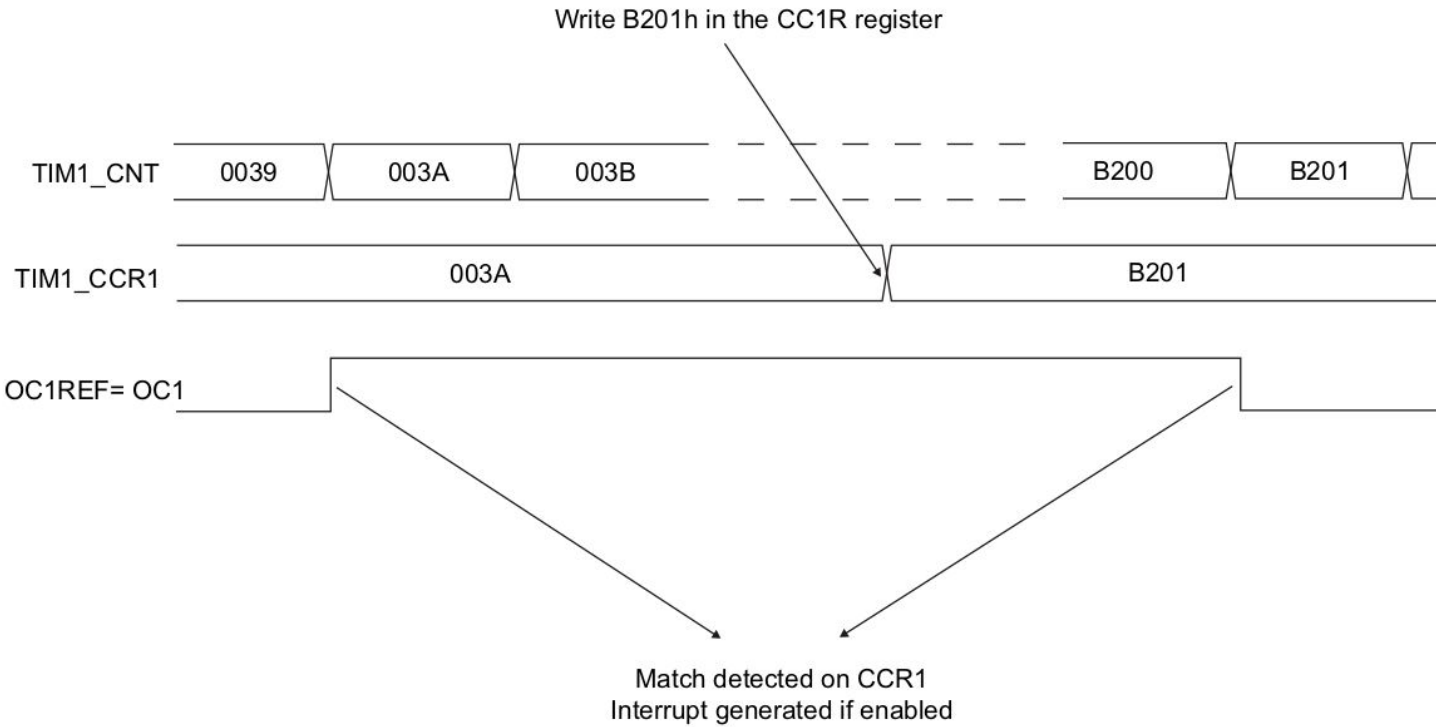
Режим по сравнению (output compare mode)

TIM2, CH1



GPIOA 5

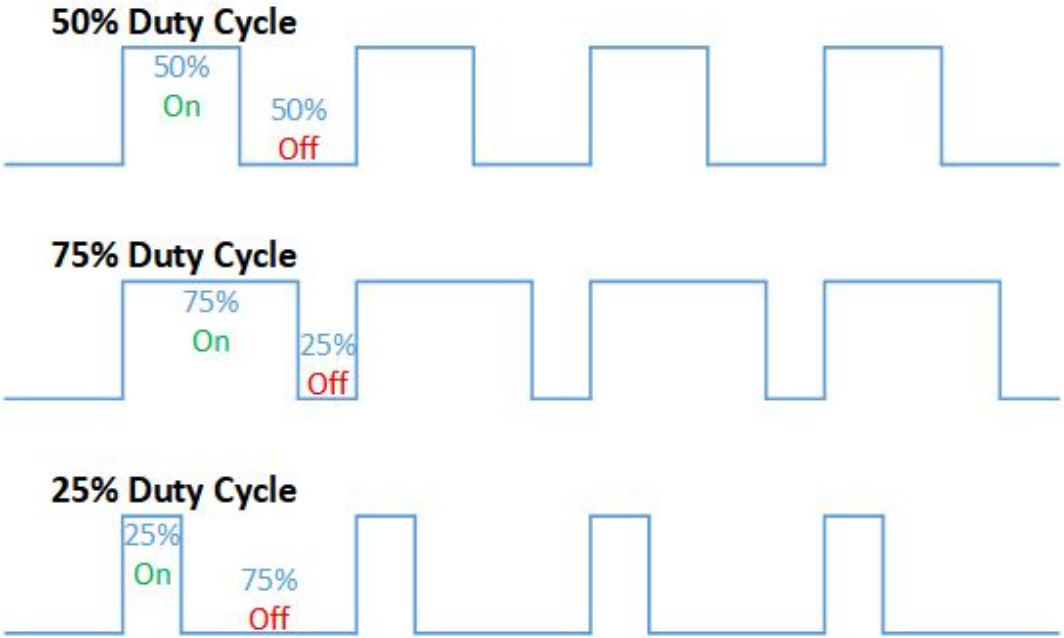
Режим по сравнению



Режим по сравнению. Инициализация

- Инициализация выхода GPIO (альт. функция)
- Включение тактирования таймера
- Установка предделителя -> *LL_TIM_SetPrescaler* [PSC]
- Уст. регистра предзагрузки -> *LL_TIM_SetAutoReload* [ARR]
- Уст. значения для сравнения -> *LL_TIM_OC_SetCompareCHx* [CCRx]
- Включение канала -> *LL_TIM_CC_EnableChannel* [CCER]
- Полярность выхода -> *LL_TIM_OC_SetPolarity* [CCER]
- Режим выхода -> *LL_TIM_OC_SetMode* [CCMR]
    - LL_TIM_OCMODE_TOGGLE
- Включение прерывания -> *LL_TIM_EnableIT_CC1* [DIER]
- Включение счетчика -> *LL_TIM_EnableCounter* [CR1]
- Настройка NVIC

Широтно-импульсная модуляция (Pulse-width modulation)



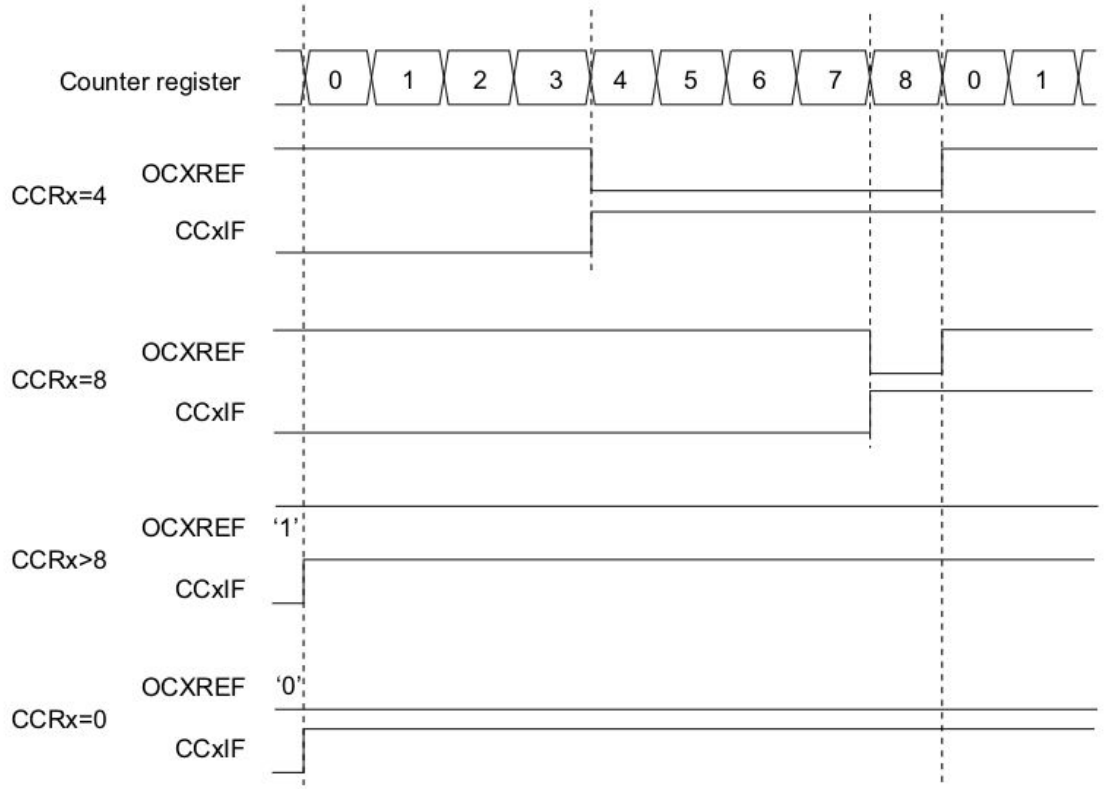
credits: wikipedia

ШИМ и таймер

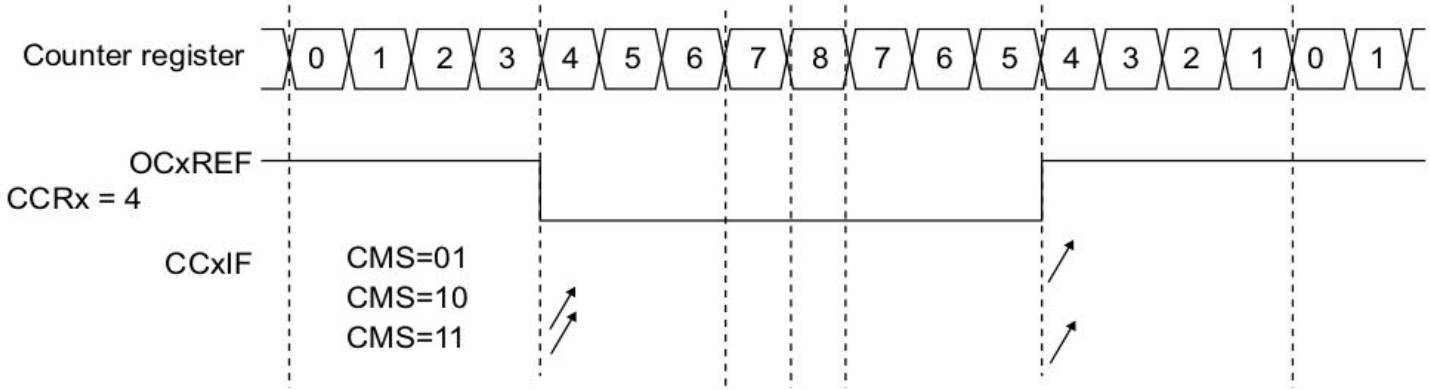
- Частота ШИМа (pwm frequency) -> регистр предзагрузки ARR
- Скважность ШИМа (duty cycle) -> регистр по сравнению CCRx
- Два режима ШИМ
  - ШИМ 1
    - **upcounting** mode: выход активен пока CNT < CCR1, в противном случае неактивен
    - **downcounting** mode: выход неактивен пока CNT > CCR1, в противном случае активен
  - ШИМ 2:
    - **upcounting** mode: выход неактивен пока CNT < CCR1, в противном случае активен
    - **downcounting** mode: выход активен пока CNT > CCR1, в противном случае неактивен

ШИМ. Выравнивание по краю (edge-aligned)

ARR=8, PWM 1
upcounting (DIR=0)
“1”: CNT < CCR1
“0: CNT >= CCR1



ШИМ. Выравнивание по центру (center-aligned)



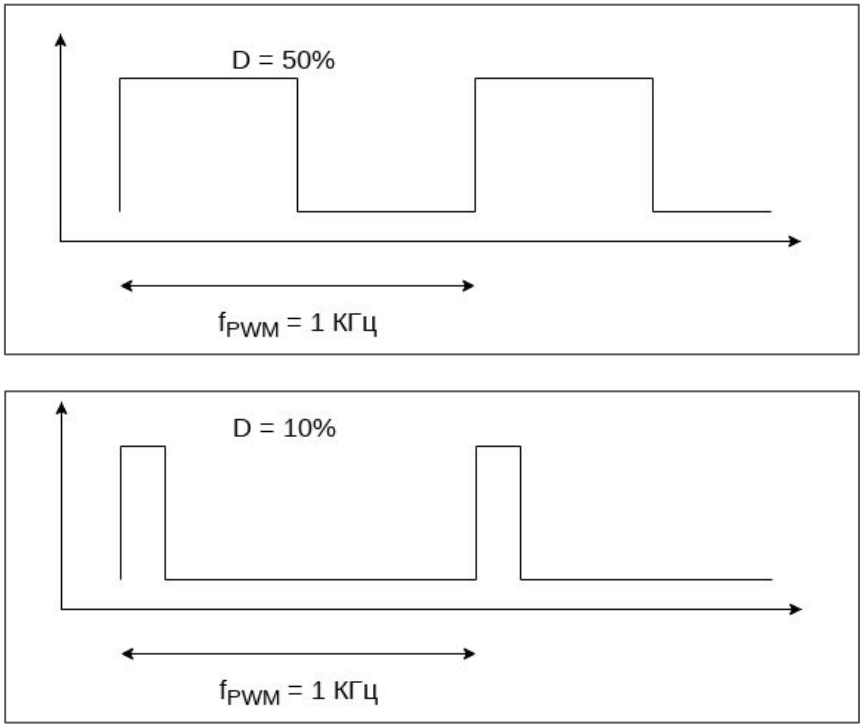
ARR=8, Режим PWM 1: сначала upcounting, потом downcounting

ШИМ. Инициализация

- Режим выхода -> *LL_TIM_OC_SetMode* [CCMR]
  - LL_TIM_OCMODE_PWM1/2
- Режим счетчика -> *LL_TIM_SetCounterMode* [CR1]
  - LL_TIM_COUNTERMODE_UP [DIR]
  - LL_TIM_COUNTERMODE_DOWN [DIR]
  - LL_TIM_COUNTERMODE_CENTER_UP [CMS]
  - LL_TIM_COUNTERMODE_CENTER_DOWN [CMS]
  - LL_TIM_COUNTERMODE_CENTER_UP_DOWN [CMS]

ШИМ. Пример расчета

- PWM 1, upcounting
- f_{TIM} = 48 МГц
- PSC = 479 -> f_{CNT} = 1 МГц
- ARR = 999 -> f_{PWM} = 1 КГц
- CCR1 = 499 -> D = 50%
- CCR2 = 99 -> D = 10%



Репозиторий

https://github.com/edosedgar/stm32f0_ARM