

Протокол передачи данных для UART

Петров А. А.

*Петров Алексей Александрович / Petrov Alexey Aleksandrovich – студент,
кафедра систем автоматического управления и контроля,
факультет интеллектуальных технических систем,
Национальный исследовательский университет,
Московский институт электронной техники, г. Зеленоград*

Аннотация: в статье рассматривается создание относительно простого протокола передачи данных для UART, недостатки и его преимущества, краткая история UART, а также логика передачи информации.

Ключевые слова: протокол передачи данных, UART.

Универсальный асинхронный приёмопередатчик UART, та же известный как УАПП, является одним из самых распространенных интерфейсов в современной электронике. Протокол RS-232 является одним самым известных из семейства UART. Родиной данного протокола, как ни странно, является США. Изначально данный протокол предназначался для отправки теле графических сообщений. Механическое устройство являлось передатчиком. И для передачи данных использовалось всего пять бит. Но в современной электронике встречаются такие реализации UART, которые позволяют отправлять пять, шесть, семь, восемь, девять бит информации.

Все процедуры обмена данными по последовательному интерфейсу осуществляются посредством специальных микросхем, которые получили название универсальных асинхронных приемопередатчиков (от англ. UART — Universal Asynchronous Receiver/Transmitter). Все устройства, которые выполняют последовательный обмен данными, имеют тот или иной тип микросхемы UART. Тем не менее, аппаратная часть большинства таких приемопередатчиков базируется на топологии кристалла 8250 и его более продвинутой версии 8250A. Устройства серии 8250 включают такие популярные микросхемы асинхронных приемопередатчиков, как 16450, 16550, 16650 и т. д. Практически любую из этих микросхем можно обнаружить в персональных компьютерах и других коммуникационных устройствах [1].

Рассмотрим кратко основные порты UART. Основными линиями передачи и приема данных являются RXD и TXD. Порт RXD (Received Data) является приемником сообщений, а TXD (Transmitted Data) является передатчиком. Также присутствуют другие порты, такие как DTS, RTS и пр. Но это уже зависит от версии UART. Приемник и передатчик работают независимо друг от друга, это означает, что UART является полнодуплексным интерфейсом.

Передача данных в UART осуществляется кадрами. Кадр - это последовательность бит фиксированной длины. Биты передаются в равные промежутки времени. Скорость UART и определяет данный интервал времени. Существует общепринятый ряд стандартных скоростей: 300; 600; 1200; 2400; 4800; 9600; 19200; 38400; 57600; 115200; 230400; 460800; 921600 бод. Рассмотрим кадр подробнее, он состоит из стартового бита, информационных битов, бита проверки четности, **одного или двух стоп-битов. В случае если количество информационных бит превышает максимальное количество возможных передаваемых битов, излишек битов будет передан следующей посылкой. Ключевым моментом в посылке является бит проверки четности. Ведь благодаря ему можно проверить целостность отправленной посылки.**

Стоит пояснить, что такое бит четности. Битом четности в электронной [вычислительной технике](#) и [сетях передачи данных](#) называют [контрольный бит](#), который служит для проверки чётности числа в двоичной системе счисления.

Пожалуй, самым сильным недостатком бита четности является его полная беспомощность при возникновении двух, четырех и т. д. ошибок в битах данных. Но что делать? Если канал передачи данных подвержен сильным помехам и практически, чуть ли не каждое сообщение приходит с множеством ошибок, а единственным интерфейсом для передачи данных является UART. Стандартная логика передачи данных UART уже будет бессильна.

Для того чтобы обеспечить сохранность передаваемых данных, необходим некоторый протокол передачи данных. Далее о его реализации и пойдет речь.

Пожалуй, первое с чего стоит начать. Это описание структуры протокола. Протокол должен состоять из трех ключевых сообщений: сообщение о начале передачи данных, данные, сообщение, позволяющее исправлять ошибки в данных.

Сообщение о начале передачи данных. Я бы советовал использовать последовательность бита типа 01010101 или же 10101010, ведь такую последовательность бит сложнее сбить помехой. Да и даже в том случае, если будут искажения в передаче, то не составит особого труда определить, что это сообщение о начале передаче данных.

Вторым пунктом идут данные, которые необходимо передать. Тут ничего сложного, главное их передать.

Третий пункт, один из самых важных. Сообщение, которое поможет исправить ошибки в переданных данных. Есть различные способы исправления ошибок, на этот счет написано много различной литературы. Но самым распространенным является код Рида-Соломона.

Основным преимуществом данного протокола является то, что сохранность информационных бит обеспечивается не только благодаря биту четности, но и благодаря коду Рида-Соломона. Недостатком же является его избыточность.

Литература

1. Магда Ю. С. Программирование последовательных интерфейсов. - М: БХВ-Петербург, 2009 – 304 стр.
-