



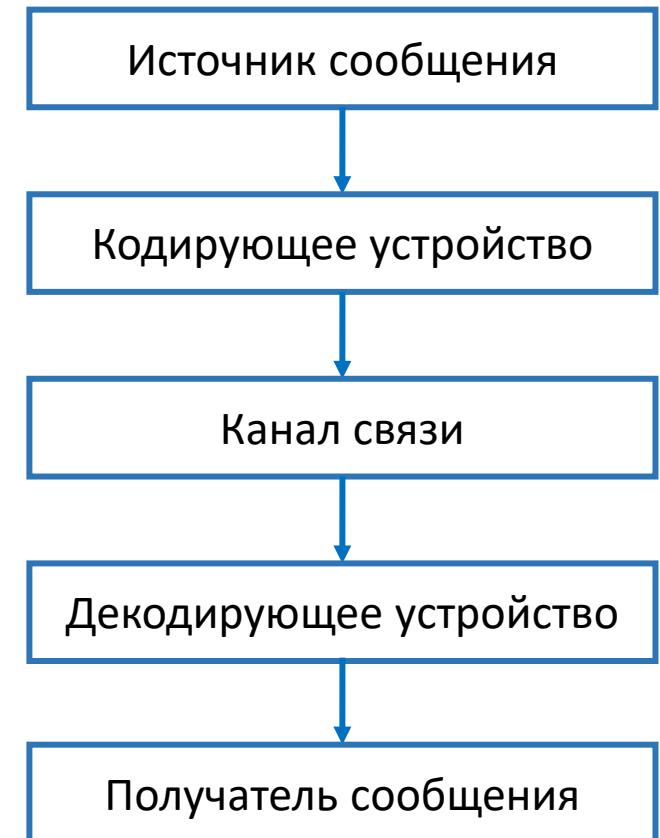
# Цифровая передача данных. UART

Кафедра ЭО  
Петрухин О.М.

# Что такое передача данных?

**Передача данных** — это процесс **перемещения информации** (в виде сигналов, символов, битов) **от одного устройства к другому** по какому-либо каналу связи.

1. **Источник данных** – устройство, генерирующее исходную информацию.
2. **Интерфейс передачи данных** – это способ физического соединения устройств для обмена данными. Он определяет какие сигналы, контакты, разъёмы и линии связи используются, и как именно данные передаются.
3. **Приёмник** – устройство, получающее и обрабатывающее информацию.
4. **Протоколы передачи** – набор правил и форматов, по которым устройства обмениваются информацией через интерфейс (UART, TCP/IP, UDP, HTTP, FTP и др.).



## По способу обмена

**Последовательно**  
(биты информации  
передаются по одному  
каналу последовательно)

**Параллельно**  
(несколько бит одновременно  
по разным проводникам)

## По направлению



- **Скорость передачи** – количество бит, передаваемых за единицу времени. Измеряется в битах в секунду (бит/с, Kbps, Mbps, Gbps).
- **Скорость модуляции (символьная скорость)**

Количество символов информации, передаваемых в единицу времени.

Измеряется в **бодах (Bd)**.

Связана со скоростью передачи данных, но не всегда равна ей (зависит от того, сколько бит кодируется в одном символе). Пример: 2400 Bd при 4 битах на символ = 9600 бит/с.

- **Пропускная способность** – максимальный объём данных, который может быть передан по каналу за определённое время.
- **Задержка (латентность)** – время, затрачиваемое на доставку одного сообщения.
- **Надёжность** – степень защиты данных от потери, искажения или несанкционированного доступа.

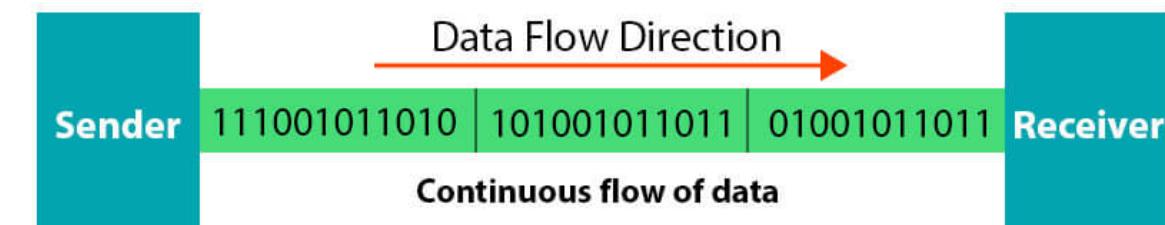
# Режимы передачи данных

**Синхронный режим** — это способ, при котором передатчик и приёмник **работают в одном ритме**, синхронизируясь **по общему тактовому сигналу** (clock). Передача идёт непрерывным потоком битов.

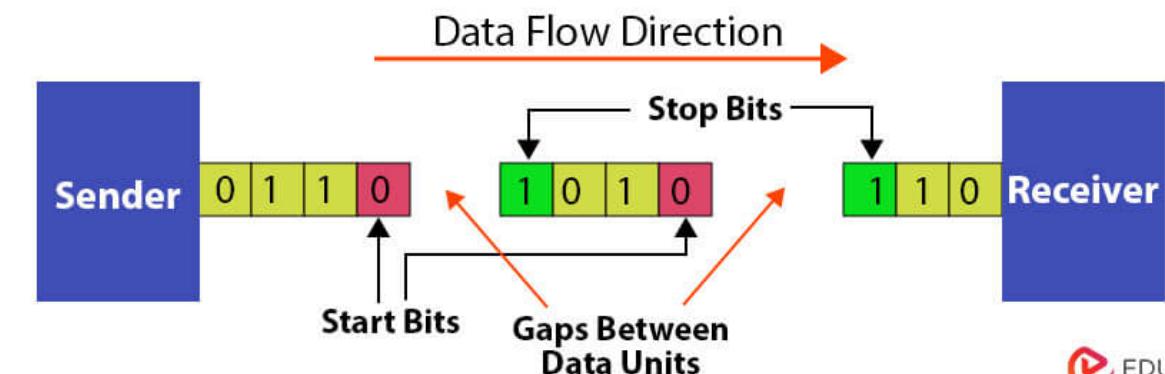
**Асинхронный режим** — это способ, при котором данные передаются **по одному символу** (байту), и каждый символ передаётся независимо от других. Устройства **не делят общий тактовый сигнал**, а **синхронизируются временно**, только на время передачи одного байта.

## Synchronous and Asynchronous Transmission

### Synchronous Transmission



### Asynchronous Transmission



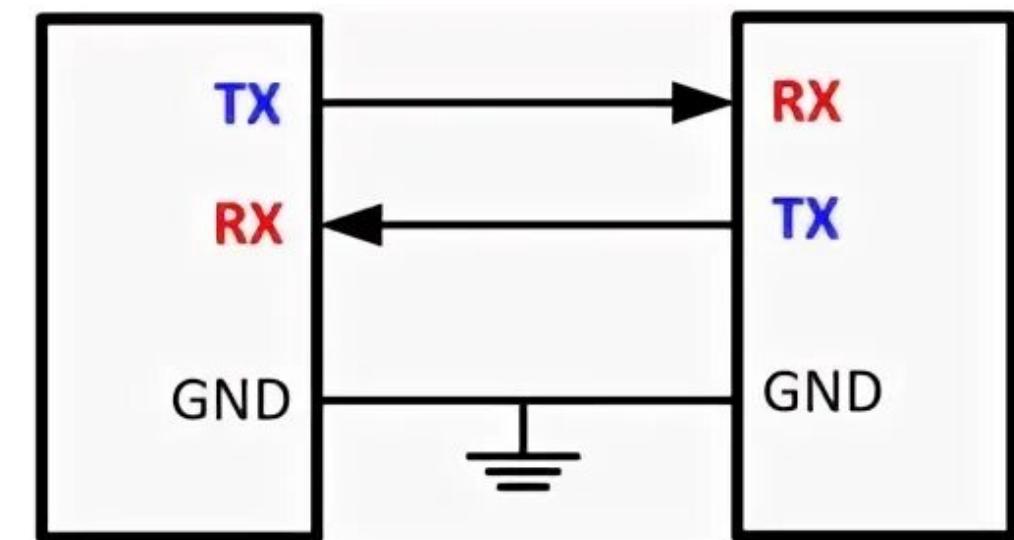
**Универсальный асинхронный приёмопередатчик** (*Universal Asynchronous Receiver-Transmitter, UART*) — это один из наиболее фундаментальных и распространенных протоколов последовательной передачи данных в цифровой электронике.

**Основное назначение** — обеспечить простую и надежную **двустороннюю связь "точка-точка"** между двумя цифровыми устройствами.

## Область применения:

- Обмен данными между микроконтроллерами
- Загрузка прошивки (Bootloader)
- Промышленные датчики и панели операторов
- Bluetooth-модули
- Wi-Fi модули
- GPS-приемники
- COM-порты
- IoT (Интернет Вещей)

- **Асинхронность.** UART не использует внешнее тактирующее устройство для синхронизации передачи данных.
- **Последовательная передача.** Данные передаются бит за битом по одному проводу (или паре проводов), в отличие от параллельных интерфейсов. Поэтому для передачи в одном направлении требуется один проводник; для полнодуплексной двунаправленной связи потребуется два проводника.
- **Выход обозначают** TD или TX (transmitted data), **вход** - RD или RX (received data). Для подключения двух устройств выход одного подключают ко входу другого и вход первого - к выходу второго.

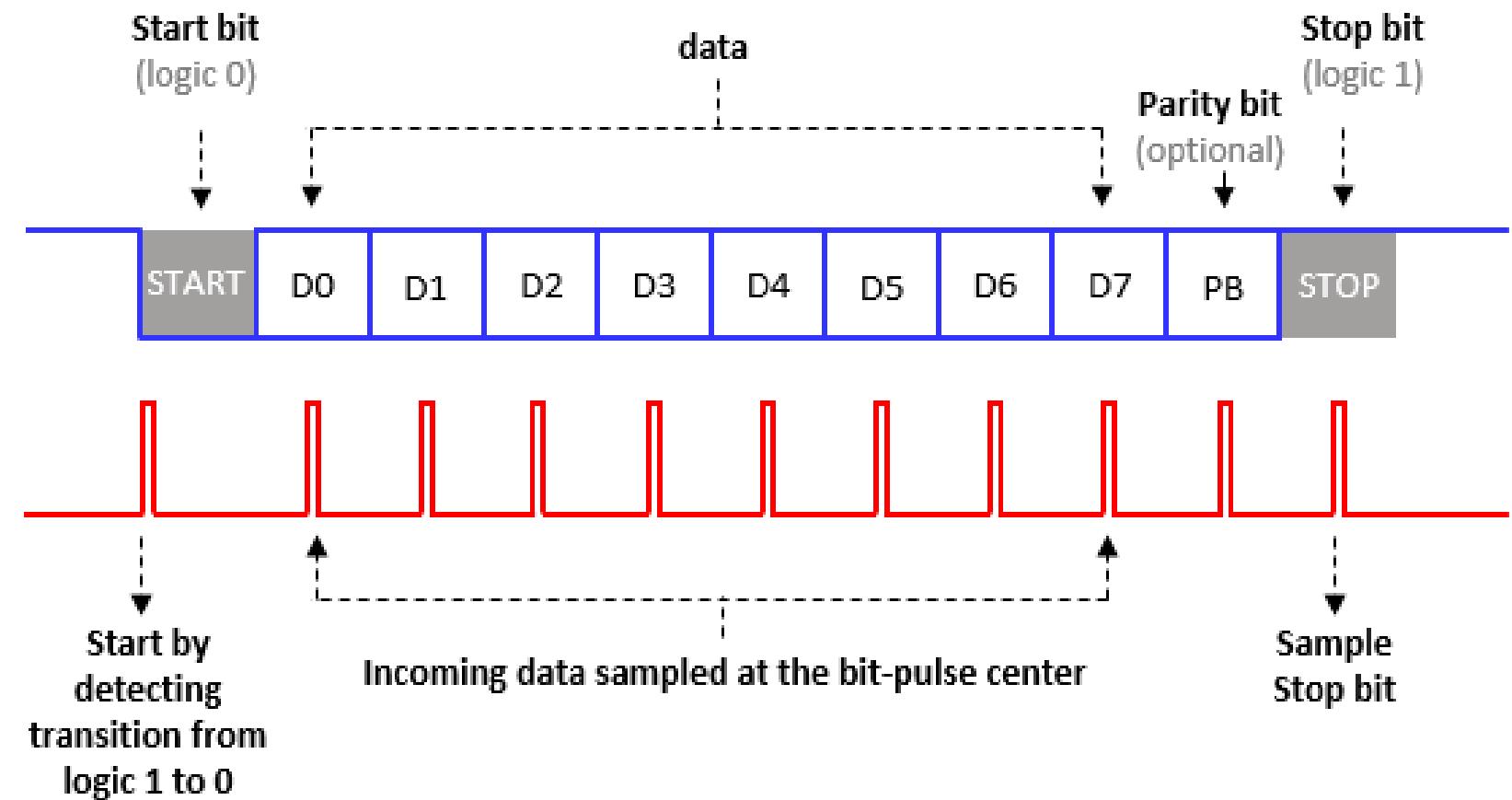


# Принцип передачи данных

Асинхронная передача данных по протоколу UART — это строго регламентированный процесс. Информация не передается сплошным потоком, вместо этого она разбивается на небольшие, стандартизованные пакеты, называемые **кадрами (frames)**.

**Кадр делится на:**

- 1) Стартовый бит
- 2) Биты данных
- 3) Бит чётности
- 4) Стоповый бит



## Стартовый бит (Start Bit)

**Назначение:** Синхронизация и оповещение о начале передачи.

**Механизм:** Когда линия передачи находится в состоянии покоя (**Idle**), она удерживается на высоком логическом уровне (**HIGH**). Для инициации передачи данных **передатчик** (TX) **принудительно переводит линию** в низкий логический уровень (LOW) ровно на один бит-период. Этот резкий переход с HIGH на LOW является сигналом для приемника (RX). Обнаружив спадающий фронт, **приемник "пробуждается"**, запускает свой внутренний таймер и готовится к приему последующих битов данных с заранее оговоренной скоростью.

**Стартовый бит всегда один**, и он всегда имеет значение **LOW**.

# Принцип передачи данных

## Биты данных (Data Bits)

**Назначение:** Передача полезной информации.

**Механизм:** Сразу после стартового бита следуют биты данных. В стандартном применении их может быть от 5 до 8, но в подавляющем большинстве современных приложений используется 8 бит для передачи одного байта.

**Порядок передачи данных: LSB (least significant bit) first.** Это означает, что если мы передаем байт **0b11001010**, то последовательность на линии будет 0, 1, 0, 1, 0, 0, 1, 1.

## Бит чётности (Parity Bit) - опционально

**Назначение:** Простейшая форма проверки целостности данных.

**Механизм:** Это **необязательный бит**, который добавляется после битов данных. Его **значение вычисляется** таким образом, чтобы общее количество единиц (логический HIGH) в битах данных было либо чётным (**Even Parity**), либо нечётным (**Odd Parity**).

**Приемник выполняет тот же подсчет и сравнивает результат.** Если он не совпадает, значит, в процессе передачи произошла ошибка (например, из-за помех).

# Принцип передачи данных

## Стоповый бит (Stop Bit)

**Назначение:** Обозначение конца кадра (frame) и гарантия корректного начала следующей передачи.

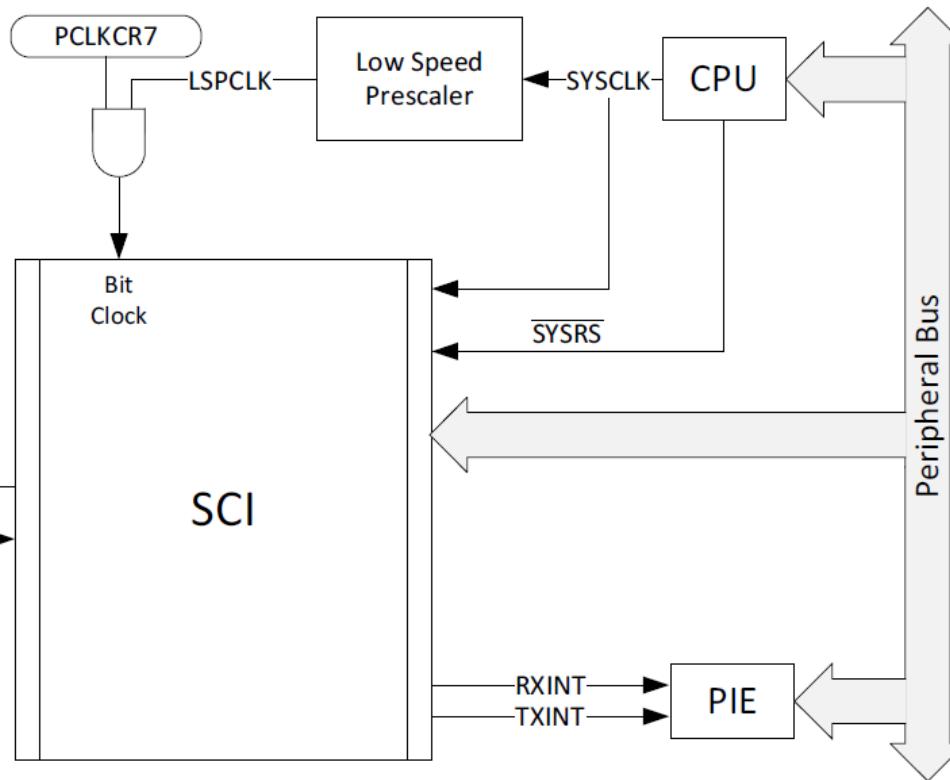
**Механизм:** После передачи последнего бита данных (или бита четности) **передатчик переводит линию обратно в высокий логический уровень (HIGH)** на определенный период времени. Длительность этого периода может составлять 1, 1.5 или 2 бит-периода. Наиболее распространенным является один стоповый бит.

Стоповый бит **критически важен**, так как он гарантирует, что линия вернется в состояние HIGH перед началом следующей передачи. Это **обеспечивает четкий и однозначный спадающий фронт для следующего стартового бита**, даже если последний бит данных был LOW.

**Скорость передачи данных (Baud Rate, или бод)** — это фундаментальный параметр UART, определяющий, сколько раз в секунду изменяется состояние сигнала на линии передачи. В контексте UART это эквивалентно количеству **бит в секунду**.

Оба устройства — передатчик и приемник — должны быть заранее настроены на **абсолютно одинаковую скорость**. Если скорости не совпадают, приемник будет считывать состояние линии в неправильные моменты времени, что приведет к полной потере и искажению данных.

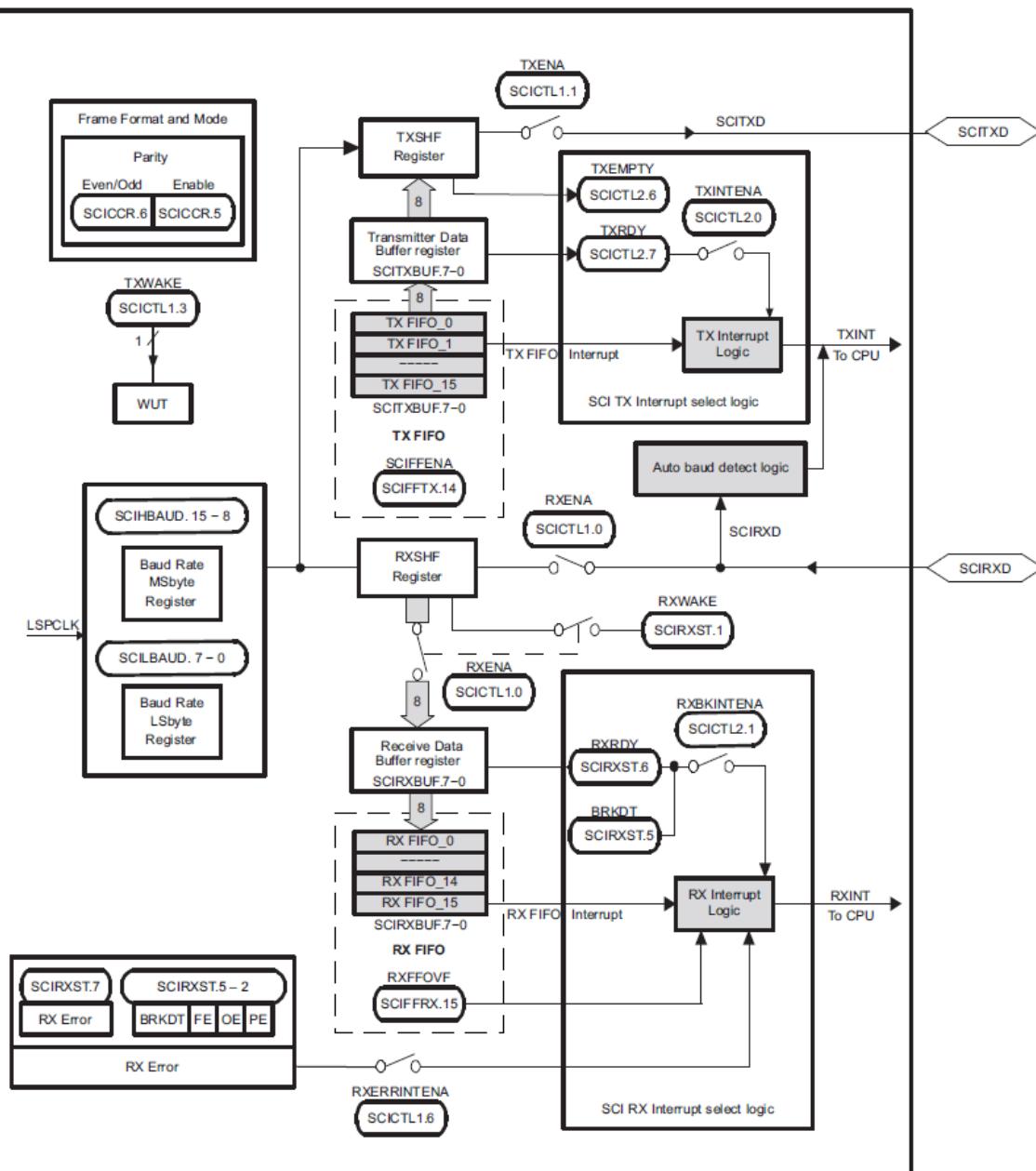
Существует ряд стандартных скоростей, поддерживаемых большинством устройств: 1200, 2400, 4800, **9600** (часто используется по умолчанию), 19200, 38400, 57600, **115200** (распространенная скорость для прошивки и быстрой передачи данных) и выше.



## Основные особенности:

- 3 модуля SCI – SCIA, SCIB, SCIC
- Четыре флага обнаружения ошибок:  
четность, переполнение, ошибка фрейма и  
обнаружение обрыва линии
- Полу- или полнодуплексный режим работы
- Отдельные прерывания для TX и RX
- Автодetectирование скорости передачи  
данных
- 16-уровневый FIFO приема/передачи

# Структура блока SCI



## Основные элементы:

### 1. Передатчик (TX)

- **SCITXBUF** — регистр буфера данных передатчика. Содержит данные (загруженные ЦП) для передачи.
- **Регистр TXSHF** — регистр сдвига передатчика. Принимает данные из регистра SCITXBUF и сдвигает их на вывод SCITXD по одному биту за раз.

### 2. Приемник (RX)

- **Регистр RXSHF** — регистр сдвига приёмника. Смещает данные с вывода SCIRXD по одному биту за раз.
- **SCIRXBUFF** — буферный регистр данных приёмника.

### 3. Программируемый генератор скорости передачи данных

### 4. Регистры управления и состояния

# Основные регистры конфигурации SCI

## SCICCR

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
STOPBITS	PARITY	PARITYENA	LOOPBKENA	ADDRIDLE_M ODE	SCICHAR		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		

## SCIHBAUD

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
BAUD							
R/W-0h							

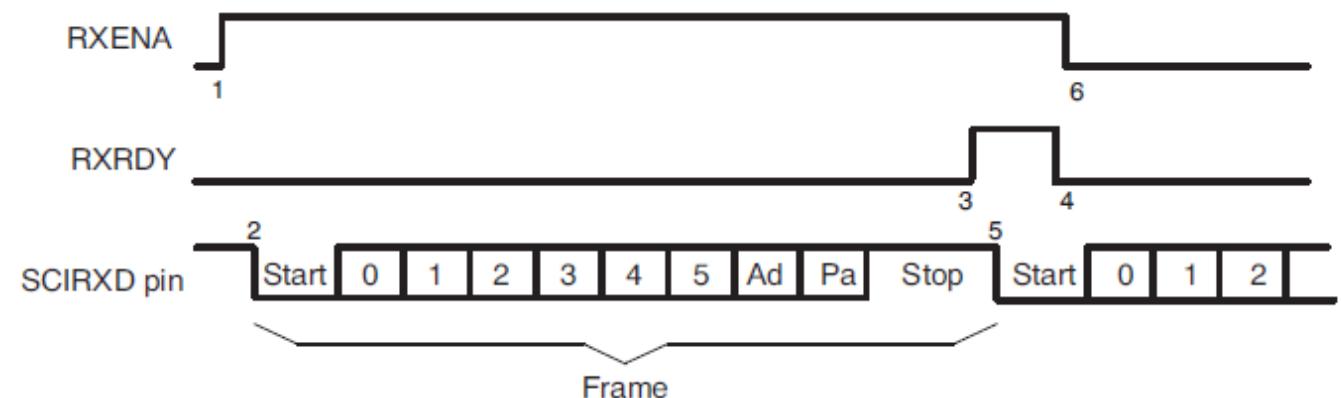
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
BAUD							
R/W-0h							

## SCICTL1

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	RXERRINTEN A	SWRESET	RESERVED	TXWAKE	SLEEP	TXENA	RXENA
R-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

# Принцип работы RX

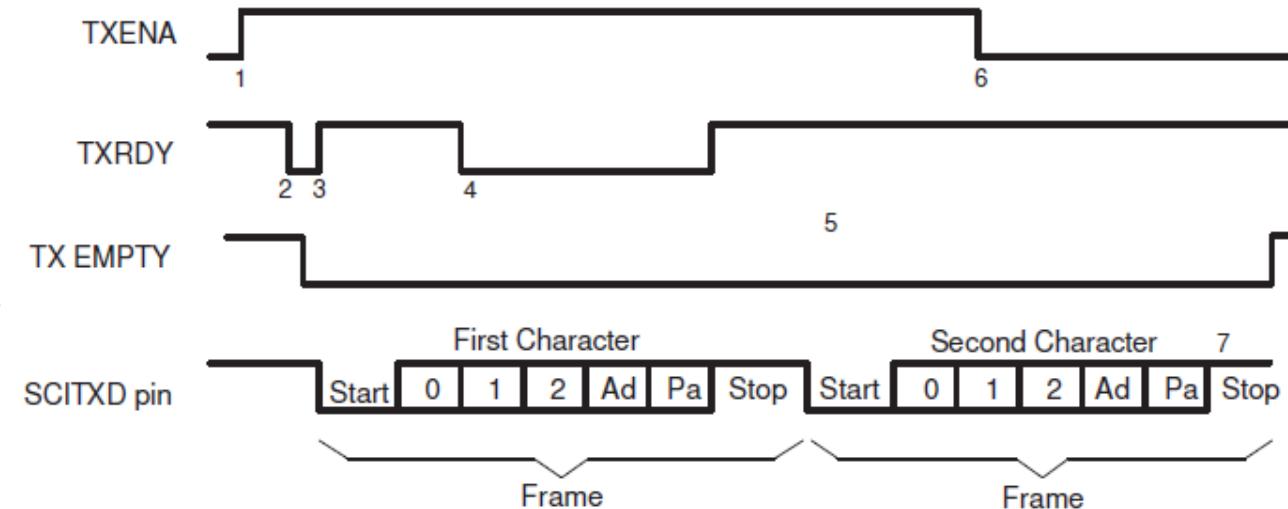
1. Флаг RXENA (SCICCTL1, бит 0) в высокий уровень, чтобы включить приемник.
2. Данные поступают на вывод SCIRXD. Обнаружен стартовый бит.
3. Данные перемещаются из RXSHF в буферный регистр приёмника (SCIRXBUF); генерируется прерывание. Флаг RXRDY (SCIRXST, бит 6) переходит в высокий уровень, сигнализируя о получении нового сообщения.
4. Программа считывает SCIRXBUF; флаг RXRDY автоматически очищается.



5. Следующий байт данных поступает на ножку SCIRXD; Детектируется стартовый бит.
6. Бит RXENA устанавливается в низкий уровень для отключения приёмника. Сборка данных продолжается в RXSHF, но не переносится в буферный регистр приёмника (SCIRXBUF).

# Принцип работы TX

1. Бит TXENA (SCICCTL1, бит 1) переходит в высокий уровень, позволяя передатчику отправлять данные.
2. Данные записываются в SCITXBUF. Буфер передатчика больше не пуст и TXRDY переходит в низкий уровень.
3. SCI передаёт данные в сдвиговый регистр (TXSHF). Передатчик готов к приёму второго сообщения (TXRDY переходит в высокий уровень) и генерирует прерывание.
4. Программа записывает второй символ в SCITXBUF после того, как TXRDY переходит на высокий уровень (3). (TXRDY снова переходит в 0 после записи второго символа в SCITXBUF).



5. Передача первого символа завершена. Начинается передача второго символа в сдвиговый регистр TXSHF.
6. Бит TXENA переходит в 0, отключая передатчик; SCI завершает передачу текущего символа.



Спасибо за внимание!