
22. USART

22.1 Особенности

- Полнодуплексный режим (независимые последовательные регистры приема и передачи)
- Асинхронная или синхронная работа
- Синхронная работа с тактовой частотой ведущего или ведомого устройства
- Генератор скорости передачи данных с высоким разрешением
- Поддерживает последовательные кадры с 5, 6, 7, 8 или 9 битами данных и 1 или 2 стоповыми битами
- Генерация нечетной или четной четности и проверка четности, поддерживаемая аппаратно
- Обнаружение переполнения данных
- Обнаружение ошибок кадрирования
- Фильтрация шума включает обнаружение ложного начального бита и цифровой фильтр нижних частот
- Три отдельных прерывания по завершению передачи, пустому регистру данных передачи и завершению приема
- Многопроцессорный режим связи
- Двухскоростной асинхронный режим связи

22.2 Обзор

Универсальный синхронный и асинхронный последовательный приемник и передатчик (USART) представляет собой очень гибкое устройство последовательной связи.

ATmega640/1280/2560 имеет четыре USART: USART0, USART1, USART2 и USART3. Функциональность всех четырех USART описана ниже. USART0, USART1, USART2 и USART3 имеют разные регистры ввода-вывода, как показано на рис. [«Сводка регистрации» на стр. 399](#).

Упрощенная блок-схема передатчика USART показана на [Рис. 22-1 на стр. 201](#). Доступные ЦП регистры ввода-вывода и контакты ввода-вывода выделены жирным шрифтом.

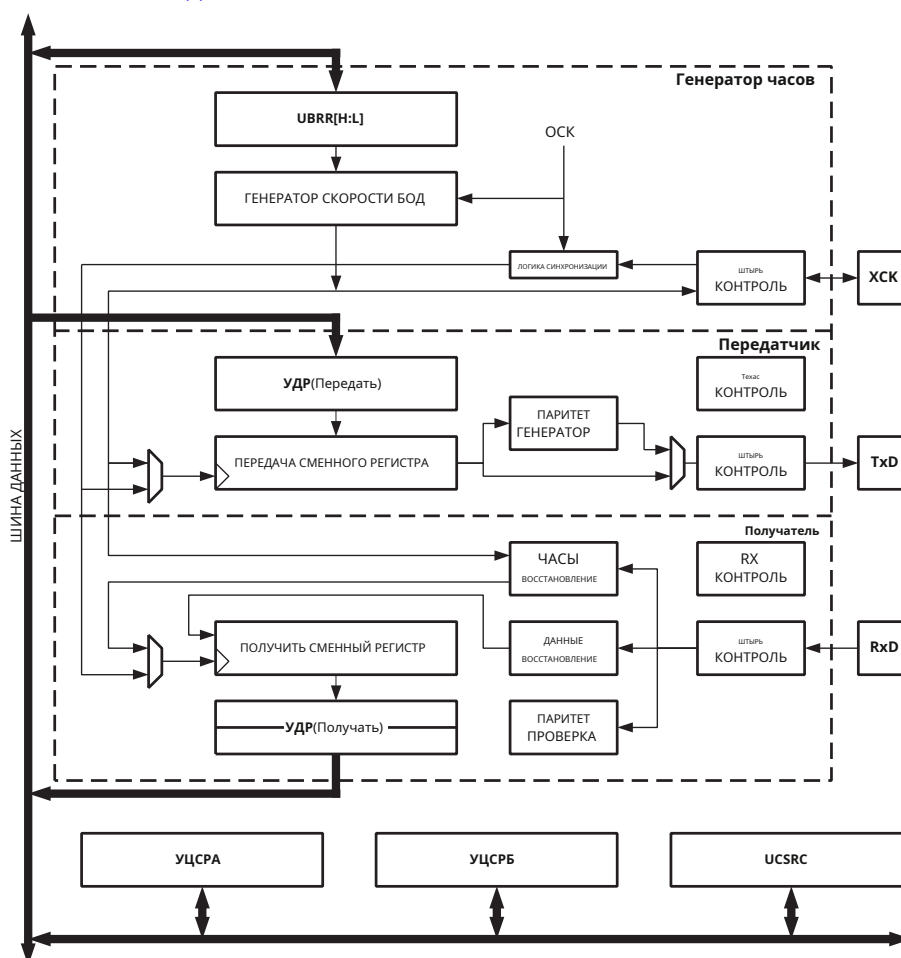
Бит Power Reduction USART0, PRUSART0, в [«PRR0 — Регистр снижения мощности 0» на стр. 55](#) необходимо отключить, записав в него логический ноль.

Бит Power Reduction USART1, PRUSART1, в [«PRR1 — Регистр снижения мощности 1» на стр. 56](#) необходимо отключить, записав в него логический ноль.

Бит Power Reduction USART2, PRUSART2, в [«PRR1 — Регистр снижения мощности 1» на стр. 56](#) необходимо отключить, записав в него логический ноль.

Бит Power Reduction USART3, PRUSART3, в [«PRR1 — Регистр снижения мощности 1» на стр. 56](#) необходимо отключить, записав в него логический ноль.

Рисунок 22-1. Блок-схема USART⁽¹⁾



Примечание: 1. См. Рисунок 1-1 на странице 2, Рис. 1-3 на стр. 4, Табл. 13-12 на стр. 80, Табл. 13-15 на стр. 82, Табл. 13-24 на стр. 88а также Табл. 13-27 на стр. 90 для размещения штифта USART.

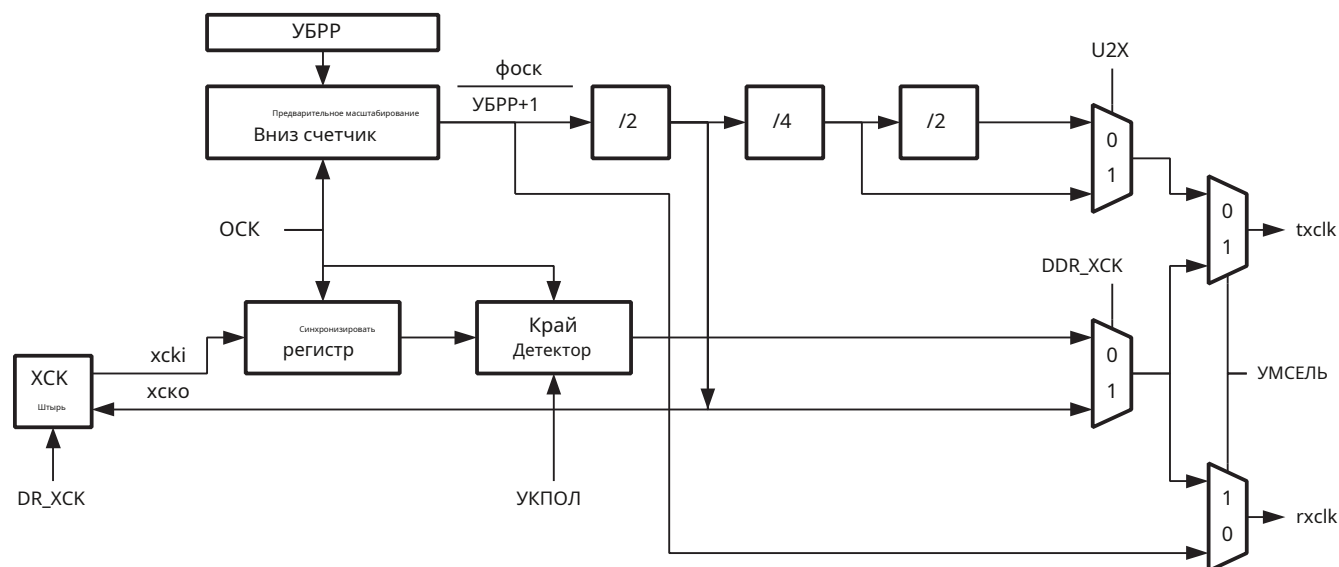
Пунктирные прямоугольники на блок-схеме разделяют три основные части USART (перечислены сверху): генератор тактовых импульсов, передатчик и приемник. Регистры управления являются общими для всех модулей. Логика генерации тактового сигнала состоит из логики синхронизации для внешнего тактового входа, используемого синхронным ведомым устройством, и генератора скорости передачи данных. Вывод ХСКп (Transfer Clock) используется только в режиме синхронной передачи. Передатчик состоит из одного буфера записи, последовательного регистра сдвига, генератора четности и логики управления для обработки различных форматов последовательных кадров. Буфер записи позволяет осуществлять непрерывную передачу данных без задержки между кадрами. Приемник является наиболее сложной частью модуля USART из-за его часов и блоков восстановления данных. Блоки восстановления используются для асинхронного приема данных. Помимо восстановительных установок, Приемник включает в себя средство проверки четности, управляющую логику, сдвиговый регистр и двухуровневый приемный буфер (UDRn). Приемник поддерживает те же форматы фреймов, что и передатчик, и может обнаруживать ошибки фрейма, превышения данных и ошибки четности.

22.3 Генерация часов

Логика генерации часов генерирует базовые часы для передатчика и приемника. USARTn поддерживает четыре режима работы часов: нормальный асинхронный, асинхронный с удвоенной скоростью, синхронный ведущий и синхронный режим ведомого. Бит UMSELn в регистре управления и состояния USART C (UCSRnC) выбирает между асинхронной и синхронной работой. Двойная скорость (только в асинхронном режиме) управляется U2Xn, находящимся в регистре UCSRnA. При использовании синхронного режима (UMSELn = 1) регистр направления данных для вывода ХСКп (DDR_XCKn) определяет, является ли источник синхронизации внутренним (ведущий режим) или внешним (ведомый режим). Вывод ХСКп активен только при использовании синхронного режима.

Рисунок 22-2 показывает блок-схему логики генерации тактового сигнала.

Рисунок 22-2. Логика генерации тактовых импульсов, блок-схема



Описание сигнала:

txclk	Часы передатчика (внутренний сигнал).
rxclk	Базовые часы приемника (внутренний сигнал).
xcki	Вход с вывода XCK (внутренний сигнал). Используется для синхронной работы ведомого устройства.
xsko	Тактовый выход на вывод XCK (внутренний сигнал). Используется для синхронной ведущей операции.
фоск	Частота выводов XTAL (системные часы).

22.3.1 Генерация внутреннего тактового сигнала — генератор скорости передачи

Генерация внутренних часов используется для асинхронного и синхронного режимов ведущего устройства. Описание в этом разделе относится к [Рисунок 22-2](#).

Регистр скорости передачи данных USART (UBRRn) и подключенный к нему обратный счетчик функционируют как программируемый делитель или генератор скорости передачи. Счетчик вниз, работающий по системным часам ($f_{оск}$), загружается со значением UBRRn каждый раз, когда счетчик досчитывается до нуля или когда записывается регистр UBRRn. Часы генерируются каждый раз, когда счетчик достигает нуля. Эти часы являются тактовым выходом генератора скорости передачи данных ($= f_{оск}/(UBRRn+1)$). Передатчик делит тактовый сигнал генератора скорости передачи данных на 2, 8 или 16 в зависимости от режима. Выход генератора скорости передачи используется непосредственно блоками синхронизации и восстановления данных приемника. Однако блоки восстановления используют конечный автомат, который использует 2, 8 или 16 состояний в зависимости от режима, установленного состоянием битов UMSELn, U2Xn и DDR_XCKn.

Табл. 22-1 на стр. 203 содержит уравнения для расчета скорости передачи данных (в битах в секунду) и для расчета значения UBRRn для каждого режима работы с использованием внутреннего источника синхронизации.

Таблица 22-1. Уравнения для расчета настройки регистра скорости передачи данных

Режим работы	Уравнение для расчета скорости передачи(1)	Уравнение для расчета значения UBRR
Асинхронный нормальный режим (U2Xn = 0)	$БОД_{\text{знак равно}} \frac{\phi}{16 \cdot UBRRn + 1} \cdot C$	$UBRRn_{\text{знак равно}} \frac{\phi}{16 \cdot БОД} \cdot C - 1$
Асинхронный двухскоростной режим (U2Xn = 1)	$БОД_{\text{знак равно}} \frac{\phi}{8 \cdot UBRRn + 1} \cdot C$	$UBRRn_{\text{знак равно}} \frac{\phi}{8 \cdot БОД} \cdot C - 1$
Синхронный мастер-режим	$БОД_{\text{знак равно}} \frac{\phi}{2 \cdot UBRRn + 1} \cdot C$	$UBRRn_{\text{знак равно}} \frac{\phi}{2 \cdot БОД} \cdot C - 1$

Примечание: 1. Скорость передачи определяется как скорость передачи в битах в секунду (бит/с).

- БОД

Скорость передачи данных (в битах в секунду, bps).
- фоск

Тактовая частота системного генератора.
- UBRRn

Содержимое регистров UBRRHn и UBRRLn (0-4095).

Некоторые примеры значений UBRRn для некоторых системных тактовых частот можно найти вТабл. 22-9 на стр. 223.

22.3.2 Двухскоростной режим (U2Xn)

Скорость передачи можно удвоить, установив бит U2Xn в UCSRnA. Установка этого бита влияет только на асинхронную операцию. Установите этот бит в ноль при использовании синхронной операции.

Установка этого бита уменьшит делитель скорости передачи с 16 до 8, фактически удвоив скорость передачи для асинхронной связи. Однако обратите внимание, что в этом случае приемник будет использовать только половину числа выборок (уменьшено с 16 до 8) для выборки данных и восстановления тактовой частоты, поэтому при использовании этого режима требуется более точная настройка скорости передачи данных и системных тактовых импульсов. Для передатчика минусов нет.

22.3.3 Внешние часы

Внешняя синхронизация используется синхронными ведомыми режимами работы. Описание в этом разделе относится к Рис. 22-2 на стр. 202для деталей.

Вход внешнего тактового сигнала с вывода ХСКп дискретизируется регистром синхронизации, чтобы свести к минимуму вероятность метастабильности. Выход из регистра синхронизации должен затем пройти через детектор фронта, прежде чем он сможет быть использован передатчиком и приемником. Этот процесс вводит задержку в два тактовых периода ЦП, поэтому максимальная внешняя тактовая частота ХСКп ограничивается следующим уравнением:

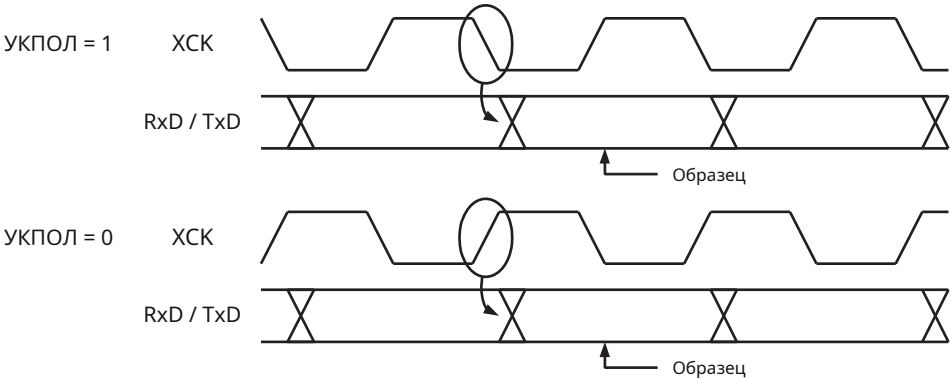
$$\phi_{ХСК} \leq \frac{\phi}{4} \cdot C$$

Обратите внимание, что фоскзависит от стабильности источника системных часов. Поэтому рекомендуется добавить некоторый запас, чтобы избежать возможной потери данных из-за изменений частоты.

22.3.4 Синхронная работа часов

Когда используется синхронный режим (UMSELn = 1), контакт ХСКn будет использоваться либо как вход тактового сигнала (Slave), либо как выход тактового сигнала (Master). Зависимость между фронтами тактовых импульсов и выборкой данных или изменением данных такая же. Основной принцип заключается в том, что входные данные (на RxDn) дискретизируются на фронте тактового сигнала ХСКn, противоположном фронту, на котором выход данных (TxDn) изменен.

Рисунок 22-3.Синхронный режим ХСКn Timing.



Бит UCPOLn UCRSC выбирает, какой фронт тактового сигнала ХСКn используется для выборки данных, а какой используется для изменения данных. В качествеРисунок 22-3показывает, что когда UCPOLn равен нулю, данные будут изменены по переднему фронту ХСКn, а выборка — по заднему фронту ХСКn. Если установлен UCPOLn, данные будут изменены по заднему фронту ХСКn, а выборка — по нарастающему фронту ХСКn.

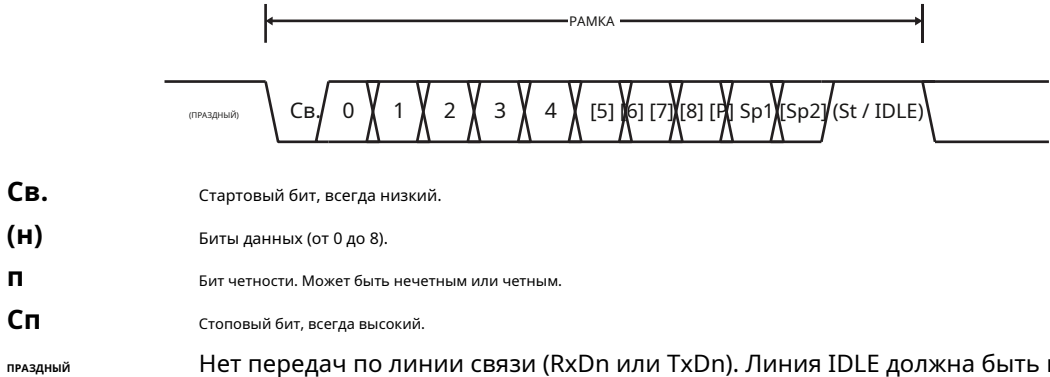
22.4 Форматы кадров

Последовательный кадр определяется как один символ битов данных с битами синхронизации (стартовый и стоповый биты) и, необязательно, битом четности для проверки ошибок. USART принимает все 30 следующих комбинаций в качестве допустимых форматов кадров:

- 1 стартовый бит
- 5, 6, 7, 8 или 9 бит данных
- нет, четный или нечетный бит четности
- 1 или 2 стоповых бита

Кадр начинается со стартового бита, за которым следует младший бит данных. Затем следуют следующие биты данных, всего до девяти, заканчиваясь самым старшим битом. Если разрешено, бит четности вставляется после битов данных перед стоповыми битами. Когда передается полный кадр, за ним может непосредственно следовать новый кадр, или линия связи может быть установлена в состояние ожидания (высокий уровень).Рисунок 22-4иллюстрирует возможные комбинации форматов кадров. Биты внутри скобок являются необязательными.

Рисунок 22-4.Форматы кадров



Формат кадра, используемый USART, устанавливается битами UCSZn2:0, UPMn1:0 и USBSn в UCSRnB и UCS-RnC. Приемник и передатчик используют одинаковые настройки. Обратите внимание, что изменение настройки любого из этих битов приведет к повреждению всей текущей связи как для приемника, так и для передатчика.

Биты размера символа USART (UCSZn2:0) выбирают количество битов данных в кадре. Биты режима четности USART (UPMn1:0) включают и устанавливают тип бита четности. Выбор между одним или двумя стоповыми битами осуществляется битом выбора стопового бита USART (USBSn). Получатель игнорирует второй стоповый бит. Таким образом, FE (ошибка кадра) будет обнаружена только в тех случаях, когда первый стоповый бит равен нулю.

22.4.1 Вычисление бита четности

Бит четности вычисляется путем выполнения исключающего ИЛИ всех битов данных. Если используется нечетная четность, результат исключающего или инвертируется. Бит четности расположен между последним битом данных и первым стоповым битом последовательного кадра. Связь между битом четности и битами данных следующая:

$$\begin{aligned} \text{Пдажезнак равно } D_{n-1} \oplus D_{n-2} \oplus \dots \oplus D_1 \oplus D_0 &= 0 \text{ п} \\ \text{Пнечетныйзнак равно } D_{n-1} \oplus D_{n-2} \oplus \dots \oplus D_1 \oplus D_0 &= 1 \end{aligned}$$

Пдаже	Бит четности с использованием четности.
Пнечетный	Бит четности с использованием нечетной четности.
Дn	Бит данных n символа.

22.5 Инициализация USART

USART должен быть инициализирован до того, как может иметь место какая-либо связь. Процесс инициализации обычно состоит из установки скорости передачи данных, установки формата кадра и включения передатчика или приемника в зависимости от использования. Для операции USART, управляемой прерыванием, флаг глобального прерывания должен быть очищен (и прерывания глобально отключены) при выполнении инициализации.

Перед выполнением повторной инициализации с измененной скоростью передачи данных или форматом кадра убедитесь, что в течение периода изменения регистров не было текущих передач. Флаг TXCn можно использовать для проверки того, что передатчик завершил все передачи, а флаг RXC можно использовать для проверки отсутствия непрочитанных данных в приемном буфере. Обратите внимание, что флаг TXCn должен очищаться перед каждой передачей (перед записью UDRn), если он используется для этой цели.

В следующих простых примерах кода инициализации USART показаны одна сборка и одна функция C, которые имеют одинаковую функциональность. В примерах предполагается асинхронная работа с использованием опроса (прерывания не разрешены) и фиксированный формат кадра. Скорость передачи задается как функциональный параметр. Для ассемблерного кода предполагается, что параметр скорости передачи хранится в регистрах r17:r16.

Пример ассемблерного кода⁽¹⁾

```
USART_Init:
    ;Установить скорость передачи
    сц    УБРРНХ, r17
    сц    УБРРНЛ, r16
    лди   r16, (1<<U2Xn)
    сц    УЦРНА, r16
    ;Включить приемник и передатчик лди
    r16, (1<<RXENn)|(1<<TXENn) сцUCSRnB,r16

    ;Установите формат кадра: 8 данных, 1 стоповый бит
    лди   r16, (2<<UMSELn)|(3<<UCSZn0)
    сц    UCSRnC,r16
    рет
```

Пример кода C⁽¹⁾

```
# определить FOSC 1843200// Тактовая частота
# определить BAUD 9600
# определить (MYUBRR FOSC/16/
BAUD-1) void main( void )
{...
USART_Init ( MYUBRR ); . . } //
главный
void USART_Init( unsigned int ubrr){ /* Установить
скорость передачи */
UBRRH = (символ без знака) (ubrr>>8);
UBRRL = (беззнаковый символ)ubrr;
/* Включить приемник и передатчик */ UCSRB
= (1<<RXEN)|(1<<TXEN);
/* Установить формат кадра: 8 data, 2 стоповых
бита */ UCSRC = (1<<USBS)|(3<<UCSZ0);
} // USART_Init
```

Примечание: 1.См. «Примеры кода» на стр. 10.

Могут быть созданы более сложные процедуры инициализации, которые включают формат кадра в качестве параметров, отключают прерывания и так далее. Однако во многих приложениях используются фиксированные значения регистров скорости и управления, и для этих типов приложений код инициализации может быть помещен непосредственно в основную процедуру или объединен с кодом инициализации для других модулей ввода-вывода.

22.6 Передача данных — передатчик USART

Передатчик USART активируется установкой *Включить передачу*(TXEN) в регистре UCSRnB. Когда передатчик включен, нормальная работа порта вывода TxDn переопределяется USART и получает функцию последовательного выхода передатчика. Скорость передачи данных, режим работы и формат кадра должны быть установлены один раз перед любой передачей. Если используется синхронная работа, часы на выводе XCKn будут переопределены и будут использоваться в качестве часов передачи.

22.6.1 Отправка кадров с 5-8 битами данных

Передача данных инициируется путем загрузки буфера передачи данными, которые должны быть переданы. ЦП может загрузить буфер передачи, записав в ячейку ввода-вывода UDRn. Буферизованные данные в буфере передачи будут перемещены в сдвиговый регистр, когда сдвиговый регистр будет готов для отправки нового кадра. Сдвиговый регистр загружается новыми данными, если он находится в состоянии ожидания (передача не ведется) или сразу после передачи последнего стопового бита предыдущего кадра. Когда сдвиговый регистр загружается новыми данными, он будет передавать один полный кадр со скоростью, заданной бодовым регистром, битом U2Xn или XCKn, в зависимости от режима работы.

В следующих примерах кода показана простая функция передачи USART, основанная на опросе *Регистр данных пуст*(UDREN) Флаг. При использовании кадров с менее чем восемью битами старшие значащие биты, записанные в UDRn, игнорируются. Перед использованием функции USART должен быть инициализирован. Предполагается, что для ассемблерного кода отправляемые данные хранятся в регистре R16.

Пример ассемблерного кода(1)
<div>USART_Передача: ;Подождите, пока буфер передачи будет пуст. лдсr17, UCSRnA sbrsr17, UDREN ржмпUSART_Передача ;Помещает данные (r16) в буфер, отправляет данные сц UDRn,r16 рет</div>
Пример кода C(1)
<div>пустотаUSART_Передача(беззнаковый символ данные){ /*Подождите, пока буфер передачи будет пуст.*/ пока(!(UCSRnA & (1<<UDREN))) ; /*Помещает данные в буфер, отправляет данные*/ UDRn = данные; }</div>

Примечание: 1.См. «Примеры кода» на стр. 10.

Функция просто ожидает, пока буфер передачи не станет пустым, проверяя флаг UDREN, прежде чем загрузить в него новые данные для передачи. Если используется прерывание о пустом регистре данных, процедура обработки прерывания записывает данные в буфер.

22.6.2 Отправка кадров с 9 битами данных

Если используются 9-битные символы (UCSZn = 7), девятый бит должен быть записан в бит TXB8 в UCSRnB до того, как младший байт символа будет записан в UDRn. В следующих примерах кода показана функция передачи, которая обрабатывает 9-битные символы. Для ассемблерного кода предполагается, что отправляемые данные хранятся в регистрах R17:R16.

Пример ассемблерного кода(1)(2)
<div>USART_Передача: ;Подождите, пока буфер передачи будет пуст. сбисУЦSRnA,УДREn ржмпUSART_Передача ;Скопируйте 9-й бит из r17 в TXB8 цбиUCSRnB, TXB8 sbrсp17,0 сбиUCSRnB, TXB8 ;Помещает данные LSB (r16) в буфер, отправляет данные сцУДРn,r16 рет</div>
Пример кода C(1)(2)
<div>пустотаUSART_Передача(беззнаковое целоеданные) { /* Подождите, пока буфер передачи будет пуст.*/ пока(!(UCSRnA & (1<<UDREn)))) ; /* Скопируйте 9-й бит в TXB8*/ UCSRnB &= ~(1<<TXB8); если (данные и 0x0100) UCSRnB = (1<<TXB8); /* Помещает данные в буфер, отправляет данные*/ UDRn = данные; }</div>

Примечания: 1. Эти функции передачи записываются как общие функции. Их можно оптимизировать, если содержимое UCSRnB статично. Например, после инициализации используется только бит TXB8 регистра UCSRnB.
2.См. «Примеры кода» на стр. 10.
Девятый бит может использоваться для указания кадра адреса при использовании многопроцессорного режима связи или для другой обработки протокола, например синхронизации.

22.6.3 Флаги передатчика и прерывания

Передатчик USART имеет два флага, которые указывают на его состояние: пустой регистр данных USART (UDREn) и передача завершена (TxCn). Оба флага могут использоваться для генерации прерываний.
Флаг пустого регистра данных (UDREn) показывает, готов ли буфер передачи к приему новых данных. Этот бит устанавливается, когда буфер передачи пуст, и сбрасывается, когда буфер передачи содержит данные для передачи, которые еще не были перемещены в регистр сдвига. Для совместимости с будущими устройствами всегда записывайте этот бит в ноль при записи регистра UCSRnA.
Когда бит разрешения прерывания по пустому регистру данных (UDRIEn) в UCSRnB записывается в единицу, прерывание по пустому регистру данных USART будет выполняться до тех пор, пока установлен UDREn (при условии, что глобальные прерывания разрешены).

UDRep очищается записью UDRn. Когда используется передача данных, управляемая прерыванием, подпрограмма прерывания опустошения регистра данных должна либо записать новые данные в UDRn, чтобы очистить UDREn, либо отключить прерывание опустошения регистра данных, иначе после завершения подпрограммы обработки прерывания возникнет новое прерывание.

Бит флага завершения передачи (TXCn) устанавливается в единицу, когда весь кадр в регистре сдвига передачи был сдвинут наружу и в настоящее время в буфере передачи нет новых данных. Бит флага TXCn автоматически очищается, когда выполняется прерывание завершения передачи, или его можно очистить, записав единицу в его битовую позицию. Флаг TXCn полезен в полудуплексных коммуникационных интерфейсах (таких как стандарт RS-485), где передающее приложение должно перейти в режим приема и освободить коммуникационную шину сразу после завершения передачи.

Когда бит разрешения прерывания передачи (TXCIEn) в UCSRnB установлен, прерывание завершения передачи USART будет выполнено, когда флаг TXCn станет установлен (при условии, что глобальные прерывания разрешены). Когда используется прерывание завершения передачи, процедура обработки прерывания не должна очищать флаг TXCn, это делается автоматически при выполнении прерывания.

22.6.4 Генератор четности

Генератор четности вычисляет бит четности для данных последовательного кадра. Когда бит четности включен (UPMn1 = 1), логика управления передатчиком вставляет бит четности между последним битом данных и первым стоповым битом отправляемого кадра.

22.6.5 Отключение передатчика

Отключение передатчика (установка TXEN на ноль) не вступит в силу до тех пор, пока не будут завершены текущие и ожидающие передачи, то есть когда сдвиговый регистр передачи и буферный регистр передачи не содержат данных для передачи. При отключении передатчик больше не будет переопределять вывод TxDn.

22.7 Прием данных – приемник USART

Приемник USART включается путем записи бита разрешения приема (RXENn) в регистре UCSRnB в единицу. Когда приемник включен, нормальная работа вывода RxDn переопределяется USART и получает функцию последовательного входа приемника. Скорость передачи данных, режим работы и формат кадра должны быть установлены один раз перед выполнением любого последовательного приема. Если используется синхронная работа, часы на выводе XCKn будут использоваться в качестве часов передачи.

22.7.1 Прием кадров с 5-8 битами данных

Приемник начинает прием данных, когда обнаруживает действительный стартовый бит. Каждый бит, следующий за начальным битом, будет опрашиваться со скоростью передачи данных или тактовой частотой XCKn и сдвигаться в сдвиговый регистр приема до тех пор, пока не будет получен первый стоповый бит кадра. Второй стоповый бит будет игнорироваться приемником. Когда получен первый стоповый бит, то есть в приемном сдвиговом регистре присутствует полный последовательный кадр, содержимое сдвигового регистра будет перемещено в приемный буфер. Затем приемный буфер может быть прочитан путем чтения адреса ввода/вывода UDRn.

В следующем примере кода показана простая функция приема USART, основанная на опросе флага завершения приема (RXCn). При использовании кадров с менее чем восемью битами самые значащие биты данных, считанных из UDRn, будут замаскированы до нуля. Перед использованием функции USART должен быть инициализирован.

Пример ассемблерного кода ⁽¹⁾
<pre> USART_Receive: ;Дождитесь получения данных лдс p17, UCSRnA sbrrsr17, RxCn ржмпUSART_Receive ;Получить и вернуть полученные данные из буфера лдс r16, UDRn рет </pre>
Пример кода C ⁽¹⁾
<pre> беззнаковый символ USART_Receive(пустота) { /* Дождитесь получения данных*/ пока (! (UCSRnA и (1<<RxCn))) ; /* Получить и вернуть полученные данные из буфера*/ возвращатьсяUDRn; } </pre>

Примечание: 1. См. «Примеры кода» на стр. 10.

Функция просто ожидает появления данных в буфере приема, проверяя флаг RxCn, прежде чем читать буфер и возвращать значение.

22.7.2 Прием кадров с 9 битами данных

Если используются 9-битные символы (UCSZn=7), девятый бит должен быть прочитан из бита RxB8n в UCSRnB. **до** чтение младших битов из UDRn. Это правило также применяется к флагам состояния FEn, DORn и UPEn. Прочитать статус из UCSRnA, затем данные из UDRn. Чтение местоположения ввода-вывода UDRn изменит состояние FIFO приемного буфера и, следовательно, изменятся биты TXB8n, FEn, DORn и UPEn, которые хранятся в FIFO.

В следующем примере кода показана простая функция приема USART, которая обрабатывает как девятибитные символы, так и биты состояния.

Пример ассемблерного кода⁽¹⁾

```

USART_Receive:
    ;Дождитесь получения данных лдс
    p17, UCSRnA
    sbrr17, RXCn
    ржмп USART_Receive
    ;Получить статус и 9-й бит, затем данные из буфера лдс
    p18, UCSRnA
    лдс r17, UCSRnB
    лдс r16, UDRn
    ;Если ошибка, вернуть -1
    и я r18, (1<<FEn)|(1<<DORn)|(1<<UPEn) перерыв
    USART_ReceiveNoError лди
    r17, ВЫСОКИЙ(-1)
    лди r16, НИЗКИЙ (-1)
USART_ReceiveNoError:
    ;Отфильтровать 9-й бит, затем вернуться
    лсrr17
    и я r17, 0x01
    рет
    
```

Пример кода C⁽¹⁾

```

беззнаковое целое USART_Receive(пустота) {

    беззнаковый символ статус, реш, ресл; /*
    Дождитесь получения данных*/ пока( !(
    UCSRnA и (1<<RXCn)) )

    ;

    /* Получить статус и 9-й бит, затем данные*/ /*
    из буфера*/ статус = UCSRnA;

    реш = UCSRnB;
    ресл = UDRn;
    /*Если ошибка, вернуть -1*/
    если(статус & (1<<FEn)|(1<<DORn)|(1<<UPEn))
        возвращаться-1;
    /*Отфильтровать 9-й бит, затем вернуться*/
    реш = (реш >> 1) & 0x01; возвращаться((реш <<
    8) | ресл);
}
    
```

Примечание: 1.См. «Примеры кода» на стр. 10.

Пример функции приема считывает все регистры ввода-вывода в файл регистров до того, как будут выполнены какие-либо вычисления. Это обеспечивает оптимальное использование приемного буфера, поскольку считанное местоположение буфера будет свободно для приема новых данных как можно раньше.

22.7.3 Получение флага конкуренции и прерывание

Приемник USART имеет один флаг, который указывает состояние приемника.

Флаг завершения приема (RXCn) указывает на наличие непрочитанных данных в приемном буфере. Этот флаг равен единице, если в приемном буфере есть непрочитанные данные, и нулю, когда приемный буфер пуст (то есть не содержит непрочитанных данных). Если приемник отключен (RXENn = 0), буфер приема будет сброшен, и, следовательно, бит RXCn станет равным нулю.

Когда в UCSRnB установлено разрешение завершения прерывания приема (RXCIEn), прерывание завершения приема USART будет выполняться до тех пор, пока установлен флаг RXCn (при условии, что глобальные прерывания разрешены). Когда используется прием данных, управляемый прерыванием, процедура завершения приема должна считать полученные данные из UDRn, чтобы сбросить флаг RXCn, в противном случае после завершения процедуры обработки прерывания возникнет новое прерывание.

22.7.4 Флаги ошибок приемника

Приемник USART имеет три флага ошибок: ошибка кадра (FEn), переполнение данных (DORn) и ошибка четности (UPEn). Ко всем можно получить доступ, прочитав UCSRnA. Общим для флагов ошибок является то, что они находятся в приемном буфере вместе с кадром, для которого они указывают на состояние ошибки. Из-за буферизации флагов ошибок UCS-RnA должен считываться перед приемным буфером (UDRn), поскольку чтение местоположения ввода-вывода UDRn изменяет местоположение чтения буфера. Другое равенство для флагов ошибок заключается в том, что они не могут быть изменены программным обеспечением, выполняющим запись в местоположение флага. Тем не менее, все флаги должны быть установлены на ноль, когда UCSRnA записывается для восходящей совместимости будущих реализаций USART. Ни один из флагов ошибок не может генерировать прерывания.

Флаг ошибки кадра (FEn) указывает состояние первого стопового бита следующего доступного для чтения кадра, сохраненного в приемном буфере. Флаг FEn равен нулю, если стоповый бит был правильно прочитан (как единица), а флаг FEn будет равен единице, если стоповый бит был неправильным (ноль). Этот флаг можно использовать для обнаружения условий рассинхронизации, обнаружения условий прерывания и обработки протокола. На флаг FEn не влияет установка бита USBSn в UCSRnC, поскольку приемник игнорирует все стоповые биты, кроме первого. Для совместимости с будущими устройствами всегда устанавливайте этот бит в ноль при записи в UCSRnA.

Флаг Data OverRun (DORn) указывает на потерю данных из-за переполнения буфера приемника. Data OverRun возникает, когда приемный буфер заполнен (два символа), это новый символ, ожидающий в приемном сдвиговом регистре, и обнаружен новый стартовый бит. Если флаг DORn установлен, то между последним кадром, считанным из UDRn, и следующим кадром, считанным из UDRn, был потерян один или несколько последовательных кадров. Для совместимости с будущими устройствами всегда записывайте этот бит в ноль при записи в UCSRnA. Флаг DORn очищается, когда полученный кадр был успешно перемещен из регистра сдвига в приемный буфер.

Флаг ошибки четности (UPEn) указывает, что следующий кадр в приемном буфере имел ошибку четности при приеме. Если проверка четности не включена, бит UPEn всегда будет считываться как ноль. Для совместимости с будущими устройствами всегда устанавливайте этот бит в ноль при записи в UCSRnA. Подробнее см. [«Вычисление бита четности» на стр. 205а](#) также [«Проверка четности»](#).

22.7.5 Проверка четности

Проверка четности активна, когда установлен старший бит режима проверки четности USART (UPMn1). Тип выполняемой проверки четности (нечетная или четная) выбирается битом UPMn0. Когда функция проверки четности включена, она вычисляет четность битов данных во входящих кадрах и сравнивает результат с битом четности из последовательного кадра. Результат проверки сохраняется в приемном буфере вместе с полученными данными и стоп-битами. Флаг ошибки четности (UPEn) затем может быть прочитан программным обеспечением, чтобы проверить, была ли в кадре ошибка четности.

Бит UPEn устанавливается, если следующий символ, который может быть прочитан из приемного буфера, имел ошибку четности при приеме и в этот момент была включена проверка четности (UPMn1 = 1). Этот бит действителен до тех пор, пока буфер приема (UDRn) не будет прочитан.

22.7.6 Отключение приемника

В отличие от передатчика, отключение приемника будет немедленным. Таким образом, данные текущих приемов будут потеряны. При отключении (то есть, RXENn установлен на ноль) приемник больше не будет отменять обычную функцию вывода порта RxDn. Буфер приемника FIFO будет очищен, когда приемник отключен. Остальные данные в буфере будут потеряны.

22.7.7 Очистка буфера приема

FIFO буфера приемника будет сброшен, когда приемник отключен, то есть буфер будет очищен от своего содержимого. Непрочитанные данные будут потеряны. Если буфер необходимо очистить во время нормальной работы, например, из-за состояния ошибки, считывайте местоположение ввода-вывода UDRn до тех пор, пока флаг RXCn не будет очищен. В следующем примере кода показано, как очистить приемный буфер.

Пример ассемблерного кода ⁽¹⁾
<pre>USART_Flush: сбисUCSRnA, RXCn рет в r16, UDRn ржмпUSART_Flush</pre>
Пример кода C ⁽¹⁾
<pre>пустотаUSART_Flush(пустота) { беззнаковый символ дурачок; пока(UCSRnA & (1<<RXCn)) фиктивный = UDRn; }</pre>

Примечание: 1. См. «Примеры кода» на стр. 10.

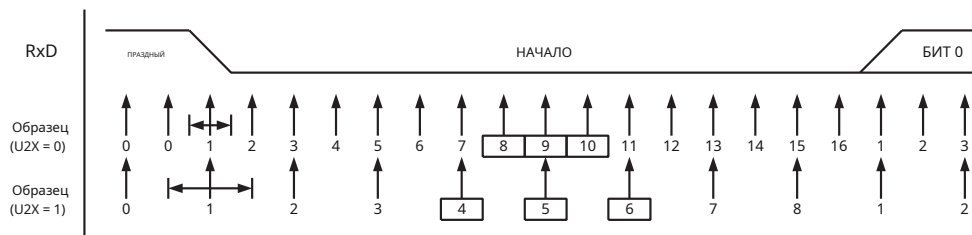
22.8 Асинхронный прием данных

USART включает в себя блок восстановления часов и блок восстановления данных для обработки асинхронного приема данных. Логика восстановления часов используется для синхронизации внутренне сгенерированных часов скорости передачи данных с входящими асинхронными последовательными кадрами на выводе RxDn. Логика восстановления данных осуществляет выборку и фильтрацию нижних частот каждого входящего бита, тем самым повышая помехозащищенность приемника. Рабочий диапазон асинхронного приема зависит от точности внутренних часов скорости передачи данных, скорости входящих кадров и размера кадра в битах.

22.8.1 Асинхронное восстановление часов

Логика восстановления часов синхронизирует внутренние часы с поступающими последовательными кадрами. Рис. 22-5 на стр. 214 иллюстрирует процесс выборки начального бита входящего кадра. Скорость выборки в 16 раз превышает скорость передачи данных в нормальном режиме и в восемь раз превышает скорость передачи данных в режиме двойной скорости. Горизонтальные стрелки иллюстрируют изменение синхронизации из-за процесса дискретизации. Обратите внимание на большую разницу во времени при использовании режима двойной скорости (U2Xn = 1). Образцы, обозначенные нулем, представляют собой образцы, выполняемые, когда линия RxDn простаивает (то есть нет активности связи).

Рисунок 22-5.Начать выборку битов

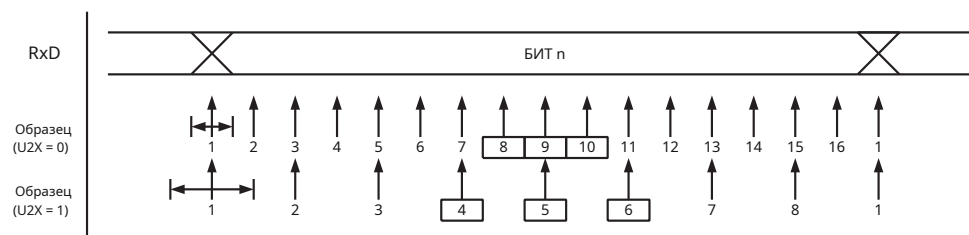


Когда логика восстановления тактового сигнала обнаруживает переход с высокого уровня (бездействия) на низкий уровень (запуск) на линии RxDn, инициируется последовательность обнаружения стартового бита. Пусть образец 1 обозначает первую нулевую выборку, как показано на рисунке. Затем логика восстановления тактового сигнала использует выборки 8, 9 и 10 для нормального режима и выборки 4, 5 и 6 для режима двойной скорости (обозначенные номерами выборок внутри прямоугольников на рисунке), чтобы определить, получен ли допустимый стартовый бит. Если две или более из этих трех выборок имеют логически высокие уровни (выигрывает большинство), стартовый бит отбрасывается как шумовой пик, и приемник начинает искать следующий переход от высокого к низкому. Однако, если обнаружен допустимый стартовый бит, логика восстановления часов синхронизируется, и можно начинать восстановление данных. Процесс синхронизации повторяется для каждого стартового бита.

22.8.2 Асинхронное восстановление данных

Когда часы приемника синхронизируются со стартовым битом, можно начинать восстановление данных. Модуль восстановления данных использует конечный автомат, который имеет 16 состояний для каждого бита в нормальном режиме и восемь состояний для каждого бита в режиме двойной скорости. [Рисунок 22-6](#)показывает выборку битов данных и бита четности. Каждому из образцов присваивается номер, равный состоянию блока восстановления.

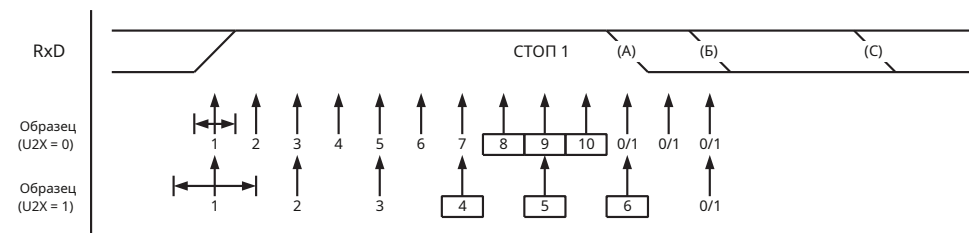
Рисунок 22-6.Выборка данных и бит четности



Решение о логическом уровне принятого бита принимается путем голосования большинством голосов логического значения по трем выборкам в центре принятого бита. Образцы в центре выделены на рисунке номером образца внутри прямоугольников. Процесс мажоритарного голосования выполняется следующим образом: если две или все три выборки имеют высокие уровни, принятый бит регистрируется как логическая 1. Если две или все три выборки имеют низкие уровни, принятый бит регистрируется как логический 0. Этот процесс голосования по большинству действует как фильтр нижних частот для входящего сигнала на выводе RxDn. Затем процесс восстановления повторяется до тех пор, пока не будет получен полный кадр. Включая первый стоповый бит. Обратите внимание, что Receiver использует только первый стоповый бит кадра.

[Рисунок 22-7](#)показывает выборку стопового бита и самое раннее возможное начало стартового бита следующего кадра.

Рисунок 22-7.Выборка стоп-бита и выборка следующего стартового бита



Для стопового бита применяется такое же большинство голосов, как и для других битов кадра. Если для стопового бита зарегистрировано значение логического 0, будет установлен флаг ошибки кадра (FEn).

Новый переход с высокого на низкий, указывающий на начальный бит нового кадра, может появиться сразу после последнего бита, используемого для голосования большинством. В режиме нормальной скорости первая выборка низкого уровня может быть в точке, отмеченной (А) на Рис. 22-7 на стр. 214. Для режима двойной скорости первый низкий уровень должен быть задержан до (В). (С) обозначает стоповый бит полной длины. Обнаружение раннего стартового бита влияет на рабочий диапазон приемника.

22.8.3 Асинхронный рабочий диапазон

Рабочий диапазон приемника зависит от несоответствия между принимаемой скоростью передачи данных и внутренней скоростью передачи данных. Если Передатчик отправляет кадры со слишком высокой или слишком низкой скоростью передачи данных, или внутренняя скорость передачи данных Приемника не имеет аналогичного значения (см. Таблица 22-2) базовой частоты, приемник не сможет синхронизировать кадры по начальному биту.

Следующие уравнения можно использовать для расчета соотношения скорости входящих данных и внутренней скорости передачи данных приемника.

$$r_{медленный} = \frac{D+1-C}{C-1+D-C+S\Phi}$$

$$r_{быстрый} = \frac{D+2-C}{D+1-C+S\Phi}$$

Д Сумма размера символа и размера четности (D = от 5 до 10 бит).

С Выборки на бит. S = 16 для режима нормальной скорости и S = 8 для режима двойной скорости.

Сф Номер первого образца, используемый для голосования по большинству. Сф= 8 для нормальной скорости и Сф= 4 для двойной скорости режим.

См Средний номер выборки, используемый для голосования по большинству. См= 9 для нормальной скорости и См= 5 для двойного Скоростной режим.

$r_{медленный}$ отношение наименьшей скорости входящих данных, которая может быть принята, по отношению к скорости приемника оценивать.

$r_{быстро}$ отношение максимальной скорости входящих данных, которая может быть принята, к скорости передачи приемника.

Таблица 22-2а также Табл. 22-3 на стр. 216 укажите максимальную допустимую ошибку скорости передачи данных приемника. Обратите внимание, что режим нормальной скорости более устойчив к изменениям скорости передачи данных.

Таблица 22-2. Рекомендуемая максимальная ошибка скорости передачи приемника для режима нормальной скорости (U2Xn = 0)

Д # (данные + бит четности)	r _{медленный} (%)	r _{быстро} (%)	Максимум. общая ошибка (%)	Рекомендуемая макс. ошибка приемника (%)
5	93,20	106,67	+ 6,67/-6,8	±3,0
6	94,12	105,79	+ 5,79/-5,88	±2,5
7	94,81	105.11	+ 5,11/-5,19	±2,0
8	95,36	104,58	+ 4,58/-4,54	±2,0
9	95,81	104.14	+ 4,14/-4,19	±1,5
10	96,17	103,78	+ 3,78/-3,83	±1,5

Таблица 22-3. Рекомендуемая максимальная ошибка скорости передачи приемника для двухскоростного режима (U2Xn = 1)

Д # (данные + бит четности)	рмедленный(%)	рбыстро(%)	Максимум. общая ошибка (%)	Рекомендуемая макс. получатель ошибка (%)
5	94,12	105,66	+ 5,66/-5,88	±2,5
6	94,92	104,92	+ 4,92/-5,08	±2,0
7	95,52	104,35	+ 4,35/-4,48	±1,5
8	96,00	103,90	+ 3,90/-4,00	±1,5
9	96,39	103,53	+ 3,53/-3,61	±1,5
10	96,70	103,23	+ 3,23/-3,30	±1,0

Рекомендации по максимальной ошибке скорости передачи приемника были сделаны в предположении, что приемник и передатчик поровну делят максимальную общую ошибку.

Есть два возможных источника ошибки скорости передачи приемника. Системные часы приемника (XTAL) всегда будут иметь незначительную нестабильность в диапазоне напряжения питания и диапазона температур. При использовании кристалла для генерации системных часов это редко является проблемой, но для резонатора системные часы могут отличаться более чем на 2% в зависимости от допуска резонаторов. Второй источник ошибки более контролируем. Генератор скорости передачи данных не всегда может выполнить точное деление системной частоты для получения желаемой скорости передачи данных. В этом случае, если возможно, можно использовать значение UBRR, которое дает приемлемую низкую ошибку.

22.9 Многопроцессорный режим связи

Установка бита режима многопроцессорной связи (MPCMn) в UCSRnA включает функцию фильтрации входящих кадров, полученных приемником USART. Кадры, не содержащие адресной информации, будут игнорироваться и не помещаться в приемный буфер. Это эффективно уменьшает количество входящих кадров, которые должен обрабатывать ЦП, в системе с несколькими MCU, которые взаимодействуют через одну и ту же последовательную шину. Настройка MPCMn не влияет на датчик, но его следует использовать по-другому, если он является частью системы, использующей режим многопроцессорной связи.

Если приемник настроен на прием кадров, содержащих от 5 до 8 битов данных, то первый стоповый бит указывает, содержит ли кадр данные или адресную информацию. Если приемник настроен на кадры с девятью битами данных, то девятый бит (RXB8n) используется для идентификации кадров адреса и данных. Когда бит типа кадра (первый стоповый или девятый бит) равен единице, кадр содержит адрес. Когда бит типа кадра равен нулю, кадр является кадром данных.

Режим многопроцессорной связи позволяет нескольким ведомым MCU получать данные от ведущего MCU. Это делается путем декодирования кадра адреса, чтобы выяснить, к какому MCU был адресован. Если был адресован конкретный подчиненный MCU, он будет получать следующие кадры данных как обычно, в то время как другие подчиненные MCU будут игнорировать полученные кадры до тех пор, пока не будет получен другой адресный кадр.

22.9.1 Использование MPCMn

Чтобы MCU действовал как ведущий MCU, он может использовать 9-битный формат символьного кадра (UCSZn = 7). Девятый бит (TXB8n) должен быть установлен при передаче кадра адреса (TXB8n = 1) или очищен при передаче кадра данных (TXB = 0). В этом случае ведомые MCU должны быть настроены на использование 9-битного формата символьного кадра.

Для обмена данными в режиме многопроцессорной связи следует использовать следующую процедуру:

1. Все ведомые MCU находятся в режиме многопроцессорной связи (установлено MPCMn в UCSRnA).
2. Главный MCU отправляет кадр адреса, и все подчиненные устройства получают и читают этот кадр. В ведомых MCU флаг RXCn в UCSRnA будет установлен как обычно.
3. Каждый ведомый MCU считывает регистр UDRn и определяет, был ли он выбран. Если это так, он очищает бит MPCMn в UCSRnA, в противном случае он ожидает следующего байта адреса и сохраняет настройку MPCMn.

4. Адресуемый MCU будет получать все кадры данных до тех пор, пока не будет получен новый адресный кадр. Другие ведомые MCU, у которых все еще установлен бит MPCMp, будут игнорировать кадры данных.

5. Когда последний фрейм данных получен адресуемым MCU, адресованный MCU устанавливает бит MPCMp и ожидает от ведущего нового кадра адреса. Затем процесс повторяется с 2.

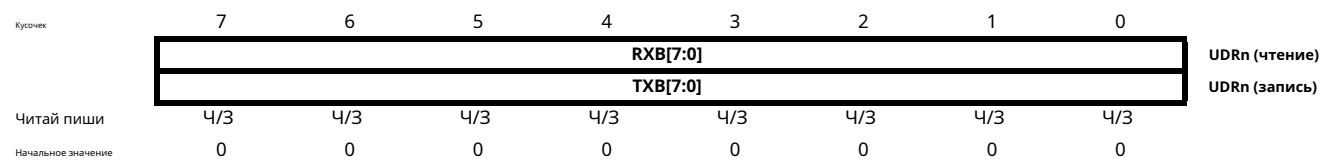
Использование любого из форматов кадра символов от 5 до 8 бит возможно, но нецелесообразно, поскольку приемник должен переключаться между использованием форматов кадра символов n и $n+1$. Это затрудняет работу в полнодуплексном режиме, поскольку передатчик и приемник используют одинаковые настройки размера символов. Если используются кадры символов от 5 до 8 бит, передатчик должен быть настроен на использование двух стоповых битов ($USBSn = 1$), поскольку первый стоповый бит используется для указания типа кадра.

Не используйте инструкции чтения-модификации-записи (SBI и CBI) для установки или сброса бита MPCMp. Бит MPCMp находится в том же месте ввода-вывода, что и флаг TXCn, и он может быть случайно сброшен при использовании инструкций SBI или CBI.

22.10 Описание регистра

В следующем разделе описываются регистры USART.

22.10.1 UDRn — Регистр данных ввода/вывода USART n



Регистр буфера передачи данных USART и регистр буфера приема данных USART имеют один и тот же адрес ввода-вывода, называемый регистром данных USART или UDRn. Регистр буфера передачи данных (TXB) будет местом назначения для данных, записанных в ячейку регистра UDRn. Чтение местоположения регистра UDRn вернет содержимое регистра буфера приема данных (RXB).

Для 5-битных, 6-битных или 7-битных символов неиспользуемые старшие биты будут игнорироваться передатчиком и устанавливаться в ноль приемником.

Буфер передачи может быть записан только тогда, когда флаг UDREn в регистре UCSRnA установлен. Данные, записываемые в UDRn, когда флаг UDREn не установлен, будут игнорироваться передатчиком USART. Когда данные записываются в буфер передачи и передатчик включен, передатчик загружает данные в сдвиговый регистр передачи, когда сдвиговый регистр пуст. Затем данные будут последовательно передаваться на вывод TxDn.

Буфер приема состоит из двухуровневого FIFO. FIFO будет изменять свое состояние всякий раз, когда осуществляется доступ к приемному буферу. Из-за такого поведения буфера приема не используйте инструкции чтения-модификации-записи (SBI и CBI) в этом месте. Будьте осторожны при использовании команд проверки битов (SBIC и SBIS), так как они также изменяют состояние FIFO.

22.10.2 UCSRnA — Регистр управления и состояния USART A

Кусочек	7	6	5	4	3	2	1	0	
	RXCn	TXCn	UDREN	FEEn	DORn	UPEn	U2Xn	MPCMn	UCSRnA
Читай пиши	р	Ч/З	р	р	р	р	Ч/З	Ч/З	
Начальное значение	0	0	1	0	0	0	0	0	

• Бит 7 – RXCn: прием USART завершен

Этот бит флага устанавливается, когда в приемном буфере есть непрочитанные данные, и сбрасывается, когда приемный буфер пуст (то есть не содержит непрочитанных данных). Если приемник отключен, буфер приема будет сброшен, и, следовательно, бит RXCn станет равным нулю. Флаг RXCn может использоваться для генерации прерывания о завершении приема (см. описание бита RXCIEн).

• Бит 6 – TXCn: передача USART завершена

Этот бит флага устанавливается, когда весь кадр в регистре сдвига передачи был сдвинут, и в буфере передачи (UDRn) в настоящее время нет новых данных. Бит флага TXCn автоматически очищается, когда выполняется прерывание завершения передачи, или его можно очистить, записав единицу в его битовую позицию. Флаг TXCn может генерировать прерывание завершения передачи (см. описание бита TXCIEн).

• Бит 5 – UDREN: регистр данных USART пуст.

Флаг UDREN указывает, готов ли буфер передачи (UDRn) к приему новых данных. Если UDREN равен единице, буфер пуст и, следовательно, готов к записи. Флаг UDREN может генерировать прерывание о пустом регистре данных (см. описание бита UDRIEn).

UDREN устанавливается после сброса, чтобы указать, что датчик готов.

• Бит 4 – FEEn: ошибка кадра

Этот бит устанавливается, если следующий символ в буфере приема имел ошибку кадра при приеме, то есть когда первый стоповый бит следующего символа в буфере приема равен нулю. Этот бит действителен до тех пор, пока буфер приема (UDRn) не будет прочитан. Бит FEEn равен нулю, когда стоповый бит полученных данных равен единице. Всегда устанавливайте этот бит в ноль при записи в UCSRnA.

• Бит 3 – DORn: переполнение данных

Этот бит устанавливается, если обнаружено состояние Data OverRun. Data OverRun возникает, когда приемный буфер заполнен (два символа), это новый символ, ожидающий в приемном сдвиговом регистре, и обнаружен новый стартовый бит. Этот бит действителен до тех пор, пока буфер приема (UDRn) не будет прочитан. Всегда устанавливайте этот бит в ноль при записи в UCSRnA.

• Бит 2 — UPEн: ошибка четности USART

Этот бит устанавливается, если следующий символ в приемном буфере имел ошибку четности при приеме и в этот момент была включена проверка четности (UPMn1 = 1). Этот бит действителен до тех пор, пока буфер приема (UDRn) не будет прочитан. Всегда устанавливайте этот бит в ноль при записи в UCSRnA.

• Бит 1 – U2Xn: удвоение скорости передачи USART.

Этот бит влияет только на асинхронную операцию. Запишите этот бит в ноль при использовании синхронной операции.

Запись этого бита в единицу уменьшит делитель скорости передачи с 16 до 8, эффективно удваивая скорость передачи для асинхронной связи.

• Бит 0 – MPCMn: режим многопроцессорной связи.

Этот бит включает режим многопроцессорной связи. Когда бит MPCMn записывается в единицу, все входящие кадры, полученные приемником USART, которые не содержат информации об адресе, будут игнорироваться. Настройка MPCMn не влияет на передатчик. Для получения более подробной информации см. [«Многопроцессорный режим связи» на стр. 216](#).

22.10.3 UCSRnB — Регистр управления и состояния USART n B

Кусочек	7	6	5	4	3	2	1	0	
	RXCIEн	TXCIEн	UDRIEn	RXENн	TXENн	UCSZн2	RXB8н	TXB8н	UCSRnB
Читай пиши	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	р	Ч/З	
Начальное значение	0	0	0	0	0	0	0	0	

• **Бит 7 – RXCIEн: Разрешение прерывания завершения приема n**
Запись этого бита в единицу разрешает прерывание по флагу RXCн. Прерывание завершения приема USART будет генерироваться только в том случае, если бит RXCIEн установлен в единицу, флаг глобального прерывания в SREG установлен в единицу и бит RXCн в UCSRnA установлен.

• **Бит 6 – TXCIEн: Разрешение прерывания завершения передачи n**
Запись этого бита в единицу разрешает прерывание по флагу TXCн. Прерывание завершения передачи USART будет генерироваться только в том случае, если бит TXCIEн установлен в единицу, флаг глобального прерывания в SREG установлен в единицу и установлен бит TXCн в UCSRnA.

• **Бит 5 – UDRIEn: разрешение прерывания пустого регистра данных USART n**
Запись этого бита в единицу разрешает прерывание по флагу UDREн. Прерывание о пустом регистре данных будет генерироваться только в том случае, если бит UDRIEn установлен в единицу, флаг глобального прерывания в SREG установлен в единицу и установлен бит UDREн в UCSRnA.

• **Бит 4 – RXENн: приемник включен n**
Запись этого бита в единицу включает приемник USART. Приемник отменит нормальную работу порта для вывода RxDн, если он включен. Отключение приемника приведет к очистке буфера приема, что сделает недействительными флаги FEn, DORн и UPEn.

• **Бит 3 – TXENн: передатчик включен n**
Запись этого бита в единицу включает передатчик USART. Передатчик отменит нормальную работу порта для вывода TxDн, если он включен. Отключение передатчика (запись TXENн в ноль) не вступит в силу до тех пор, пока не будут завершены текущие и ожидающие передачи, то есть когда регистр сдвига передачи и регистр буфера передачи не содержат данных для передачи. При отключении передатчик больше не будет переопределять порт TxDн.

• **Бит 2 – UCSZн2: Размер символа n**
Биты UCSZн2 в сочетании с битом UCSZн1:0 в UCSRnC устанавливают количество битов данных (Character Size) в кадре, используемом приемником и передатчиком.

• **Бит 1 – RXB8н: прием данных, бит 8 n**
RXB8н — это девятый бит данных полученного символа при работе с последовательными кадрами с девятью битами данных. Должен быть прочитан перед чтением младших битов из UDRн.

• **Бит 0 – TXB8н: передача данных, бит 8 n**
TXB8н — это девятый бит данных в передаваемом символе при работе с последовательными кадрами с девятью битами данных. Должен быть записан перед записью младших битов в UDRн.

22.10.4 UCSRnC — Регистр управления и состояния USART n C

Кусочек	7	6	5	4	3	2	1	0	UCSRnC
	UMSELn1	UMSELn0	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	
Читай пиши	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	
Начальное значение	0	0	0	0	0	1	1	0	

• Биты 7:6 – UMSELn1:0 Выбор режима USART

Эти биты выбирают режим работы USARTn, как показано на рис. [Таблица 22-4](#).

Таблица 22-4. Настройки битов UMSELn

UMSELn1	UMSELn0	Режим
0	0	Асинхронный USART
0	1	Синхронный USART
1	0	(Сдержанный)
1	1	Мастер SPI (MSPIM) ⁽¹⁾

Примечание: 1. См. «USART в режиме SPI» на стр. 227 для полного описания работы режима Master SPI (MSPIM).

• Биты 5:4 – UPMn1:0: Режим четности

Эти биты включают и устанавливают тип генерации и проверки четности. Если эта функция включена, передатчик будет автоматически генерировать и отправлять четность передаваемых битов данных в каждом кадре. Получатель сгенерирует значение четности для входящих данных и сравнит его с настройкой UPMn. При обнаружении несоответствия будет установлен флаг UPEn в UCSRnA.

Таблица 22-5. Настройки битов UPMn

UPMn1	UPMn0	Режим четности
0	0	Неполноценный
0	1	Сдержанный
1	0	Включено, даже паритет
1	1	Включено, нечетная четность

• Бит 3 – USBSn: выбор стопового бита

Этот бит выбирает количество стоповых битов, которые должны быть вставлены преобразователем. Получатель игнорирует этот параметр.

Таблица 22-6. Настройки битов USBS

USBSn	Стоповые биты
0	1 бит
1	2-битный

• Бит 2:1 – UCSZn1:0: Размер символа

Биты UCSZn1:0 в сочетании с битом UCSZn2 в UCSRnB устанавливают количество битов данных (размер символа) в кадре, используемом приемником и передатчиком.

Таблица 22-7. Настройки битов UCSZn

УКСЗn2	UCSZn1	UCSZn0	Размер символа
0	0	0	5-битный
0	0	1	6-битный
0	1	0	7-битный
0	1	1	8-битный
1	0	0	Сдержанный
1	0	1	Сдержанный
1	1	0	Сдержанный
1	1	1	9-битный

• Бит 0 – UCSPOLn: полярность синхронизации

Этот бит используется только для синхронного режима. Запишите этот бит в ноль, когда используется асинхронный режим. Бит UCSPOLn устанавливает взаимосвязь между изменением выходных данных и выборкой входных данных и синхронными часами (XCKn).

Таблица 22-8. Настройки бита UCSPOLn

UCSPOLn	Передаваемые данные изменены (Выход вывода TxDn)	Образец полученных данных (Вход на контакте RxDn)
0	Восходящий край XCKn	Падение XCKn Edge
1	Падение XCKn Edge	Восходящий край XCKn

22.10.5 UBRRnL и UBRRnH — регистры скорости передачи USART

Кусочек	15	14	13	12	11	10	9	8	УБРРН
	-	-	-	-	УБРР[11:8]				
	УБРР[7:0]								УБРРЛ
	7	6	5	4	3	2	1	0	
Читай пиши	р	р	р	р	Ч/З	Ч/З	Ч/З	Ч/З	
	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	
Начальное значение	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

• Бит 15:12 – Зарезервированные биты

Эти биты зарезервированы для использования в будущем. Для совместимости с будущими устройствами эти биты должны быть записаны в ноль при записи UBRRH.

• Бит 11:0 – UBRR11:0: Регистр скорости передачи USART

Это 12-битный регистр, который содержит скорость передачи данных USART. UBRRH содержит четыре старших бита, а UBRRL содержит восемь младших битов скорости передачи данных USART. Текущие передачи передатчика и приемника будут повреждены, если скорость передачи данных изменится. Запись UBRRL вызовет немедленное обновление прескалера скорости передачи.

22.11 Примеры настройки скорости передачи

Для стандартных частот кварца и резонатора наиболее часто используемые скорости передачи для асинхронной работы могут быть сгенерированы с помощью настроек UBRR в Таблица 22-9 Табл. 22-12 на стр. 226. Значения UBRR, которые дают фактическую скорость передачи данных, отличающуюся менее чем на 0,5% от целевой скорости передачи данных, выделены в таблице жирным шрифтом. Допустимы более высокие рейтинги ошибок, но приемник будет иметь меньшую помехоустойчивость при высоких рейтингах ошибок, особенно для больших последовательных кадров (см. «Асинхронный рабочий диапазон» на стр. 215). Значения ошибок рассчитываются по следующему уравнению:

Ошибка[%] = $\frac{\text{BaudRate} - \text{Целевая скорость}}{\text{BaudRate}} \cdot 100\%$

Таблица 22-9. Примеры настроек UBRRn для часто используемых частот генератора

Бод <small>Оценивать [бит/с]</small>	ф _{оск} = 1,0000 МГц				ф _{оск} = 1,8432 МГц				ф _{оск} = 2,0000 МГц			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	УБРР	Ошибка	УБРР	Ошибка	УБРР	Ошибка	УБРР	Ошибка	УБРР	Ошибка	УБРР	Ошибка
2400	25	0,2%	51	0,2%	47	0,0%	95	0,0%	51	0,2%	103	0,2%
4800	12	0,2%	25	0,2%	23	0,0%	47	0,0%	25	0,2%	51	0,2%
9600	6	- 7,0%	12	0,2%	11	0,0%	23	0,0%	12	0,2%	25	0,2%
14,4 К	3	8,5%	8	- 3,5%	7	0,0%	15	0,0%	8	- 3,5%	16	2,1%
19,2 тыс.	2	8,5%	6	- 7,0%	5	0,0%	11	0,0%	6	- 7,0%	12	0,2%
28,8 К	1	8,5%	3	8,5%	3	0,0%	7	0,0%	3	8,5%	8	- 3,5%
38,4 К	1	- 18,6%	2	8,5%	2	0,0%	5	0,0%	2	8,5%	6	- 7,0%
57,6 К	0	8,5%	1	8,5%	1	0,0%	3	0,0%	1	8,5%	3	8,5%
76,8 К	–	–	1	- 18,6%	1	- 25,0%	2	0,0%	1	- 18,6%	2	8,5%
115,2 тыс.	–	–	0	8,5%	0	0,0%	1	0,0%	0	8,5%	1	8,5%
230,4К	–	–	–	–	–	–	0	0,0%	–	–	–	–
250 тыс.	–	–	–	–	–	–	–	–	–	–	0	0,0%
Максимум.(1)	62,5 Кбит/с		125Кбит/с		115,2 Кбит/с		230,4 Кбит/с		125Кбит/с		250Кбит/с	

Примечание: 1. UBRR = 0, ошибка = 0,0%

Таблица 22-10. Примеры настроек UBRRn для часто используемых частот генератора

Бод <small>Оценивать [бит/с]</small>	ф _{оск} = 3,6864 МГц				ф _{оск} = 4,0000 МГц				ф _{оск} = 7,3728 МГц			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	УБРР	Ошибка	УБРР	Ошибка	УБРР	Ошибка	УБРР	Ошибка	УБРР	Ошибка	УБРР	Ошибка
2400	95	0,0%	191	0,0%	103	0,2%	207	0,2%	191	0,0%	383	0,0%
4800	47	0,0%	95	0,0%	51	0,2%	103	0,2%	95	0,0%	191	0,0%
9600	23	0,0%	47	0,0%	25	0,2%	51	0,2%	47	0,0%	95	0,0%
14,4 К	15	0,0%	31	0,0%	16	2,1%	34	- 0,8%	31	0,0%	63	0,0%
19,2 тыс.	11	0,0%	23	0,0%	12	0,2%	25	0,2%	23	0,0%	47	0,0%
28,8 К	7	0,0%	15	0,0%	8	- 3,5%	16	2,1%	15	0,0%	31	0,0%
38,4 К	5	0,0%	11	0,0%	6	- 7,0%	12	0,2%	11	0,0%	23	0,0%
57,6 К	3	0,0%	7	0,0%	3	8,5%	8	- 3,5%	7	0,0%	15	0,0%
76,8 К	2	0,0%	5	0,0%	2	8,5%	6	- 7,0%	5	0,0%	11	0,0%
115,2 тыс.	1	0,0%	3	0,0%	1	8,5%	3	8,5%	3	0,0%	7	0,0%
230,4К	0	0,0%	1	0,0%	0	8,5%	1	8,5%	1	0,0%	3	0,0%
250 тыс.	0	- 7,8%	1	- 7,8%	0	0,0%	1	0,0%	1	- 7,8%	3	- 7,8%
0,5М	–	–	0	- 7,8%	–	–	0	0,0%	0	- 7,8%	1	- 7,8%
1М	–	–	–	–	–	–	–	–	–	–	0	- 7,8%
Максимум ⁽¹⁾	230,4 Кбит/с		460,8 Кбит/с		250Кбит/с		0,5 Мбит/с		460,8 Кбит/с		921,6 Кбит/с	

Примечание: 1. UBRR = 0, ошибка = 0,0%

Таблица 22-11. Примеры настроек UBRRn для часто используемых частот генератора

Бод <small>Оценивать</small> [бит/с]	ф _{оск} = 8,0000 МГц				ф _{оск} = 11,0592 МГц				ф _{оск} = 14,7456 МГц			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	УБРР	Ошибка	УБРР	Ошибка	УБРР	Ошибка	УБРР	Ошибка	УБРР	Ошибка	УБРР	Ошибка
2400	207	0,2%	416	- 0,1%	287	0,0%	575	0,0%	383	0,0%	767	0,0%
4800	103	0,2%	207	0,2%	143	0,0%	287	0,0%	191	0,0%	383	0,0%
9600	51	0,2%	103	0,2%	71	0,0%	143	0,0%	95	0,0%	191	0,0%
14,4 К	34	- 0,8%	68	0,6%	47	0,0%	95	0,0%	63	0,0%	127	0,0%
19,2 тыс.	25	0,2%	51	0,2%	35	0,0%	71	0,0%	47	0,0%	95	0,0%
28,8 К	16	2,1%	34	- 0,8%	23	0,0%	47	0,0%	31	0,0%	63	0,0%
38,4 К	12	0,2%	25	0,2%	17	0,0%	35	0,0%	23	0,0%	47	0,0%
57,6 К	8	- 3,5%	16	2,1%	11	0,0%	23	0,0%	15	0,0%	31	0,0%
76,8 К	6	- 7,0%	12	0,2%	8	0,0%	17	0,0%	11	0,0%	23	0,0%
115,2 тыс.	3	8,5%	8	- 3,5%	5	0,0%	11	0,0%	7	0,0%	15	0,0%
230,4К	1	8,5%	3	8,5%	2	0,0%	5	0,0%	3	0,0%	7	0,0%
250 тыс.	1	0,0%	3	0,0%	2	- 7,8%	5	- 7,8%	3	- 7,8%	6	5,3%
0,5М	0	0,0%	1	0,0%	–	–	2	- 7,8%	1	- 7,8%	3	- 7,8%
1М	–	–	0	0,0%	–	–	–	–	0	- 7,8%	1	- 7,8%
Максимум(1)	0,5 Мбит/с		1 Мбит/с		691,2 Кбит/с		1,3824 Мбит/с		921,6 Кбит/с		1,8432 Мбит/с	

Примечание: 1. UBRR = 0, ошибка = 0,0%

Таблица 22-12. Примеры настроек UBRRn для часто используемых частот генератора

Бод <small>Оценивать [бит/с]</small>	ф _{оск} = 16,0000 МГц				ф _{оск} = 18,4320 МГц				ф _{оск} = 20,0000 МГц			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	УБРР	Ошибка	УБРР	Ошибка	УБРР	Ошибка	УБРР	Ошибка	УБРР	Ошибка	УБРР	Ошибка
2400	416	- 0,1%	832	0,0%	479	0,0%	959	0,0%	520	0,0%	1041	0,0%
4800	207	0,2%	416	- 0,1%	239	0,0%	479	0,0%	259	0,2%	520	0,0%
9600	103	0,2%	207	0,2%	119	0,0%	239	0,0%	129	0,2%	259	0,2%
14,4 К	68	0,6%	138	- 0,1%	79	0,0%	159	0,0%	86	- 0,2%	173	- 0,2%
19,2 тыс.	51	0,2%	103	0,2%	59	0,0%	119	0,0%	64	0,2%	129	0,2%
28,8 К	34	- 0,8%	68	0,6%	39	0,0%	79	0,0%	42	0,9%	86	- 0,2%
38,4 К	25	0,2%	51	0,2%	29	0,0%	59	0,0%	32	- 1,4%	64	0,2%
57,6 К	16	2,1%	34	- 0,8%	19	0,0%	39	0,0%	21	- 1,4%	42	0,9%
76,8 К	12	0,2%	25	0,2%	14	0,0%	29	0,0%	15	1,7%	32	- 1,4%
115,2 тыс.	8	- 3,5%	16	2,1%	9	0,0%	19	0,0%	10	- 1,4%	21	- 1,4%
230,4К	3	8,5%	8	- 3,5%	4	0,0%	9	0,0%	4	8,5%	10	- 1,4%
250 тыс.	3	0,0%	7	0,0%	4	- 7,8%	8	2,4%	4	0,0%	9	0,0%
0,5М	1	0,0%	3	0,0%	–	–	4	- 7,8%	–	–	4	0,0%
1М	0	0,0%	1	0,0%	–	–	–	–	–	–	–	–
Максимум(1)	1 Мбит/с		2 Мбит/с		1,152 Мбит/с		2,304 Мбит/с		1,25 Мбит/с		2,5 Мбит/с	

Примечание: 1. UBRR = 0, ошибка = 0,0%

23. USART в режиме SPI

Универсальный синхронный и асинхронный последовательный приемник и передатчик (USART) можно настроить на режим работы, совместимый с главным SPI. Режим Master SPI (MSPIM) имеет следующие особенности:

- Полный дуплекс, трехпроводная синхронная передача данных
- Основная операция
- Поддерживает все четыре режима работы SPI (режим 0, 1, 2 и 3)
- LSB First или MSB First Передача данных (настраиваемый порядок данных)
- Операция в очереди (двойная буферизация)
- Генератор скорости передачи данных с высоким разрешением
- Работа на высокой скорости ($f_{XCKmax} = f_{CK}/2$)
- Гибкая генерация прерываний

23.1 Обзор

Установка обоих битов UMSELn1:0 в единицу включает USART в логике MSPIM. В этом режиме работы основная управляющая логика SPI получает прямой контроль над ресурсами USART. Эти ресурсы включают сдвиговый регистр и буферы передатчика и приемника, а также генератор скорости передачи данных. Генератор и проверка четности, логика восстановления данных и часов, а также логика управления RX и TX отключены. Логика управления USART RX и TX заменена общей логикой управления передачей SPI. Однако логика управления контактами и логика генерации прерываний идентичны в обоих режимах работы.

Расположение регистров ввода-вывода одинаково в обоих режимах. Однако некоторые функции регистров управления изменяются при использовании MSPIM.

23.2 Сравнение USART MSPIM и SPI

AVR USART в режиме MSPIM полностью совместим с AVR SPI в отношении:

- Временная диаграмма ведущего режима
- Функциональность бита UCPOln идентична биту SPI CPOL.
- Функциональность бита UCPHAn идентична биту SPI CPHA.
- Функциональность бита UDORDn идентична биту SPI DORD.

Однако, поскольку USART в режиме MSPIM повторно использует ресурсы USART, использование USART в режиме MSPIM несколько отличается от использования SPI. Помимо различий в битах регистра управления и того, что USART в режиме MSPIM поддерживает только основную операцию, два модуля различаются следующими функциями:

- USART в режиме MSPIM включает (двойную) буферизацию передатчика. SPI не имеет буфера
- Приемник USART в режиме MSPIM включает дополнительный уровень буфера
- Бит SPI WCOL (Write Collision) не включен в USART в режиме MSPIM.
- Бит режима двойной скорости SPI (SPI2X) не включен. Однако тот же эффект достигается соответствующей установкой UBRRn
- Время прерывания несовместимо
- Управление контактами отличается из-за того, что USART работает только ведущим в режиме MSPIM.

Сравнение USART в режиме MSPIM и выводов SPI показано на рис. [Табл. 23-4 на стр. 235](#).

23.2.1 Генерация тактового сигнала

Логика генерации часов генерирует базовые часы для передатчика и приемника. Для режима работы USART MSPIM поддерживается только внутренняя генерация тактовых импульсов (т. е. основная операция). Следовательно, регистр направления данных для контакта XCKn (DDR_XCKn) должен быть установлен в единицу (то есть как выход), чтобы USART в MSPIM работал.

правильно. Предпочтительно, чтобы DDR_XCKn был настроен до включения USART в MSPIM (то есть биты TXENn и RXENn установлены в единицу).

Генерация внутренних часов, используемая в режиме MSPIM, идентична синхронному ведущему режиму USART. Таким образом, скорость передачи данных или настройку UBRRn можно рассчитать с использованием тех же уравнений, см.Таблица 23-1.

Таблица 23-1. Уравнения для расчета настройки регистра скорости передачи данных

Режим работы	Уравнение для расчета скорости передачи(1)	Уравнение для расчета значения UBRRn
Синхронный мастер-режим	$БОД_{знак} равно \frac{\phi}{2 \cdot UBRRn + 1}$	$UBRRn_{знак} равно \frac{\phi}{2 \cdot БОД} - 1$

Примечание: 1. Скорость передачи определяется как скорость передачи в битах в секунду (бит/с).

- БОД**
Скорость передачи данных (в битах в секунду, bps).
- фоск**
Тактовая частота системного генератора.
- UBRRn**
Содержимое регистров UBRRnH и UBRRnL (0-4095).

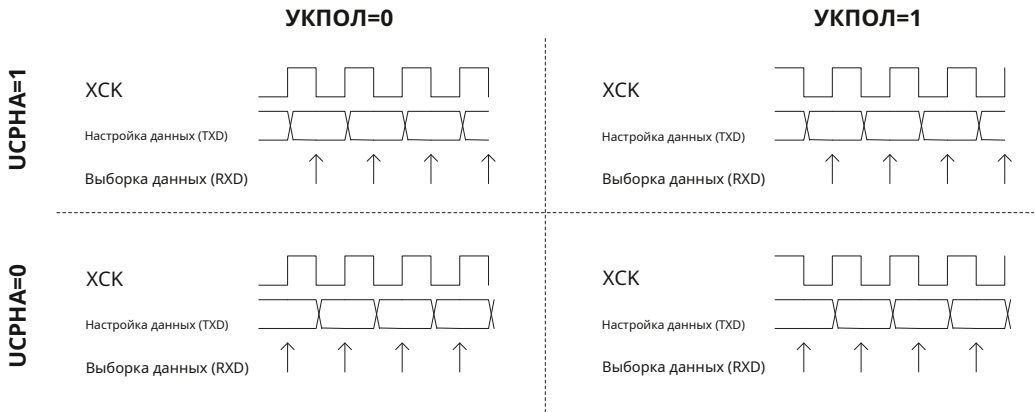
23.3 Режимы данных SPI и синхронизация

Существует четыре комбинации фазы и полярности XCKn (SCK) по отношению к последовательным данным, которые определяются управляющими битами UCSRNn и UCSPOLn. Временные диаграммы передачи данных показаны наРисунок 23-1. Биты данных сдвигаются и фиксируются на противоположных фронтах сигнала XCKn, обеспечивая достаточное время для стабилизации сигналов данных. Функциональность UCSPOLn и UCSRNn кратко изложена вТаблица 23-2. Обратите внимание, что изменение настройки любого из этих битов приведет к повреждению всей текущей связи как для приемника, так и для передатчика.

Таблица 23-2. Функциональность UCSPOLn и UCSRNn.

UCSPOLn	UCSRNn	SPI-режим	Передний край	Задний край
0	0	0	Образец (восходящий)	Настройка (падение)
0	1	1	Настройка (восходящая)	Образец (падение)
1	0	2	Образец (падение)	Настройка (восходящая)
1	1	3	Настройка (падение)	Образец (восходящий)

Рисунок 23-1.Временные диаграммы передачи данных UCSRNn и UCSPOLn.



23.4 Форматы кадров

Последовательный кадр для MSPIM определяется как один символ из 8 битов данных. USART в режиме MSPIM имеет два допустимых формата кадра:

- 8-битные данные с первым старшим битом
- 8-битные данные с первым младшим битом

Кадр начинается с младшего или старшего бита данных. Затем следуют следующие биты данных, всего до восьми, заканчивающиеся соответственно старшим или наименее значащим битом. Когда передается полный кадр, за ним может непосредственно следовать новый кадр, или линия связи может быть переведена в состояние ожидания (высокий уровень).

Бит UDORDn в UCSRnC устанавливает формат кадра, используемый USART в режиме MSPIM. Приемник и передатчик используют одинаковые настройки. Обратите внимание, что изменение настройки любого из этих битов приведет к повреждению всей текущей связи как для приемника, так и для передатчика.

Передача 16-битных данных может быть достигнута путем записи двух байтов данных в UDRn. Затем прерывание завершения передачи UART будет сигнализировать о том, что 16-битное значение было смещено.

23.4.1 Инициализация USART MSPIM

USART в режиме MSPIM должен быть инициализирован до того, как может иметь место какая-либо связь. Процесс инициализации обычно состоит из установки скорости передачи данных, установки основного режима работы (путем установки DDR_XCKn в единицу), установки формата кадра и включения передатчика и приемника. Только передатчик может работать независимо. Для операции USART, управляемой прерыванием, флаг глобального прерывания должен быть очищен (и, таким образом, прерывания глобально отключены) при выполнении инициализации.

Примечание: Чтобы гарантировать немедленную инициализацию выхода XCKn, регистр скорости передачи данных (UBRRn) должен быть равен нулю в момент включения передатчика. В отличие от работы USART в нормальном режиме, UBRRn должен быть записан в нужное значение после включения передатчика, но до начала первой передачи. Установка UBRRn на ноль перед включением передатчика не требуется, если инициализация выполняется сразу после сброса, поскольку UBRRn сбрасывается на ноль.

Перед выполнением повторной инициализации с измененной скоростью передачи данных, режимом данных или форматом кадра убедитесь, что в период изменения регистров не ведется передача данных. Флаг TXCn можно использовать для проверки того, что передатчик завершил все передачи, а флаг RXCn можно использовать для проверки отсутствия непрочитанных данных в приемном буфере. Обратите внимание, что флаг TXCn должен очищаться перед каждой передачей (перед записью UDRn), если он используется для этой цели.

В следующих простых примерах кода инициализации USART показаны одна сборка и одна функция C, которые имеют одинаковую функциональность. В примерах предполагается опрос (прерывания не разрешены). Скорость передачи задается как функциональный параметр. Для ассемблерного кода предполагается, что параметр скорости передачи хранится в регистрах r17:r16.

Пример ассемблерного кода ⁽¹⁾
<pre> USART_Init: клрр18 внеУБРРnX, r18 внеУБРРnЛ, r18 ; Установка вывода порта XCKn в качестве вывода включает основной режим. сбиXCKn_DDR, XCKn ; Установите режим работы MSPI и режим данных SPI 0. лди r18, (1<<UMSELn1) (1<<UMSELn0) (0<<UCPHAn) (0<<UCPOLn) внеUCSRnC, r18 ; Включите приемник и передатчик. лди r18, (1<<RXENn) (1<<TXENn) внеUCSRnB, r18 ; Установите скорость передачи данных. ; ВАЖНО: Скорость передачи должна быть установлена после включения передатчика! внеУБРРnX, r17 внеУБРРnЛ, r18 рет </pre>
Пример кода C ⁽¹⁾
<pre> пустотаUSART_Init(беззнаковое целоебод) { УБРРn = 0; /* Установка вывода порта XCKn в качестве вывода, включает режим ведущего. */ XCKn_DDR = (1<<XCKn); /* Установить режим работы MSPI и режим данных SPI 0. */ UCSRnC = (1<<UMSELn1) (1<<UMSELn0) (0<<UCPHAn) (0<<UCPOLn); /* Включить приемник и передатчик. */ UCSRnB = (1<<RXENn) (1<<TXENn); /* Установить скорость передачи данных. */ /* ВАЖНО: Скорость передачи должна быть установлена после включения передатчика */ UBRRn = бод; } </pre>

Примечание: 1. См. «Примеры кода» на стр. 10.

23.5 Передача данных

Для использования USART в режиме MSPI требуется, чтобы передатчик был включен, то есть бит TXENn в регистре UCSRnB установлен в единицу. Когда передатчик включен, нормальная работа порта вывода TxDn переопределяется и получает функцию последовательного выхода передатчика. Включение приемника является необязательным и выполняется установкой бита RXENn в регистре UCSRnB в единицу. Когда приемник включен, нормальная работа вывода RxDn переопределяется и получает функцию последовательного входа приемника. XCKn в обоих случаях будет использоваться в качестве тактового генератора передачи.

После инициализации USART готов к передаче данных. Передача данных инициируется записью в ячейку ввода/вывода UDRn. Это относится как к отправке, так и к приему данных, поскольку передатчик управляет часами передачи. Данные, записанные в UDRn, перемещаются из буфера передачи в сдвиговый регистр, когда сдвиговый регистр готов к отправке нового кадра.

Примечание: Чтобы входной буфер синхронизировался с количеством переданных байтов данных, регистр UDRn должен считываться один раз для каждого переданного байта. Работа входного буфера идентична обычному режиму USART, то есть, если происходит переполнение, будет потерян последний полученный символ, а не первые данные в буфере. Это означает, что если передаются четыре байта, сначала байт 1, затем байты 2, 3 и 4, и UDRn не считывается до завершения всех передач, то будет потерян байт 3, который должен быть получен, а не байт 1.

В следующих примерах кода показана простая функция передачи USART в режиме MSPIM, основанная на опросе флага пустого регистра данных (UDREN) и флага завершения приема (RXCN). Перед использованием функции USART должен быть инициализирован. Для ассемблерного кода предполагается, что отправляемые данные хранятся в регистре R16, а полученные данные будут доступны в том же регистре (R16) после возврата из функции.

Функция просто ожидает, пока буфер передачи не станет пустым, проверяя флаг UDREN, прежде чем загрузить в него новые данные для передачи. Затем функция ожидает наличия данных в буфере приема, проверяя флаг RXCN, прежде чем читать буфер и возвращать значение.

Пример ассемблерного кода⁽¹⁾

```
USART_MSPIM_Transfer:
; Подождите, пока буфер передачи будет пуст.
сбисUCSRnA, UDREN
ржмпUSART_MSPIM_Transfer
; Помещает данные (r16) в буфер, отправляет данные
внеUDRn,r16
; Дождитесь получения данных
USART_MSPIM_Wait_RXCn:
сбисUCSRnA, RXCn
ржмпUSART_MSPIM_Wait_RXCn
; Получить и вернуть полученные данные из буфера
вр16, UDRn
рет
```

Пример кода C⁽¹⁾

```
беззнаковый символUSART_Receive(пустота) {

/* Ожидание опустошения буфера передачи
*/ пока (!(UCSRnA & (1<<UDREN)) );
/* Помещаем данные в буфер, отправляем данные */
UDRn = data;
/* Ожидаем получения данных */ пока !(
UCSRnA & (1<<RXCn)) );
/* Получить и вернуть полученные данные из буфера */
возвращатьсяUDRn;
}
```

Примечание: 1. См. «Примеры кода» на стр. 10.

23.5.1 Флаги и прерывания передатчика и приемника

Флаги RXCn, TXCn и UDREN и соответствующие прерывания в USART в режиме MSPIM по функциям идентичны нормальной работе USART. Однако флаги состояния ошибки приемника (FE, DOR и PE) не используются и всегда считываются как ноль.

23.5.2 Отключение передатчика или приемника

Отключение передатчика или приемника в режиме USART в режиме MSPIM по функциям идентично нормальной работе USART.

23.6 Описание регистра USART MSPIM

В следующем разделе описаны регистры, используемые для работы SPI с использованием USART.

23.6.1 UDRn — Регистр данных ввода/вывода USART MSPIM

Описание функций и битов регистра данных USART (UDRn) в режиме MSPIM идентично нормальной работе USART. См. «UDRn — регистр данных ввода/вывода USART n» на стр. 218.

23.6.2 UCSRnA — Регистр управления и состояния USART MSPIM n A

Кусочек	7	6	5	4	3	2	1	0	
	RXCn	TXCn	UDRen	-	-	-	-	-	UCSRnA
Читай пиши	Ч/З	Ч/З	Ч/З	р	р	р	р	р	
Начальное значение	0	0	0	0	0	1	1	0	

• Бит 7 — RXCn: прием USART завершен

Этот бит флага устанавливается, когда в приемном буфере есть непрочитанные данные, и сбрасывается, когда приемный буфер пуст (то есть не содержит непрочитанных данных). Если приемник отключен, буфер приема будет сброшен, и, следовательно, бит RXCn станет равным нулю. Флаг RXCn может использоваться для генерации прерывания о завершении приема (см. описание бита RXCIEн).

• Бит 6 — TXCn: передача USART завершена

Этот бит флага устанавливается, когда весь кадр в регистре сдвига передачи был сдвинут, и в буфере передачи (UDRn) в настоящее время нет новых данных. Бит флага TXCn автоматически очищается, когда выполняется прерывание завершения передачи, или его можно очистить, записав единицу в его битовую позицию. Флаг TXCn может генерировать прерывание завершения передачи (см. описание бита TXCIEн).

• Бит 5 — UDRen: регистр данных USART пуст.

Флаг UDRen указывает, готов ли буфер передачи (UDRn) к приему новых данных. Если UDRen равен единице, буфер пуст и, следовательно, готов к записи. Флаг UDRen может генерировать прерывание о пустом регистре данных (см. описание бита UDRIE). UDRen устанавливается после сброса, чтобы указать, что датчик готов.

• Бит 4:0 — зарезервированные биты в режиме MSPI

В режиме MSPI эти биты зарезервированы для использования в будущем. Для совместимости с будущими устройствами эти биты должны быть записаны в ноль при записи UCSRnA.

23.6.3 UCSRnB — регистр управления и состояния USART MSPIM n B

Кусочек	7	6	5	4	3	2	1	0	
	RXCIEн	TXCIEн	UDRIE	RXENн	TXENн	-	-	-	UCSRnB
Читай пиши	Ч/З	Ч/З	Ч/З	Ч/З	Ч/З	р	р	р	
Начальное значение	0	0	0	0	0	1	1	0	

• Бит 7 — RXCIEн: разрешение прерывания завершения приема

Запись этого бита в единицу разрешает прерывание по флагу RXCп. Прерывание завершения приема USART будет генерироваться только в том случае, если бит RXCIEн установлен в единицу, флаг глобального прерывания в SREG установлен в единицу и бит RXCп в UCSRnA установлен.

• Бит 6 — TXCIEн: разрешение прерывания завершения передачи

Запись этого бита в единицу разрешает прерывание по флагу TXCп. Прерывание завершения передачи USART будет генерироваться только в том случае, если бит TXCIEн установлен в единицу, флаг глобального прерывания в SREG установлен в единицу и установлен бит TXCп в UCSRnA.

• Бит 5 — UDRIE: разрешение прерывания пустого регистра данных USART

Запись этого бита в единицу разрешает прерывание по флагу UDREн. Прерывание о пустом регистре данных будет генерироваться только в том случае, если бит UDRIE установлен в единицу, флаг глобального прерывания в SREG установлен в единицу и бит UDREн в UCSRnA установлен.

• Бит 4 — RXENн: Включение приемника

Запись этого бита в единицу включает приемник USART в режиме MSPIM. Приемник отменит нормальную работу порта для вывода RxDп, если он включен. Отключение приемника приведет к очистке буфера приема. Только включение приемника в режиме MSPI (т. е. установка RXENн=1 и TXENн=0) не имеет смысла, поскольку именно передатчик управляет тактовой частотой передачи и поскольку поддерживается только ведущий режим.

• Бит 3 — TXENн: включение передатчика

Запись этого бита в единицу включает передатчик USART. Передатчик отменит нормальную работу порта для вывода TxDп, если он включен. Отключение передатчика (запись TXENн в ноль) не вступит в силу до тех пор, пока не будут завершены текущие и ожидающие передачи, то есть когда регистр сдвига передачи и регистр буфера передачи не содержат данных для передачи. При отключении передатчик больше не будет переопределять порт TxDп.

• Бит 2:0 — зарезервированные биты в режиме MSPI.

В режиме MSPI эти биты зарезервированы для использования в будущем. Для совместимости с будущими устройствами эти биты должны быть записаны в ноль при записи UCSRnB.

23.6.4 UCSRnC — Регистр управления и состояния USART MSPIM n C

Кусочек	7	6	5	4	3	2	1	0	
	UMSELn1	UMSELn0	-	-	-	UDORDn	UCPHAн	UCPOLn	UCSRnC
Читай пиши	Ч/З	Ч/З	р	р	р	Ч/З	Ч/З	Ч/З	
Начальное значение	0	0	0	0	0	1	1	0	

• Бит 7:6 - UMSELn1:0: Выбор режима USART

Эти биты выбирают режим работы USART, как показано на рис.Таблица 23-3. Видеть«UCSRnC — Регистр управления и состояния USART n C» на стр. 221для полного описания нормальной работы USART. MSPIM включен, когда оба бита UMSELn установлены в единицу. UDORDn, UCPHAн и UCPOLn могут быть установлены в той же операции записи, где включен MSPIM.

Таблица 23-3. Настройки битов UMSELn

UMSELn1	UMSELn0	Режим
0	0	Асинхронный USART
0	1	Синхронный USART
1	0	(Сдержанный)
1	1	Мастер SPI (MSPIM)

• Бит 5:3 — зарезервированные биты в режиме MSPI.

В режиме MSPI эти биты зарезервированы для использования в будущем. Для совместимости с будущими устройствами эти биты должны быть записаны в ноль при записи UCSRnC.

• Бит 2 — UDORDn: Порядок данных

Если установлено значение 1, первым передается младший бит слова данных. При нулевом значении старший бит слова данных передается первым. Ссылаться на«Режимы данных SPI и синхронизация» на стр. 228для деталей.

• Бит 1 — UCPHAн: фаза синхронизации

Установка бита UCPHAн определяет выборку данных на арендованном фронте (первом) или хвостовом (последнем) фронте ХСКн. Ссылаться на «Режимы данных SPI и синхронизация» на стр. 228для деталей.

• Бит 0 — UCPOLn: полярность синхронизации

Бит UCPOLn устанавливает полярность тактового сигнала ХСКн. Комбинация настроек битов UCPOLn и UCPHAн определяет синхронизацию передачи данных. Ссылаться на«Режимы данных SPI и синхронизация» на стр. 228для деталей.

23.6.5 UBRRnL и UBRRnH — регистры скорости передачи USART MSPIM

Функции и битовое описание регистров скорости передачи данных в режиме MSPI идентичны нормальной работе USART. См. «UBRRnL и UBRRnH — регистры скорости передачи USART» на стр. 222.

Таблица 23-4. Сравнение USART в режиме MSPIM и контактов SPI.

USART_MSPIM	СПИ	Комментарий
TxDn	МОСИ	Только мастер выход
RxDn	МИСО	Только мастер-вход
ХСКн	СКК	(Функционально идентичны)
(нет данных)	\overline{SS}	Не поддерживается USART в MSPIM