

# UART

---

Метод передачи данных

# UART

## Метод передачи и приёма данных

- Универсальный асинхронный приёмопередатчик (*Universal Asynchronous Receiver-Transmitter, UART*) — узел вычислительных устройств, предназначенный для организации связи с другими цифровыми устройствами. Преобразует передаваемые данные в последовательный код так, чтобы было возможно передать их по цифровой линии другому аналогичному устройству. Метод преобразования хорошо стандартизован и широко применяется в компьютерной технике, особенно во встраиваемых устройствах.



- Передача данных в UART осуществляется по одному биту в равные промежутки времени. Этот временной промежуток определяется заданной скоростью UART и для конкретного соединения указывается в бодах (что в данном случае соответствует битам в секунду). Существует общепринятый ряд стандартных скоростей: 300; 600; 1200; 2400; 4800; 9600; 19200; 38400; 57600; 115200; 230400; 460800; 921600 бод.

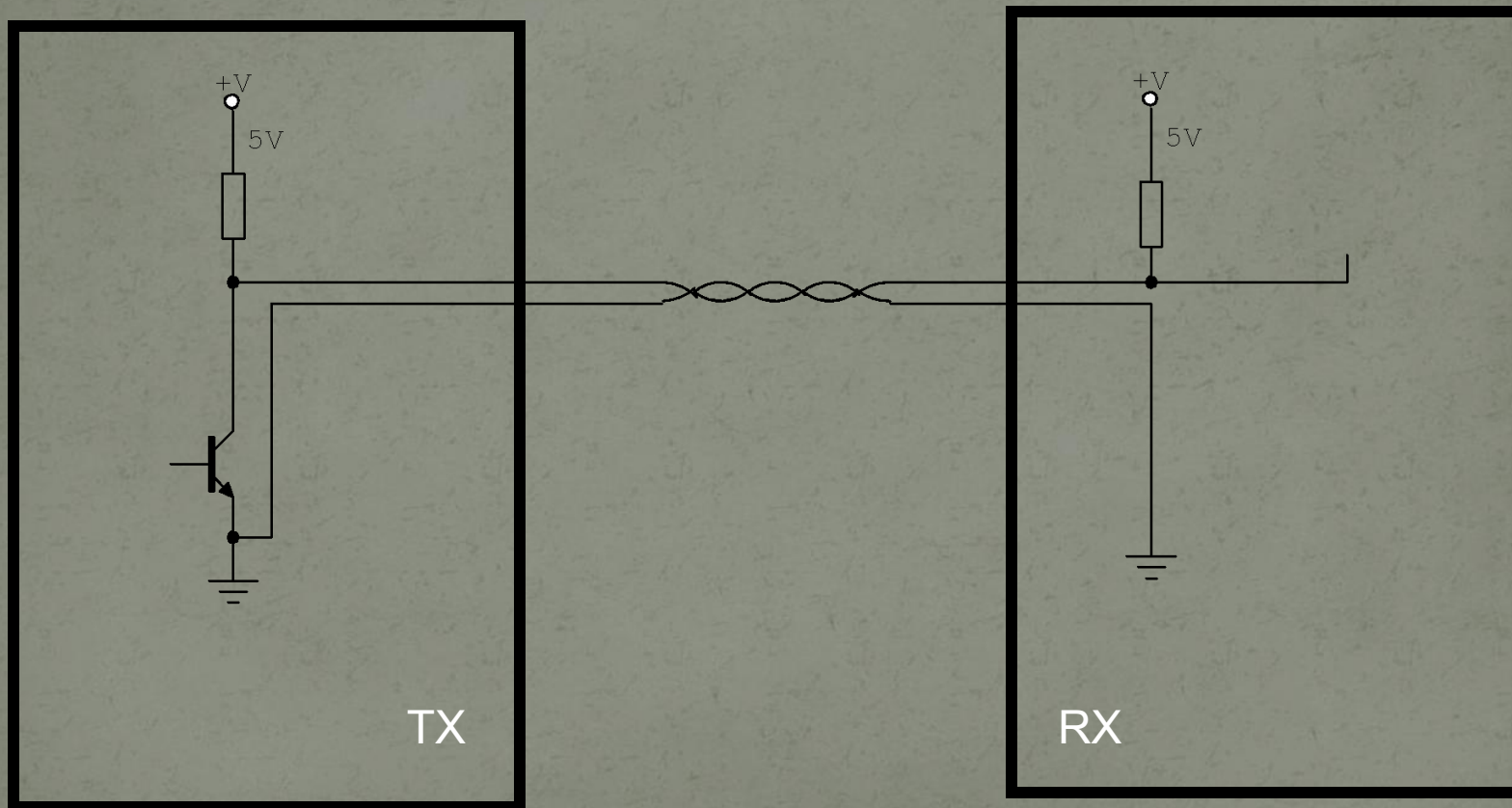
- Помимо собственно информационного потока UART автоматически вставляет в поток синхронизирующие метки, так называемые **стартовый и стоповый биты**, а также биты проверки на четность/нечетность. При приёме эти лишние биты удаляются из потока. Обычно полезная информация передается байтами (8 бит), однако встречаются реализации UART, которые позволяют передавать по 5, 6, 7, 8 или 9 бит. Могут вставляться два стоповых бита вместо одного, что делается для уменьшения вероятности рассинхронизации приёмника и передатчика при плотном трафике. Приёмник игнорирует второй стоповый бит, воспринимая его как короткую паузу на линии.



# Протокол UART

- Принято соглашение, что пассивным (в отсутствие потока данных) состоянием входа и выхода UART является логическая 1. Стартовый бит всегда логический 0, поэтому приёмник UART ждёт перепада из 1 в 0 и отсчитывает от него временной промежуток в половину длительности бита (середина передачи стартового бита). Если в этот момент на входе всё ещё 0, то запускается процесс приёма минимальной посылки. Для этого приёмник отсчитывает 9 битовых длительностей подряд (для 8-бит данных) и в каждый момент фиксирует состояние входа. Первые 8 значений являются принятыми данными, последнее значение проверочное (стоп-бит). Значение стоп-бита всегда 1, если реально принятое значение иное, UART фиксирует ошибку.

# Рецессивный и доминантный биты



TX

RX

Трансмиттер

Ресивер



- Для формирования временных интервалов передающий и приёмный UART имеют источник точного времени (тактирования). Точность этого источника должна быть такой, чтобы сумма погрешностей (приёмника и передатчика) установки временного интервала от начала стартового импульса до середины стопового импульса не превышала половины (а лучше хотя бы четверти) битового интервала. Для 8-бит посылки  $0,5/9,5 = 5\%$  (в реальности не более  $3\%$ ). Поскольку эта сумма ошибок приёмника и передатчика плюс возможные искажения сигнала в линии, то рекомендуемый допуск на точность тактирования UART не более  $1,5\%$ .

- Поскольку синхронизирующие биты занимают часть битового потока, то результирующая пропускная способность UART не равна скорости соединения. Например, для 8-битных посылок синхронизирующие биты занимают 20 % потока, что для физической скорости 115 200 бод даёт битовую скорость данных 92160 бит/с или 11 520 байт/с.



# Контроль чётности

- Многие реализации UART имеют возможность автоматически контролировать целостность данных методом контроля битовой чётности. Когда эта функция включена, последний бит данных («бит чётности») контролируется логикой UART и содержит информацию о чётности количества единичных бит в посылке.
- Для четного количества единичных бит в передаваемом байте  $P=1$
- Для нечетного количества единичных бит в передаваемом байте  $P=0$

# Управление потоком данных

- В старые времена устройства с UART могли быть настолько медлительными, что не успевали обрабатывать поток принимаемых данных. Для решения этой проблемы модули UART иногда снабжались отдельными выходами и входами управления потоком. При заполнении входного буфера логика принимающего UART выставляла на соответствующем выходе запрещающий уровень, и передающий UART приостанавливал передачу.
- Позже управление потоком возложили на коммуникационные протоколы, и необходимость в отдельных линиях управления потоком постепенно исчезла.



# Физический уровень

- Логическая схема UART имеет входы-выходы с логическими уровнями, соответствующими полупроводниковой технологии схемы: КМОП, ТТЛ и т. д. Такой физический уровень может быть использован в пределах одного устройства, однако непригоден для коммутируемых длинных соединений по причине низкой защищённости от электрического разрушения и помехоустойчивости. Для таких случаев были разработаны специальные физические уровни, такие, как токовая петля, RS-232, RS-485, LIN и тому подобные.

- Специфической разновидностью физического уровня асинхронного интерфейса является физический уровень IrDA.
- Протокол IrDA (Infra red Data Association) позволяет соединяться с периферийным оборудованием без кабеля при помощи ИК-излучения с длиной волны 880 nm. Порт IrDA позволяет устанавливать связь на коротком расстоянии до 1 метра в режиме точка-точка.



- Существуют физические уровни UART для сложных сред. В некотором смысле стандартный компьютерный телефонный модем также можно назвать специфическим физическим уровнем асинхронного интерфейса. Существуют специальные микросхемы проводных модемов, сделанных специально как физический уровень асинхронного интерфейса (то есть протоколно прозрачные).
- Выпускается также радиоканальный физический уровень в виде модулей радиоприёмников и радиопередатчиков.

# Стандартизация

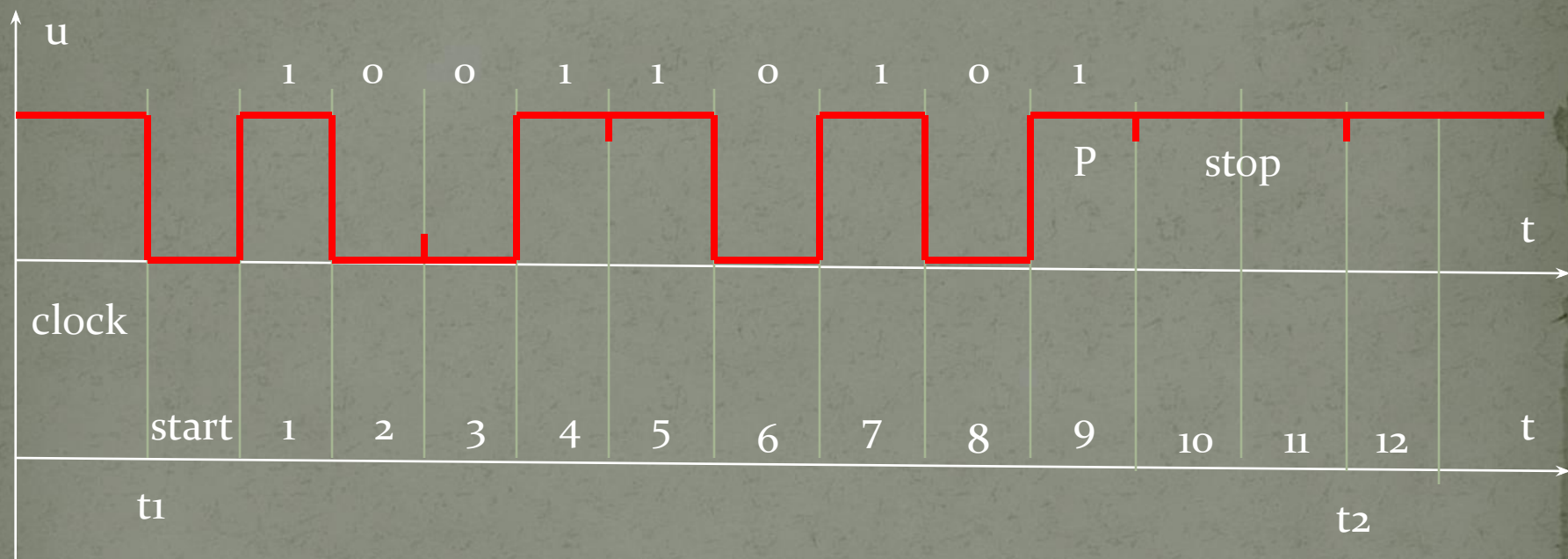
- Идея асинхронной передачи данных появилась в те далёкие времена, когда о стандартизации ещё мало заботились и лучшее, что можно было ожидать от поставщиков разрозненных решений, так это открытой публикации алгоритмов работы своих изделий. Собственно, поэтому стандарта UART как такового нет, но логика работы UART описана как составная часть во многих других стандартах.



# Пример передачи байта

$9A_h = 10011010_2$

$P=1$



# Пример передачи байта