

Универсальный синхронно-
асинхронный
последовательный приемо-
передатчик (USART)
микроконтроллера ATtiny2313

Универсальный синхронно-асинхронный последовательный приемо-передатчик (USART) – устройство последовательной передачи информации.

Особенности:

- полно-дуплексная организация (независимые регистраторы последовательного приема и передачи);
- синхронный и асинхронный режимы работы;
- выбор скорости передачи информации;
- поддержка кадров длиной 5-9 бит и 1-2 стоп-бита;
- аппаратная поддержка генерации и проверки сигнала четности;
- обнаружение переполнения данных;
- обнаружение ошибок кадрирования;
- режим межпроцессорных связей;
- двухскоростной режим асинхронной передачи.

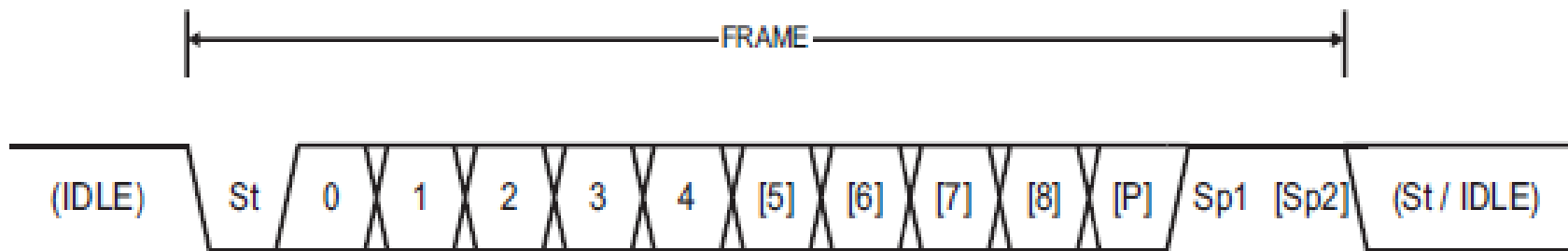
Формат кадров

Единицей передачи данных является кадр.

Кадр – одно слово данных и сопутствующие ему биты синхронизации (стартовый бит, стоповые биты). Сюда же может быть добавлен бит четности, который применяется для проверки правильности передачи информации.

Поддерживается 30 разных вариантов формата кадров. Любой допустимый формат имеет следующие элементы:

- один стартовый бит (всегда «0»);
- 5, 6, 7, 8 или 9 бит данных;
- бит четности (если включен контроль четности);
- один или два стоп-бита (всегда «1»).



Самый старший разряд передается последним.

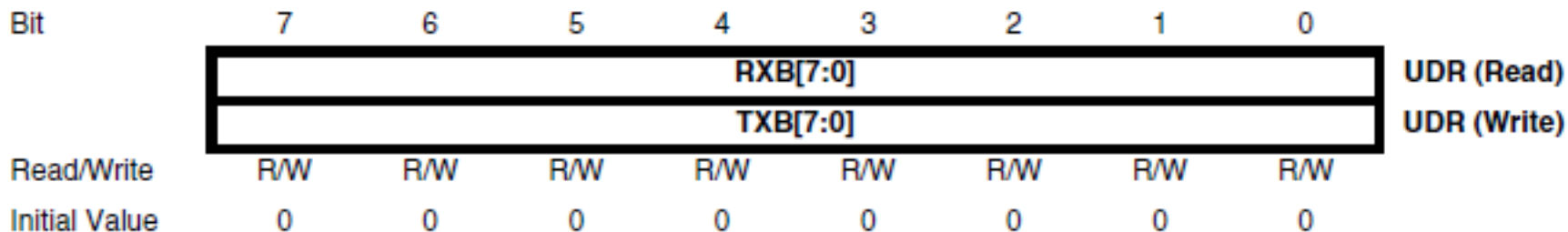
Инициализация USART

1. Установка скорости передачи информации.
2. Установка формата кадра.
3. Включение передатчика или приемника в зависимости от выполняемой операции.

Флаг глобального разрешения прерываний должен быть сброшен до окончания процесса инициализации.

Регистр ввода-вывода данных - UDR

5



Регистр буфера передаваемых данных и регистр буфера принимаемых данных совместно используют один и тот же адрес в пространстве ввода-вывода, воспринимаемый как единый регистр данных UDR.

При записи данных в регистр UDR данные попадают в буфер передачи (TXB). При чтении из регистра UDR возвращается содержимое буфера приема (RXB).

При 5-, 6- и 7-битовых режимах работы старшие неиспользуемые разряды передатчиком будут игнорироваться, а приемником будут установлены в ноль.

Запись в буфер передачи может производиться только когда флаг UDRE регистра UCSRA установлен. Если данные загружены в буфер передачи и передача разрешена, то передатчик загружает данные в передающий сдвиговый регистр, если он пуст. Далее данные начинают побитово передаваться на выход TXD.

Регистр статуса и управления «А» - UCSRA

Bit	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	UPE	U2X	MPCM	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

Бит 7 – RXC : Флаг завершения приема.

Флаг устанавливается, если в буфере приемника есть непрочитанные данные. Флаг очищается, когда буфер приемника пуст. Флаг может быть использован для вызова прерывания по событию «Прием завершен».

Бит 6 – TXC : Флаг завершения передачи.

Флаг устанавливается, если очередной кадр в сдвиговом регистре передатчика был полностью передан, а в буфере передатчика (UDR) нет никаких новых данных, предназначенных для передачи. Флаг очищается, когда начинается выполнение соответствующей процедуры обработки прерывания. Флаг может быть использован для вызова прерывания по событию «Передача завершена».

Бит 5 – UDRE : Флаг «Регистр данных пуст».

Флаг указывает на готовность буфера передачи (UDR) принимать новые данные. Если флаг установлен в «1», то буфер пуст и готов к записи новых данных. Флаг может быть использован для вызова прерывания по событию «Регистр данных пуст».

Бит 4 – FE : Флаг ошибки кадрирования.

Бит устанавливается, если очередная принятая посылка в буфере имеет ошибку кадрирования, то есть если первый стоповый бит посылки в буфере приема оказался нулевым. Флаг равен нулю, если стоповый бит в принятом кадре равен единице. При перезаписи регистра UCSRA бит FE рекомендуется устанавливать в «0».

Бит 3 – DOR : Флаг переполнения.

Бит устанавливается, если обнаружено переполнение данных. Переполнение данных происходит в том случае, когда буфер приема полон, в приемном сдвиговом регистре находится ещё одна посылка и обнаружен новый стартовый бит. Флаг сохраняет свое значение до тех пор, пока приемный буфер (UDR) не будет прочитан. При перезаписи регистра UCSRA бит DOR рекомендуется устанавливать в «0».

Бит 2 – UPE : Флаг ошибки контроля четности.

Флаг устанавливается, если очередное слово данных, находящееся в приемном буфере, имеет ошибку четности и проверка четности в момент приема этого слова была разрешена ($UPM1 = 1$). Флаг действителен до тех пор, пока не прочитан буфер приема. При перезаписи регистра UCSRA бит UPE рекомендуется устанавливать в «0».

Бит 1 – U2X : Удвоение скорости обмена.

Бит используется только в асинхронном режиме работы. При работе в синхронном режиме данный бит рекомендуется устанавливать в «0». При установке бита уменьшается коэффициент деления делителя в формирователе скорости обмена, что приводит к удвоению скорости передачи/приема информации.

Бит 0 – MPCSM : Режим мультипроцессорного обмена.

Бит включает режим мультипроцессорного обмена. При установке бита, все входящие кадры, полученные приемником и не являющиеся адресом, будут игнорироваться. Установка бита не влияет на работу передатчика.

Регистр статуса и управления «В» - UCSRB

Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Бит 7 – RXCIE : Разрешение прерывания по завершению приема.

При установке бита разрешается генерация прерывания при установке флага RXC. Прерывание будет сгенерировано, если установлены бит RXCIE, бит I регистра SREG и бит RXC регистра UCSRA.

Бит 6 – TXCIE : Разрешение прерывания по завершению передачи.

При установке бита разрешается генерация прерывания при установке флага TXC. Прерывание будет сгенерировано, если установлены бит TXCIE, бит I регистра SREG и бит TXC регистра UCSRA.

Бит 5 – UDRIE : Разрешение прерывания по событию «Регистр данных пуст».

При установке бита разрешается генерация прерывания при установке флага UDRE. Прерывание будет сгенерировано, если установлены бит UDRIE, бит I регистра SREG и бит UDRE регистра UCSRA.

Бит 4 – RXEN : Разрешение приема.

При установке бита разрешается работа приемника.

Бит 3 – TXEN : Разрешение передачи.

При установке бита разрешается работа передатчика.

Бит 2 – UCSZ2 : Формат посылок.**Бит 1 – RXB8 : Восьмой разряд приемного буфера.**

Предназначен для хранения девятого информационного разряда принимаемого слова данных (кадр в 9 разрядов). Бит должен быть прочитан до того, как будет прочитан буфер UDR.

Бит 0 – TXB8 : Восьмой разряд буфера передачи.

Девятый информационный разряд слова данных, предназначенный для передачи кадра в 9 бит. Разряд должен быть записан перед тем, как младшие 8 разрядов будут записаны в регистр UDR.

Регистр статуса и управления «С» - UCSRC

Bit	7	6	5	4	3	2	1	0	
	–	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	UCSRC
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

Бит 6 – UMSEL : Выбор режима работы.

«0» - асинхронный режим; «1» - синхронный режим.

Бит 3 – USBS : Выбор количества стоповых бит.

Разряд позволяет выбирать количество стоповых бит, которые будут вставлены передатчиком в конец каждой посылки:

- «0» - один бит;
- «1» - два бита.

На работе приемника это не отражается.

Биты 5:4 – UPM1:0 : Выбор режима контроля четности.

Если контроль четности включен, передатчик автоматически генерирует и посылает биты контроля четности в каждом кадре переданных данных. Приемник генерирует значение четности в каждом кадре для входных данных и сравнивает полученное значение со значением флага UPM0. Если обнаружено несоответствие, то устанавливается флаг UPE регистра UCSRA.

Выбор режима контроля четности

UPM1	UPM0	Режим контроля четности
0	0	Отключено
0	1	Зарезервировано
1	0	Включено, проверка на четность
1	1	Включено, проверка на нечетность

Биты 2:1 – UCSZ1:0 : Формат посылки.

Биты UCSZ2:0 (UCSZ2 из регистра UCSRB) определяют количество информационных разрядов в кадре.

UCSZ2	UCSZ1	UCSZ0	Размер посылки
0	0	0	5 бит
0	0	1	6 бит
0	1	0	7 бит
0	1	1	8 бит
1	1	1	9 бит

Бит 0 – UC POL : Полярность тактового сигнала.

Бит используется только в синхронном режиме. При записи в регистр нового значения бит рекомендуется устанавливать в «0». Бит устанавливает связь между фронтами тактового сигнала (ХСК) и моментами передачи/приема очередного бита информации.

UC POL	Момент передачи очередного бита данных (на выход TXD)	Момент приема очередного бита данных (на вход RXD)
0	Передний фронт сигнала ХСК	Задний фронт сигнала ХСК
1	Задний фронт сигнала ХСК	Передний фронт сигнала ХСК

Регистр скорости обмена информации – UBRRL и UBRRH

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	UBRR[11:8]				UBRRH
	UBRR[7:0]								UBRRL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Baud Rate (bps)	$f_{osc} = 1.0000 \text{ MHz}$			
	U2X = 0		U2X = 1	
	UBRR	Error	UBRR	Error
2400	25	0.2%	51	0.2%
4800	12	0.2%	25	0.2%
9600	6	-7.0%	12	0.2%
14.4k	3	8.5%	8	-3.5%
19.2k	2	8.5%	6	-7.0%
28.8k	1	8.5%	3	8.5%
38.4k	1	-18.6%	2	8.5%

Пример передачи данных

```
#define F_CPU 1000000UL
```

```
#include <avr/io.h>
```

```
#include <util/delay.h>
```

```
int main(void)
```

```
{
```

```
    UCSRB = 1 << TXEN;
```

```
    UBRRH = 0;
```

```
    UBRRL = 12;
```

```
    while(1)
```

```
    {
```

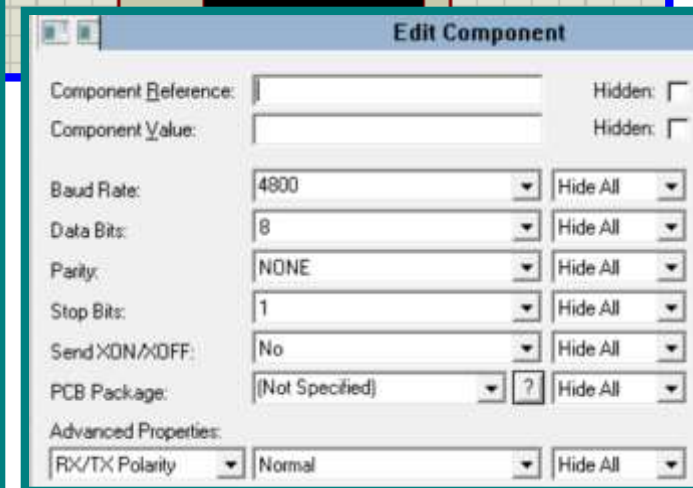
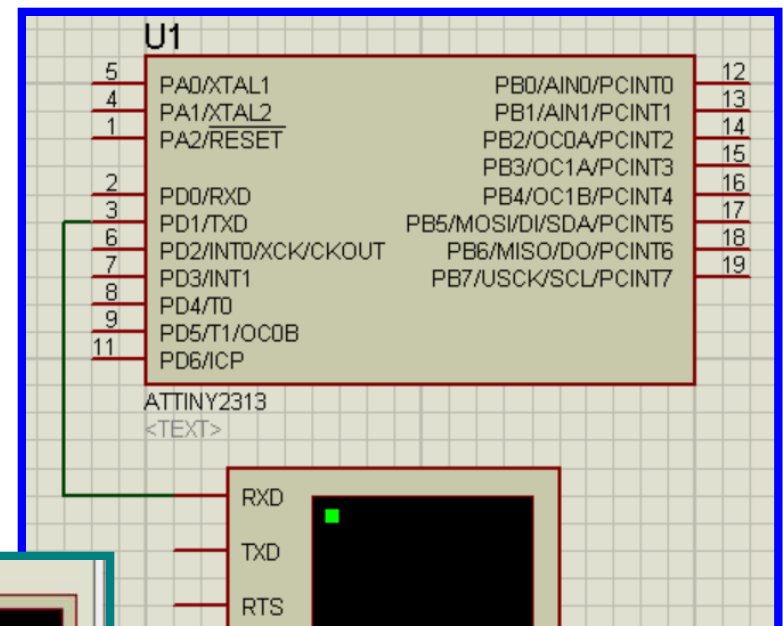
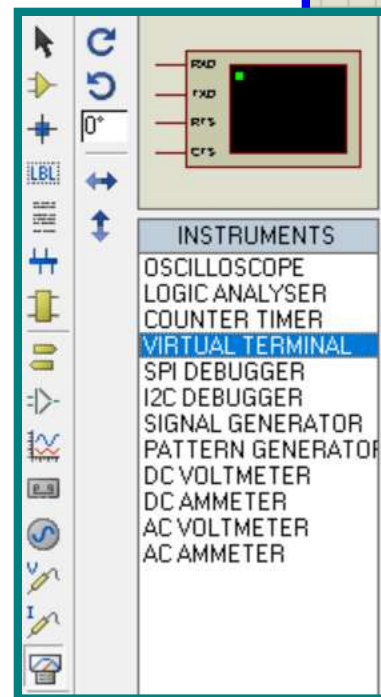
```
        while(!(UCSRA & (1<<UDRE)));
```

```
        UDR = 3;
```

```
        _delay_ms(200);
```

```
    }
```

```
}
```




```
#include <avr/io.h>
```

```
int main(void)
```

```
{
```

```
    DDRB=0xFF;
```

```
    UCSRB = 1 << RXEN;
```

```
    UBRRH = 0;
```

```
    UBRRL = 12;
```

```
    while(1)
```

```
    {
```

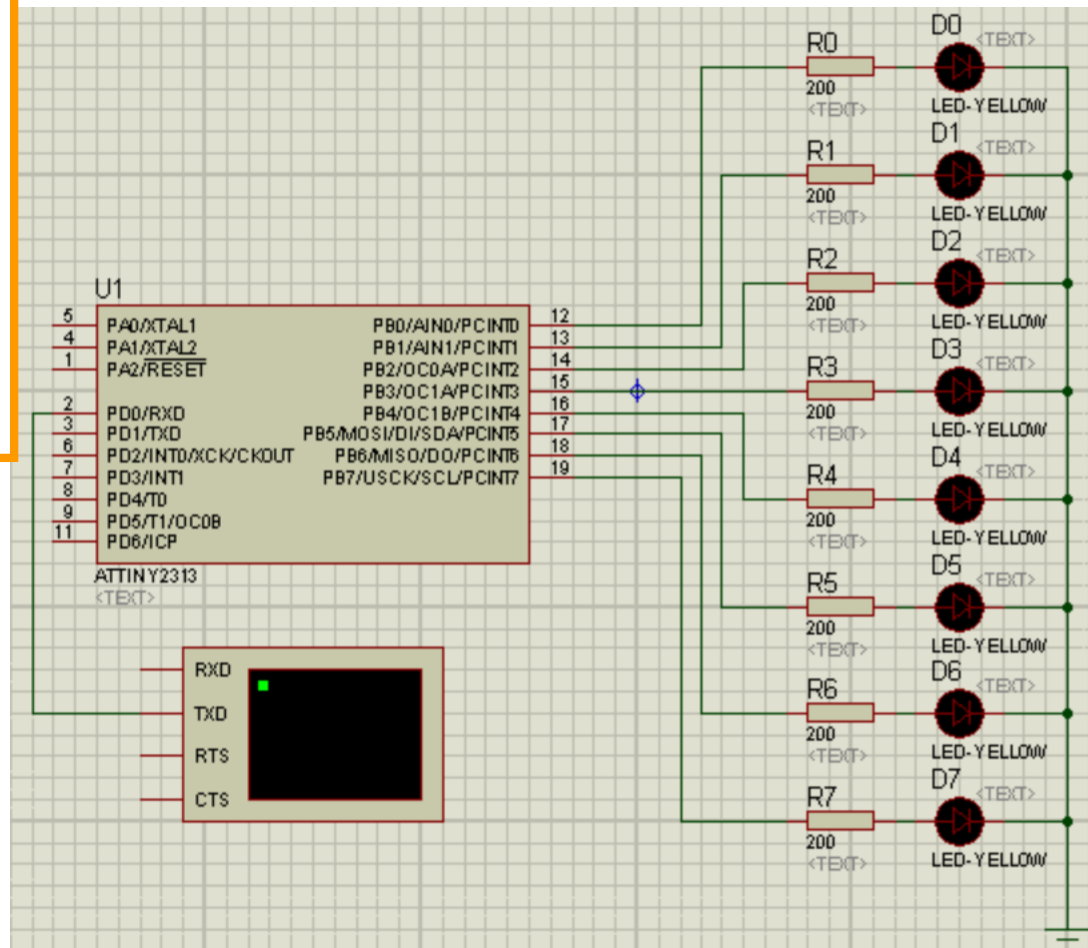
```
        while(!(UCSRA & (1 << RXC)));
```

```
        PORTB = UDR;
```

```
    }
```

```
}
```

Пример приема данных



```

#include <avr/io.h>
#include <avr/interrupt.h>

ISR(USART_UDRE_vect)
{
    UDR = 3;
    UCSRB &= ~(1<<5);
}

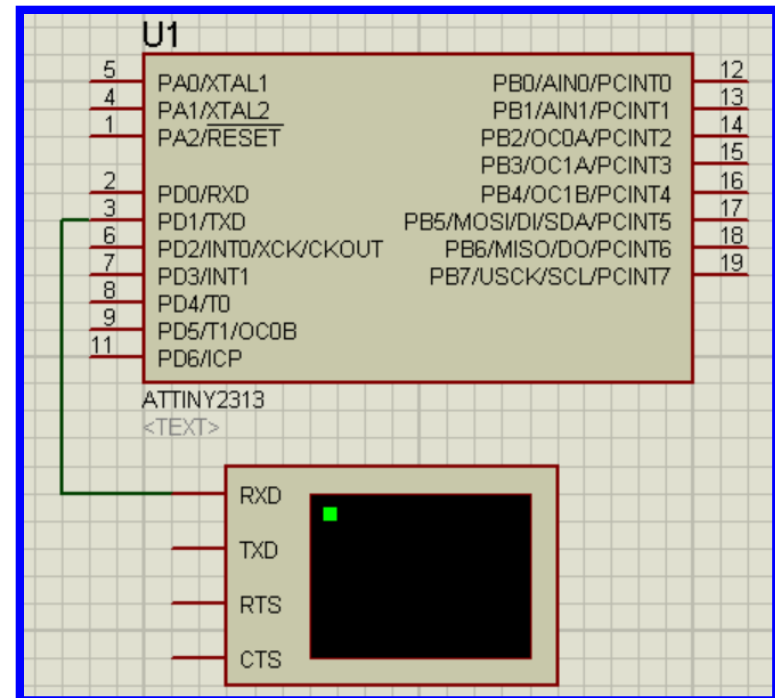
int main(void)
{
    UCSRB = (1 << TXEN) | (1 << UDRIE);
    UBRRH=0;
    UBRRL=12;

    sei();

    while(1)
    { }
}

```

Пример передачи данных (используется прерывание)



```
#include <avr/io.h>
#include <avr/interrupt.h>
```

```
ISR(USART_RX_vect)
{
    PORTB=UDR;
}
```

```
int main(void)
{
    DDRB=0xFF;

    UCSRB = (1 << RXEN) | (1 << RXCIE);
    UBRRH = 0;
    UBRRL = 12;

    sei();

    while(1)
    { }
```

Пример приема данных (используется прерывание)

