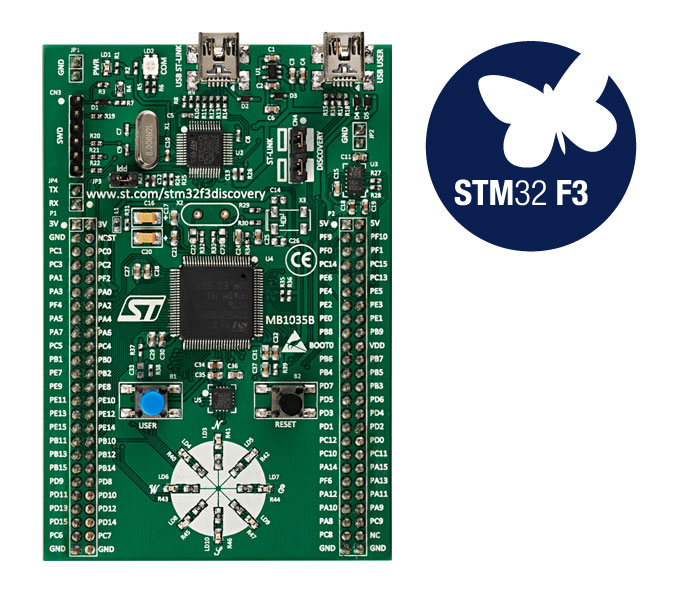
Основы работы с RTC в STM32F3



# Основные сведенье о CRC в STM32

Циклический избыточный код (сyclic redundancy check, CRC) - алгоритм нахождения контрольной суммы, предназначенный для проверки целостности данных. В отличие от кодов Рида-Соломона он не позволяют исправлять ошибки, но имеет простую аппаратную реализацию, за счет чего получили широкое распространение.

Вычисляемая сумма является остатком от деления исходного сообщения на порождающий многочлен в конечном поле. Более детальное описание процедуры деления, и связных с ней алгоритмов, а также параметры основных CRC алгоритмов можно найти по ссылкам:

* <http://www.sunshine2k.de/articles/coding/crc/understanding_crc.html>
* <https://pycrc.org/models.html>

В микроконтроллерах STM32 имеется аппаратный блок расчета CRC сумм. Для большинства серий (F0, F1, F2, F3, L1) блок имеет фиксированные параметры, и предназначен для расчета наиболее распространённой суммы - CRC32, которая используется в ряде протоколов (ethernet), и форматах данных (gzip, png, etc.). В микроконтроллерах STM32F3 размер и коэффициента полиному можно настраивать.

# Работа с CRC блоком

## Инициализация

Включение CRC блока является простым по сравнению с другими устройствами. Для этого нужно только включить CRC в регистре RCC->AHBENR (бит CRCEN)

## Подготовка к расчету

Для начала расчета нужно сбросить блок и установить параметры CRC суммы выполнив следующие действия:

1. сбросить CRC блок (регистр CRC->CR, бит RESET);
2. выбрать, нужно ли инвертировать порядок бит входной последовательности (регистр CRC->CR, биты REV\_IN);
3. выбрать, нужно ли инвертировать порядок бит выходной суммы (регистр CRC->CR, бит REV\_OUT);
4. выбрать порядок (степень) полинома (регистр CRC->CR, биты POLYSIZE);
5. задать сам полином в нормальном представлении (регистр RCC->POL);
6. задать начальное состояния регистра сдвига (регистр RCC->INIT).

На этом подготовка CRC закончена, и можно подавать данные. Отметим, что у блока отсутствует функциональность по сложение по модулю 2 выходного значения с заранее заданной константой. Поэтому если вы захотите реализовать стандартный **CRC32** и сравнить его с какой-либо реализацией, то значение, полученное от STM32F3 **нужно** будет не забыть сложить с 0xFFFFFFFF по модулю 2 (out ^ 0xFFFFFFFF)/

## Передача данных в CRC

Для передачи данных в CRC необходимо просто записывать их в регистр CRC->DR.

DR регистр является 32-х битным, и в него можно записывать сразу по 4 байта. Если нужно записать только 1 байт, то его нужно записать в младший байт регистра, не трогая старшие байты. При реализации программы для этого нужно правильно привести тип регистра к uint8\_t, чтобы сгенерированный код не пытался записать что-либо в другие байты.

## Получения CRC суммы

Для этого нужно просто считать данные из CRC->DR регистра. CRC блок понимает, что происходит операция чтения, и выдает CRC сумму.