Современные 32-разрядные ARM-микроконтроллеры серии STM32: цифро-аналоговый преобразователь

Олег Вальпа

В статье приведено описание цифро-аналогового преобразователя 32-разрядных ARM-микроконтроллеров серии STM32 от компании STMicroelectronics. Рассмотрена архитектура и состав его регистров, а также приведены практические примеры программ.

ВВЕДЕНИЕ

Цифро-аналоговый преобразователь (ЦАП) представляет собой устройство, позволяющее получить аналоговый сигнал необходимой формы из соответствующего цифрового кода. Фактически он производит обратную операцию, выполняемую аналого-цифровым преобразователем (АЦП). ЦАП и АЦП являются интерфейсами между дискретным цифровым миром и аналоговыми сигналами. Любой ЦАП характеризуется разрядностью, производительностью и динамическим диапазоном.

ОПИСАНИЕ ЦАП

Цифро-аналоговый преобразователь микроконтроллеров серии STM32 [1] представляет собой 12-разрядный преобразователь цифровых данных в выходное напряжение от 0 В до опорного напряжения Vref+. ЦАП поддерживает как 12-разрядный режим, так и 8-разрядный. Кроме того, он может быть использован в сочетании с блоком прямого доступа к памяти DMA.

ЦАП имеет два канала, каждый из которых содержит независимый преобразователь. Канал 1 подключен к выводу PA4, а канал 2 — к выводу PA5.

В двухканальном режиме преобразование может выполняться независимо или одновременно, когда оба канала группируются для синхронного запуска.

В 12-разрядном режиме данные могут выравниваться по правому или по левому краю 16-разрядных слов.

Запуск преобразования возможен программно либо от внешних источников. В качестве таких источников запуска могут служить таймеры или внешний вход EXTI_9.

Вход опорного напряжения Vref+ является общим с блоком АЦП. ЦАП микроконтроллеров серии STM32 имеет следующие основные характеристики:

- два цифро-аналоговых преобразователя с самостоятельным выходом;
- поддержка 8-разрядного и 12-разрядного режимов;
- возможность выравнивания данных по левому или правому краю в 12-разрядном режиме;
- обеспечение возможности синхронного обновления;
- наличие встроенного генератора шума;
- наличие встроенного генератора треугольных импульсов;
- допустимость независимого или одновременного преобразования в каналах;
- возможность использования DMA для обоих каналов;

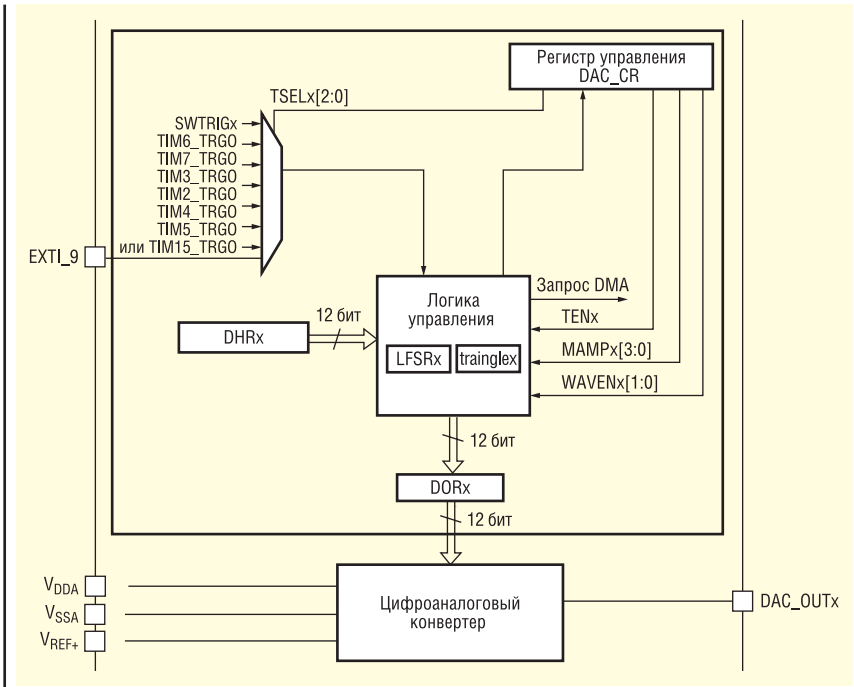

Рис. 1. Структурная схема ЦАП

- обеспечение внешнего запуска преобразований;
- наличие входа опорного напряжения.

Структурная схема цифро-аналогового преобразователя приведена на рисунке 1. Выводы, показанные на нем, имеют следующие назначения:

- EXTI_9 — вход внешнего прерывания с номером 9;
- Vref+ — вход опорного напряжения;
- Vdda — вход аналогового питания;
- Vssa — вход общего потенциала аналогового питания;
- DAC_OUTx — выход аналогового сигнала, где x принимает значение 1 или 2 для соответствующего канала.

В верхней левой части схемы показаны источники внешнего запуска преобразования: программный запуск, таймеры и внешний вход EXTI_9. Логика управления позволяет формировать сигнал шума и треугольный сигнал, а также запрос DMA. Регистр управления предназначен для программного управления блоком ЦАП. Сигнал DMAENx регистра управления предназначен для разрешения работы первого или второго канала ЦАП. Сигнал TENx включает внешний запуск преобразования от источника, выбираемого сигналами TSELx[2:0]. С помощью разрядов WAVENx[1:0] включаются и отключаются функции формирования шума или треугольного сигнала. Разряды MAMPx[3:0] задают битовую маску, которая используется для формирования сигнала шума с помощью регистра сдвига и логических вентилей. Если формируется треугольный сигнал, тогда в этих разрядах задается амплитуда выходного сигнала.

ОПИСАНИЕ РЕГИСТРОВ ЦАП

Блок ЦАП включает в свой состав 14 регистров для двух каналов. Карта размещения регистров ЦАП в пространстве памяти представлена в таблице 1.

Рассмотрим подробнее назначение всех разрядов этих регистров.

Регистр управления DAC_CR имеет нулевое смещение адреса и обнуляется после сброса микроконтроллера. Этот регистр состоит из двух частей: разряды 0...15 управляют работой канала 1, а разряды 16...31 управляют работой канала 2. Они устанавливаются и очищаются программно.

Биты 31...30 зарезервированы.

Бит 29 с именем DMAUDRIE2 разрешает прерывания в канале 2 при отставании запроса DMA. Состояние 0 этого бита запрещает прерывание, а состояние 1 — разрешает.

Бит 28 DMAEN2 разрешает запрос DMA в канале 2.

Биты 27...24 MAMP2[3:0] служат для выбора маски в режиме генерации шума или амплитуды в режиме генерации треугольных импульсов в канале 2. Они могут принимать следующие значения:

- 0000 — не маскирован бит 0 LFSR или амплитуда треугольных импульсов равна 1;
- 0001 — не маскированы биты [1:0] LFSR или амплитуда треугольных импульсов равна 3;
- 0010 — не маскированы биты [2:0] LFSR или амплитуда треугольных импульсов равна 7;
- 0011 — не маскированы биты [3:0] LFSR или амплитуда треугольных импульсов равна 15;
- 0100 — не маскированы биты [4:0] LFSR или амплитуда треугольных импульсов равна 31;
- 0101 — не маскированы биты [5:0] LFSR или амплитуда треугольных импульсов равна 63;
- 0110 — не маскированы биты [6:0] LFSR или амплитуда треугольных импульсов равна 127;
- 0111 — не маскированы биты [7:0] LFSR или амплитуда треугольных импульсов равна 255;
- 1000 — не маскированы биты [8:0] LFSR или амплитуда треугольных импульсов равна 511;
- 1001 — не маскированы биты [9:0] LFSR или амплитуда треугольных импульсов равна 1023;
- 1010 — не маскированы биты [10:0] LFSR или амплитуда треугольных импульсов равна 2047;
- 1011 — не маскированы биты [11:0] LFSR или амплитуда треугольных импульсов равна 4095.

Биты 23...22 WAVE2[1:0] разрешают генерацию шума или треугольного сигнала в канале 2 и могут принимать следующие значения:

Таблица 1. Карта регистров ЦАП

Сдвиг адреса	Имя регистра	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DAC_CR	Резерв		DMAUDRIE2	DMAEN2	MAMP2[3:0]				WAVE2		TSEL2[2:0]			TEN2	BOFF2	EN2	Резерв		DMAUDRIE1	DMAEN1	MAMP1[3:0]				WAVE1[2:0]		TSEL1[2:0]			TEN1	BOFF1	EN1
0x04	DAC_SWT RIGR	Резерв																														SWTRIG2	SWTRIG1
0x08	DAC_DHR1 2R1	Резерв																				DACC1DNR[11:0]											
0x0C	DAC_DHR1 2L1	Резерв																DACC1DHR[11:0]														Резерв	
0x10	DAC_DHR8 R1	Резерв																								DACC1DHR[7:0]							
0x14	DAC_DHR1 2R2	Резерв																				DACC2DHR[11:0]											
0x18	DAC_DHR1 2L2	Резерв																DACC2DHR[11:0]														Резерв	
0x1C	DAC_DHR8 R2	Резерв																								DACC2DHR[7:0]							
0x20	DAC_DHR1 2RD	Резерв				DACC2DHR[11:0]												Резерв				DACC1DHR[11:0]											
0x24	DAC_DHR1 2LD	DACC2DHR[11:0]												Резерв				DACC1DHR[11:0]														Резерв	
0x28	DAC_DHR8 RD	Резерв																DACC2DHR[7:0]								DACC1DHR[7:0]							
0x2C	DAC_DOR1	Резерв																				DACC1DOR[11:0]											
0x30	DAC_DOR2	Резерв																				DACC2DOR[11:0]											
0x34	DAC_SR	Резерв		DMAUDR2		Резерв														DMAUDR1		Резерв											

- 00 — генерация запрещена;
- 01 — разрешена генерация шума;
- 1X — разрешена генерация треугольного сигнала.

Генерация возможна, если бит TEN2 регистра DAC_CR установлен в 1, т.е. разрешен запуск в канале 2.

Биты 21...19 TSEL2[2:0] позволяют выбрать источник запуска для канала 2. Эти биты определяют тип внешнего события, используемого для запуска канала 2 ЦАП. Они могут принимать следующие значения:

- 000 — таймер 6;
- 001 — зарезервирован;
- 010 — таймер 7;
- 011 — таймер 9;
- 100 — таймер 2;
- 101 — таймер 4;
- 110 — внешняя линия 9;
- 111 — программный запуск.

Бит 18 TEN2 разрешает запуск канала 2 ЦАП.

Бит 17 BOFF2 разрешает использование выходного буфера для канала 2.

Бит 16 EN2 разрешает работу канала 2.

Биты 15...0 имеют аналогичное назначение для канала 1.

Регистр DAC_SWTRIGR служит для программного запуска.

Биты 31...2 зарезервированы.

Бит 1 SWTRIG2 обеспечивает программный запуск канала 2.

Бит 0 SWTRIG1 производит программный запуск канала 1.

Регистр DAC_DHR12R1 служит для записи 12-разрядных данных с выравниванием по правому краю и позволяет использовать биты 11...0 в качестве данных для канала 1. Остальные биты в нем зарезервированы.

Регистр DAC_DHR12L1 служит для записи 12-разрядных данных с выравниванием по левому краю и позволяет использовать биты 15...4 в качестве данных для канала 1. Остальные биты в нем зарезервированы.

Регистр DAC_DHR8R1 служит для записи 8-разрядных данных с выравниванием по правому краю и позволяет использовать биты 7...0 в качестве данных для канала 1. Остальные биты в нем зарезервированы.

Структура регистров для второго канала ЦАП DAC_DHR12R2, DAC_DHR12L2 и DAC_DHR8R2 такая же, как и для первого.

Следующие три регистра позволяют записывать данные в два канала одновременно.

Регистр DAC_DHR12RD служит для записи 12-разрядных данных с выравниванием по правому краю. Биты 11...0 служат для записи данных для первого канала, а биты 27...16 — для данных второго канала.

Регистр DAC_DHR12LD предназначен для записи 12-разрядных данных с выравниванием по левому краю. Биты 15...4 служат для записи данных для первого канала, а биты 31...20 для данных второго канала.

Регистр DAC_DHR8RD нужен для записи 8-разрядных данных с выравниванием по правому краю. В биты 7...0 записываются данные для первого канала, а в биты 16...8 для второго.

Регистры выходных данных DAC_DOR1 и DAC_DOR2 содержат выходные данные соответственно для первого и второго каналов. Они доступны только для чтения. Структура этих регистров одинаковая. Данные содержатся в разрядах 11...0.

Регистр статуса DAC_SR хранит информацию о состоянии ЦАП. В этом регистре зарезервированы все биты кроме двух.

Бит 29 DMAUDR2 – это флаг, сигнализирующий отставание запроса DMA в канале 2. Такая ситуация происходит, когда частота запуска преобразования выше, чем возможность обслуживания запроса DMA.

Бит 13 DMAUDR1 — аналогичный флаг отставания запроса DMA в канале 1.

РЕЖИМЫ РАБОТЫ ЦАП

Каждый канал может быть включен установкой разряда ENx соответствующего канала в регистре DAC_CR. Канал включается с задержкой времени Twakeup.

ЦАП имеет два выходных буферных элемента, которые используются для снижения выходного сопротивления и подключения напрямую к нагрузке без применения внешнего операционного усилителя. Управление выходными буферами осуществляется с помощью разрядов BOFFx в регистре DAC_CR.

В зависимости от содержимого регистра DAC_DORx изменяется напряжение на выходе DAC. Но непосредственная запись данных в этот регистр невозможна, допускается только чтение. В режиме одиночных каналов данные загружаются в регистр DAC_DORx через специальные регистры:

- DAC_DHR12Rx — для записи 12 бит данных с выравниванием по правому краю;
- DAC_DHR12Lx — для записи 12 бит данных с выравниванием по левому краю;
- DAC_DHR8Rx — для записи 8 бит данных с выравниванием по правому краю.

На рисунке 2 наглядно показаны все эти способы загрузки регистра DAC_DORx.

В режиме сдвоенных каналов данные загружаются в DORx через регистры:

- DAC_DHR12RD — для записи 12 бит данных с выравниванием по правому краю;
- DAC_DHR12LD — для записи 12 бит данных с выравниванием по левому краю;
- DAC_DHR8RD — для записи 8 бит данных с выравниванием по правому краю.

На рисунке 3 показаны представленные здесь способы загрузки регистра DORx.

Данные, загруженные в регистр DAC_DHRx, автоматически переписываются в регистр DAC_DORx через один цикл APB1, если разряд TENx регистра DAC_CR сброшен, или через три цикла, если разряд TENx регистра DAC_CR установлен.

Когда регистр DAC_DORx загружается содержимым DAC_DHRx, сформированное напряжение появляется на аналоговом выходе спустя время Tsettling, которое зависит от напряжения питания и нагрузки, подключенной к выходу.

Цифровые данные преобразуются с помощью ЦАП в выходное напряжение, находящееся в диапазоне от 0 В до Vref+. Зависимость выходного сигнала от входных данных линейная. Выходное напряжение вычисляется по следующей формуле: DACoutput = Vref+. Зависимость выходного сигнала от входных

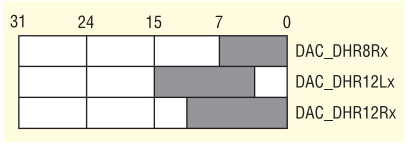

Рис. 2. Способы загрузки регистра DAC_DORx в режиме одиночных каналов

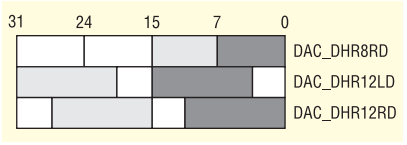

Рис. 3. Способы загрузки регистра DAC_DORx в режиме сдвоенных каналов

Таблица 2. Выбор источника запуска

Источник запуска	Тип	Содержимое TSEL[2:0]
Таймер 6	Внутренний сигнал от таймера	000
Таймер 3		001
Таймер 7		010
Таймер 5 или таймер 15		011
Таймер 2		100
Таймер 4		101
EXTI_9	Внешний сигнал	110
SWTRIG	Программно	111

данных линейная. Выходное напряжение вычисляется по следующей формуле: DACoutput = Vref+ * (DOR / 4095).

Если управляющий разряд TENx установлен, включается внешний запуск преобразования от таймера или внешнего вывода контроллера. С помощью разрядов TSELx[2:0] выбирается источник запуска, как показано в таблице 2.

Каждый раз при обнаружении фронта запускающего сигнала данные, хранящиеся в регистре DAC_DHRx, будут переданы в регистр DAC_DORx, и он обновится через три цикла APB1.

Если выбран программный запуск, преобразование начнется сразу после установки разряда SWTRIG, который сбросится аппаратно после передачи данных из регистра DAC_DHRx в регистр DAC_DORx.

Разряды TSELx[2:0] нельзя изменять, если установлен бит ENx. Если выбран программный запуск, то передача данных из регистра DAC_DHRx в регистр DAC_DORx занимает один цикл шины APB1.

ИСПОЛЬЗОВАНИЕ DMA

Каждый канал может работать с DMA. Для обслуживания запросов от ЦАП в модуле DMA предусмотрено два канала. Запрос DMA будет генерироваться от ЦАП, когда приходит не программный сигнал внешнего запуска. Затем значение регистра DAC_DHRx будет перенесено в регистр DAC_DORx.

В сдвоенном режиме, если установлены оба разряда DMAENx, генерируются запросы DMA от обоих каналов. Если необходим только один запрос, то следует установить соответствующий разряд DMAENx.

Запросы от ЦАП к DMA не имеют очереди, поэтому если второй внешний запуск пришел до подтверждения последнего запроса, последний без сообщения об ошибке обслужен не будет.

ГЕНЕРАТОР ШУМА

Для генерирования шума доступен регистр LFSR (Linear Feedback Shift Register). Генератор шума выбирается установкой разрядов WAVEx[1:0] = 01. Предустановленное значение регистра LFSR равно 0xAAA. Этот регистр обновляется спустя три такта APB1 после каждого внешнего запуска, после выполнения специального алгоритма вычисления сигнала шума, который приведен на рисунке 4.

Значение регистра LFSR, которое может быть скрыто полностью или частично с помощью разрядов MAMPx[3:0] регистра DAC_CR, добавляется без переполнения к содержимому регистра DAC_DHRx, который затем будет сохранен в регистре DAC_DORx. Если LFSR = 0, к нему добавляется 1, реализуя, тем самым, антиблокировочный механизм. На рисунке 5 представлена временна́я диаграмма процедуры генерации шума. Генератор шума включается установкой разряда TENx регистра DAC_CR.

ГЕНЕРАТОР ТРЕУГОЛЬНОГО СИГНАЛА

Генерирование треугольного сигнала посредством ЦАП производится за-

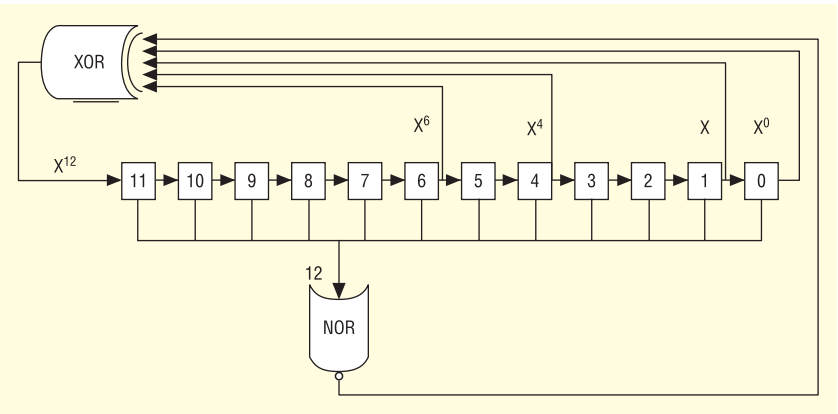

Рис. 4. Алгоритм вычисления сигнала шума

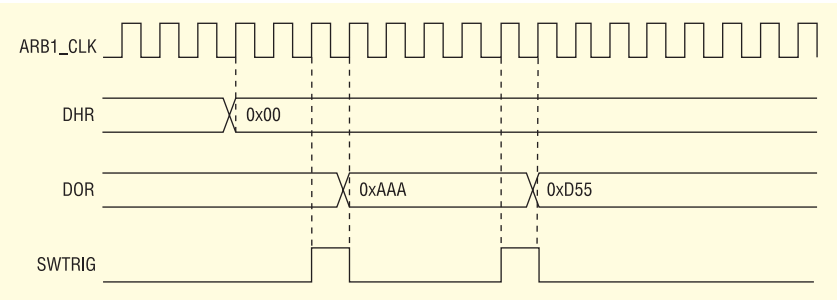

Рис. 5. Временна́я диаграмма процедуры генерации шума

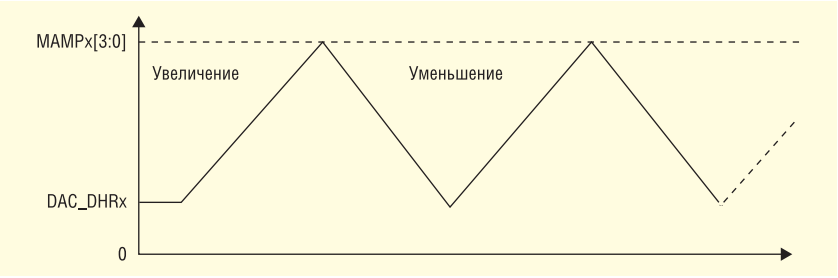

Рис. 6. Осциллограмма генерации шума

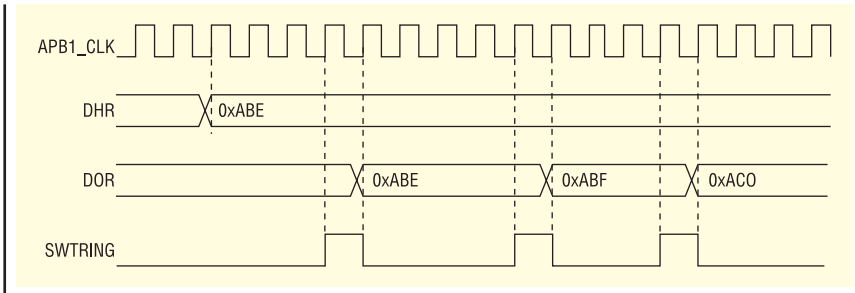

Рис. 7. Временна́я диаграмма процедуры генерации треугольника

писью в разряды WAVEx[1:0] двоичного значения 10. Амплитуда сигнала задается разрядами MAMPx[3:0] регистра DAC_CR. После каждого внешнего запуска через три цикла APB1 происходит инкремент специального счетчика. Значение этого счетчика затем добавляется к содержимому регистра DAC_DHRx и сохраняется в DAC_DORx. Этот счетчик будет инкрементироваться до тех пор, пока не достигнет максимального значения, которое определяется разрядами MAMPx[3:0]. После достижения этого значения счетчик переключится на вычитание, а при достижении нуля — обратно на сложение. Сброс генерирования треугольного сигнала произойдет после сброса разрядов WAVEx[1:0].

На рисунке 6 представлена осциллограмма генерации шума.

Временна́я диаграмма процедуры генерации треугольника представлена на рисунке 7.

РЕЖИМ ДВУХКАНАЛЬНОГО ПРЕОБРАЗОВАНИЯ

Чтобы эффективно использовать пропускную способность шины в приложениях, требующих работу двух каналов ЦАП, имеется три регистра (DHR8RD, DHR12RD и DHR12LD), которые обеспечивают одновременный доступ к обоим каналам.

Существует 11 способов преобразования с помощью двух каналов и этих трех регистров. Ниже приводятся описания этих режимов.

НЕЗАВИСИМЫЙ ЗАПУСК БЕЗ ГЕНЕРАЦИИ

Для настройки этого режима преобразования необходимо выполнить следующие действия:

- включить оба канала установкой разрядов TEN1 и TEN2;
- задать источник запуска каждого из каналов установкой разрядов TSEL1[2:0] и TSEL2[2:0];
- загрузить данные в нужный регистр DHR (DAC_DHR12RD, DAC_DHR12LD или DAC_DHR8RD).

Когда появится сигнал запуска канала 1, данные из регистра DHR1 будут перенесены в регистр DAC_DOR1. Когда появится сигнал запуска канала 2, данные из регистра DHR2 будут перенесены в регистр DAC_DOR2. Перенос данных происходит через три такта APB1.

НЕЗАВИСИМЫЙ ЗАПУСК С LFSR-ГЕНЕРАЦИЕЙ

Для настройки этого режима преобразования необходимо выполнить следующие операции:

- включить оба канала установкой разрядов TEN1 и TEN2;
- задать источник запуска каждого из каналов установкой разрядов TSEL1[2:0] и TSEL2[2:0];
- сконфигурировать оба канала установкой разрядов WAVEx[1:0] в 01 и задать значение маски регистра LFSR с помощью разрядов MAMPx[3:0];
- загрузить данные в нужный регистр DHR (DAC_DHR12RD, DAC_DHR12LD или DAC_DHR8RD).

Листинг 1

```
#include «stm32f10x.h»
#include «stm32f10x_rcc.h»
#include «stm32f10x_gpio.h»
// Массив данных для генератора гармоничного сигнала синуса
const uint16_t sin[32] = {
2047, 2447, 2831, 3185, 3498, 3750, 3939, 4056, 4095, 4056,
3939, 3750, 3495, 3185, 2831, 2447, 2047, 1647, 1263, 909,
599, 344, 155, 38, 0, 38, 155, 344, 599, 909, 1263, 1647};
unsigned char i=0;
// Функция обработчика прерывания от таймера 6
void TIM6_DAC_IRQHandler(void)
{
TIM6->SR &= ~TIM_SR_UIF; // Сбросить флаг UIF
DAC->DHR12R1=sin[i++]; // Записать в ЦАП очередной элемент массива
if (i==32) i=0; // Если вывели в ЦАП все 32 значения - начать снова
}
int main(void)
{
// Включить порт
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
// Включить ЦАП
RCC_APB1PeriphClockCmd(RCC_APB1Periph_DAC, ENABLE);
// Включить таймер 6
RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM6,ENABLE);
// Настроить вывод ЦАП
GPIO_InitTypeDef GPIO_InitStructure;
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_4;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
GPIO_Init(GPIOA, &GPIO_InitStructure);
// Настроить частоту таймера
TIM6->PSC = 0;
TIM6->ARR = 500;
TIM6->DIER |= TIM_DIER_UIE; // Разрешить прерывание от таймера
TIM6->CR1 |= TIM_CR1_CEN; // Начать отсчет
NVIC_EnableIRQ(TIM6_DAC_IRQn); // Разрешить TIM6_DAC_IRQn прерывания
// Включить DAC1
DAC->CR |= DAC_CR_EN1;
// Организовать бесконечный цикл
while (1) {}
}
```

Листинг 2

```
#include «stm32f10x.h»
#include «stm32f10x_rcc.h»
#include «stm32f10x_gpio.h»
int main(void) {
// Включить порт A
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
// Включить ЦАП
RCC_APB1PeriphClockCmd(RCC_APB1Periph_DAC, ENABLE);
// Включить таймер 6
RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM6,ENABLE);
// Настроить вывод для ЦАП
GPIO_InitTypeDef GPIO_InitStructure;
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_4;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
GPIO_Init(GPIOA, &GPIO_InitStructure);
// Настроить частоту таймера
TIM6->PSC = 0;
TIM6->ARR = 500;
TIM6->CR2=TIM_CR2_MMS_1; // Таймер назначить источником событий для ЦАП
TIM6->CR1 |= TIM_CR1_CEN; // Начать отсчет
// Включить DAC1
DAC->CR |= DAC_CR_TEN1; // Преобразование по возникновению события
DAC->CR &= ~DAC_CR_TSEL1; // от таймера 6
DAC->CR |= DAC_CR_WAVE1_0; // Генерация шума
// DAC->CR |= DAC_CR_WAVE1_1; // Генерация сигнала треугольной формы
DAC->CR |= DAC_CR_MAMP1; // Максимальная амплитуда
DAC->CR |= DAC_CR_EN1; // Включить ЦАП1
// Организовать бесконечный цикл
while(1) {}
}
```

Когда появится сигнал запуска канала 1, значение счетчика LFSR1 с той же маской добавляется в регистр DHR1 и сумма переносится в регистр DAC_DOR1 через три такта APB1. После этого счетчик LFSR1 обновляется.

Когда появится сигнал запуска канала 2, значение счетчика LFSR2 с той же маской добавляется в регистр DHR2 и сумма через три такта APB1 переносится в регистр DAC_DOR2. После этого счетчик LFSR2 обновляется.

НЕЗАВИСИМЫЙ ЗАПУСК С РАЗДЕЛЬНОЙ LFSR-ГЕНЕРАЦИЕЙ

Для настройки этого режима преобразования необходимы следующие действия:

- включить оба канала установкой разрядов TEN1 и TEN2;
- задать источник запуска каждого из каналов установкой разрядов TSEL1[2:0] и TSEL2[2:0];
- конфигурировать оба канала установкой разрядов WAVEx[1:0] в 01 и задать различные значения маски регистра LFSR с помощью разрядов MAMP1[3:0] и MAMP2[3:0];
- загрузить данные в нужный регистр DHR (DAC_DHR12RD,DAC_DHR12LD или DAC_DHR8RD).

Когда сформируется сигнал запуска канала 1, значение счетчика LFSR1 с маской, заданной в MAMP1[3:0], добавляется в регистр DHR1 и спустя три такта APB1 сумма переносится в регистр DAC_DOR1. После этого счетчик LFSR1 обновляется.

Когда придет сигнал запуска канала 2, значение счетчика LFSR2 с маской, заданной в MAMP2[3:0], добавляется в регистр DHR2 и через три такта APB1 сумма переносится в регистр DAC_DOR2. После чего счетчик LFSR2 обновляется.

НЕЗАВИСИМЫЙ ЗАПУСК С ГЕНЕРИРОВАНИЕМ ТРЕУГОЛЬНЫХ ИМПУЛЬСОВ

Для настройки этого режима преобразования необходимо выполнить следующие действия:

- включить оба канала установкой разрядов TEN1 и TEN2;
- задать источник запуска каждого из каналов установкой разрядов TSEL1[2:0] и TSEL2[2:0];
- сконфигурировать оба канала установкой разрядов WAVEx[1:0] в 10 и задать значение максимальной амплитуды сигнала разрядами MAMPx[3:0];
- загрузить данные в нужный регистр DHR (DAC_DHR12RD, DAC_DHR12LD или DAC_DHR8RD).

НЕЗАВИСИМЫЙ ЗАПУСК С ГЕНЕРИРОВАНИЕМ РАЗЛИЧНЫХ ТРЕУГОЛЬНЫХ ИМПУЛЬСОВ

Настройка этого режима будет произведена, если выполнить следующие действия:

- включить оба канала установкой разрядов TEN1 и TEN2;
- задать источник запуска каждого из каналов установкой разрядов TSEL1[2:0] и TSEL2[2:0];
- сконфигурировать оба канала установкой разрядов WAVEx[1:0] в 10 и задать различное значение максимальной амплитуды сигнала разрядами MAMP1[3:0] и MAMP2[3:0];
- загрузить данные в нужный регистр DHR (DAC_DHR12RD, DAC_DHR12LD или DAC_DHR8RD).

ОДНОВРЕМЕННЫЙ ПРОГРАММНЫЙ ЗАПУСК

Для настройки этого режима преобразования необходимо загрузить данные в нужный регистр DHR (DAC_DHR12RD, DAC_DHR12LD или DAC_DHR8RD). При такой конфигурации, спустя один такт APB1, данные из регистров DHR1 и DHR2 будут перезаписаны в DAC_DOR1 и DAC_DOR2 соответственно.

ПРИМЕРЫ ПРОГРАММ

Рассмотрим программы, позволяющие генерировать сигнал одной из трех форм: синус, треугольник и шум. Для генерации синуса использован табличный метод. То есть для генерации синуса значения не вычисляются, а извлекаются из подготовленного заранее массива и выводятся в ЦАП каждый раз, когда возникает прерывание от таймера. Вариант этой программы приведен в листинге 1.

После запуска этой программы на выводе PA4 будет сформирован аналоговый сигнал с формой синуса. При этом на сигнале будут наблюдаться «ступеньки». Чтобы сделать сигнал более гладким,

потребуется в несколько раз увеличить массив с предварительно вычисленными значениями синуса. Это позволит уменьшить разницу между соседними цифровыми значениями для формируемого сигнала.

Для генерации любой другой формы сигнала нужно просто заменить в массиве данных имеющиеся значения на значения для генерации нужного сигнала.

Рассмотрим программу для генерации сигнала шума. Здесь программа потребуется лишь для инициализации ЦАП и таймера, после чего формирование сигнала будет происходить без участия программы, то есть аппаратно. Для этого следует включить внешний запуск преобразования, а источником его запуска назначить таймер, который будет досчитывать до записанного в его регистр значения, а затем обнуляться и генерировать событие, которое запустит новое преобразование. И так до бесконечности. В результате на соответствующем выводе микроконтроллера будет сформирован непрерывный сигнал шума. Вариант такой программы приведен в листинге 2.

После запуска этой программы на экране осциллографа, подключенного к выводу PA4 микроконтроллера, появится характерный сигнал шума.

Для генерации сигнала треугольника необходимо закомментировать в программе строку:

```
DAC->CR |= DAC_CR_WAVE1_0; //
Генерация шума
```

и удалить символы комментирования для строки:

```
DAC->CR |= DAC_CR_WAVE1_1; // Генерация
сигнала треугольной формы.
```

В режиме генерации шума или треугольника можно задавать смещение сигнала относительно нуля. Это делается записью значения смещения в соответствующий каналу регистр DAC_DHRx.

Подробнее ознакомиться с блоком ЦАП микроконтроллера STM32 можно на сайте www.st.com [2].

Литература

1. www.st.com
2. www.st.com/web/en/resource/technical/document/reference_manual/CD00246267.pdf

CNY

* Статья перепечатана из журнала «Современная электроника», № 2, 2015 г., с разрешения редакции, тел. +7 (495) 232-00-87, www.soel.ru