

# Лекция 5

## Синхронный и асинхронный интерфейсы

**Асинхронный способ передачи данных** — такой способ передачи цифровых данных от передатчика к приемнику по последовательному интерфейсу, при котором данные передаются в любой момент времени. Синхронизация идет по времени — приемник и передатчик заранее договариваются о том на какой частоте будет идти обмен

**Синхронный способ передачи данных** - способ передачи цифровых данных по последовательному интерфейсу, при котором приемнику и передатчику известно время передачи данных, то есть, передатчик и приемник работают синхронно, в такт.

## Асинхронный способ передачи

### Кодирование информации в RS-232C



## Асинхронный интерфейс UART

**Universal Asynchronous Receiver Transmitter** – «универсальный асинхронный приемопередатчик».

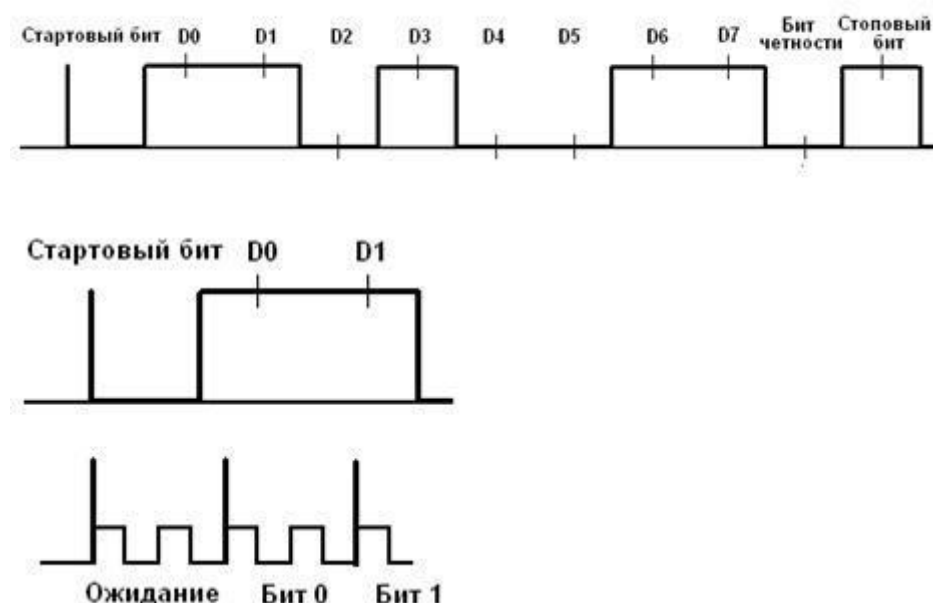
UART, в виду своей простоты долгое время являлся стандартным последовательным интерфейсом для передачи данных, как в микроконтроллерных устройствах, так и в микропроцессорных.

Сейчас он больше распространен в микроконтроллерных устройствах, а в компьютерах заменен на USB.

Существуют микросхемы преобразования USB-UART для подсоединения устройств с UART интерфейсом к компьютерам

## Протокол передачи данных

Для асинхронной передачи UART достаточно всего двух сигнальных линий – TX (Transmit) и RX (Receive). В начале передачи передатчик устанавливает линию в низкий уровень — это старт бит. Приемник, определив, что линия просела, интервал равный времени одного такта и считывает первый бит. С каждым тактом, передатчик выставляет новый бит, а приемник принимает их. Последний бит это стоп бит.



## Прием и передач данных

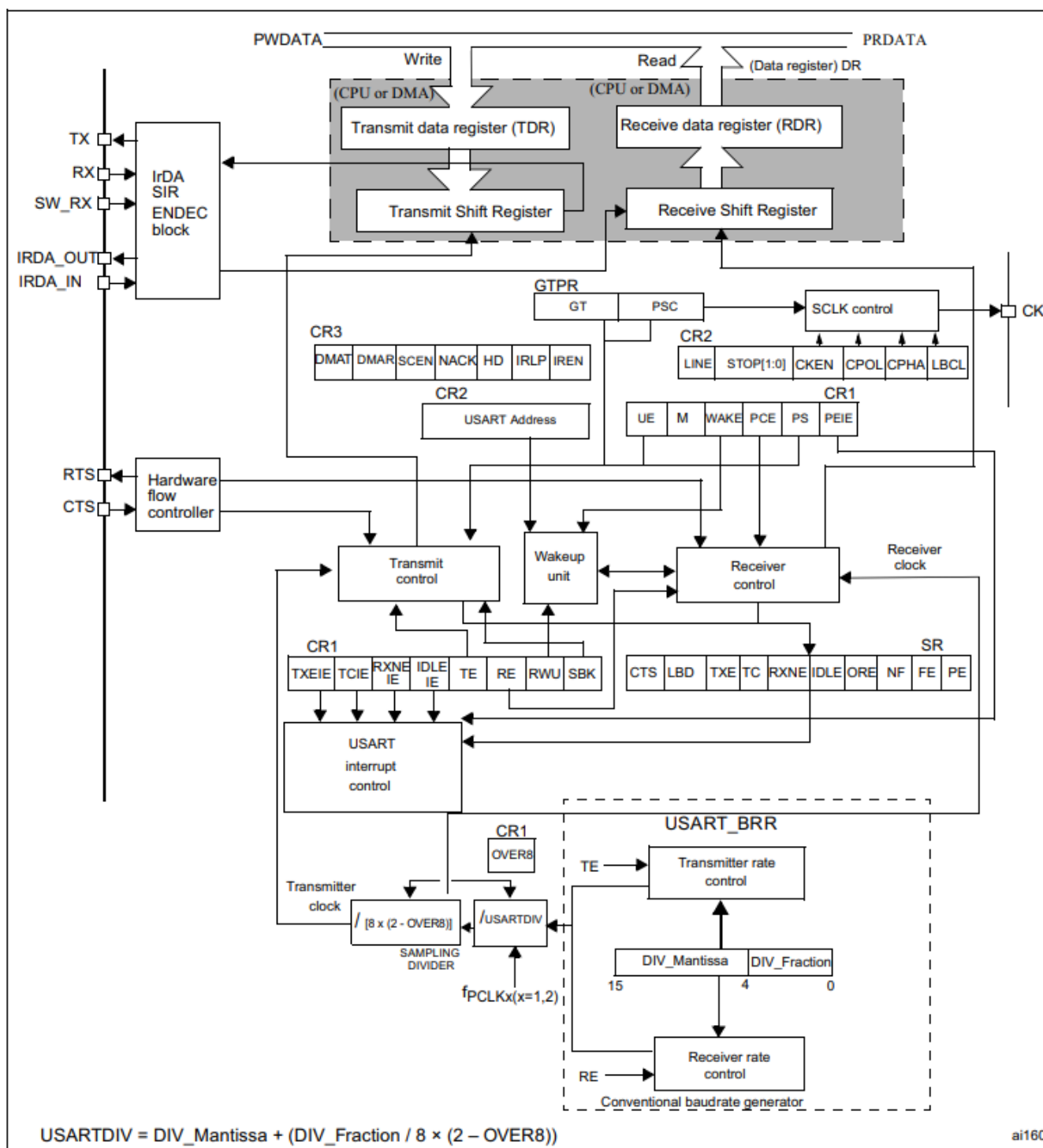
При приеме биты данных из линии с каждым тактом попадают в сдвиговый регистр. Как только сдвиговый регистр заполнился, данные переходят в регистр данных, откуда их можно считать программой.

[Image5] | *Image5.gif*

Тоже самое, но в обратном порядке происходит во время передачи.

# Модуль UART в микроконтроллера STM32F411

3 Модуля USART - USART1, USART2 и USART6



## Общее описание модуля USART

- Гибкая система установки скорости передачи
- Программируемая длина слова (8 или 9 бит)
- Возможность конфигурации количества стоп битов (1 или 2)

- Контроль честности (четное количество 1 или нечетное)
- Независимое включение передатчика и приемника
- Конфигурируемый DMA для приема и передачи сообщений
- 4 Флага дектора ошибок: (Overrun error, Noise detection, Frame error, Parity error)
- 10 флагов прерываний:
  - Transmit data register empty
  - Transmission complete
  - Receive data register full
  - Overrun error, Framing error, Noise error, Parity error
  - CTS changes, LIN break detection, Idle line received
- Мульти процессорная коммуникация
- Поддержка LIN протокола, Поддержка ИК порта IrDA SIR(кодер и декодер), Поддержка SmartCard (возможность общения с SIM карта)

## Особенности USART

USART STM микроконтроллера очень обширный, но мы рассмотрим только то, что относится к UART

В модуле USART можно настраивать следующие параметры:

- Скорость обмена до 4 мбит/с
- Контроль четности
- 1 или 2 стоповых битов
- 8 или 9 бит данных
- Запросы на детектирование ошибок приемо-передачи
- Прерывания по приему, передачи, ошибкам передачи

Для настройки и работы модуля UART нужны всего несколько регистра

- USART\_CR1/CR2/CR3 - регистр настройки 1
- USART\_DR - регистр принятого символа (регистр данных)
- USART\_BRR – регистр настройки скорости передачи
- USART\_SR - регистр состояния

## Регистр CR1 - регистр управления

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	Reserved	UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
rw	Res.	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

**Bit15: OVER8** Режим дискретизации

- **0:** 1/16
- **1:** 1/8

**Bit13: UE** Включение модуля USART

- **0:** Отключить
- **1:** Включить

**Bit12: M** Длина символа

- **0:** 1 Стартовый бит, 8 бит данных
- **1:** 1 Стартовый бит, 9 бит данных

**Bit7: TXEIE** Разрешить прерывание по передаче

**Bit6: TCIE** Разрешить прерывание по концу передачи

**Bit5: RXNEIE** Разрешить прерывание по приему

**Bit3: TE** Разрешить передачу

**Bit2: RE** Разрешить прием

## Регистр SR - регистр статуса

Reset value: 0x00C0 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
						rc_w0	rc_w0	r	rc_w0	rc_w0	r	r	r	r	r

**Bit7: TXE** Регистр данных передачи пуст. Этот бит устанавливается аппаратно, когда содержимое регистра данных передачи перемещается в сдвиговый регистр. Установка этого бита может генерировать прерывание, если установлен TXEIE бит = 1 в регистре USART\_CR1. Этот бит очищается когда происходит запись в регистр данных USART\_DR.

- 0: Данные не перемещены в сдвиговый регистр
- 1: Данные перемещены в сдвиговый регистр

**Bit6: TC** Передача завершена. Этот бит устанавливается когда сдвиговый регистр тоже опустошался и стоит бит TXE. Установка этого бита также может генерировать прерывание если установлен бит TCIE=1 в регистре USART\_CR1. Очищается программно, путем записи 0

- 0: Передача не завершена
- 1: Передача завершена

**Bit5: RXNE** Регистр данных чтения не пуст. Этот бит устанавливается когда содержимое сдвигового регистра перемещается в регистр данных USART\_DR. Установка этого бита генерирует прерывание, если установлен бит RXNEIE=1 в регистре USART\_CR1. Этот бит очищается, сразу после чтения из регистра данных USART\_DR. Также этот бит может быть очищен посредством записи 0 в него

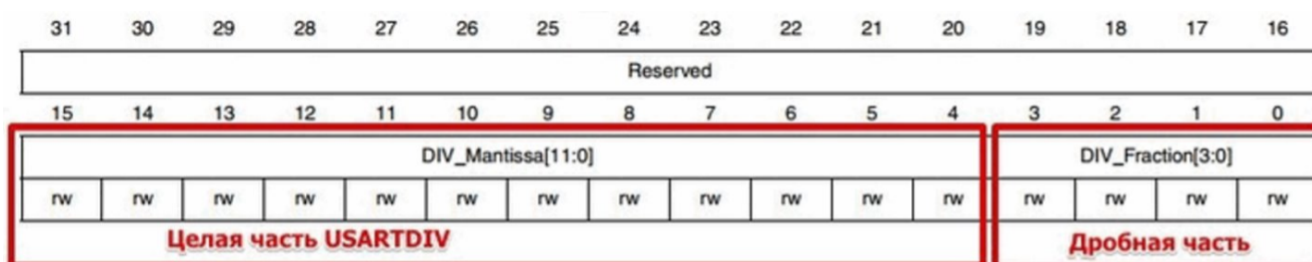
- 0: Данные не приняты
- 1: Данные готовы для чтения

## Регистр данных

**USART\_DR** — Регистр данных. При передачи, посылаемый символ должен быть записан в этот регистр. При приеме, принятый символ нужно прочитать из этого регистра. Занимает 32 бита из которых используются только 9 (!) первых бит, остальные принудительно зануляются аппаратно.

## Регистра настройки скорости передачи

**USART\_BRR** — Регистр настройки скорости передачи. Первые его два байта определяют частоту передачи. Вторые принудительно ноль.



Для вычисления скорости используется следующая формула

- $USARTDIV = CLK / (BaudRate * 8 * (2 - OVER8))$

## Порядок запуска модуля UART

- Подключить USART к источнику тактирования – устанавливаем бит USART2EN в регистре APB1ENR (АЦП тактируется от матрицы шин APB1).
- Необходимо сконфигурировать порты. Настроить порты, на альтернативную функцию нужного модуля USART
- Настроить формат передачи байт, с помощью регистра CR1 и CR2
- Задать скорость передачи с помощью регистра BRR
- Разрешить передачу помощью бита TE и если надо прием, с помощью бита RE в модуле USART с помощью регистра CR1
- Включить сам модуль USART битом UE в регистре CR1
- Если работаем через прерывание, то разрешить глобальное прерывание для нужного USART, в регистре ISER[1] модуля NVIC
- Если работаем через прерывание, в зависимости от того, что нам нужно, разрешить прерывание по сигналу модуля UART (например, от сигнала регистр данных передатчика пуст (бит TXEIE в регистре CR1))

## Задание

Передавать раз в 0.5 секунды фразу Hello World!

- Подключиться к внешнему источнику тактирования
- Настроить таймер 2 на 0.5 секунды
- Подключить модуль UART2 к шине тактирования
- Настроить порты PORT A.2 как TX, Port A.3 как RX на альтернативную функцию работы с UART в режим Push-Pull(двухтактный выход) + Pull Up(подтяжка к 1)
- Настроить USART2 на скорость 9600 бит/с, 1 стоп бит, 1 старт бит, без проверки четности, режим дискретизации 1/16, 8 бит данных.
- Включить USART2
- Включить передачу данных
- Послать сообщение используя программный буфер

## Дома сделать тоже самое с использованием прерывания

- Подключиться к внешнему источнику тактирования
- Настроить таймер 2 на 0.5 секунды

- Подключить модуль UART2 к шине тактирования
- Настроить порты PORT A.2 как TX, Port A.3 как RX на альтернативную функцию работы с UART в режим Push-Pull(двухтактный выход) + Pull Up(подтяжка к 1)
- Настроить USART2 на скорость 19200 бит/с, 2 стоп бит, 1 старт бит, без проверки четности, режим дискретизации 1/8, 8 бит данных.
- Разрешить глобальное прерывание по USART
- Разрешить прерывание по передаче
- Включить USART2
- Включить передачу данных