UART

A shift register can convert parallel data into serial data, send it to

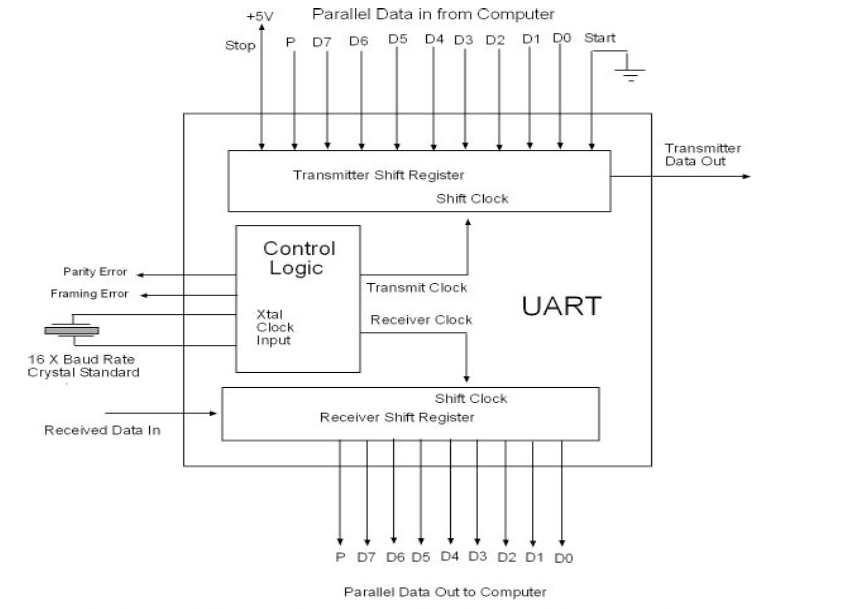
another computer over two wires, and then convert it back to

parallel data. The device that contains two shift registers and the

control circuitry to send and receive serial data is called a Universal

Asynchronous Receiver/Transmitter (UART). The block diagram of a

UART is shown below:



If you put a parallel in serial out shift register in the same package as a

serial in parallel out shift register and add some control and clock logic, you

have created a device known as a UART or Universal Asynchronous Receiver

Transmitter. A UART is used with a computer to convert the computer

parallel data into serial data, to send it out serially, and to then to receiver

data serially and convert it to parallel data back into the computer.

The UART is called an asynchronous device because it does not require the

data to have a clock to identify the location of each serial bit. The clock is

embedded in the signal using an ANSII Standard Format. ANSII stands for

American National Standard for Information Interchange. When data is sent

out, the first bit goes low signaling the UART that data is coming. The eight

data bits follow in serial as D0, D1, D2, D3, D4, D5, D6, and D7. Next

comes a parity bit that is either odd or even. An odd parity bit is added as a

one or a zero to guarantee that there are always an odd number of 1’s in the

bit stream. Even party would do the same but make the number of 1’s in

the serial bit stream an even number. After the parity bit comes a high stop

bit. The process continues for the next eight bit serial byte, continuing until

all of the data is sent.

The rate the data is sent out is called the BAUD rate and it is the number of

transitions per second. Since this rate is known, the UART has its own clock

that is 16 times the BAUD rate. It uses this 16 X over clock system to count

to where the middle of each data position should be and then to sample the

data stream. If at the end of the eleven transitions required to send eight

bits of data, the signal is not high, a framing error flag is sent out. If it is

high, the UART assumes the data is good and continues. The UART next

checks the parity. If the parity is odd and the received data has an even

number of 1’s, the UART sends out a parity error flag. If the parity is also

good, the UART sends the next eight bit byte of data.

Every system that sends and or receives serial data has a UART at the serial

to parallel interface to process the data. Serial data has the advantage of

needing only one wire and a signal return ground to transmit and receive

data from one computer or data system to the next. The UART does the

parallel to serial and serial to parallel conversion needed by most computers

and also controls how fast the data is sent and received.

RS232 is the original COM port standard used by the personal

computer. It is a single line standard that ignores electrical noise on

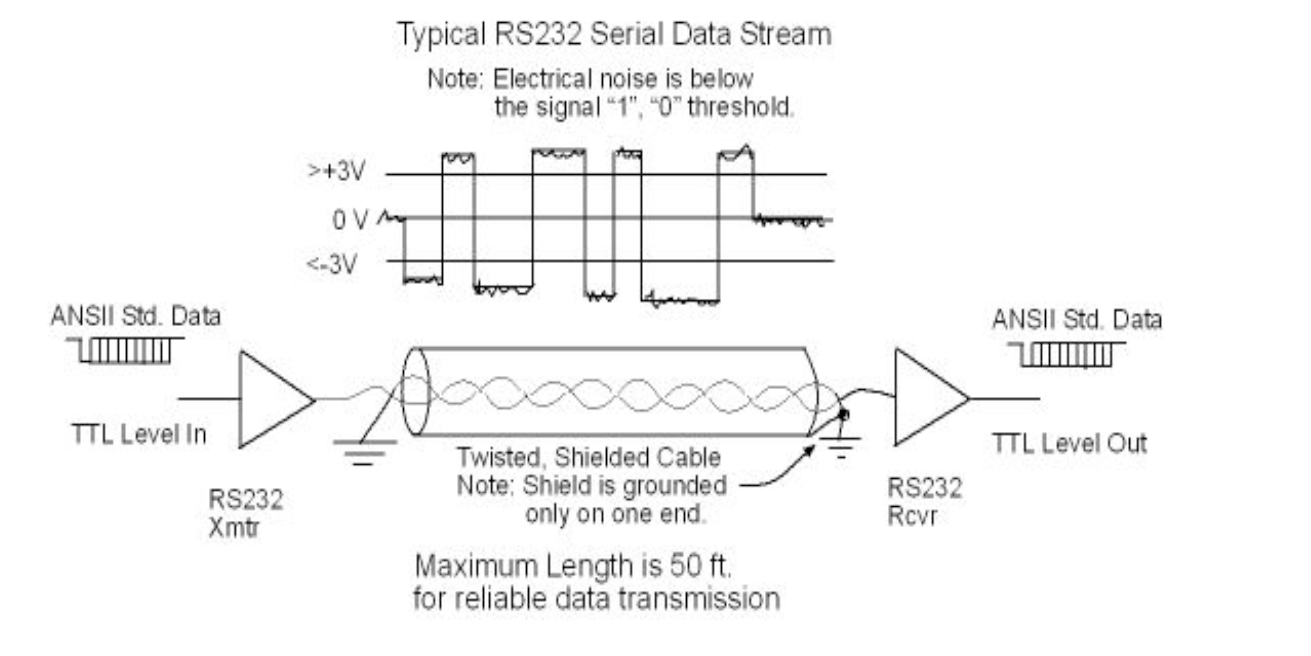
the line by setting the signal threshold beyond the noise. A logical 1

must be at a voltage level between +3 volts to +12 volts. A logical 0

is at a negative voltage level of –3 volts to – 12 volts. There is a

+/- 3 volt noise band in between a logical 1 voltage level and a

logical 0 voltage level.



The electrical standard has nothing to do with the data format or with ANSI

standard for asynchronous data. It only provided a way the data can be

sent as electrically noise immune as possible. The electrical data

transmission standard RS232, works well to ignore electrical disturbances

caused by 60 Hz power as an example. As the distance between two

computers increases, the level of the electrical noise picked up increases

until at around 50 feet, RS232 cables start to have too many errors detected

to work reliably.

The top data bit rate or BAUD rate for RS232 is somewhere around 100

kilobaud. Up until 20 years ago, this was considered a fast serial data

transfer rate. Presently, top serial data transmission rates can exceed a billion bits per second. This has come about through advances in MODEM

technology and the adoption of differential signal transmission standards.

Both of these subjects are discussed in the pages that follow.