

Микроконтроллер MDR32F9Q2I

Часть 3. Системный таймер SysTick

Михаил Голубцов (Москва)

Цикл статей предназначен для тех читателей, которые хотят ознакомиться с отечественным микроконтроллером, совместимым с Cortex-M3, и научиться его использовать. Рассмотрены основные аппаратные узлы микроконтроллера, а также их программирование. В каждой статье приводится пример реального использования, со схемой и полным текстом программы. Микроконтроллер Cortex-M3 сложнее многих популярных приборов, и данная публикация может облегчить его освоение.

Процессор имеет 24-разрядный системный таймер SysTick, который считает вниз от загруженного в него значения до нуля. Перезагрузка (возврат в начало) значения в регистр LOAD происходит по очередному фронту синхросигнала, затем счёт продолжается по следующему фронту. Когда процессор остановлен для отладки, таймер не декрементируется.

ОПИСАНИЕ РЕГИСТРОВ СИСТЕМНОГО ТАЙМЕРА SysTick

Все регистры системного таймера SysTick доступны для записи и чтения. Имеются следующие регистры:

- SysTick->CTRL – управление функциями таймера;
- SysTick->LOAD – перегружаемое значение таймера;
- SysTick->VAL – текущее значение системного таймера;
- SysTick->CAL – калибровочное значение таймера.

Регистр SysTick->CTRL разрешает основные функции системного таймера. Назначение битов регистра:

- 31 – 17 – не используются;
- 16 – COUNTFLAG, возвращает 1, если таймер досчитал до нуля с последнего момента чтения;
- 15 – 3 – не используются;
- 2 – CLKSOURCE, указывает источник синхросигнала: 0 – LSI, 1 – HCLK;
- 1 – TCKINT, разрешает запрос на прерывание от системного таймера: 0 – таймер досчитает до нуля и прерыва-

ние не возникнет, 1 – возникает запрос на прерывание;

- 0 – ENABLE, разрешает работу таймера: 0 – работа таймера запрещена, 1 – работа таймера разрешена.

Когда бит ENABLE установлен в единицу, таймер загружает значение RELOAD из регистра LOAD и начинает декрементироваться. По достижению значения 0 таймер устанавливает бит COUNTFLAG и, если бит TCKINT установлен в 1, генерирует запрос на прерывание. Затем загружается значение RELOAD и продолжается счёт.

Регистр LOAD устанавливает стартовое значение, загружаемое в регистр VAL. Это значение загружается в регистр VAL, когда таймер разрешён и когда достигается значение нуля. Назначение битов регистра VAL:

- 31 – 24 – не используются;
- 23 – 0 – RELOAD.

Значение RELOAD может быть любым в диапазоне 0x00000001 – 0x00FFFFFF. Значение 0 допустимо, но не оказывает влияния, потому что запрос на прерывание и активизация бита COUNTFLAG происходят только при переходе таймера из состояния 1 в 0.

Расчёт значения RELOAD происходит в соответствии с использованием таймера:

- для формирования таймера с периодом N процессорных циклов применяется значение RELOAD, равное N – 1. Например, если требуется прерывание каждые 100 циклов, то уста-

навливается значение RELOAD, равное 99;

- для формирования одиночного прерывания после задержки в N тактов процессора используется значение N. Например, если требуется прерывание после 400 тактов процессора, то устанавливается RELOAD, равное 400.

Регистр VAL содержит текущее значение системного таймера. Назначение битов регистра:

- 31 – 24 – не используются;
- 23 – 0 – CURRENT.

Чтение значения CURRENT возвращает текущее значение системного таймера. Запись любого значения сбрасывает регистр в ноль и также очищает бит COUNTFLAG регистра CTRL.

Регистр CAL показывает калибровочное значение системного таймера. Назначение битов регистра:

- 31 – NOREF, всегда читается как 0;
- 30 – SKEW, всегда читается как 0;
- 29 – 24 – не используются;
- 23 – 0 – TENMS, читается как 0x0002904.

Калибровочное значение фиксировано и равно 0x0002904 (10500), что позволяет генерировать базовое время 1 мс с частотой 10,5 МГц (84/8 = 10,5 МГц).

ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ СИСТЕМНОГО ТАЙМЕРА

Системный таймер работает от процессорного синхросигнала. Если синхросигнал останавливается в режиме пониженного энергопотребления, то системный таймер останавливается. Необходимо, чтобы программа использовала доступ к регистрам системного таймера, выровненный по словам.

Системный таймер удобно использовать для формирования требуемых временных интервалов во время выполнения программы. Один из возможных вариантов – завести переменную счётчика «тиков». В обработчике

прерывания системного таймера значение этой переменной увеличивается на единицу. Это прерывание возникает с частотой, которая зависит от настроек системного таймера. Например, можно сделать его таким, что переменная счётчика «тиков» будет увеличиваться на единицу каждую миллисекунду.

В любом месте программы можно присвоить вспомогательной переменной текущее значение системного таймера и начать подсчитывать требуемое количество «тиков», сравнивая сохранённое в момент начала интервала значение с текущим значением системного таймера. Например, если надо сформировать паузу длительностью 1 с, следует ждать до тех пор, пока разность между сохранённым и текущим значением системного таймера станет меньше 1000.

Обратите внимание, что перед использованием системного таймера необходимо настроить тактирование микроконтроллера.

На сайте журнала приведён пример возможного использования системного таймера.

Следует учитывать, что действия, выполняемые во время ожидания окончания интервала времени, должны быть короткими, так как в этом случае проверка факта окончания интервала времени происходит реже, – фактически, с частотой, которая зависит от длительности действий внутри цикла ожидания завершения паузы. Пример: надо засечь интервал 1000 мс, и в цикле происходит проверка нажатия на кнопку (проверка состояния вывода порта и условный оператор). Такое действие происходит очень быстро и почти не влияет на точность отсчёта интервала времени.

Если внутри цикла вставить длительные действия, например, передачу блока данных по последовательному порту, которая происходит в течение 400 мс, то в данном случае проверка факта окончания заданного интервала времени происходит очень редко – примерно раз в 400 мс (через 400, 800 и 1200 мс). Очевидно, что в итоге получится интервал времени длительностью 1200 мс, а не требуемый интервал в 1000 мс.

В следующей части цикла будут рассмотрены источники тактовых сигналов для микроконтроллера и работа с ними.

