

Современные 32-разрядные ARM-микроконтроллеры серии STM32: универсальный последовательный порт USART

Олег Вальпа (г. Миасс, Челябинская обл.)

В статье приведено описание универсального последовательного порта USART 32-разрядных ARM-микроконтроллеров серии STM32 от компании STMicroelectronics. Рассмотрена архитектура, состав и назначение регистров конфигурирования порта, а также приведены примеры программ инициализации.

ВВЕДЕНИЕ

Наряду с портами ввода-вывода GPIO универсальный синхронно-асинхронный последовательный приёмопередатчик USART (Universal Synchronous Asynchronous Receiver Transmitter) является одним из самых востребованных портов для связи любого микроконтроллера с внешними устройствами. С помощью порта USART можно легко организовать связь микроконтроллера с компьютером, провести его программирование, а также связать микроконтроллеры между собой по интерфейсам RS-232, RS-485, RS-422 и т.д.

Преимуществами порта USART являются простота в использовании, а также, в отличие от интерфейса USB, гибкость в настройке и надёжность работы.

ОПИСАНИЕ ПОРТОВ USART

Микроконтроллеры STM32 [1] имеют обычно несколько портов USART. Например, STM32F103RBT имеет три последовательных порта, обозначаемых USART1, USART2 и USART3.

Кроме обычного асинхронного режима работы с использованием сигналов RXD и TXD, порты USART STM32 поддерживают несколько расширенных режимов работы. В отличие от стандартных портов USART они могут:

- работать с однопроводной полудуплексной линией связи;
- поддерживать интерфейсы Smartcard стандарта ISO7618-3, LIN (local interconnection network) и IrDA (infrared data association);
- связываться с внешними устройствами, оснащёнными SPI-совместимым интерфейсом, по 3-проводной линии.

К особенностям порта USART микроконтроллеров STM32 также относится возможность дробного деления такто-

вой частоты для формирования заданной скорости работы. Благодаря этому можно получить стандартные скорости связи порта при любой частоте тактового сигнала.

Кроме того, с помощью блока DMA (Direct Memory Access) для любого порта USART может быть организован прямой доступ к памяти, как для приёма, так и для передачи данных.

Порты USART STM32 способны поддерживать скорость обмена до 4,5 Мбит/с. Формат слова USART может составлять 8 или 9 бит данных и 0,5; 1; 1,5 или 2 стоповых бит. Дробные значения стоповых бит применяются в режиме порта Smartcard. Первым передаётся и принимается младший бит данных.

Некоторые порты USART STM32 можно программно переназначать на другие выходы микроконтроллера.

На рисунке 1 приведена структурная схема одного порта USART, которая помогает понять изложенное выше описание, а также даёт представление об архитектуре порта USART и составе регистров для его инициализации.

РЕГИСТРЫ ПОРТОВ USART

В микроконтроллерах STM32 для настройки каждого порта USART и работы с ним имеется по 7 регистров:

1. USART_SR – регистр статуса, указывающий на состояние порта USART;
2. USART_DR – регистр данных для записи передаваемых и чтения принимаемых данных;
3. USART_BRR – регистр, определяющий скорость обмена;
4. USART_CR1 – первый управляющий регистр;
5. USART_CR2 – второй управляющий регистр;
6. USART_CR3 – третий управляющий регистр;

7. USART_GTPR – регистр делителя и задержки.

Формат регистров с названием входящих в них разрядов представлен в таблице 1.

Передача и приём информации через порт USART производится по кадрам, как показано на рисунке 2.

Каждый кадр начинается со стартового бита, после которого следует несколько бит информации, бит паритета, если он задан, и стоповые биты.

Рассмотрим поочередно назначение и содержимое разрядов этих регистров.

РЕГИСТР USART_SR

Регистр статуса USART_SR хранит разряды, отображающие текущее состояние порта USART.

Разряд 0 PE отображает ошибку паритета при приёме байта. Данный разряд устанавливается в единичное состояние при аппаратном выявлении ошибки паритета, т.е. несовпадении числа единиц в принимаемом байте с ожидаемым чётным или нечётным количеством, заданным для контроля.

Разряд 1 FE указывает на ошибку кадра при приёме байта. Ошибкой кадра является, например, отсутствие стопового бита или несоответствие их количества заданному при настройке порта значению. Данный разряд также устанавливается аппаратно.

Разряд 2 NE фиксирует наличие шума в кадре. Шумом считается изменение состояния линии связи в середине приёма очередного бита информации.

Разряд 3 ORE отображает ошибку переполнения приёмного буфера.

Разряд 4 IDLE фиксирует обнаружение кадра ожидания. Кадром ожидания является отсутствие очередного стартового бита после стоповых бит.

Следующая группа флагов позволяет определить окончание передачи или приёма данных.

Разряд 5 RXNE фиксирует приём данных в регистр USART_DR.

Разряд 6 TC сигнализирует о завершении передачи очередного байта данных, записанных в регистр данных USART_DR.

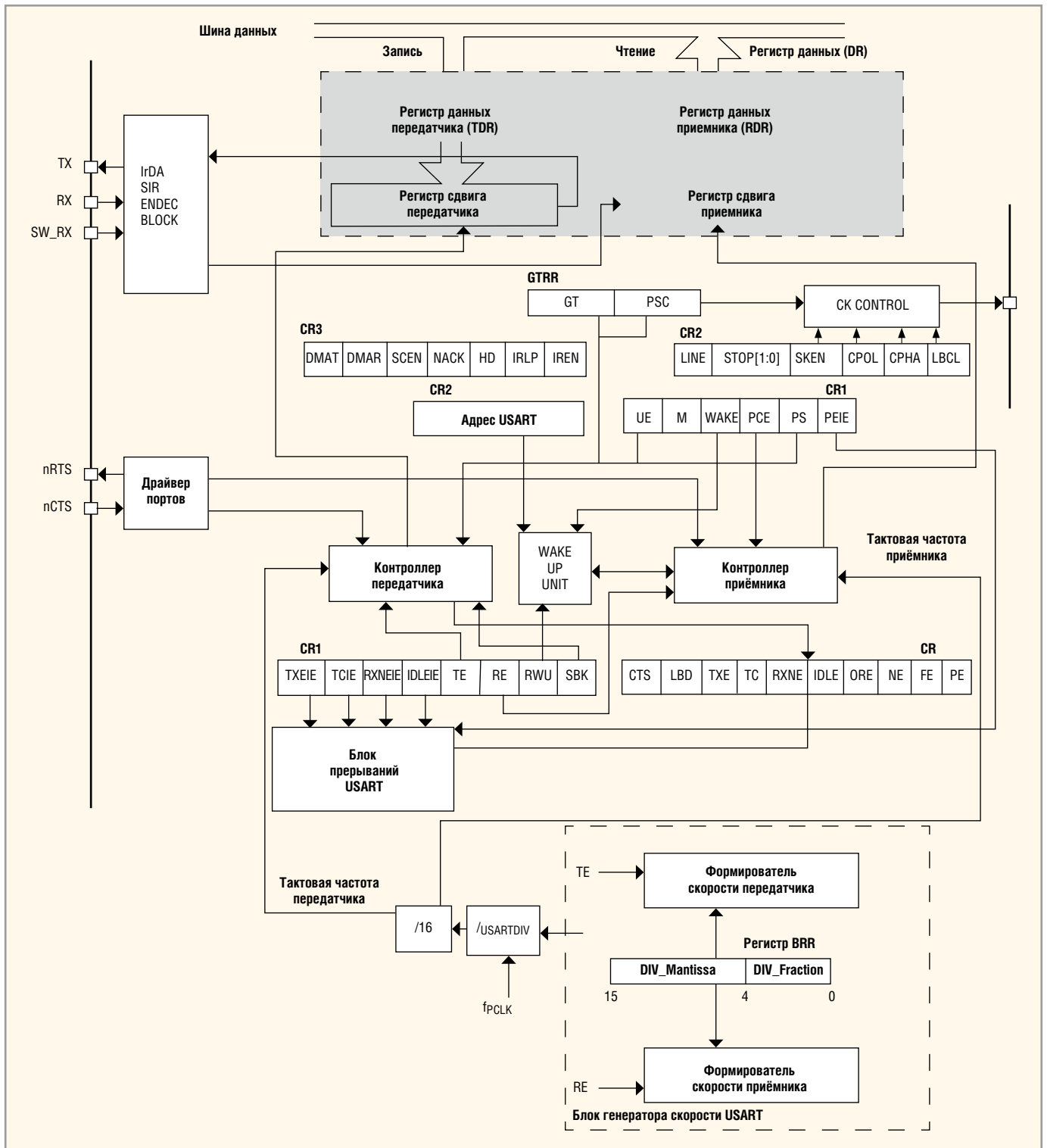


Рис. 1. Структурная схема порта USART

Разряд 7 TXE сообщает о том, что сдвиговый регистр передачи данных пуст.

Порт USART передаёт данные из сдвигового регистра, в который они поступают из регистра USART_DR. После записи данных в USART_DR они находятся в нём до тех пор, пока сдвиговый регистр не освободится. Как только данные передаются, сдвиговый регистр освобождается, и в него переписывается информация из USART_DR. При

этом устанавливается флаг TXE, означающий, что регистр данных пуст. Если же очередные данные не будут записаны в регистр данных USART_DR, то после освобождения сдвигового регистра будет установлен также флаг TC.

Разряд 8 LBD фиксирует обнаружение разрыва связи.

Разряд 9 CTS отображает изменение состояния сигнала CTS.

Разряды 10...31 зарезервированы для будущих версий микроконтроллеров.

Очищаются все разряды программно, последовательностью операций чтения регистра USART_SR, с последующим чтением регистра USART_DR.

РЕГИСТР USART_DR

Регистр USART_DR содержит байт принятых или передаваемых данных. Несмотря на то, что в программе производится обращение к одному и тому же регистру USART_DR как для приёма данных, так и для их переда-

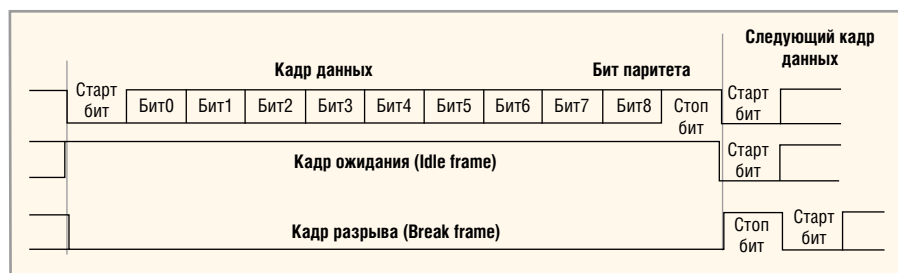


Рис. 2. Формат кадра порта USART

чи, на самом деле при чтении информация поступает из регистра приёмника порта USART, а при записи данные заносятся в регистр передатчика порта USART.

Для работы с регистром данных используются следующие операторы:
 USART1->DR = tx1; // Передать байт переменной tx1 через USART
 rx1 = USART1->DR; // Считать принятый байт из USART в переменную rx1

Регистр USART_BRR

Регистр USART_BRR служит для задания скорости обмена через порт USART и содержит определяющую эту скорость переменную USARTDIV с фиксированной точкой. Данная переменная состоит из двух частей: целой 12-разрядной части DIV_mantissa и дробной 4-разрядной части DIV_fraction. Дробная часть позволяет довольно точно подстроить скорость обмена для любой частоты тактового сигнала.

Скорость обмена вычисляется по формуле:

$$baud = \frac{fck}{16 \times USARTDIV}$$

где *fck* – тактовый сигнал (PCLK2 для USART1 и PCLK1 для USART2, 3). Соответственно, значение переменной

$$USARTDIV = \frac{fck}{16 \times baud}$$

Рассмотрим пример вычисления значения переменной USARTDIV для скорости порта USART, равной 9600 бод с тактовой частотой 8 МГц.

$$USARTDIV = \frac{8000000}{16 \times 9600} = 52,0833$$

Целая часть USARTDIV имеет значение 52, а в 16-ричном формате 0x34. Для перевода дробной части 0,0833 в 16-ричный формат, необходимо умножить эту часть на 16 и округлить полученное значение до целого. Таким образом, получим: 0,0833 × 16 = 1,3328. После округления и перевода этого значения в 16-ричный формат получим число 0x01. Теперь, поместив обе части вычисляемого числа на свои разряды в соответствии с таблицей 1, получим значение USARTDIV = 0x0341, которое следует записать в регистр задания скорости USART_BRR.

Такая запись осуществляется с помощью следующих операторов:

```
USART1->BRR = 0x0341; // Записать значение скорости для USART1
```

При округлении дробной части возникла погрешность, которую можно вычислить с помощью обратного преобразования. Для этого необходимо преобразовать полученное значение в десятичное представление. В нашем случае целая часть равна 0x34 = 52, а дробная часть равна 1/16 = 0,0625 и USARTDIV = 52,0625 вместо расчётного значения 52,0833.

Ошибка в процентном выражении составит: error = (52,0833 – 52,0625) / 52,0833 × 100% = 0,04%. Такая ошибка не превышает предельно допустимое значение, составляющее 5%, и вполне приемлема для нормальной работы порта USART.

Теперь рассмотрим разряды трёх управляющих регистров: USART_CR1, USART_CR2 и USART_CR3.

Регистр USART_CR1

Регистр USART_CR1 имеет следующие разряды:

- разряд 0 SBK управляет отправкой символа разрыва связи (0 – не посылать, 1 – посылать).
- разряд 1 RWU определяет режим приёмника USART (0 – приёмник активен, 1 – приёмник в «спящем» режиме);
- разряд 2 RE разрешает работу приёмника;
- разряд 3 TE разрешает работу передатчика;

Таблица 1. Формат регистров порта USART

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	USART_SR	Резерв												CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE																					
	Reset value													0	0	1	1	0	0	0	0	0	0																					
0x04	USART_DR	Резерв												DR[8:0]																														
	Reset value													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			
0x08	USART_BRR	Резерв												DIV_Mantissa[15:4]										DIV_Fraction [3:0]																				
	Reset value													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		
0x0C	USART_CR1	Резерв												UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK																	
	Reset value													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	0	
0x10	USART_CR2	Резерв												LINEN	STOP [1:0]	CLKEN	CPOL	CPHA	LBCL	Резерв	LBDE	LBOL	Резерв	ADD[3:0]																				
	Reset value													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	0	0
0x14	USART_CR3	Резерв												STSE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSSEL	IRLP	IREN	EIE																				
	Reset value													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	0	0
0x18	USART_GTPR	Резерв												GT[7:0]							PSC[7:0]																							
	Reset value													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	0	0

- разряд 4 IDLEIE разрешает прерывания от флага IDLE регистра USART_SR;
 - разряд 5 RXNEIE разрешает прерывания от флага RXNE регистра USART_SR;
 - разряд 6 RXNEIE разрешает прерывания от флага RXNE регистра USART_SR, когда в регистр данных перемещён принятый байт;
 - разряд 7 TXEIE разрешает прерывания от флага TXE регистра USART_SR, т.е. когда регистр передачи пуст;
 - разряд 8 PEIE разрешает прерывание при обнаружении ошибки паритета;
 - разряд 9 PS управляет типом паритета (0 – чётный, 1 – нечётный);
 - разряд 10 PCE разрешает контроль паритета (0 – отключён, 1 – включён);
 - разряд 11 WAKE определяет метод «пробуждения» порта (0 – по состоянию линии связи, 1 – по выделению адреса);
 - разряд 12 M определяет длину послыки данных (0 – 8 бит данных, 1 – 9 бит данных);
 - разряд 13 UE разрешает работу USART (0 – отключён, 1 – включён);
 - разряды 14–31 зарезервированы.
- Для наглядности в таблице 2 приведён формат кадра данных USART в зависимости от установки разрядов M и PCE.

РЕГИСТР USART_CR2

Ниже представлено назначение разрядов регистра USART_CR2:

- разряды 3...0 ADD[3:0] задают адрес узла USART для многопроцессорной связи с целью его пробуждения при обнаружении данного адреса;
- разряд 4 зарезервирован и всегда имеет нулевое состояние;
- разряд 5 LBDL определяет длину обнаружения разрыва LIN (0 – 10 бит, 1 – 11 бит);
- разряд 6 LBDIE разрешает прерывание от флага LBD в регистре USART_SR;
- разряд 7 зарезервирован и всегда имеет нулевое состояние;
- разряд 8 LBCL управляет последним синхроимпульсом порта в режиме SPI (0 – не влияет, 1 – определяет синхроимпульс);
- разряд 9 CPHA определяет фазу синхроимпульсов порта в режиме SPI (0 – по фронту, 1 – по спаду);
- разряд 10 CPOL определяет полярность синхроимпульсов порта в режиме SPI (0 – пассивен низкий уровень, 1 – пассивен высокий уровень);

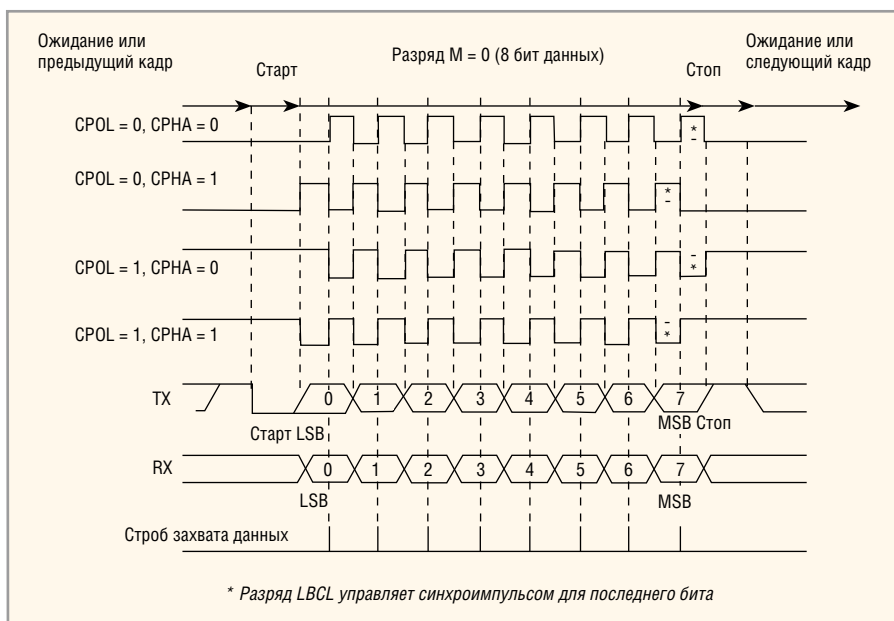


Рис. 3. Варианты настройки порта USART в режиме SPI

- разряд 11 CLKEN активирует выход СК (0 – пассивен, 1 – активен);
 - разряды 13 и 12 STOP определяют формат стоп-битов (00 – 1 стоп-бит, 01 – 0,5 стоп-бита, 10 – 2 стоп-бита, 11 – 1,5 стоп-бита);
 - разряд 14 LINEN разрешает режим LIN USART (0 – запрещён, 1 – разрешён);
 - разряды 15...31 зарезервированы.
- Разряды CPOL, CPHA и LBCL не должны изменяться во время активности передатчика.

Варианты настройки порта USART в режиме SPI с помощью разрядов CPOL, CPHA и LBCL наглядно представлены на рисунке 3.

РЕГИСТР USART_CR3

Регистр USART_CR3 содержит следующие разряды:

- разряд 0 EIE разрешает прерывания от ошибок;
- разряд 1 IREN разрешает режим работы порта IrDA;
- разряд 2 IRLP осуществляет выбор между нормальным и низко потребляющим режимом IrDA (0 – нормальный режим, 1 – низко потребляющий режим);
- разряд 3 HDSEL разрешает полудуплексный режим для однопроводного интерфейса;
- разряд 4 NACK разрешает формирование сигнала NACK в режиме Smartcard;
- разряд 5 SCEN разрешает работу порта в режиме Smartcard;
- разряд 6 DMAR разрешает работу приёмника через DMA;

- разряд 7 DMAT разрешает работу передатчика через DMA;
- разряд 8 RTSE разрешает формирование сигнала RTS;
- разряд 9 CTSE разрешает формирование сигнала CTS;
- разряд 10 CTSIE разрешает прерывания от флага CTS регистра USART_SR;
- разряды 11...31 зарезервированы.

Некоторые разряды регистров USART могут быть использованы для формирования прерывания. Эти разряды представлены на рисунке 4.

РЕГИСТР USART_GTPR

Последний регистр порта USART_CTPR содержит следующие разряды:

- разряды 7...0 PSC[7:0] определяют величину предварительного делителя для задания скорости порта в режиме IrDA и Smartcard;
- разряды 15...8 GT[7:0] задают значение таймера защиты для работы порта USART в режиме Smartcard;
- разряды 31...16 зарезервированы.

Более подробное описание назначения всех регистров USART и их разрядов можно найти в источнике [2].

Таблица 2. Формат кадра данных USART

Бит M	Бит PCE	Формат USART
0	0	S012345678F
0	1	S01234567PF
1	0	S0123456789F
1	1	S012345678PF

Примечания:

S – стартовый бит, F – стоповый бит, P – бит паритета, 0...8 – биты данных.

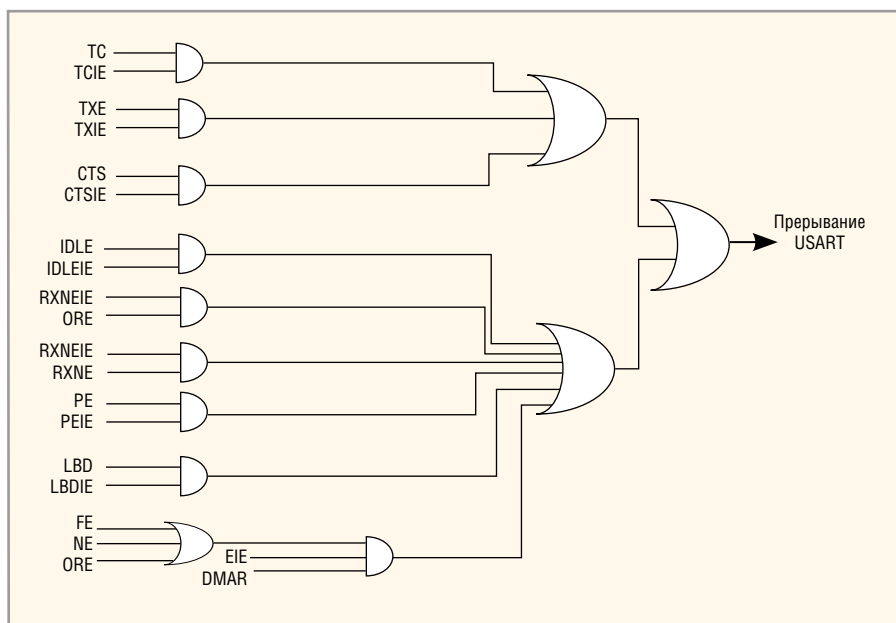


Рис. 4. Структурная схема формирования прерываний от порта USART

ПРОГРАММНОЕ ИСПОЛЬЗОВАНИЕ ПОРТОВ USART

После знакомства с назначением регистров порта USART рассмотрим пример его инициализации. Допустим необходимо настроить порт USART1 для связи с компьютером со скоростью обмена 9600 бод. Определим формат кадра порта с длиной посылки данных 8 бит, одним стоп-битом, без контроля паритета. Частоту тактового сигнала примем равной 24 МГц.

Вначале необходимо включить тактирование порта USART и портов GPIO с помощью следующего набора команд:

```
RCC->APB2ENR|=RCC_APB2ENR_USART1EN; // Включить тактирование порта USART1
RCC->APB2ENR|=RCC_APB2ENR_IOPAEN; // Включить тактирование GPIO
```

```
RCC->APB2ENR|=RCC_APB2ENR_AFIOEN; // Включить тактирование альтернативных функций GPIO
```

Далее следует рассчитать скорость обмена для порта и записать полученное значение в регистр USART1_BRR. Итак: $USARTDIV = fck / (16 \times baud) = 24\,000\,000 / (16 \times 9600) = 156,25$. Переменная DIV_Mantissa составляет 156 = 0x9C, а переменная DIV_Fraction = 0,25 × 16 = 4.

Запись значения скорости в регистр USART1_BRR выполняется с помощью команды:

```
USART1->BRR = 0x09c4; // Задать скорость обмена порта USART1 равную 9600 бод
```

Количество стоп-битов задаётся командой:

```
USART1->CR2 &= ~USART_CR2_STOP; // Значение STOP = 00 соответствует 1 стоп-биту
```

Разрешение работы приёмника, передатчика и порта можно задать одной командой:

```
USART1->CR1 = USART_CR1_RE|USART_CR1_TE|USART_CR1_UE
```

Теперь инициализация порта выполнена.

Для передачи данных через порт USART необходимо записать передаваемый байт в регистр USART_DR конкретного порта:

```
USART1->DR = data; // Передать байт данных
```

Прежде чем передавать следующий байт необходимо дождаться окончания передачи предыдущего байта, анализируя состояние разряда TC регистра USART_SR:

```
while((USART1->SR & USART_SR_TC)==0) {} // Ждать окончания передачи
USART1->SR &= ~USART_SR_TC; // Очистить флаг окончания передачи
```

Приём информации от порта можно осуществлять следующим образом:

```
while((USART1->SR & USART_SR_RXNE)==0) {} // Ждать приёма информации
temp = USART1->DR; // Считать принятый байт
```

В данном случае сброс флага окончания приёма производится автоматически после чтения регистра данных.

Обмен информацией через порт USART с помощью ожидания установки флагов является неэффективным и приводит к значительным потерям

производительности микроконтроллера, а порой и к зависанию программы. Во избежание этих проблем рекомендуется использовать работу с портом USART по прерываниям.

Для работы по прерываниям необходимо написать функцию обработчика прерывания, в которой будет проводиться анализ причины прерывания, а также будут выполняться необходимые операции для его обработки.

Такая функция может иметь следующий вид:

```
void USART1_IRQHandler(void)
{
    unsigned char temp; // Временная байтовая переменная
    if((USART1->SR & USART_SR_RXNE)!=0) // Если приём завершен
    {
        temp = USART1->DR; // Считать принятый байт
        ... // Выполнить необходимые действия
    }
    if((USART1->SR & USART_SR_TC)!=0) // Если передача завершена
    {
        USART1->SR &= ~USART_SR_TC; // Сбросить флаг
        ... // Выполнить необходимые действия
    }
}
```

Для активации прерываний от порта USART необходимо сначала их разрешить и затем задать события, которые будут генерировать прерывания с помощью следующих команд:

```
NVIC_EnableIRQ(USART1_IRQn); // Разрешить прерывания от USART1
USART1->CR1 |= USART_CR1_TCIE; // Разрешить прерывание по окончанию передачи
USART1->CR1 |= USART_CR1_RXNEIE; // Разрешить прерывание по приёму данных
```

Использование прерываний освобождает процессор микроконтроллера от необходимости постоянной проверки флагов и позволяет высвободить его ресурсы для других действий.

Помимо прерываний уменьшить нагрузку на процессор микроконтроллера позволяет использование DMA. Но это уже тема отдельной статьи...

ЛИТЕРАТУРА

1. <https://www.st.com>
2. http://www.st.com/web/en/resource/technical/document/reference_manual/CD00246267.pdf