

# Современные 32-разрядные ARM-микроконтроллеры серии STM32: сторожевые таймеры

Олег Вальпа (г. Миасс, Челябинская обл.)

**В статье приведено описание сторожевых таймеров 32-разрядных ARM-микроконтроллеров серии STM32 от компании STMicroelectronics. Рассмотрена архитектура и состав регистров сторожевых таймеров, а также приведены практические примеры программ.**

## Введение

Все микропроцессорные устройства подвержены влиянию различных помех, которые могут нарушить их работу и даже привести к «зависанию» программы. Это в свою очередь приводит к останову системы управления и может повлечь за собой крайне опасные ситуации.

Источниками помех могут являться различные коммутационные устройства в питающей сети, воздействие электромагнитных полей, атмосферные явления в виде разрядов молний и тому подобное. Защититься от подобных помех не всегда возможно, поэтому в случае « зависания» процессора необходимо автоматически восстановить его работу, например, путём перезапуска. Это позволит восстановить работу системы и обеспечит высокий уровень безопасности систем.

В своё время разработчики процессоров решили эту задачу с помощью сторожевых таймеров, получивших название «сторожевая собака». Идея решения заключается в том, что сторожевой таймер инициализируется и запускается самим процессором в начале работы

программы. После чего данный таймер самостоятельно отсчитывает заданное время, и если в течение этого времени процессор ни разу не перезагрузит его, то таймер обнуляется и перезапустит процессор. Таким образом, при нормальной работе процессора таймер будет регулярно перезапускаться, а в случае « зависания» процессора таймер через заданное время сформирует сигнал сброса для процессора и перезапустит его.

Микроконтроллеры серии STM32 [1] имеют два таких таймера. Один из них называется «независимый сторожевой таймер» (Independent Watchdog, IWDG), а другой – «оконный сторожевой таймер» (Window Watchdog, WWDG). Рассмотрим каждый из них отдельно.

## Независимый сторожевой таймер

Таймер IWDG тактируется специализированным низкочастотным сигналом LSI, благодаря чему он продолжает работать, даже если пропадает системный тактовый сигнал. Таймер осуществляет обратный отсчёт от заданного значения до нуля. Сброс процессора тай-

мер производит при достижении счётчиком таймера нуля.

Этот таймер лучше всего подходит для тех приложений, которым необходимо, чтобы сторожевой таймер был запущен как полностью независимый процесс, вне основного приложения, но у которого были бы не слишком высокие требования к временным параметрам. Функциональная схема независимого сторожевого таймера IWDG приведена на рисунке 1.

Таймер IWDG запускается путём записи значения 0xCCCC в ключевой регистр IWDG\_KR. Счётчик таймера начинает обратный отсчёт от значения, заданного в регистре перезагрузки IWDG\_RLR. По умолчанию этот регистр имеет значение 0xFFFF. Когда счётчик досчитает до нуля, формируется сигнал сброса для процессора от таймера IWDG.

Каждый раз, когда в ключевой регистр IWDG\_KR записывается значение 0xAAAA, данные из регистра IWDG\_RLR перегружаются в счётчик, что возобновляет отсчёт и предотвращает сброс процессора от IWDG.

Если с помощью соответствующих битов конфигурации разрешена функция «аппаратный сторожевой таймер», то сторожевой таймер автоматически запускается после подачи питания.

Регистр предварительного делителя IWDG\_PR позволяет программно задать коэффициент деления тактовой частоты для сторожевого таймера IWDG.

По умолчанию регистры IWDG\_PR и IWDG\_RLR защищены от записи. Прежде чем изменить их значения, необходимо записать код 0x5555 в регистр IWDG\_KR. Запись в ключевой регистр любого другого значения снова заблокирует запись в регистры IWDG\_PR и IWDG\_RLR.

Период таймера вычисляется при помощи формулы:

$$T = ((4 * 2^{IWDG\_PR}) * IWDG\_RLR) / 40 \text{ [мс].}$$

Определить период отсчёта сторожевого таймера для тактовой частоты 40 кГц можно также с помощью таблицы 1.

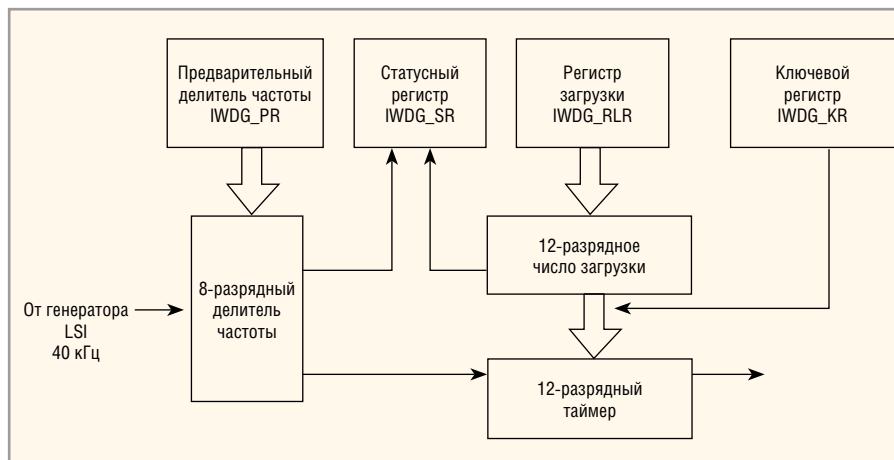


Рис. 1. Функциональная схема сторожевого таймера IWDG

В связи с нестабильностью внутреннего RC генератора микроконтроллера, указанные в таблице значения времени могут незначительно отличаться от реальных.

Регистр статуса IWDG\_SR позволяет определить, какие операции выполняет таймер в текущий момент.

Когда микроконтроллер входит в режим отладки и его ядро останавливается, счётчик IWDG либо продолжает работать, либо останавливается, в зависимости от конфигурационного бита DBG\_IWDG\_STOP в модуле DBG.

#### Регистры независимого сторожевого таймера

Карта регистров сторожевого таймера IWDG приведена в таблице 2.

Ключевой регистр IWDG\_KR имеет 16 бит для ключа KEY[15:0]. В эти биты программа должна регулярно записывать ключевое значение 0xAAAA, иначе сторожевой таймер произведёт сброс, когда счётчик достигнет значения 0. Запись ключевого значения 0x5555 в этот регистр разрешает доступ к регистрам IWDG\_PR и IWDG\_RLR. Запись ключевого значения 0xCCCC запуска-

Таблица 1. Минимальное и максимальное значения периода сторожевого таймера

Предварительный делитель	Значение регистра IWDG_PR[2:0], биты	Минимальное значение для RL[11:0]=0x000, мс	Максимальное значение для RL[11:0]=0xFFFF, мс
4	0	0,1	409,6
8	1	0,2	819,2
16	2	0,4	1638,4
32	3	0,8	3276,8
64	4	1,6	6553,6
128	5	3,2	13107,2
256	6 или 7	6,4	26214,4

ет сторожевой таймер, кроме случая, когда он уже запущен в режиме «аппаратный сторожевой таймер».

Регистр предварительного делителя IWDG\_PR содержит три бита PR[2:0], которые задают коэффициент деления частоты сторожевого таймера. Изменить значение IWDG\_PR можно только тогда, когда бит PVU регистра IWDG\_SR сброшен. Соответствие между значением бит PR[2:0] регистра IWDG\_PR и коэффициентом деления частоты сторожевого таймера приведены в таблице 1.

При чтении этого регистра его значение может быть некорректным, если операция чтения производится тогда, когда предыдущая операция запи-

си в этот регистр ещё не завершилась. По этой причине значение, прочитанное в этом регистре, действитель но только тогда, когда бит PVU в регистре IWDG\_SR сброшен.

Регистр перезагрузки IWDG\_RLR имеет 12 бит RL[11:0], которые позволяют записать в него значение от 0 до 0xFFFF. Это значение будет перезагружаться в счётчик сторожевого таймера каждый раз при записи значения 0xAAAA в регистр ключа IWDG\_KR. Период таймера вычисляется на основе этого значения и тактовой частоты после предварительного делителя. Для того чтобы изменить или прочитать значение регистра перезагрузки, необходимо



[www.rfhic.com](http://www.rfhic.com)

#### СВЧ-усилители на основе GaN компании RFHIC – новые возможности для разных применений

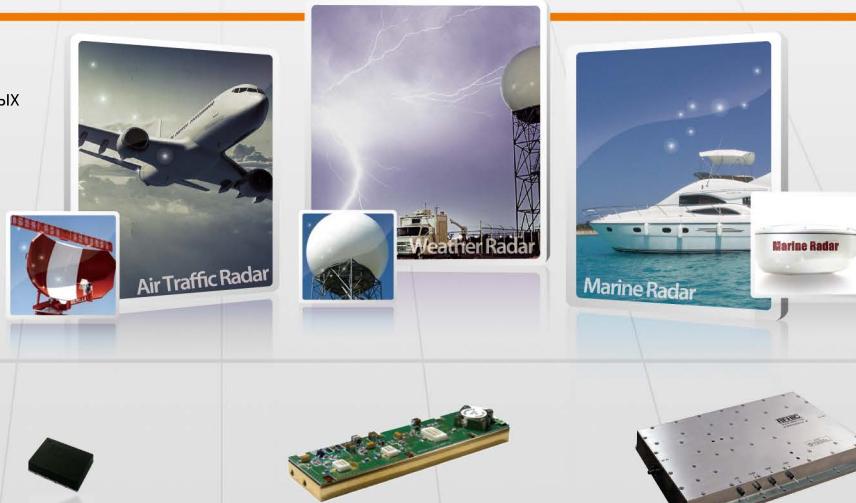
Компания RFHIC предоставляет самые эффективные решения на основе твердотельных усилителей мощности для метеорологических, морских и воздушных РЛС

RFHIC – оптимальный выбор по цене и качеству



- Усилитель высокой мощности на основе GaN

1–2 кВт



- SMD/SMT GaN-усилитель для ФАР

2,7...2,9 Гц  
50 Вт      9,2...9,5 Гц  
25 Вт

- Платный усилитель мощности

2,7...2,9 Гц  
200–400 Вт      5,6...5,7 Гц  
200 Вт      9,2...9,5 Гц  
50–100 Вт

- GaN-модуль

9,3...9,5 Гц  
200 Вт

ОФИЦИАЛЬНЫЙ ДИСТРИБЬЮТОР ПРОДУКЦИИ RFHIC

**PROCHIP**  
POWERED BY PROSOFT

Активный компонент вашего бизнеса

ТЕЛ.: (495) 232-2522 / ФАКС: (495) 234-0640 / [INFO@PROCHIP.RU](mailto:INFO@PROCHIP.RU) / [WWW.PROCHIP.RU](http://WWW.PROCHIP.RU)



Реклама

**Листинг 1**

```
// Функция инициализации сторожевого таймера IWDG
void Init_IWDG(u16 tw) // Параметр tw от 7мс до 26200мс
{
    // Для IWDG_PR=7 Tmin=6,4мс RLR=Tmc*40/256
    IWDG->KR=0x5555; // Ключ для доступа к таймеру
    IWDG->PR=7; // Обновление IWDG_PR
    IWDG->RLR=tw*40/256; // Загрузить регистр перезагрузки
    IWDG->KR=0xAAAA; // Перезагрузка
    IWDG->KR=0xCCCC; // Пуск таймера
}

// Функция перезагрузки сторожевого таймера IWDG
void IWDG_res(void)
{
    IWDG->KR=0xAAAA; // Перезагрузка
}
```

убедиться, что бит RVU регистра IWDG\_SR сброшен.

Регистр статуса IWDG\_SR имеет два информационных бита.

Бит 1 RVU устанавливается аппаратно и указывает, что идёт процесс обновления значения перезагрузки. Очищается он аппаратно, когда операция обновления значения перезагрузки завершена. Данная операция занимает до 5 циклов RC генератора на частоте 40 кГц. Значение регистра перезагрузки можно обновлять только при сброшенном бите RVU.

Бит 0 PVU устанавливается аппаратно и указывает, что идёт процесс обновления значения предварительного делителя. Очищается он аппаратно, когда операция обновления значения предварительного делителя закончена. Данная операция также занимает до 5 циклов RC генератора на частоте 40 кГц. Значение предварительного делителя можно обновлять только при сброшенном бите PVU.

Если приложение использует несколько значений перезагрузки или предварительного делителя, то, прежде

чем изменить значение перезагрузки, необходимо ждать, пока не будет сброшен бит RVU. Также, прежде чем изменить значение предварительного делителя, следует ждать, пока не будет сброшен бит PVU.

Однако, после обновления значения предварительного делителя или перезагрузки, необходимости анализировать состояние бит RVU и PVU, прежде чем продолжить выполнение кода, нет. Операция записи будет принята к исполнению и завершена даже в случае перехода в режим низкого потребления.

### **Примеры программ для независимого сторожевого таймера**

В листинге 1 приведён пример функций для работы с таймером IWDG. Комментарии позволяют понять назначение всех операций, выполняемых в этой программе.

Используя данные функции, задействовать в программе сторожевой таймер IWDG и осуществлять управление им не составит труда.

### **Оконный сторожевой таймер**

Таймер WWDG лучше всего подходит для тех приложений, которые требуют, чтобы сторожевой таймер реагировал на перезапуск в пределах точного промежутка времени, то есть окна времени.

Таймер тактируется сигналом, полученным делением сигнала тактовой частоты PCLK1 с шины APB1. Он имеет конфигурируемое окно времени, которое служит для перезагрузки сторожевого таймера, и может быть задан так, чтобы обнаруживать неправильное поведение программы, в виде запаздывания или опрежения в работе.

Обычно этот сторожевой таймер используется для того, чтобы обнаруживать возникновение некорректной работы программного обеспечения, вызванное внешним вмешательством или непредвиденными логическими условиями, которые заставляют прикладную программу отклоняться от своего нормального алгоритма.

Функциональная схема оконного сторожевого таймера WWDG приведена на рисунке 2. Фактически оконный сторожевой таймер является расширенной версией традиционного встраиваемого сторожевого таймера.

Оконный сторожевой таймер представляет собой 7-битный вычитающий счётчик, тактируемый сигналом PCLK1, поделённым на 4096 с помощью аппаратного 12-битного предварительного делителя и программно управляемого делителя от 1 до 8.

Оконный сторожевой таймер активируется путём установки бита WDGA в регистре WWDG\_CR. С этого момента его нельзя отключить, за исключением общего сброса. После активизации сторожевой таймер начинает счёт в обратном направлении и генерирует сброс при изменении состояния счёт-

**Таблица 2. Карта регистров IWDG и их значения после сброса**

Сдвиг	Регистр	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	IWDG_KR	Резерв	KEY[15:0]																								PR[2:0]		0		0		
	Исх. значение		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x04	IWDG_PR	Резерв	RL[11:0]																								0		0		0		
	Исх. значение		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
0x08	IWDG_RLR	Резерв	RVU																								PVU		0		0		
	Исх. значение		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
0x0C	IWDG_SR	Резерв	RVU																								PVU		0		0		
	Исх. значение		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

чика с 0x40 на 0x3F, то есть при обнулении шестого разряда таймера, условно обозначаемого как T6. Если программа перегружает счётчик, когда его значение больше, чем значение в регистре окна времени, это также инициирует сброс.

Чтобы предотвратить сброс процессора, прикладная программа, во время нормальной операции, должна регулярно производить запись в регистр WWDG\_CR. Эта операция должна производиться только тогда, когда значение счётчика меньше, чем значение в регистре окна времени. Значение, которое можно записать в регистр WWDG\_CR, должно быть в пределах 0xFF...0x40.

Счётчик таймера WWDG является автономным. Он всегда производит обратный отсчёт. Даже тогда, когда сторожевой таймер отключён. При включении сторожевого таймера необходимо установить бит T6 счётчика, чтобы предотвратить немедленный сброс процессора. Биты T[5:0] содержат число, которое представляет собой временнюю задержку до сброса от WWDG.

Регистр конфигурации WWDG\_CFR содержит верхний предел окна времени. Чтобы предотвратить сброс, обратный счётчик должен быть перезагружен тогда, когда его значение меньше, чем значение регистра окна времени, но больше 0x3F.

На рисунке 3 представлена временная диаграмма, демонстрирующая оконный процесс работы сторожевого таймера.

Другой способ перезагрузить счётчик состоит в том, чтобы использовать прерывание от раннего пробуждения. Это прерывание разрешается установкой бита EWI в регистре WWDG\_CFR. Когда обратный счётчик достигает значения 0x40, генерируется прерывание от раннего пробуждения, и можно использовать соответствующий обработчик, чтобы перезагрузить счётчик и предотвратить сброс от WWDG.

Запрос этого прерывания очищается записью нуля в разряд EWIF регистра WWDG\_SR.

При необходимости можно сгенерировать программный сброс процессора путём установки в единичное состояние бита WDGA и сбросив бит T6.

В таблице 3 приведён диапазон значений периода таймера WWDG для тактовой частоты сигнала PCLK равной 24 МГц, в зависимости от установленных значений регистров таймера.

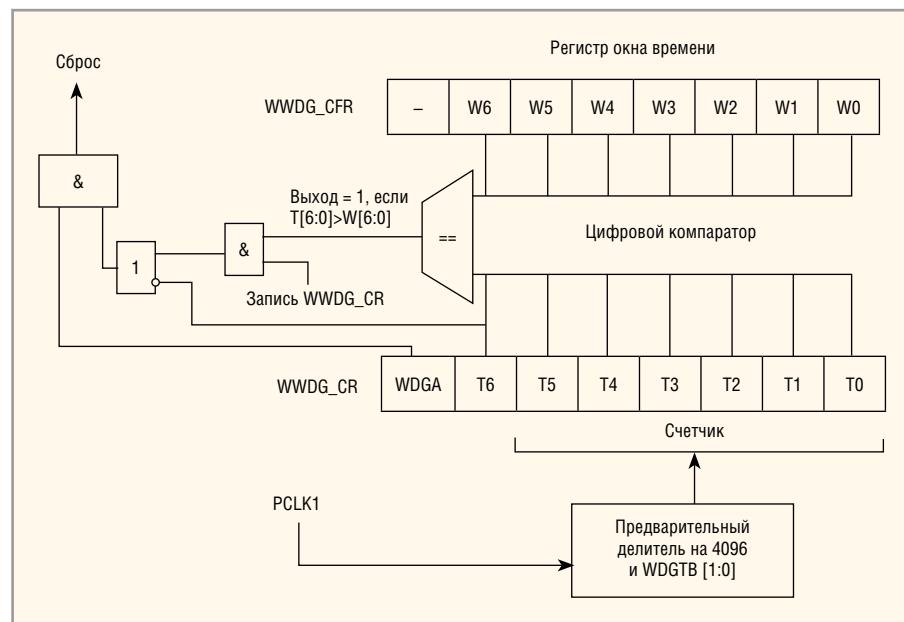


Рис. 2. Функциональная схема сторожевого таймера WWDG

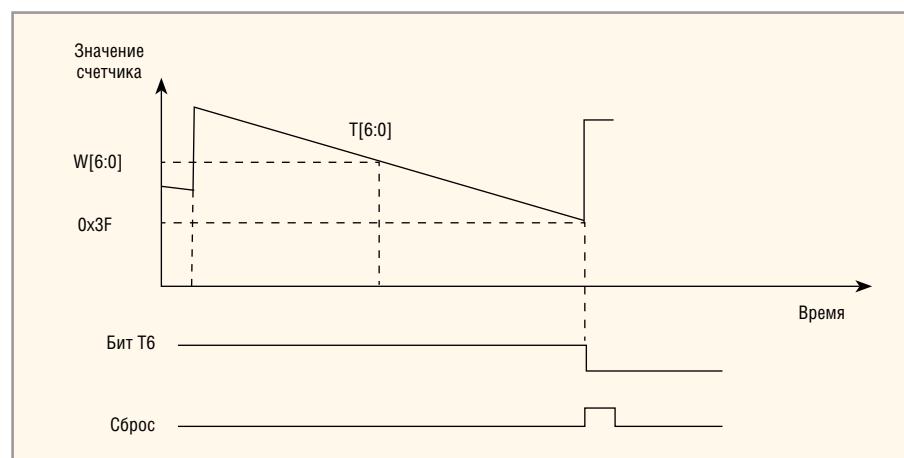


Рис. 3. Временная диаграмма сторожевого таймера WWDG

Таблица 3. Диапазон значений периода таймера WWDG

Предварительный делитель	Значение WDGTB[1:0], биты	Минимальный период для T[5:0]=0, мкс	Максимальный период для T[5:0]=0x3F, мкс
1	0	170	10920
2	1	341	21850
4	2	682	43690
8	3	1365	87380

Для более точного расчёта периода необходимо использовать следующую формулу:

$$T_{wwdg} = T_{pclk1} * 4096 * 2^{WDGTB} * (T[5:0]+1).$$

Чтобы избежать немедленного сброса процессора, при записи данных в регистр WWDG\_CR необходимо устанавливать бит T6 в единичное состояние.

Когда микроконтроллер входит в режим отладки, счётчик WWDG будет продолжать работать или останавливаться

ется, в зависимости от конфигурационного бита DBG\_WWDG\_STOP в модуле DBG.

### Регистры оконного сторожевого таймера

Сторожевой таймер WWDG имеет три регистра. Карта этих регистров приведена в таблице 4.

Регистр управления WWDG\_CR содержит бит WDGA для активации таймера и 7-разрядный счётчик T[6:0]. Бит WDGA устанавливается программно, а очищается аппаратно, только путём общего сброса. Когда WDGA равен 1,

сторожевой таймер включён и может производить сброс. С появлением каждого тактового импульса счётчик T[6:0] уменьшает своё значение на единицу. Сброс генерируется, когда счётчик меняет значение с 0x40 на 0x3F, то есть при обнулении разряда T6.

Регистр конфигурации WWDG\_CFR содержит три бита для настройки таймера и 7 разрядов W[6:0] для установки окна времени.

Разряд 9 EWI служит для разрешения прерывания раннего пробуждения процессора. Если он установлен, то

при достижении счётчиком значения 0x40 генерируется прерывание. Обнуляется этот бит аппаратно, путём общего сброса микроконтроллера.

Биты 8 и 7 группы WDGTB[1:0] назначают коэффициент дополнительного деления частоты входного сигнала счётчика. Они могут иметь следующие значения:

- 00 – деление на 1;
- 01 – деление на 2;
- 10 – деление на 4;
- 11 – деление на 8.

Разряды W[6:0] организуют 7-битное значение окна времени, с которым сравнивается значение обратного счётчика.

Регистр статуса WWDG\_SR содержит всего один бит статуса EWIF, представляющий собой флаг прерывания пробуждения процессора. Этот бит принимает единичное значение аппаратно, когда счётчик достигает значения 0x40 и тогда, когда прерывание выключено.

Очищается он программно, записью в него нулевого значения. Запись в этот разряд значения 1 эффекта не имеет.

## Примеры программ для оконного сторожевого таймера

В листинге 2 приведены функции для работы с таймером WWDG с комментариями, позволяющими понять назначение всех операций, выполняемых в этой программе.

С помощью таких несложных функций можно организовать контроль работы программы, обеспечивая тем самым защиту от сбоев. Подробнее ознакомиться с таймерами микроконтроллера STM32 можно на сайте STMicroelectronics [2].

## ЛИТЕРАТУРА

1. [www.st.com](http://www.st.com).
2. [www.st.com/web/en/resource/technical/document/reference\\_manual/CD00246267.pdf](http://www.st.com/web/en/resource/technical/document/reference_manual/CD00246267.pdf).



### Листинг 2

```

// Функция инициализации сторожевого таймера WWDG
void WWDG_Init(void)
{
    // Инициализация прерывания
    NVIC->ISER[0] |= NVIC_ISER_SETENA_0;
    // Инициализация таймера WWDG
    RCC->APB1ENR |= RCC_APB1ENR_WWDGEN; // Включить тактирование WWDG
    WWDG->CFR |= WWDG_CFR_WDGTB0; // Задать частоту тактирования WWDG
    WWDG->CFR |= 0x41; // Задать окно
    WWDG->CR |= 0x7F; // Разрешить работу WWDG и установить начальное
    // значение
    WWDG->SR &= ~WWDG_SR_EWIF; // Очистить флаг EWI
    WWDG->CFR |= WWDG_CFR_EWI; // Разрешить прерывания
}

// Функция обработки прерывания сторожевого таймера WWDG
void WWDG_IRQHandler(void)
{
    WWDG->CR |= 0x7F; // Обновить значение таймера WWDG
    WWDG->SR &= ~WWDG_SR_EWIF; // Очистить флаг EWI
    // Другие действия ...
}

// Функция запуска сторожевого таймера WWDG
void WWDT_ON(void)
{
    RCC->APB1ENR |= RCC_APB1ENR_WWDGEN; // Включить тактирование WWDG
    WWDG->CR |= 0x7F; // Обновить значение таймера WWDG
    WWDG->SR &= ~WWDG_SR_EWIF; // Очистить флаг EWI
}

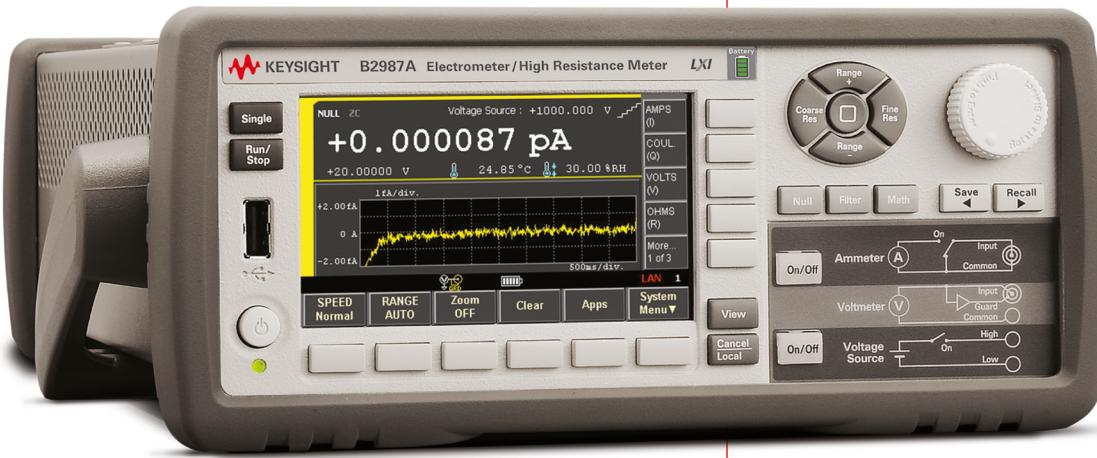
// Функция останова сторожевого таймера WWDG
void WWDT_OFF(void)
{
    RCC->APB1ENR |= RCC_APB1ENR_WWDGRST; // Отключить тактирование WWDG
}

```

Таблица 4. Карта регистров WWDG и их значения после сброса

Сдвиг	Регистр	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	WWDG_CR	Резерв															WDGA	T[6:0]																
	Исх. значение																	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x04	WWDG_CFR	Резерв															EWI	WDGTB1	WDGTB0															
	Исх. значение																	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x08	WWDG_SR	Резерв															EWIF																	
	Исх. значение																																0	

# Новый взгляд на Фемтоамперметр/Петаомметр



## ДОСТОВЕРНЫЕ ИЗМЕРЕНИЯ СИЛЫ ТОКА ОТ 0,01x10<sup>-15</sup> А И СОПРОТИВЛЕНИЯ ДО 10x10<sup>15</sup> ОМ

Измерители Keysight серии B2980A имеют широкие функциональные возможности, которые обеспечивают уверенность в результатах измерений и способствуют лучшему пониманию происходящих процессов. Благодаря встроенной аккумуляторной батареи, позволяющей устраниить влияние шумов сети переменного тока, можно обнаружить и измерить реальные сигналы, которые раньше были невидимы. Функция построения гистограмм в режиме реального времени и возможность отображения сигналов во временной области позволит вам не пропустить критически важные нюансы поведения сигнала. Программа контроля состояния схемы упрощает выявление неисправностей. Воспользуйтесь возможностями самого точного в мире фемтоамперметра/петаомметра с графическим интерфейсом для повышения достоверности измерений.

**HARDWARE + SOFTWARE + PEOPLE = INSIGHTS**

### Фемтоамперметр/Петаомметр Keysight серии B2980A

Режим работы от аккумуляторной батареи: есть

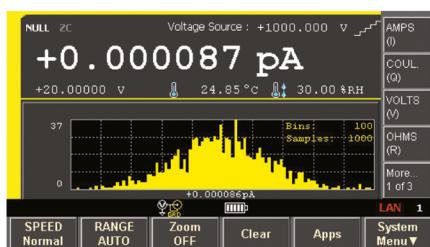
Функция построения гистограмм в режиме реального времени и представление сигналов во временной области

Измерение силы тока: от 0,01 фА до 20 мА

Измерение сопротивления: до 10 ПОм

Встроенный источник напряжения: до ±1000 В

Скорость измерений: до 20 000 отсчетов/с



Режим построения гистограммы для детального отображения сигнала

Посмотрите видеоролик  
[www.keysight.com/find/b2980a\\_info](http://www.keysight.com/find/b2980a_info)

8 800 500 9286 (звонок по России бесплатный)

© Keysight Technologies, Inc. 2014

**KEYSIGHT**  
TECHNOLOGIES

Unlocking Measurement Insights

Реклама

Группа электронных измерений Agilent – теперь Keysight Technologies