Asynchronous Serial Protocols

RS232

RS-232

- RS-232 is one of the most widely used techniques used to interface external equipment to computers.
  - It uses serial communications where one bit is sent along a line, at a time.
  - This differs from parallel communications which send one or more bytes, at a time.
- The main advantage that serial communications has over parallel communications is that a single wire is needed to transmit and another to receive.
- RS-232 is a **de facto standard** that most computer and instrumentation companies comply with.
  - It was standardised in 1962 by the Electronics Industries Association (EIA).
- Unfortunately this standard only allows short cable runs with low bit rates.
  - The standard RS-232 only allows a bit rate of 19 600 bps for a maximum distance of 20 m.
  - New serial communications standards, such as RS-422 and RS-449, allow very long cable runs and high bit rates.
  - For example, RS-422 allows
    - a bit rate of up to 10 Mbps
    - over distances up to 1 mile,
    - using twisted-pair, coaxial cable or optical fibres.
- The new standards can also be used to create computer networks.

Electrical characteristics

Line voltages

- The electrical characteristics of RS-232 defines the minimum and maximum voltages of a logic ‘1’ and ‘0’.
  - A logic ‘1’ ranges from –3 V to –25 V, but will typically be around –12 V.
  - A logical ‘0’ ranges from 3 V to 25 V, but will typically be around +12 V.
  - Any voltage between –3 V and +3 V has an indeterminate logical state.
  - If no pulses are present on the line the voltage level is equivalent to a high level, that is –12 V.
  - A voltage level of 0 V at the receiver is interpreted as a line break or a short circuit.

Frame format

- RS-232 uses asynchronous communication which has a start/stop data format.
- Each character is transmitted one at a time with a delay between them.
  - This delay is called the inactive time and is set at a logic level high (–12 V) as shown in Figure 1.
- The transmitter sends a start bit to inform the receiver that a character is to be sent in the following bit transmission.
  - This start bit is always a ‘0’.
- Next, 5, 6 or 7 data bits are sent as a 7-bit ASCII character, followed by a parity bit and finally either 1, 1.5 or 2 stop bits.
- Figure 1 shows a frame format and an example transmission of the character ‘A’, using odd parity.

RS-232 frame format

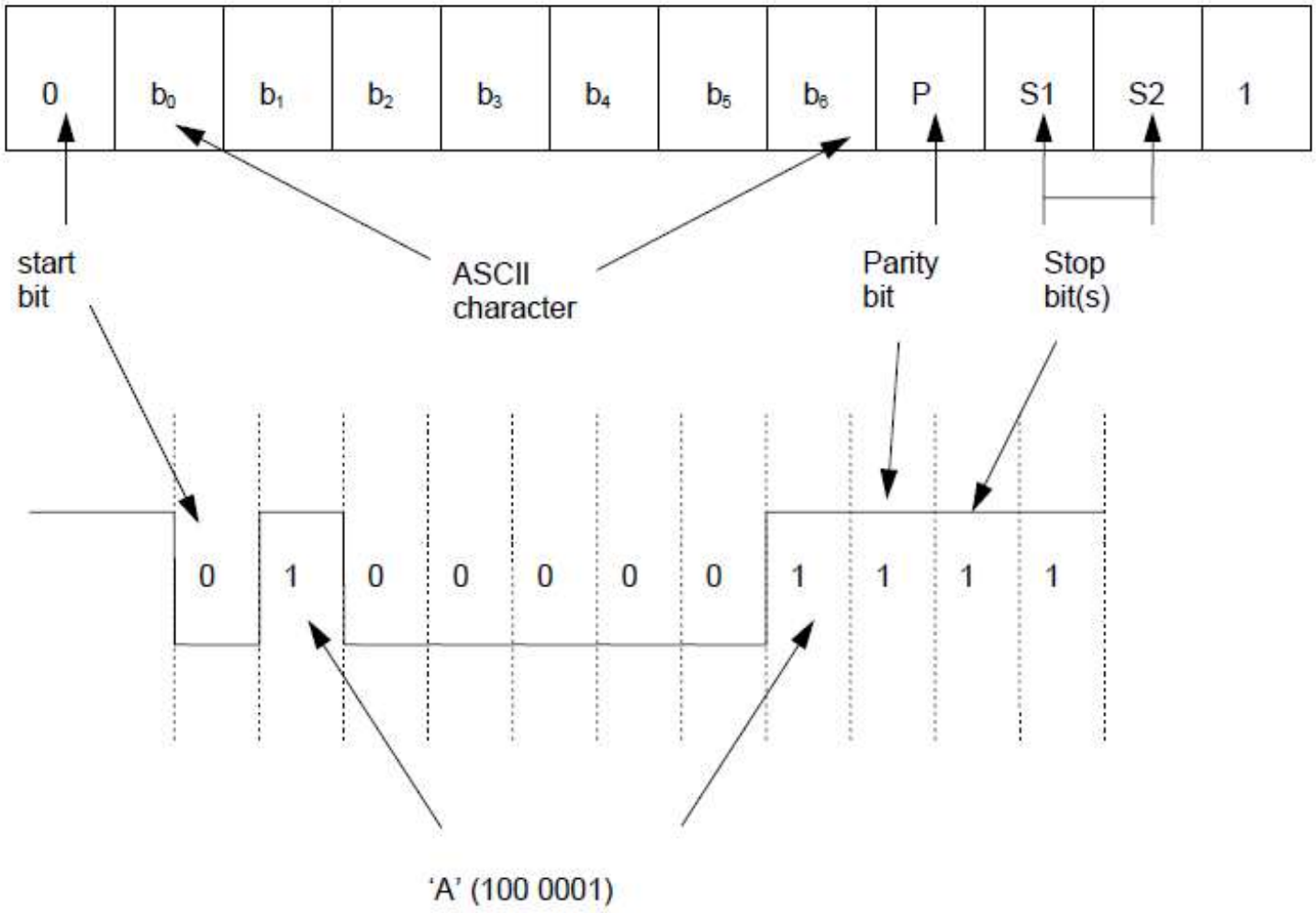


Figure 1: RS-232 frame format

Frame format

- The timing of a single bit sets the rate of transmission.
- Both the transmitter and receiver need to be set to the same bit-time interval.
- An internal clock on both sets this interval.
- These only have to be roughly synchronized at approximately the same rate as data is transmitted in relatively short bursts.

Parity

- Error control is data added to transmitted data in order to detect or correct an error in transmission.
  - RS-232 uses a simple technique known as parity to provide a degree of error detection.
- A parity bit is added to transmitted data to make the number of 1s sent either even (even parity) or odd (odd parity).
- It is a simple method of error coding and only requires exclusive-OR (XOR) gates to generate the parity bit.
- The parity bit is added to the transmitted data by inserting it into the shift register at the correct bit position.
- A single parity bit can only detect an odd number of errors, that is, 1, 3, 5, and so on.
  - If there is an even number of bits in error then the parity bit will be correct and no error will be detected.
- This type of error coding is not normally used on its own where there is the possibility of several bits being in error.

Baud rate

- One of the main parameters, which specify RS-232 communications, is the rate of transmission at which data is transmitted and received.
  - It is important that the transmitter and receiver operate at, roughly, the same speed.
- For asynchronous transmission the start and stop bits are added in addition to the 7 ASCII character bits and the parity.
  - Thus a total of 10 bits are required to transmit a single character.
  - With 2 stop bits, a total of 11 bits are required.
  - If 10 characters are sent every second and if 11 bits are used for each character, then the transmission rate is 110 bits per second (bps).
- The bit rate is measured in bits per second (bps).

Bit stream timings

- Asynchronous communications is a stop/start mode of communication and both the transmitter and receiver must be set up with the same bit timings.
  - A start bit identifies the start of transmission and is always a low logic level.
  - Next, the least significant bit is sent followed by the rest of the bits in the character.
  - After this, the parity bit is sent followed by the stop bit(s).
- The actual timing of each bit relates to the baud rate and can be found using the following formula:

$$\text{Time period of each bit} = \frac{1}{\text{baud rate}} \text{ second}$$

Bit stream timings

- Example: if the baud rate is 9600 baud (or bps) then the time period for each bit sent is 1/9600 s or 104 µs.
- Table shows some bit timings as related to baud rate.
- An example of the voltage levels and timings for the ASCII character ‘V’ is given in figure.

Bit timings related to baud rate

Baud rate	Time for each bit (µs)
1200	833
2400	417
9600	104
19200	52

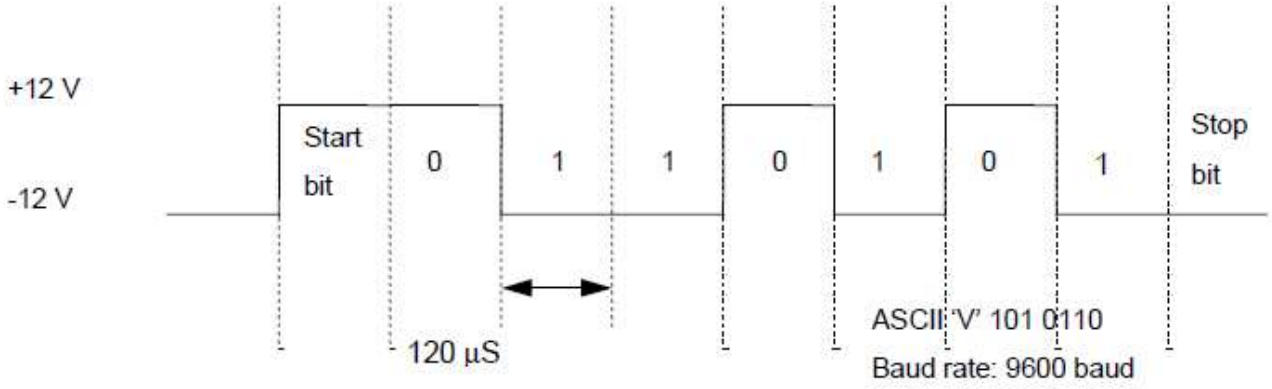


Figure 2: ASCII ‘V’ at RS-232 voltage levels

Communications between two nodes

- RS-232 is intended to be a standard but not all manufacturers abide by it.
  - Some implement the full specification while others implement just a partial specification.
- This is mainly because not every device requires the full functionality of RS-232, for example a modem requires many more control lines than a serial mouse.
- The rate at which data is transmitted and the speed at which the transmitter and receiver can transmit/receive the data dictates whether data handshaking is required.

Transmission and reception of characters

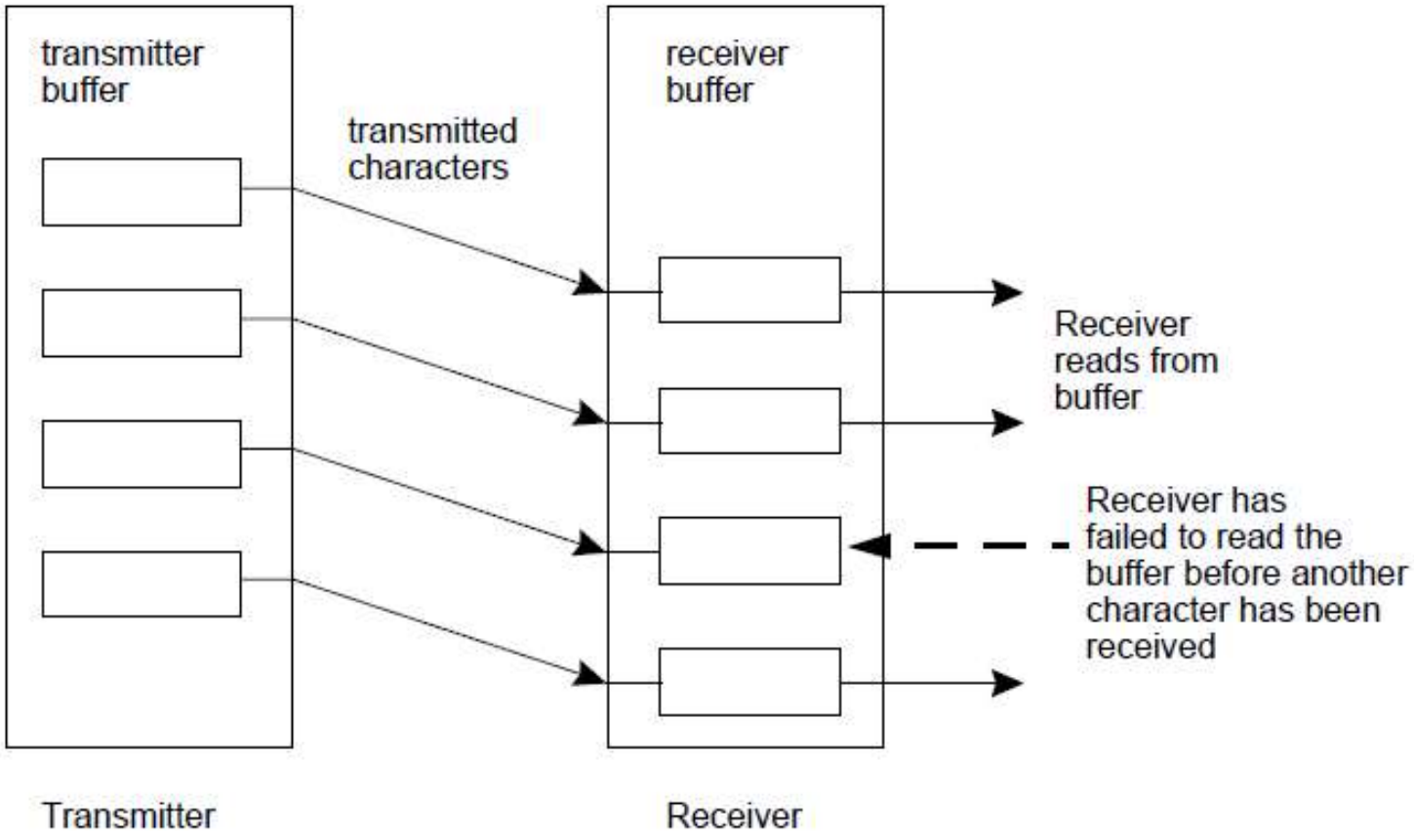


Figure 3: Transmission and reception of characters

Handshaking

- In the transmission of data, there can either be
  - no handshaking,
  - hardware handshaking or
  - software handshaking.
- If no handshaking is used then the receiver must be able to read the received characters before the transmitter sends another.
  - The receiver may buffer the received character and store it in a special memory location before it is read.
  - This memory location is named the **receiver buffer**.
  - Typically, it may only hold a single character. If it is not emptied before another character is received then any character previously in the buffer will be overwritten.
- An example of this is illustrated in Figure 3.
  - In this case, the receiver has read the first two characters successfully from the receiver buffer, but it did not read the third character as the fourth transmitted character has overwritten it in the receiver buffer.
  - If this condition occurs then some form of handshaking must be used to stop the transmitter sending characters before the receiver has had time to service the received characters.

Handshaking

- Hardware handshaking involves the transmitter asking the receiver if it is ready to receive data.
- If the receiver buffer is empty it will inform the transmitter that it is ready to receive data.
- Once the data is transmitted and loaded into the receiver buffer the transmitter is informed not to transmit any more characters until the character in the receiver buffer has been read.
- The main hardware handshaking lines used for this purpose are:
  - CTS – Clear to send.
  - RTS – Ready to send.
  - DTR – Data terminal ready.
  - DSR – Data set ready.
- Software handshaking involves sending special control characters. These include the DC1 (Xon)-DC4 (Xoff) control characters.

Simple no-handshaking communications

- In this form of communication it is assumed that the receiver can read the received data from the receive buffer before another character is received.
  - Data is sent from a TD pin connection of the transmitter and is received in the RD pin connection at the receiver.
  - When a DTE (such as a computer) connects to another DTE, then the transmit line (TD) on one is connected to the receive (RD) of the other and vice versa.

Simple no-handshaking communications

- Figure shows the connections between the nodes.

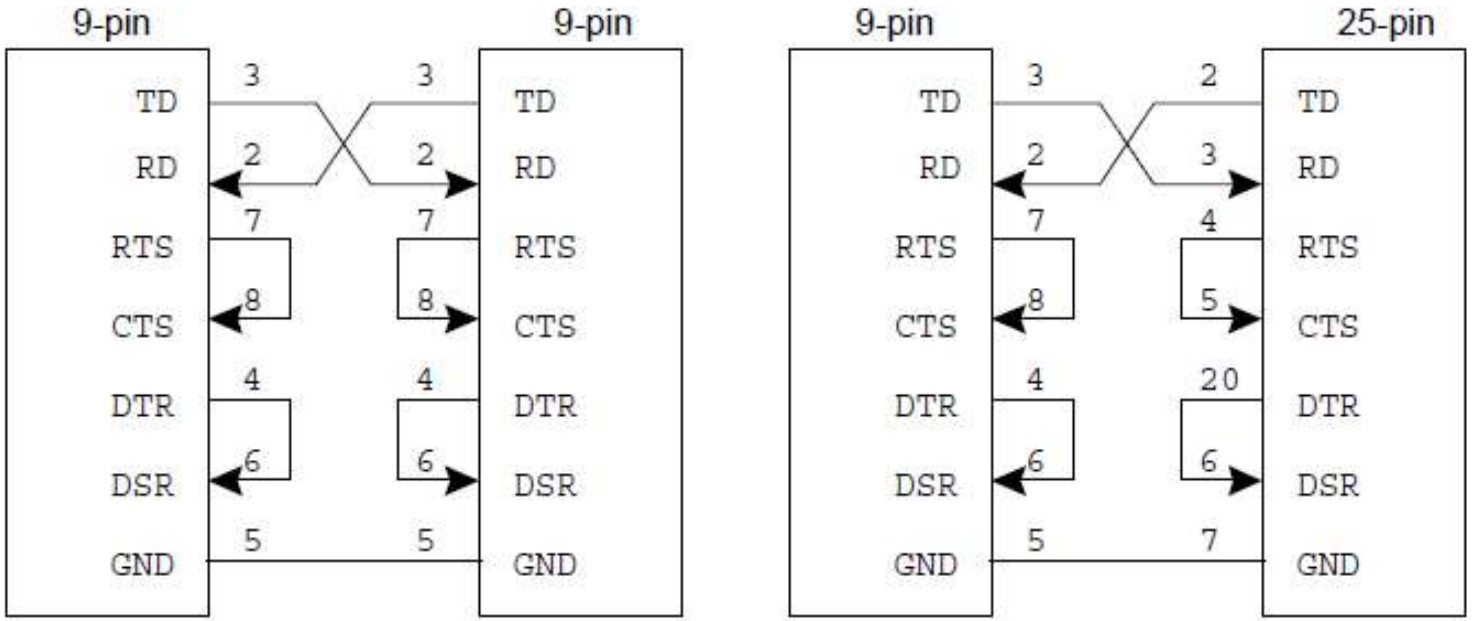


Figure 4: RS-232 connections with no hardware handshaking

Software handshaking

- There are two ASCII characters that start and stop communications.
  - X-ON (^S , Cntrl-S or ASCII 11)
  - X-OFF (^Q, Cntrl-Q or ASCII 13).
- When the transmitter receives an X-OFF character it ceases communications until an X-ON character is sent.
- This type of handshaking is normally used when the transmitter and receiver can process data relatively quickly.
  - Normally, the receiver will also have a large buffer for the incoming characters.
  - When this buffer is full, it transmits an X-OFF.
  - After it has read from the buffer the X-ON is transmitted.

Software handshaking

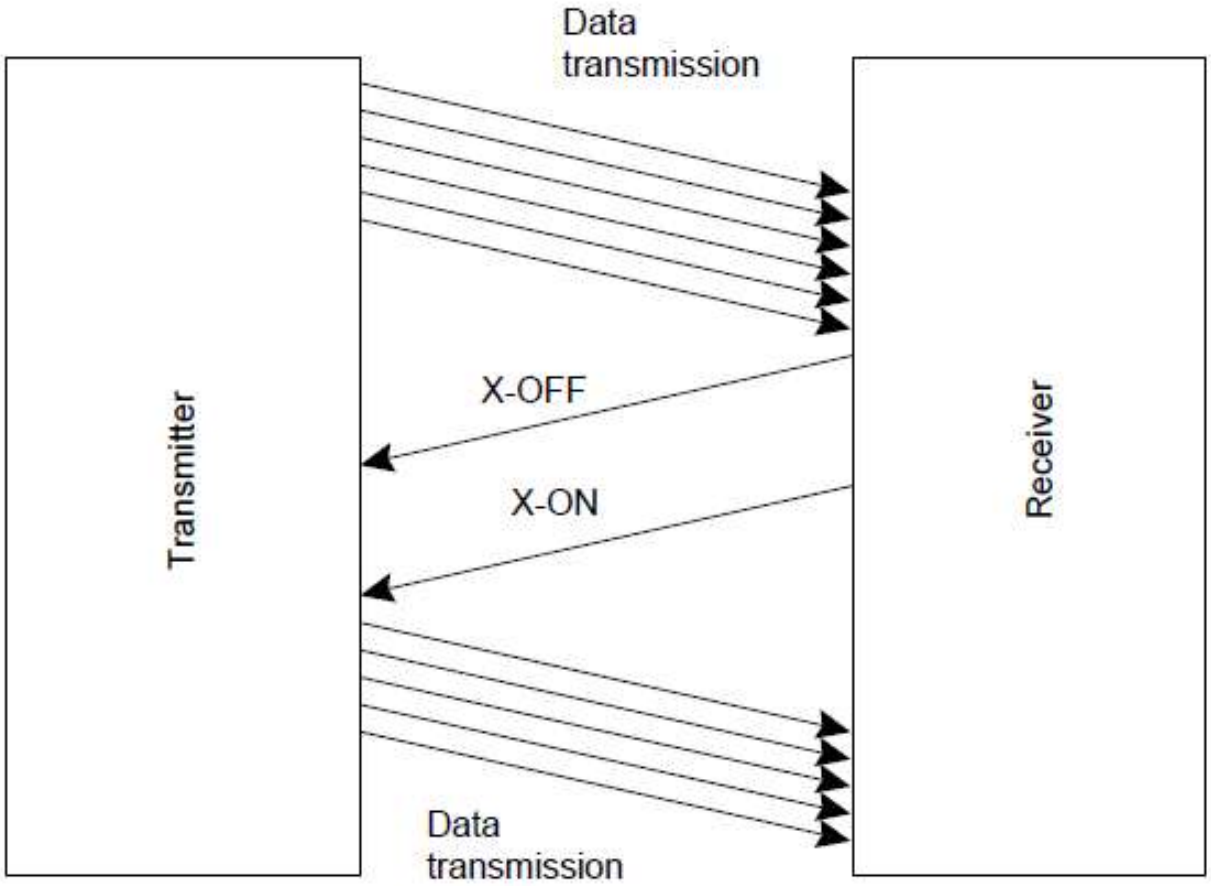


Figure 5: Software handshaking using X-ON and X-OFF

Hardware handshaking

- Hardware handshaking stops characters in the receiver buffer from being overwritten.
- The control lines used are all active HIGH.
  - When a node wishes to transmit data it asserts the RTS line active (that is, HIGH).
  - It then monitors the CTS line until it goes active (that is, HIGH).
  - If the CTS line at the transmitter stays inactive then the receiver is busy and cannot receive data, at the present.
  - When the receiver reads from its buffer the RTS line will automatically go active indicating to the transmitter that it is now ready to receive a character.

Hardware handshaking

- Receiving data is similar to the transmission of data, but the lines DSR and DTR are used instead of RTS and CTS.
- When the DCE wishes to transmit to the DTE the DSR input to the receiver will become active.
  - If the receiver cannot receive the character, it will set the DTR line inactive.
  - When it is clear to receive it sets the DTR line active and the remote node then transmits the character.
  - The DTR line will be set inactive until the character has been processed.

Hardware handshaking

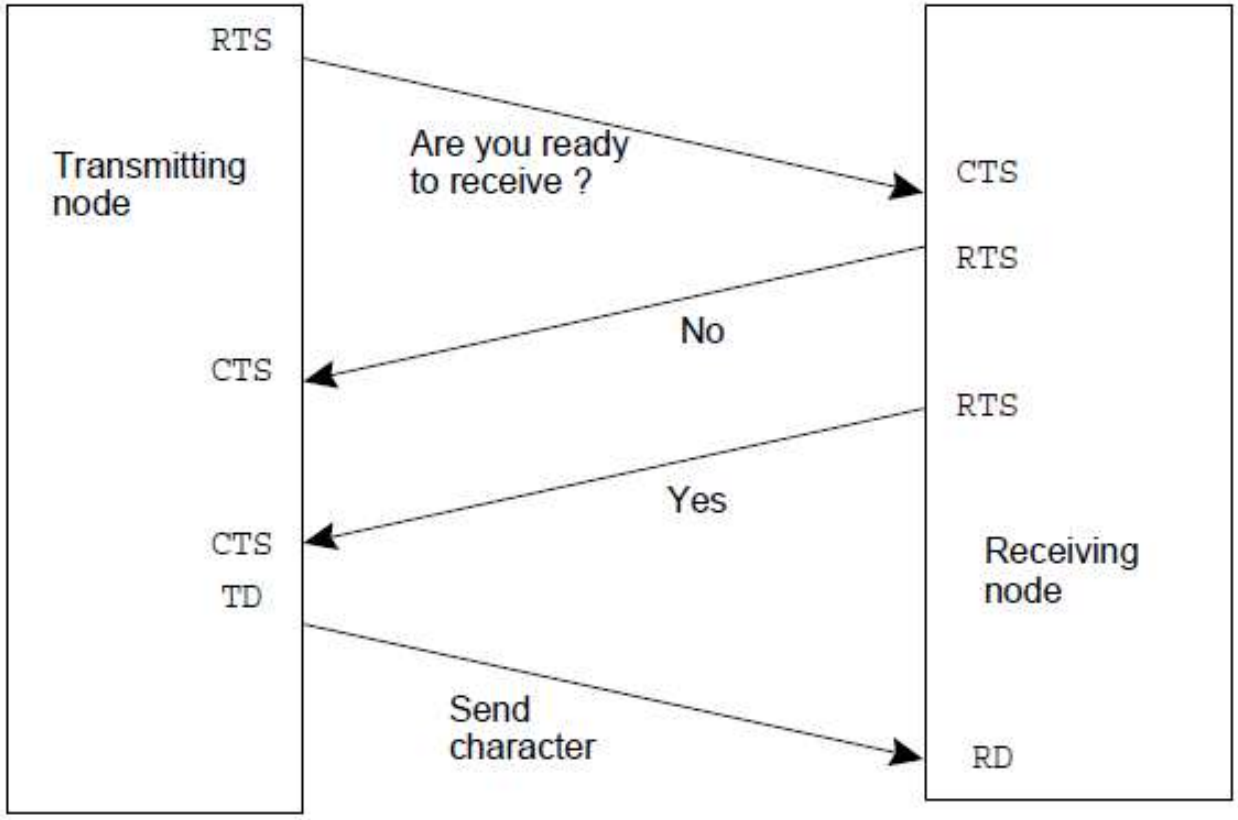


Figure 6: Handshaking lines used in transmitting data

Two-way communications with handshaking

- For full handshaking of the data between two nodes the RTS and CTS lines are crossed over (as are the DTR and DSR lines).
- This allows for full remote node feedback

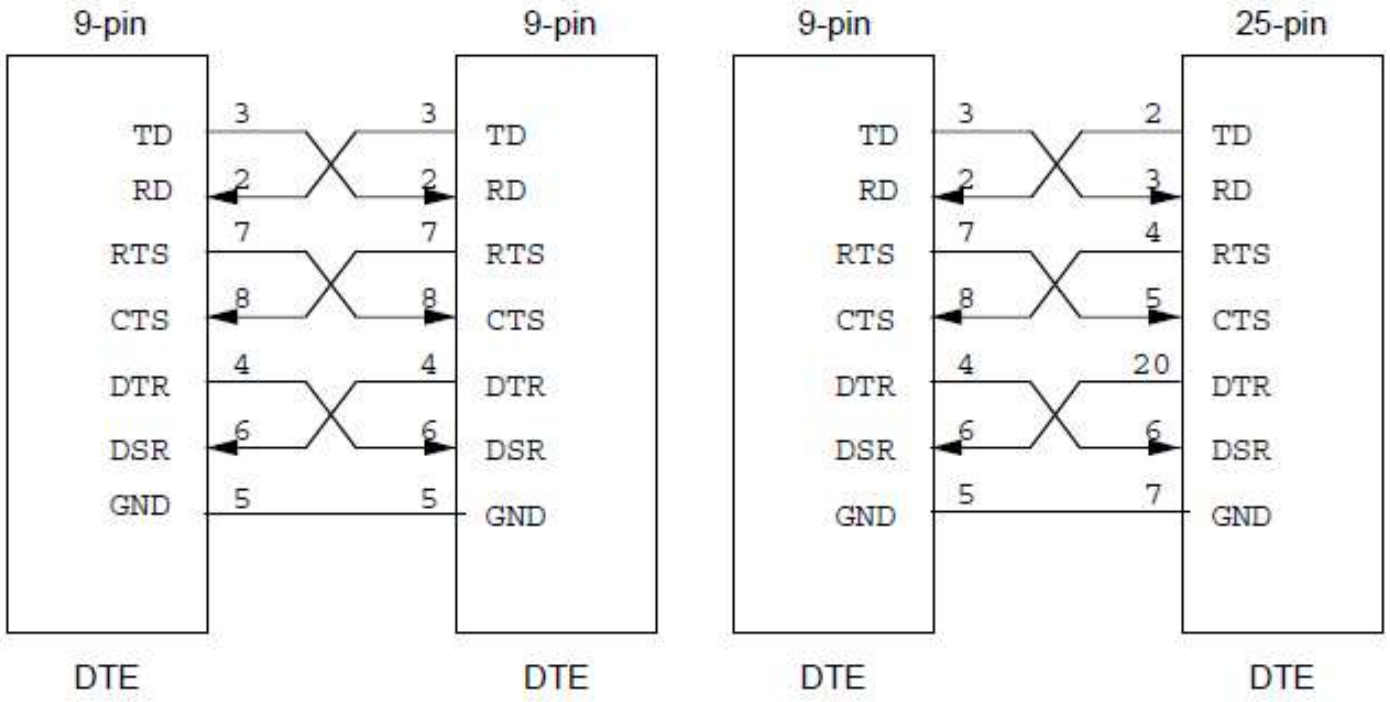


Figure 7: RS-232 communications with handshaking

DTE-DCE connections (PC to modem)

- A further problem occurs in connecting two nodes.
  - DTE/DTE connection requires crossovers on their signal lines,
  - DTE/DCE connections require straight-through lines.

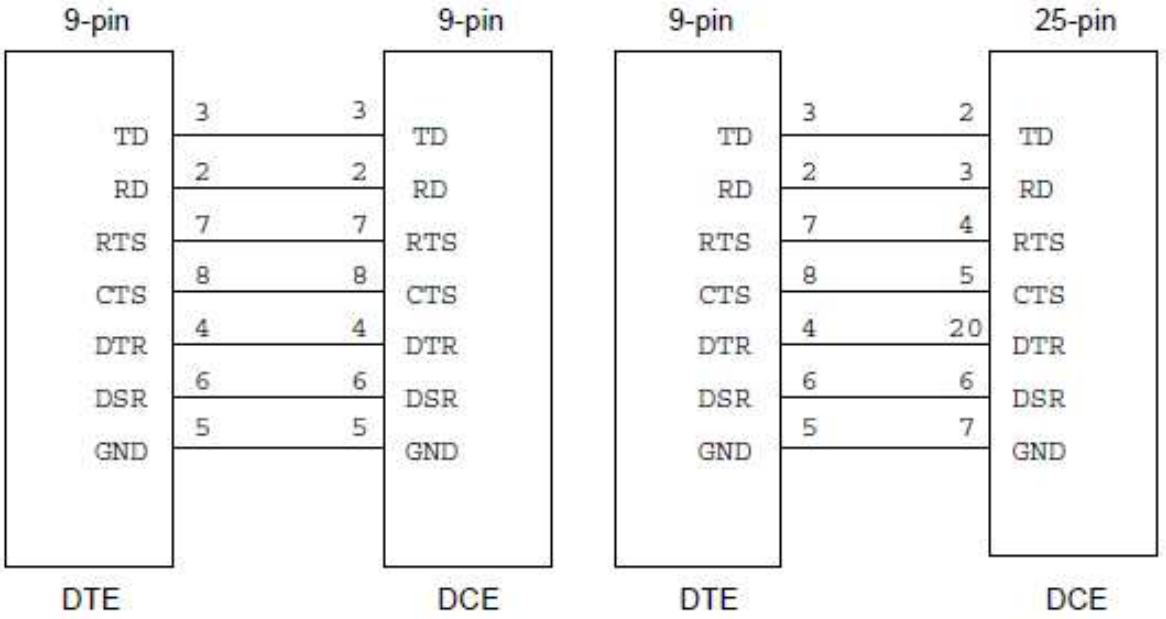


Figure 8: DTE to DCE connections

RS485

Introduction

- The main standards organizations for data communications are the
  - ITU (International Telecommunications Union), standards are the V-series
  - the EIA (Electronic Industry Association) ; standards are the RS-series.
  - ISO (International Standards Organization).
- The EIA has defined many standards for serial communications.
- RS-232 has many limitations, such as:
  - One transmitter and one receiver.
  - Maximum connection length of 20m.
  - Maximum baud rate of 20 kbps.

RS 422 and RS485

- The RS-422 and RS-423 standards replace the RS-232 standards and support higher data rates and greater immunity to electrical interference. The main standards are:
  - RS-422A – Supports multipoint connections.
    - It defines the electrical characteristics of balanced load voltage digital interface circuits.
  - RS-423A – Supports only point-to-point connections.
    - It defines electrical characteristics of unbalanced voltage digital interface circuits.
  - RS-449 – Defines the basic interface standards and refers to the RS-422/3 standards.
    - It defines a general-purpose 37-position and 9-position interface for DTE and DCE employing serial binary data interchange.
  - RS-485 – Similar to RS-422
    - can support more nodes per line because it uses lower impedance drivers and receivers.

RS-485 (ISO 8482)

- RS-485 is an upgraded version of RS-422 and extends the number of peripherals that can be interfaced.
  - It allows for bidirectional multipoint party line communications. This can be used in networking applications.
    - RS-422 and RS-232 facilitate simplex communication
    - RS-485 allows for multiple receivers on a single line, facilitating half-duplex communications.
- The maximum data rate is unlimited and is set by the rise time of the pulses, but it is usually limited to 10 Mbps.
  - A network using the RS-485 standard can have up to 32 transmitters/receivers with a maximum cable length of 1.2 km, as shown in Figure 13.
  - The maximum cable length is 1200m.

RS-485 connecting to multiple nodes

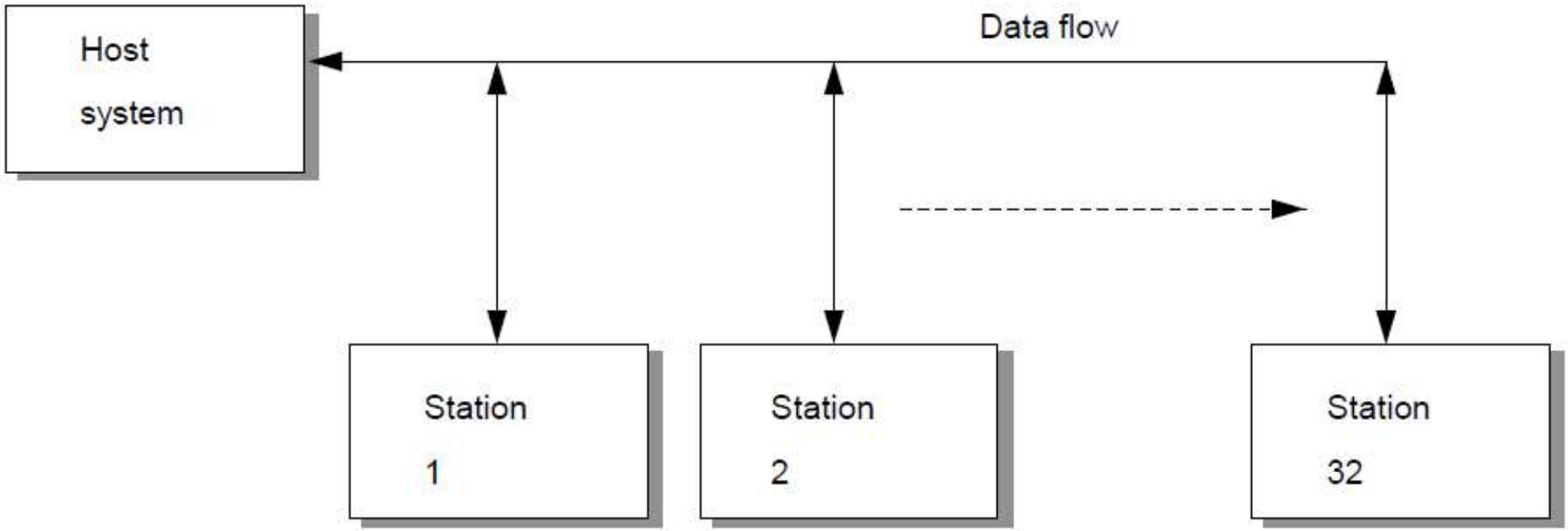


Figure 13: RS-485 connecting to multiple nodes

Multiple Nodes

- The RS-485 two-wire connection involves a half-duplex transmission mode, that is, only one device can transmit data at a time. This thus involves a polling procedure, with a master and up to 32 slaves. The slaves must wait in a high-impedance state (a stand-by mode). The control of the driver on the slave is either with:
  - An active RTS line.
  - Bit changes on the transmit data line.
  - Sends of the X-ON/X-OFF flow control characters.
- Slaves cannot send data unless they are selected.
- In a four-wire operation RS-485 devices connect with two twisted pair cables with characteristic impedance of 120Ω and general shielding. Each link requires a terminating load on the ends of the cable.

Multiple Nodes

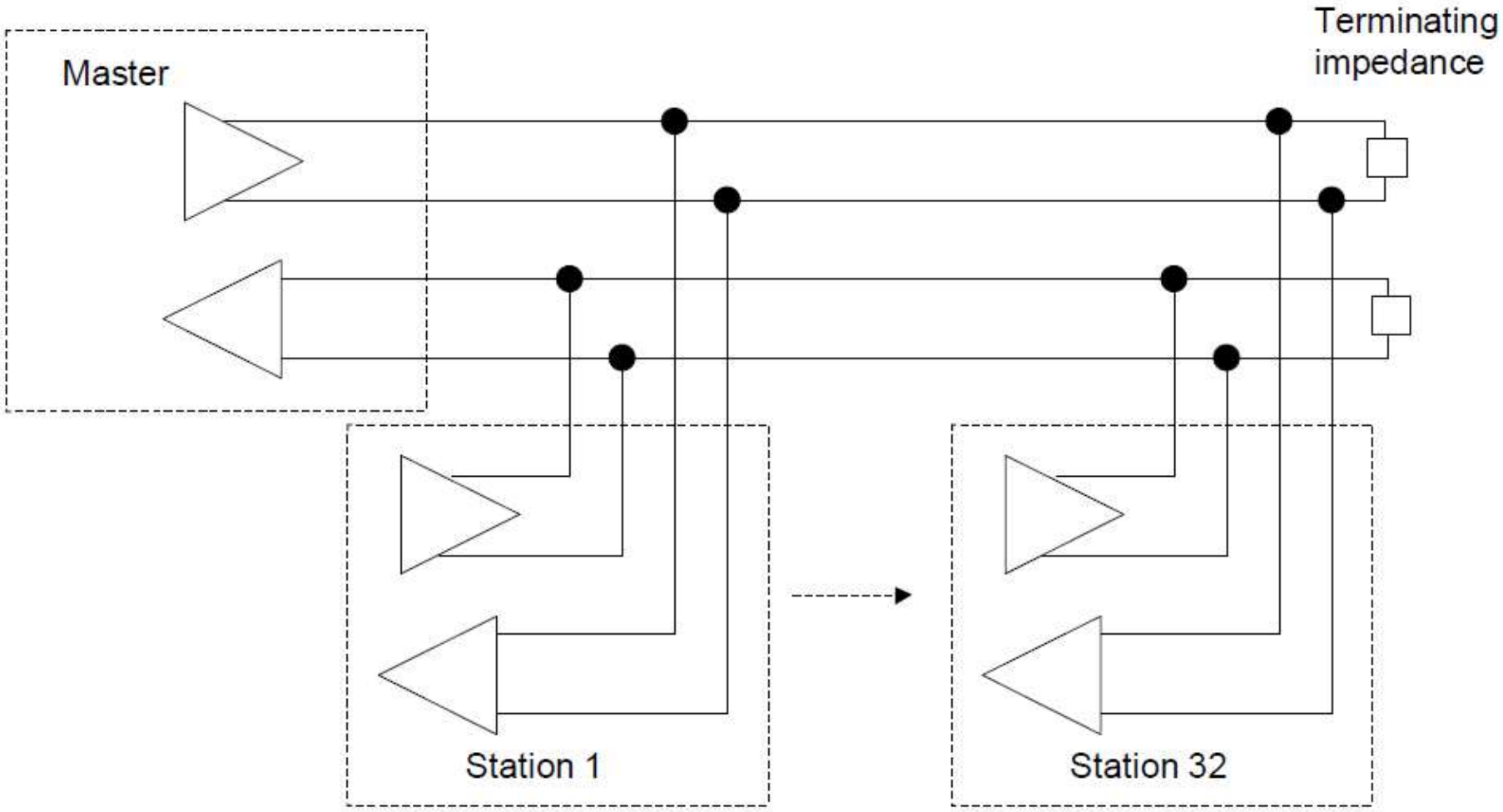


Figure 14: RS-485 connecting to multiple nodes

Operating Modes

- RS-485 operates in one of two modes:
  - Two-wire, multidrop, party line – in this mode, a balanced transmission line is used to connect to all of the stations, which share a common communications channel. Up to 32 driver/receiver pairs can share the common channel.
  - Four-wire – in this mode, each station connects to a four-wire bus, as illustrated in Figure 14. It is necessary in this mode that one node acts a master station and all others as slaves. The master then communicates with each of the slaves. All slave nodes communicate only with the master node. A master–slave network is useful when mixed protocols are used.

Line drivers

- Transmission lines have effects on digital pulses in the following ways:
  - Attenuation – The transmission line contains series resistance that causes a reduction in the pulse amplitude.
  - Pulse distortion – The transmission line insulation produces a shunt capacitance on the signal path and a series resistance and inductance of the conductors. This causes the transmission line to distort the shape of the pulse. The two main effects are the block of high frequencies in the pulse and phase distortion.
  - Noise – Noise is any unwanted electrical signals added to a signal. A digital system is less prone to noise as it has only two levels and it takes a relatively large change in voltage to cause an error.

Electrical Characteristics

- Table shows the electrical characteristics of the different serial communication standards.
- The two main standards agencies are the EIA and the ITU.
- Balanced lines use two lines for each signal line, whereas unbalanced lines use one wire for each signal and a common return circuit (see Figure 15).
  - RS-422 is a balanced interface and uses two conductors to carry the signal (see Figure 16). The electrical currents in each of the conductors are 180° out-of-phase with each other.
  - Balanced lines are generally less prone to noise as any noise induced into the conductors will be of equal magnitude. At the receiver the noise will tend to cancel out.

Unbalanced / Balanced Circuits

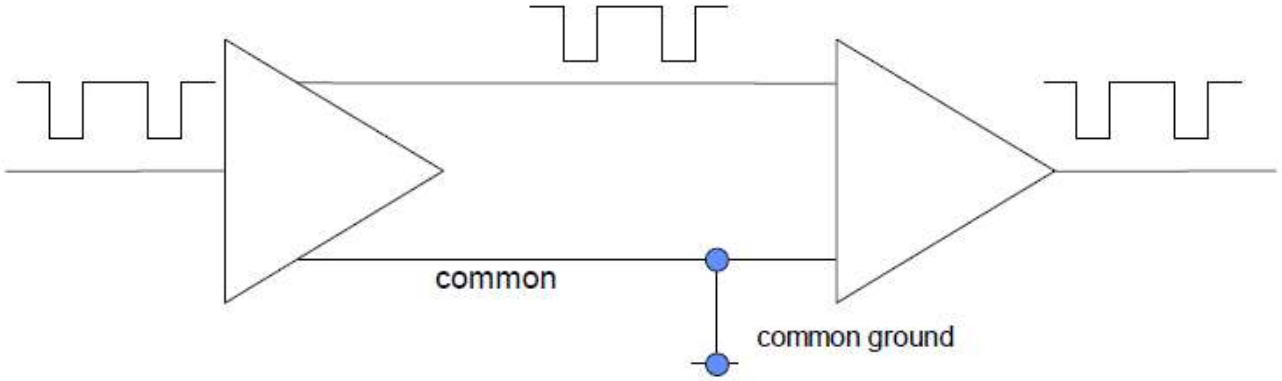


Figure 15:Unbalanced digital interface circuit (RS-423)

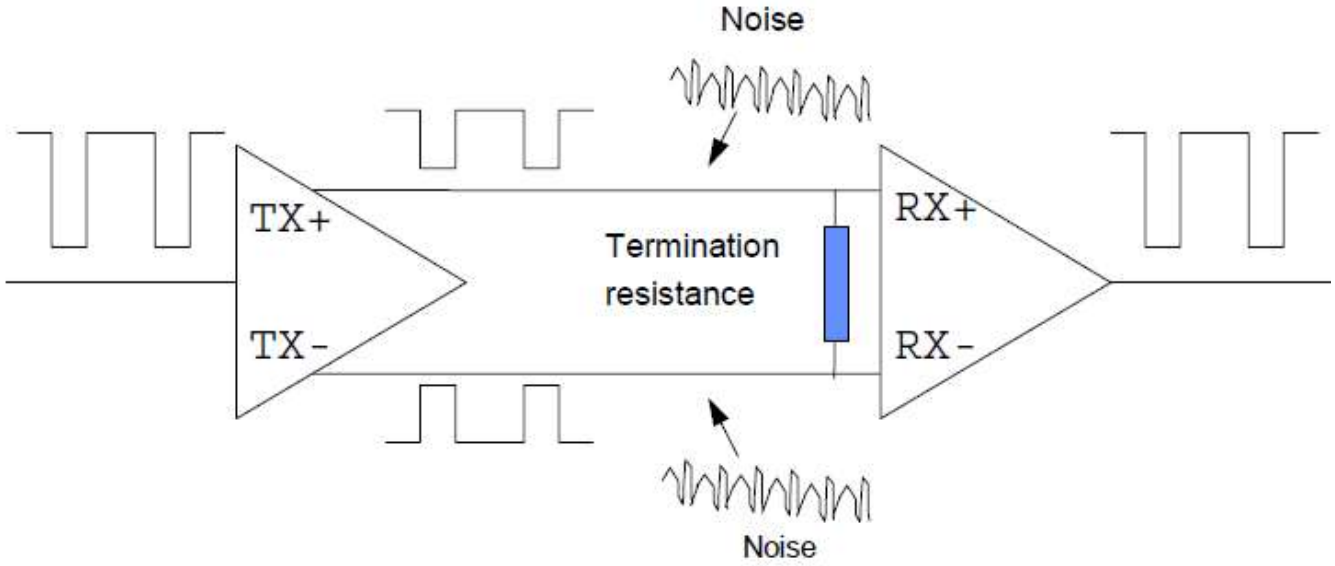


Figure 16: Balanced digital interface circuit (RS-422)

Electrical Characteristics

- The voltage levels for RS-232 range from ±3 to ±25 V, whereas, for RS422/ RS423 the voltage ranges are ±0.2 to ±6 V.
  - For very high bit rates the cable is normally terminated with the characteristic impedance of the line; for example, a 50Ω cable is terminated with a 50Ω termination.
- RS-422 interface circuits can have up to 10 receivers.
  - They have no ground connection and are thus useful in isolating two nodes.
  - For two-way communications four connections are required, the TX+ and TX- on one node connects to the RX+ and RX- on the other.

Electrical Characteristics

- Nodes may have a direct RS-422 connection or can be fitted with a special interface adaptor to convert from RS-232 to RS-422 (although the maximum data rate is likely to be limited to the maximum RS-232 rate).
- It should also be noted that the maximum connection distance relates to the maximum data rate.
  - If a lower data rate is used then the maximum distance can be increased.
  - For example, in some situations with a good quality cable and in a low noise environment, it is possible to have cable runs of 1 km using RS-232 at 1200 bps.

RS-485 Characteristics

- Like RS-422, RS-485 also uses differential signals, so there are two voltage signals to indicate the data by their relative voltages.
- RS-485 outputs can be tri-stated to allow multiple devices to transmit on the same pair of signals, but otherwise they are electrically the same as RS-422 signals.
- A master/slave arrangement is most common for RS-485 systems and this can be implemented
  - with one (see Figure 19) or
  - two (see Figure 20) pair(s) of wires.
- All devices can communicate over the single pair of wires, or the master device can transmit on a separate pair of wires which simplifies the management of the traffic on the bus.

RS-485 Four-wire Connection

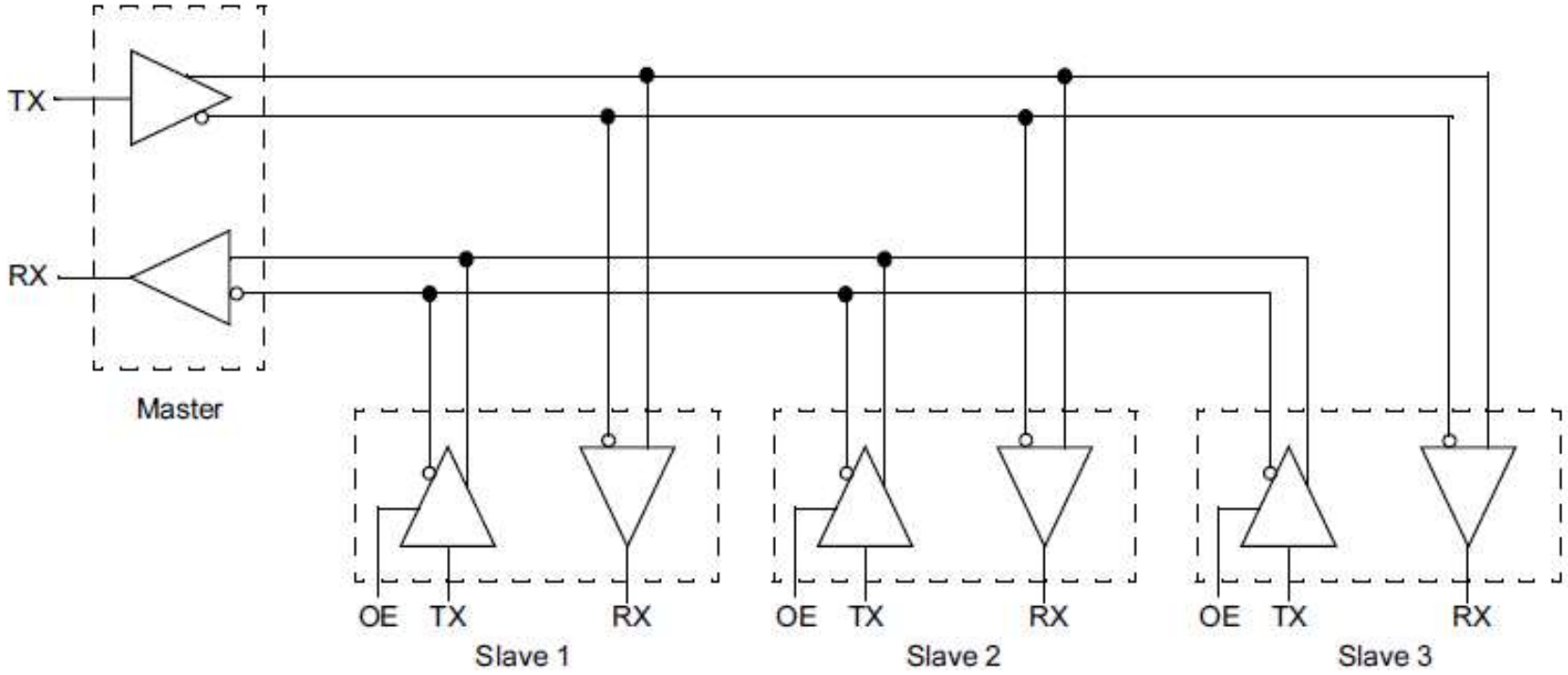


Figure 19: Typical RS-485 Four-wire Connection

RS-485 Two-wire Connection

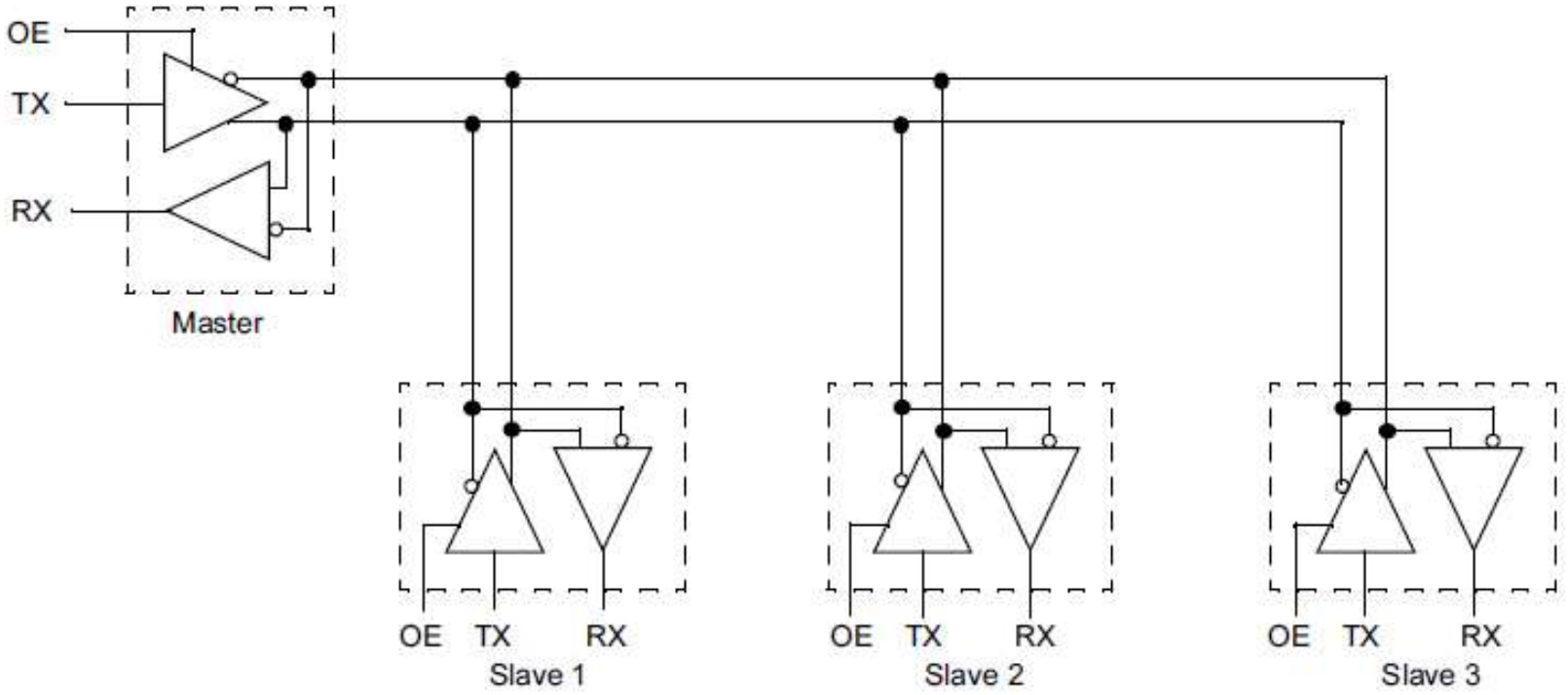


Figure 20: Typical RS-485 Two-wire Connection

RS-485 Implementation

- RS-485 software can be significantly more complex than for RS-232 and RS-422, particularly when two wire bus systems are used.
- A protocol is required to ensure that no more than one device transmits at any one time.
- RS-485 busses are often implemented with a single master and numerous slaves.
  - The master is the only device that initiates communication on the bus and this avoids bus contention problems.

RS-485 Implementation

- Typically, the master broadcasts an address and data, then receives data back from the particular slave being addressed.
  - Each slave must have a unique address and must know what data format to expect and to return.
  - In addition, each device must control the output enable signal on its transceiver to enable the output only while it is transmitting.
- Since all the devices usually use the same physical connection, it is essential to avoid driving the signal on the bus except when transmitting.
  - The user must ensure that the receivers have a valid state when no device is driving the bus.
  - Typically, this can be done with resistors but it is best to check with the manufacturer of the particular driver being used.

Using The Ninth Data Bit

- The USART can handle data lengths of 8 bits or 9 bits.
- The most common RS-232 applications use 8 bits, but there several reasons why 9 bits might be used:
  - The data is 9 bits in length
  - The data is 8 bits and two STOP bits are required
  - The data is 8 bits and parity is required
  - The data is 8 bits and 9th bit address detect is required
- The TX9 bit in TXSTA and the RX9 bit in RCSTA must be set to enable transmission and reception of the ninth bit.
  - The order of reading and writing the data is very important when doing nine-bit operations
  - As a simple rule, always read or write the ninth bit before the other eight bits of data.
  - Under certain circumstances, if you can be certain that the data will be read before the next reception completes, it is possible to read the RX9D bit after reading RCREG but this is not good practice.

Using The Ninth Data Bit

- The USART in many PICmicro devices has a 9-bit Address Detect mode that is enabled with the ADDEN bit in the RCSTA register.
  - When this bit is set, the USART ignores all received data unless the ninth bit is set.
  - This feature can be used to implement RS-485 system where the ninth bit indicates that the master is transmitting a slave address.
  - This allows the slave devices to automatically ignore all transmissions until an address is broadcast.
  - The slaves can then compare the received data to their own addresses, and if they match, the ADDEN bit can be cleared so that they receive the data that follows.
  - This reduces the software overhead of the slaves and makes slave software easier to implement and more efficient.

Main serial standards

EIA ITU	RS-232-C V.28	RS-423-A V.10/X.26	RS-422-A V.11/X.27	RS-485
Data rate	20 kbps	300 kbps	10 Mbps	10 Mbps
Max distance	15 m	1200 m	1200 m	1200 m
Type	Unbalanced	Unbalanced	Balanced differential	Balanced differential
Number of drivers and receivers	1 driver 1 receiver	1 driver 10 receivers	1 driver 10 receivers	32 drivers 32 receivers
Driver voltages	±12 V	± 6 V	±5 V	±5 V
Number of conductors per signal	1	2	2	2

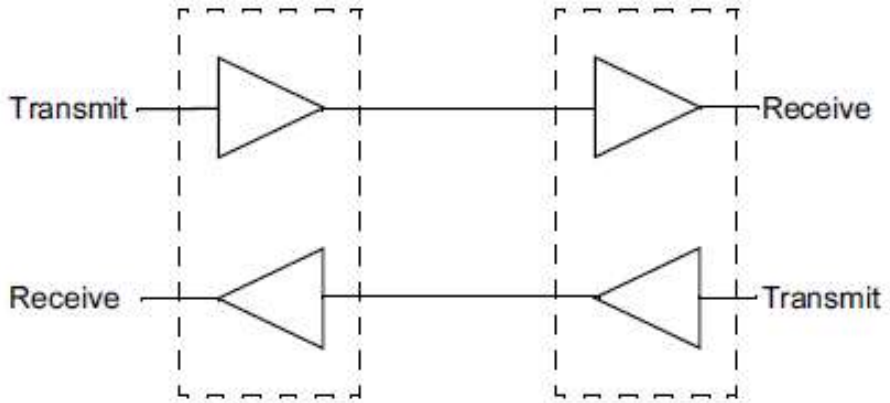
Implementing Serial Protocols with PICmicros USART

AN774

RS-232 Characteristics

- RS-232 uses a single ended signal (referenced to ground) to indicate the data.
  - Typically, there is a ground reference (GND) and two signals, a transmit (TXD) output and a receive (RXD) input.
  - There are also other signals that may be used, including hardware handshaking signals.
- The simplest bi-directional RS-232 system has two wires.
- Transceivers are usually used to convert between the logic levels of a microcontroller and the RS-232 voltage levels.

Figure 17: Typical Rs-232 Connection



RS-232 Implementation

- The program code required to perform RS-232 communications can be simple, since the USART typically transmits and receives data in the form of bytes.
- The code needs to detect whether data has been received or can be transmitted and can then read or write a register to get or send the data.
- No signals need to be enabled or disabled. RS-232 software can be more complicated if the data format changes; for example, if a parity bit or multiple STOP bits are used, or if hardware flow control is required.

Break Signal

- A special symbol defined in the RS-232 interface is the break signal.
  - The break is a zero output that is maintained for some period longer than a single transmission.
- It is usually used to indicate or request a special event such as indicating a problem, causing an interruption, hanging up a modem, etc.
- The USART has no inherent way to generate or detect a break, but it can be done.
- To transmit a break, the baud rate generator can be set to a lower baud rate for a single transmission of a zero byte.

Break Signal

- The length of a break is not defined and it is typically in the order of 100 ms to 500 ms, although anything longer than a single transmission time is valid.
- Because of the temporary baud rate change, the USART will not be able to receive data correctly while transmitting the break. Before changing baud rates to send a break, the TRMT bit should be checked to ensure that the last transmission was completed.
- When receiving a break, the first indication that a break condition exists is that a framing error has been detected.
  - This is because when the STOP bit is expected to be a one, the break will keep the signal at zero.
  - Receiving data of zero with a framing error is usually sufficient to conclude that a break has occurred.

USART

- USART stands for Universal Synchronous Asynchronous Receiver Transmitter and its main function is to transmit or receive serial data.
- Its operation can be divided into two broad categories:
  - synchronous
  - asynchronous.
- **Synchronous** operation uses a clock and data line
- **Asynchronous** operation has no separate clock accompanying the data.
- There are substantial differences between these two modes of operation and this presentation is concerned only with asynchronous operation.

Overview

- The USART can transmit and receive data serially. It can transfer a frame of data with eight or nine data bits per transmission and detect errors when data is overwritten or incorrectly framed.
- It can generate interrupts when a reception occurs (or a transmission completes) and it contains data buffers that simplify the timing requirements of the software controlling the USART.
- Some parts have an addressable USART that uses the ninth data bit to distinguish between address and data receptions.
  - This allows simple filtering of incoming data and is often used in the RS-485 protocol.

PIC Microcontroller Families

- The USART is incorporated into many PIC16, PIC17 and PIC18 parts. The USARTs in all these parts are basically the same, but there are some important differences to consider:
  - The USART is addressable in all current PIC18 parts and is not addressable in all PIC17 parts.
  - The USART is addressable in some PIC16 parts and is not addressable in other PIC16 parts.
  - The USART in PIC17 parts does not have a high speed baud rate option, while the USART in all PIC16 and PIC18 parts does have a high speed baud rate option.
  - Most devices with a USART have only one. The PIC17C7XX and some PIC18FXX20 parts have two USARTs. Future PIC18 parts may have two USARTs.
  - The various families have different interrupt control registers and the PIC18 parts allow the interrupts to be prioritized.
  - Some new PIC18 parts have an enhanced USART with added features not covered here.

Special Function Registers

- Several special function registers control the USART.
- These registers allow the various modes of operation to be selected, the baud rate (i.e., bit rate) to be set up, data to be transferred, the transmit and receive status to be monitored, etc.
- The registers that affect the USART are shown in the following tables.

TXSTA and RCSTA Registers

- The TXSTA and RCSTA registers are used to control transmission and reception but there are some overlapping functions and both registers are always used.
  - Parts with two USARTs have TXSTA1, RCSTA1, TXSTA2 and RCSTA2 registers instead of a single pair of TXSTA and RCSTA registers.

Table 1: Registers That Control Transmission And Reception

Register Name	Description
TXSTA	Transmit Status and Control
RCSTA	Receive Status and Control

TXREG and RCREG Registers

- The TXREG and RCREG registers are used to write data to be transmitted and to read the received data.
  - Parts with two USARTs have TXREG1, RCREG1, TXREG2 and RCREG2 registers instead of a single pair of TXREG and RCREG registers.

Table 2: Registers Used To Write And Read Data

Register Name	Description
TXREG	Transmit Data Register
RCREG	Receive Data Register

SPBRG register

- The SPBRG register allows the baud rate to be set.
  - Parts with two USARTs have SPBRG1 and SPBRG2 registers instead of a single SPBRG register.

Register Name	Description
SPBRG	Baud Rate Generator

Interrupt Registers

- In addition, there are interrupt registers that control the interrupts but are also used to determine whether data has been received or can be transmitted.
  - Interrupts are often used when the PICmicro MCU processor is busy executing code and data needs to be transmitted or received in the background.

Interrupt Registers For PIC16 Devices

Register Name	Description
INTCON	Interrupt Control Register
PIR1	Peripheral Interrupt Flag Register
PIE1	Peripheral Interrupt Enable Register

Interrupt Registers For Pic18 Devices

Register Name	Description
INTCON	Interrupt Control Register
RCON	RESET Control Register
PIE1, PIE3	Peripheral Interrupt Enable Registers
PIR1, PIR3	Peripheral Interrupt Flag Registers
IPR1, IPR3	Peripheral Interrupt Priority Registers

Asynchronous Operation

- The USART uses two I/O pins to transmit and receive serial data. Because there is no separate clock signal for asynchronous operation, one pin (TX) is used for transmission and the other pin (RX) is used for reception.
- Both transmission and reception can occur at the same time and this is known as 'full duplex' operation.
- Transmission and reception can be independently enabled, but when the serial port is enabled the USART will control both pins,
  - one cannot be used for general purpose I/O when the other is being used for transmission or reception.

Asynchronous Operation

- Figure 9 represents the signal on the TX or RX pins of the uC
  - The START bit is a ‘zero’ and the STOP bit is a ‘one.’
  - The data is sent Least Significant bit first, so the bit pattern looks backwards in comparison to the way it appears when written as a binary number.
  - The data is not inverted, even though RS-232 uses negative voltages to represent a logic one. Generally, when using the USART for RS-232 communications, the signals must be inverted and level-shifted through a transceiver chip of some sort.

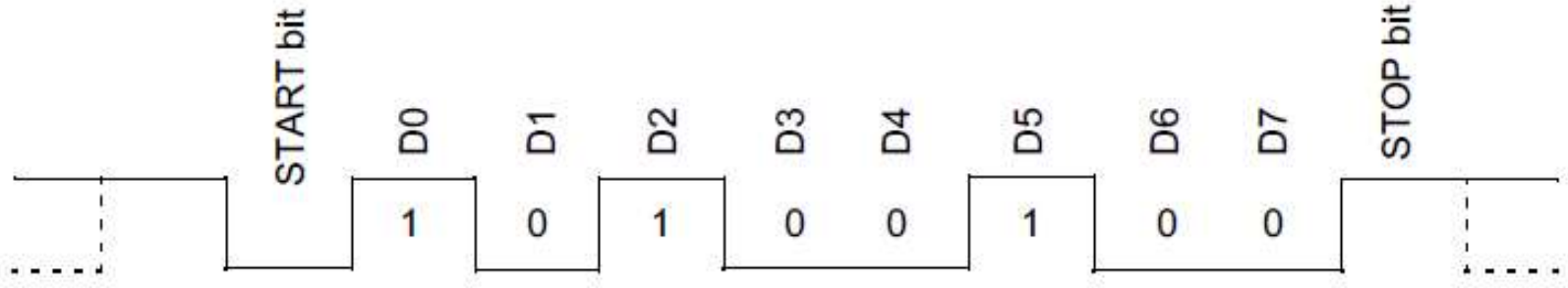


Figure 9: Asynchronous Mode Signal

Asynchronous Operation

- There are some features and uses of the USART that affect the signal.
  - The Nine-bit mode is useful when parity or an extra STOP bit is needed. It can also be used for the Addressable mode described later in this Application Note.
  - To implement parity, the ninth bit is set to make the total number of data bits either even or odd, depending on whether even or odd parity is being used.
- If two STOP bits are needed, the ninth data bit is set to one so that the signal stays high for two-bit periods after the first eight data bits.

Transmission

- The USART can be configured to transmit eight or nine data bits by the TX9 bit in the TXSTA register.
  - If nine bits are to be transmitted, the ninth data bit must be placed in the TX9D bit of the TXSTA register before writing the other eight bits to the TXREG register.
- Once data has been written to TXREG, the eight or nine bits are moved into the Transmit Shift Register.
- From there they are clocked out onto the TX pin preceded by a START bit and followed by a STOP bit.

Asynchronous Transmission

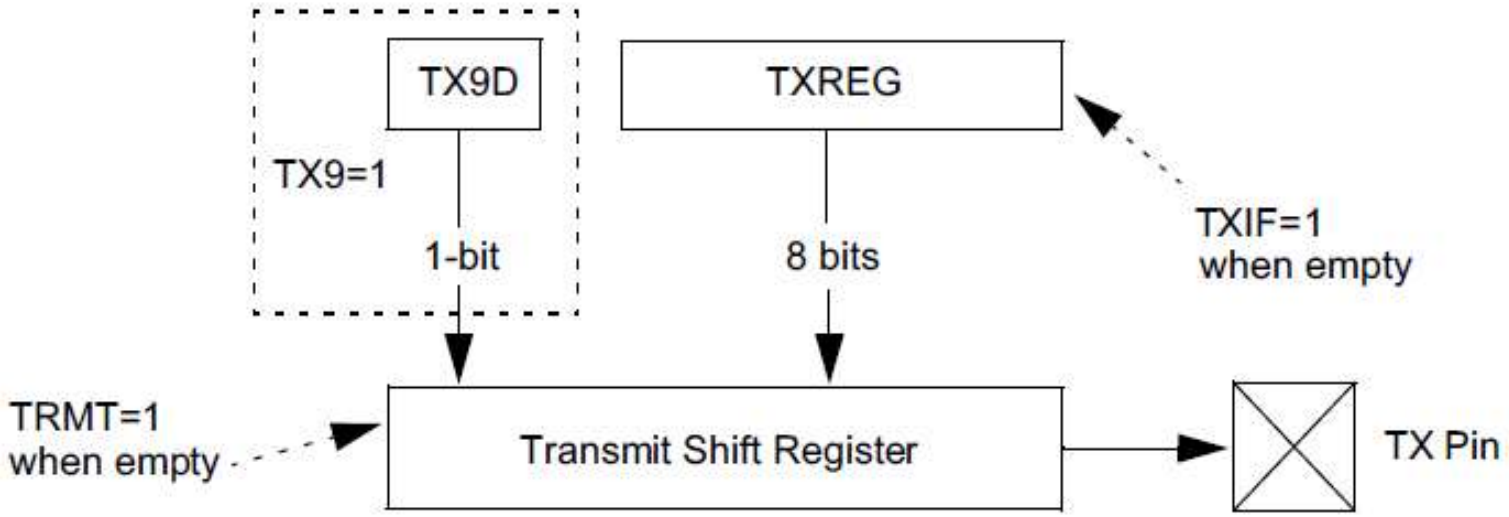


Figure 10: Asynchronous Transmission

Reception

- The USART can be configured to receive eight or nine bits by the RX9 bit in the RCSTA register.
- After the detection of a START bit, eight or nine bits of serial data are shifted from the RX pin into the Receive Shift Register, one bit at a time.
- After the last bit has been shifted in, the STOP bit is checked and the data is moved into the FIFO (First In First Out) buffer. RCREG is the output of a two element FIFO buffer.
- If another byte is received before the first byte has been read from RCREG, it will be kept in the FIFO 'behind' RCREG until the first byte has been read.
  - If nine-bit reception is enabled, the ninth bit is passed into the RX9D bit in the RCSTA register in the same way as the other eight bits of data are passed into the RCREG register.
- Two bytes can be held in the FIFO while a third is being received.
  - The user must ensure that data is read from RCREG before the third byte has been completely shifted in, otherwise the third byte will be discarded and an overrun error will be indicated.

Asynchronous Reception

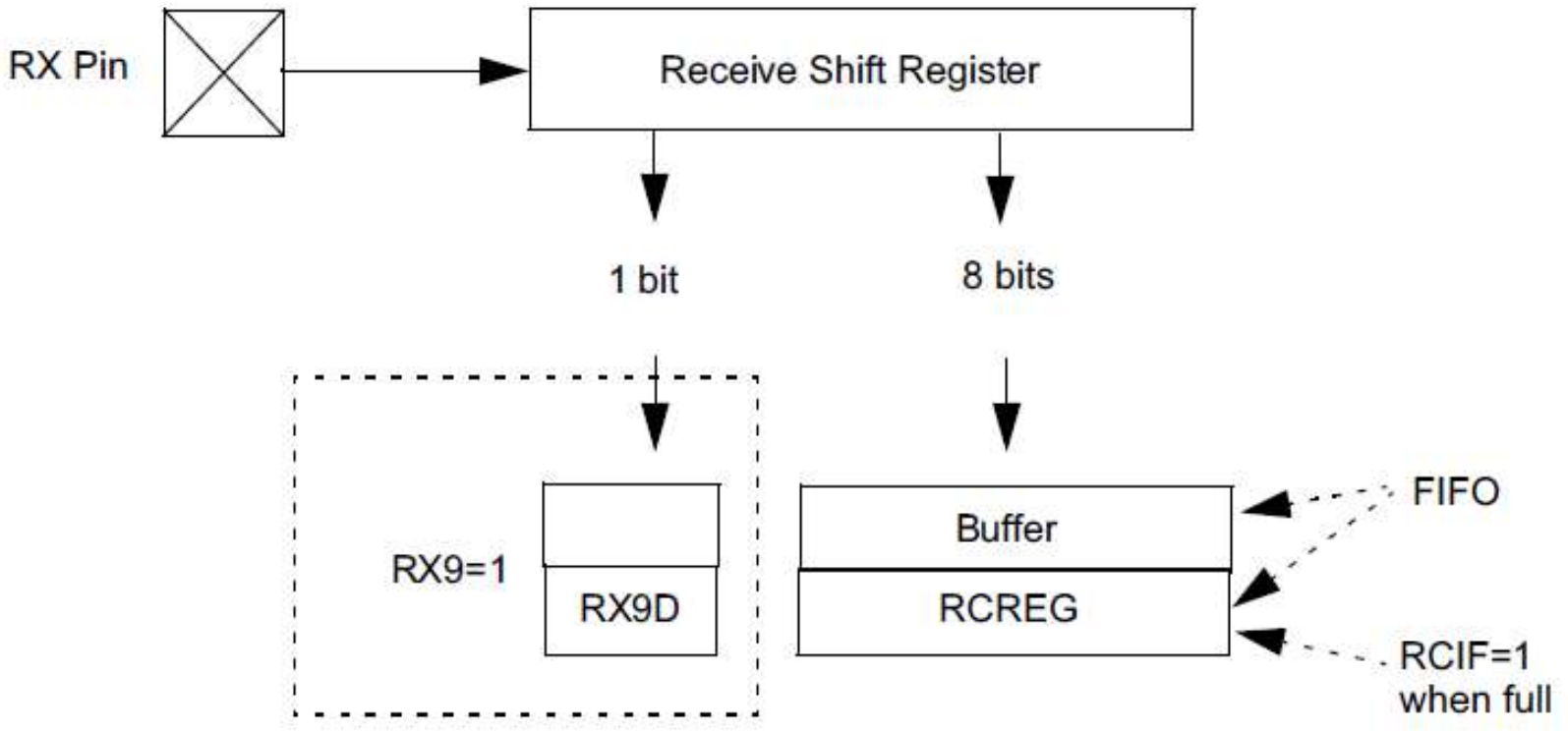


Figure 11: Asynchronous Reception

Addressable Mode

- Some devices have an addressable USART that can automatically filter certain transmissions.
- The received bytes are separated into two categories for addresses and data, indicated by the ninth data bit as shown in Figure 12.
- Only address bytes are processed by the USART, all other data is ignored.
  - This feature is usually used when there are multiple devices on a bus and transmissions are addressed to a particular device.
- The receiving devices ignore all data bytes with the ninth bit low and only receive address bytes with the ninth bit set.
- When the address byte is received and matches its own address, the receiving device can change into the normal Reception mode to receive the data that follows the address byte.
  - Nine-bit transmission and reception is always used with the addressable USART feature.

Addressable Mode

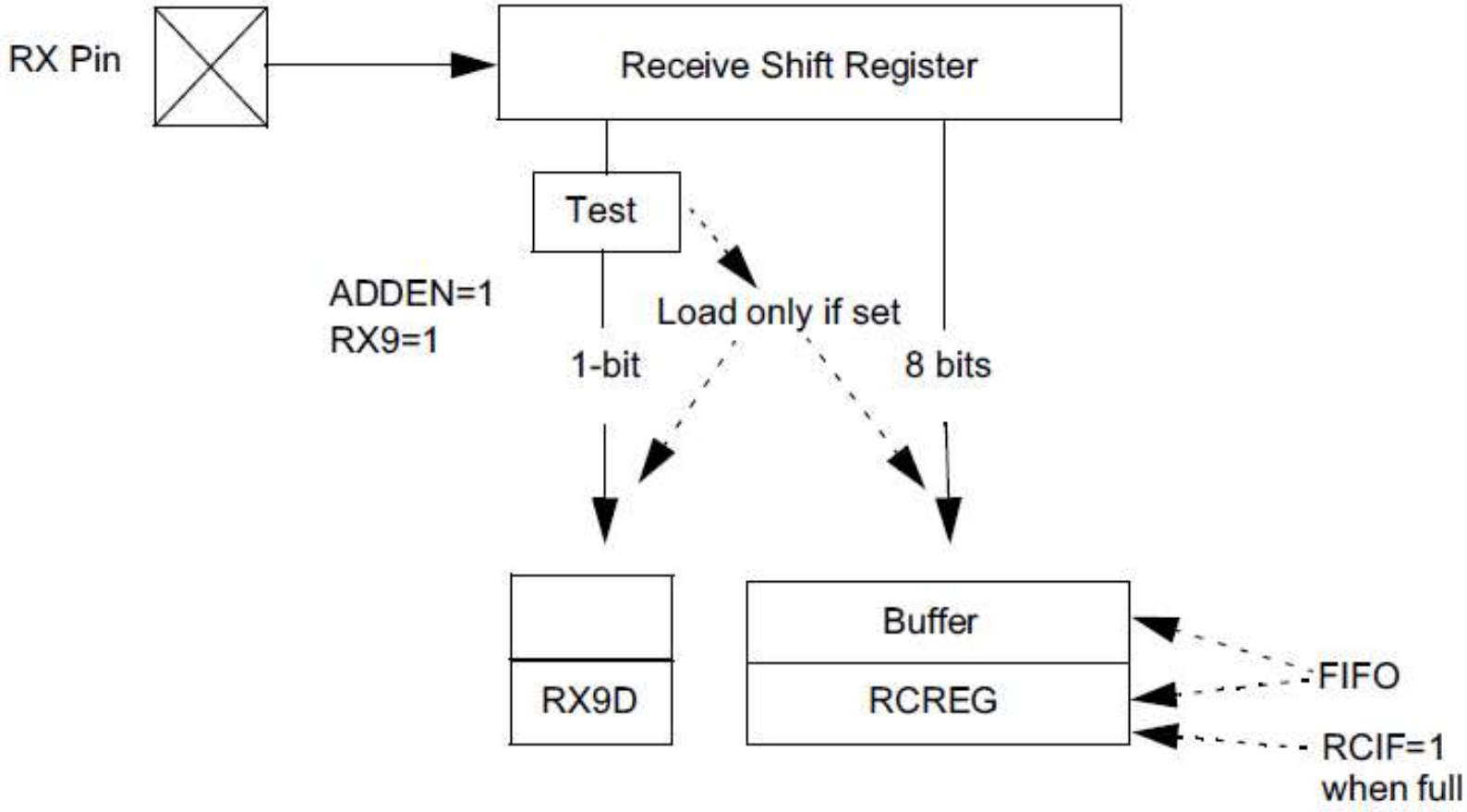


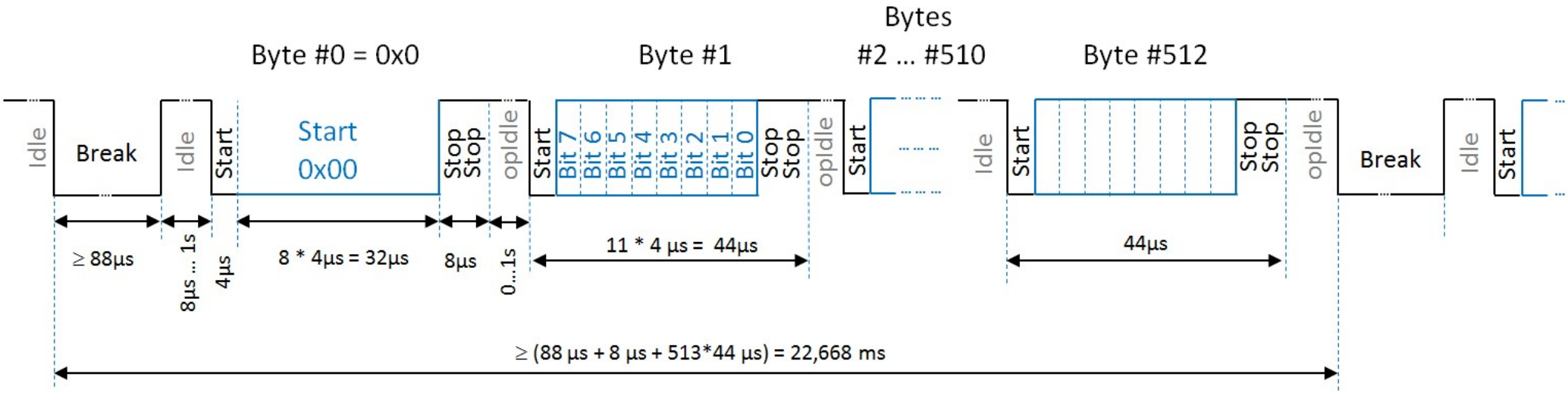
Figure 12: Addressable Mode

DMX 512

- Brilliant ideas are basically simple.
- DMX512 Specifications
  - DMX *Digital MultipleXed*,
  - Maximum 512 devices allowd on the network
  - 1986 by USITT (*United States of Institutes for Theatre Technology*)
  - Uses RS-485 at the physical layer
  - Standardized cable and connectors
  - “daisy-chain” connections, serial linked devices

DMX512

- Specs (continued)
  - E1 Master (Tx) and maxim 512 Receivers (Rx)
    - 512 devices is beyond what RS485 was designed for
    - One physical device may have more than 1 address



The DMX512 Waveforms

Chauvet Intimidator Spot LED 350 DMX Moving Light

Available versions include 8 and 14 addresses

Extensions

- RDM (*Remote Device Management*)
  - adopted in 2006
  - Allows bidirectional communication.
- ACN (*Architecture for Control Networks*)
  - Another extension
  - Allows DMX512 data transfe via IP or other ACN compatible networks

RS-232/485 converter

- RS-232 is a standard port on many systems, including PCs and many instruments.
- A common requirement is to convert from RS-232 to RS-485, as this allows for long transmission lengths.
  - If a computer connects to external equipment, it is important to isolate the grounds.
  - This is typically achieved with an opto-isolator which converts between the RS-232 interface and the RS-485 interface, as shown in Figure 21.
- Data is transmitted from the RS-232 port to the RS-485 line only if the RS-485 driver is in active mode.
  - This active mode is controlled by the RTS signal from the RS-232 port (or by detecting transitions on the transmit data line, TD).

RS-232/485 converter

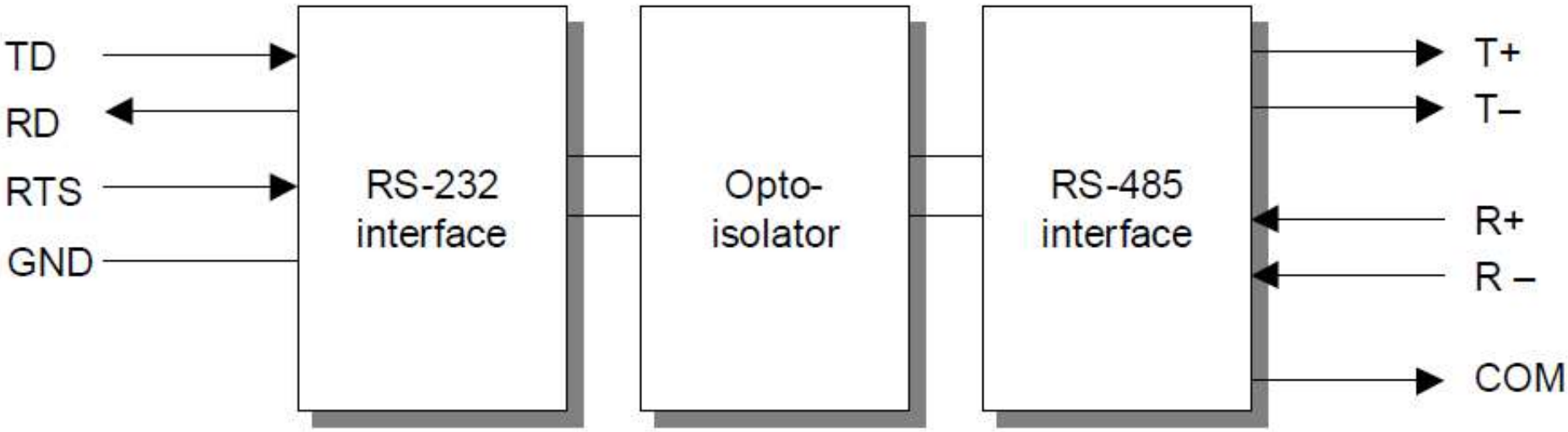


Figure 21: Isolation between RS-232 and RS-485

ADAM-4520

Isolated RS-232 to RS-422/485 Converter

- Main Features
  - Automatic RS-485 data flow control
  - 3000 VDC isolation protection
  - Surge protection RS-485 data line
  - Transmission speed up to 115.2 Kbps
  - Networking up to 1200 meters (4000 feet)
  - Reserved space for termination resistors
  - Power and data flow indicator for troubleshooting
  - Power requirement: +10 to +30 VDC
  - Mounts easily on a DIN-rail, panel or piggyback

ADAM-4561
USB to RS485 Converter

- Main Features
  - Widows 98/ME/2000, Linux drivers supported
  - Full compliance with USB V1.1 specifications.
  - RS-232/422/485 port supported
  - Transmission speed up to 115.2 kbps
  - Isolation protection 3000 VDC provided
  - Automatic RS-485 data flow control
  - No external power supply necessary; the hub derives its power from the USB port.
  - Plug & Play installation
  - No additional IRQs or I/O ports required
  - Hot attach & detach function supported