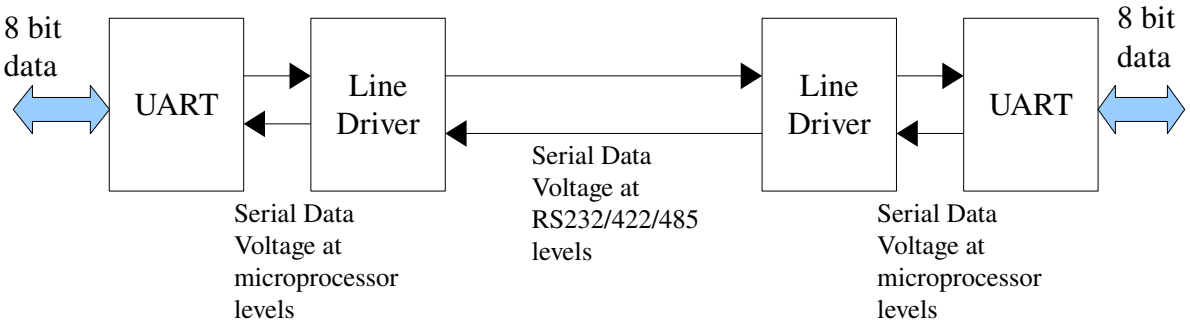
Serial communications hardware

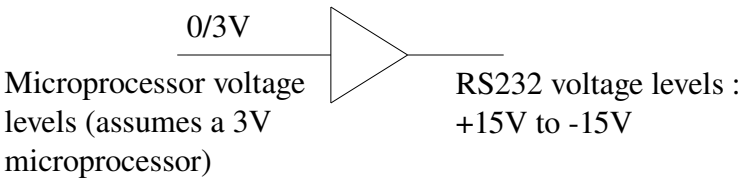
Many microcontrollers and larger computer systems include a device known as a UART or Universal Asynchronous Receiver/Transmitter. This device allows them to send serial communications in a number of formats such as RS232, RS422 and RS485. The first of these, RS232 is probably best known to PC users as it was widely used for mice in the past though most mice these days are USB or PS/2. RS232 was also used to communicate with external dial-up modems. RS422 is similar to RS232 but operates at a different voltage and uses different wiring (push-pull or differential signalling) but is otherwise just like RS232. RS485 is a serial protocol bus standard similar to 232/422 and is widely used in industry for communicating with instruments and control systems.

From a hardware perspective therefore, RS232/422/485 can be viewed as follows:

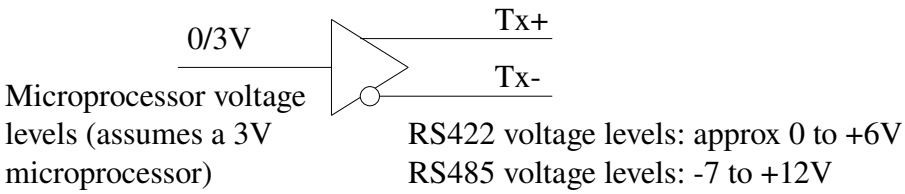


A microprocessor writes 8 bit (parallel) data to a UART which outputs that data in serial form to a line driver. The line driver amplifies the signal so that it is less susceptible to noise allowing it to be transmitted further and more reliably. At the receiving end, the line driver converts the received voltage levels back to microprocessor voltage levels and passes the serial data on to another UART. This UART reconstructs an 8 bit byte from the serial data and passes it back to the microprocessor.

Line Drivers and receivers.
Wires carrying RS232 signals can switch between -15V and +15V. Negative voltages are taken to be logic 1 while positive voltages are interpreted as logic 0. The reason for the high voltage levels is to make the data signal much larger compared to electrical interference that is always present. An RS232 line driver must therefore take microprocessor signals such as 0 and 3V (logic 0 and logic 1) and convert them to a level between +3V to +15V (logic 0) and -3V to -15V (logic 1). The line driver must also provide a certain amount of protection for the microprocessor controlling it.
RS422 and RS485 line drivers use push-pull line drivers which further improve their noise immunity.
Line receivers convert the incoming RS232/422/485 signals to the correct voltage levels for the microprocessor system they are part of.



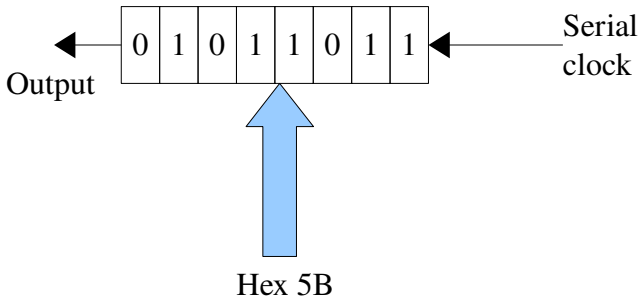
RS232 line driver



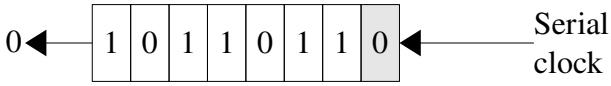
RS422/485 line driver

Inside a UART.

A UART's job is to convert data between serial and parallel formats. It does this by using a shift register. For the purposes of transmission, the shift register is used as a parallel to serial converter. An 8-bit parallel to serial converter is shown below. In the figure, a microprocessor has just written the value 0x5B into the shift register. The shift register is driven by a serial clock. At each clock tick (active clock edge), the contents of the shift register are shifted one bit to the left.



After the first clock tick, the situation is a follows:

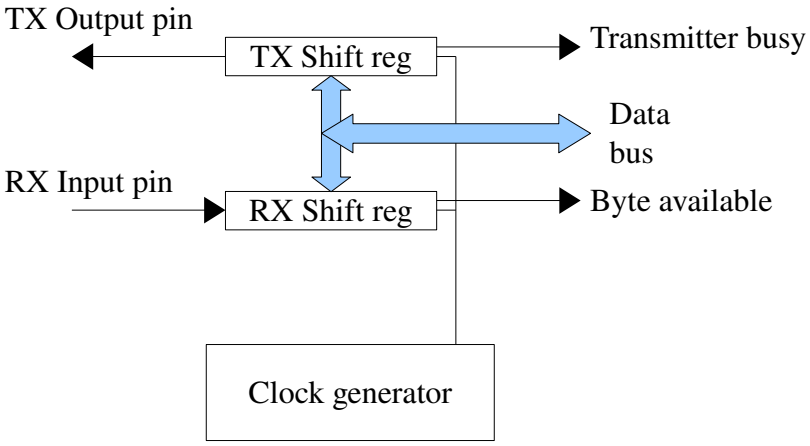


After the second tick we have

The situation continues until all 8 bits have been shifted out of the register. The output may be connected to a pin on the side of the UART which in turn is connected to a line driver for transmission on to a remote system.

Receiving serial data also necessitates the use of a shift register – in this case a serial to parallel converter. The serial to parallel converter shifts data from a serial line into a (perhaps) 8 bit register. When all bits have been received, the process stops.

A complete UART typically consists of the following components:



Transmission and reception are usually driven from the same clock source (which the microprocessor configures to run at a particular rate). The receiver (RX) and transmitter (TX) input and output pins are connected to a line driver. Transmitter and receiver status are reported to the microprocessor – often causing interrupts to be generated. A data bus is used to send/receive parallel data.