

SYNCHRONOUS SERIAL COMMUNICATION

As its name implies, synchronous communication takes place between a transmitter and a receiver operating on synchronized clocks. In a synchronous system, the communication partners have a short conversation before data exchange begins. In this conversation, they align their clocks and agree upon the parameters of the data transfer, including the time interval between bits of data. Any data that falls outside these parameters will be assumed to be either in error or a placeholder used to maintain synchronization.

Once each side knows what to expect of the other, and knows how to indicate to the other whether what was expected was received, then communication of any length can commence.

CLOCK SYNCHRONIZATION

In order to initiate a successful synchronous communication link, several distinct pieces of hardware must be configured around a common clock. This configuration must take two data lines into account, the transmission line (the line it uses to send data) and the reception line (the line it uses to receive data). It is essential not only that all devices in the system be synchronized with each other, but also that each individual device have its transmission and reception lines synchronized as well.

There are three clocking methods by which to achieve synchronization: internal, external, and recovered clocking. All three methods derive the clock signal for the reception line from the incoming data. The clock signal for the transmission line will always be generated by the device's internal oscillator, but the phase reference used by the internal oscillator differs for each of the clocking methods. When internal clocking is used, the transmit clock is phase locked to the device's own internal oscillator. For external clocking, the transmit clock is phase locked to the phase of the oscillator belonging to another device in the network. For recovered clocking, the transmit clock phase is locked to the clock derived from the incoming data.

Byte Oriented Synchronous Protocols

Synchronous communication can be implemented for full and half-duplex networks using bit- or byte-oriented protocols. Half-duplex networks, whether point-to-point or multipoint, can only support communication in one direction at a time. The most commonly used protocol for such networks is IBM's Binary Synchronous Communication Procedures (BiSync). BiSync is a byte oriented protocol, which means that it approaches transmitted data as "blocks" that must each be decoded and tracked to determine what they are, and what they are telling the receiver to do.

In a BiSync system one computer is designated as a control station. It is responsible for initiating all data transfers, and thereby controlling the direction of flow on the communication line. Byte-oriented communication begins with establishing synchronization, it then establishes communication parameters that define instructions for processing given bit sequences. Finally, the actual data will be transmitted, and then followed by several frames that validate the transmission. BiSync transmission is also governed by a strict set of rules for data transmission. These rules require frequent handshaking and validation--speaking in sentences rather than

paragraphs between pauses. As a result of the extensive handshaking, and because communication can take place in only one direction, BiSync communication is best suited for low-speed applications.

Byte Oriented Synchronous Protocols

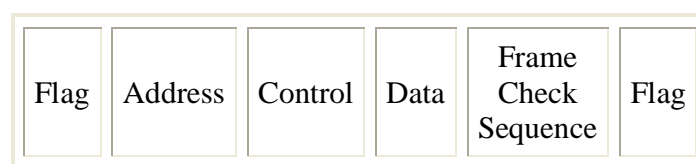
In bit-oriented protocols, data is accepted as a long string of bits whose order does not impart specific instructions to the receiver. Data is flagged by a set bit pattern at either end, and is validated by a single frame check sequence at the end of the message that either accepts it or demands retransmission. Any bits received outside of valid flag sequences are ignored as placeholders.

Clearly, bit-oriented communication requires considerably less overhead than byte-oriented communication, because it is not constantly attempting to match bit sequences to numerous predetermined arrangements. The only sequence the bit-oriented protocol is concerned with identifying is the flag sequence. Bit protocols have other advantages over byte protocols as well. In a byte-oriented system, because of the constant handshaking, communication can take place in only one direction at a time. In bit-oriented systems, both ends can talk to each other at once, enabling effective use of full-duplex networks. Further, a single master device using bit protocols can communicate with multiple slave devices by using an address field following the start of message flag. This address field is tailored to each individual slave, and slaves only process data that is specifically addressed to them. Likewise, when the master receives from the slave, it knows from precisely where the transmission originated.

The two main bit-oriented protocols used today are Synchronous Data Link Control (SDLC) and High-Level Data Link Control (HDLC). SDLC, which was developed by IBM in the 1970s, is based around a network of primary and secondary network nodes. The primary controls the network and continually polls the secondaries to determine whether they have data to transmit. Four configurations are available for SDLC networks:

- **Point to Point:** one primary connected to one secondary
- **Multipoint:** one primary connected to multiple secondaries
- **Loop:** one primary connects to the first and last secondary, and secondaries in the middle pass messages through each other to the primary.
- **Hub go-ahead:** Uses one inbound and one outbound channel. The primary sends on the outbound channel and the secondaries send on the inbound channel. Secondaries pass messages through each other back to the primary.

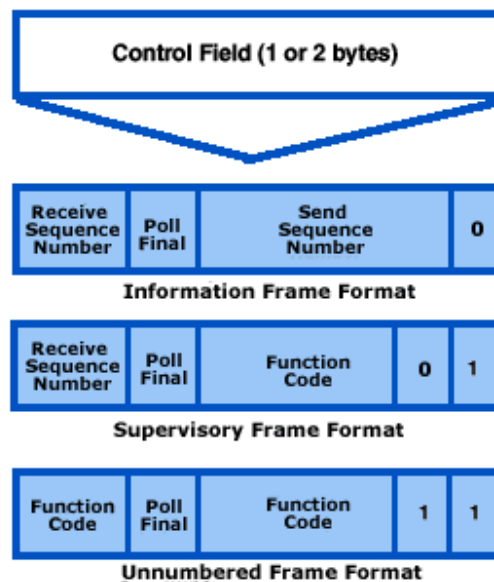
Key to SDLC communication is the control characters it uses to maintain data integrity. The figure below shows the six fields that comprise a single SDLC data frame.



(SDLC Data Frame)

The **Flag** field starts and ends error checking. The **Address** field is used to indicate the intended data destination, and can be a single address, a group of addresses, or a broadcast to the entire network. The **Data** field is the information being transmitted, and the **Frame Check Sequence (FCS)** is generally a Cyclic Redundancy Check (CRC) calculation. A calculation on the transmitted data is done by the transmitter and the result is sent in the FCS. This calculation is then performed by the receiver after data transmission is complete. If the results don't match, an error is assumed.

The **Control** field uses three different formats depending on the type of SDLC frame. The diagram below breaks out the different data bits in the control field. Explanations for the three control formats follow.



(Control Field Formats for SDLC Frames)

The **Information Frame** is used when actual data is being transmitted. It is also used to provide sequencing, flow and error control functions. The sequence number bits are used to indicate the number of the frame that will be sent/received next, and are used by both the primary and secondary nodes. The Poll Final bit is used by the primary to tell the secondary whether or not an immediate response is required. The secondary uses the Poll Final bit to indicate whether the current frame is the last in its response, or whether more frames are coming.

The **Supervisory Frame** is used to control the communication network. It can request or suspend data transfer, report status, and acknowledge receipt of data. Note that since Supervisory frames are used exclusively for control, they do not have data fields.

The **Unnumbered Frame** is unsequenced, can contain one or two bytes, and is used to provide miscellaneous control commands. For instance, it might be used by a primary node to activate the secondary nodes in the network.

Another bit-oriented synchronous communication protocol, High Level Data Link Control (HDLC) which is based on SDLC, also uses the frame format described above. However HDLC, which was approved by the International Standards Organization (ISO) in 1979, differs from SDLC in several ways. With HDLC, 32-bit checksums can be used, thereby providing an advantage over SDLC in the sophistication and accuracy of error checking, and thus data integrity. Unlike SDLC, HDLC protocols cannot operate using loop or hub go-ahead configurations.

The largest difference between the two, is that SDLC uses only a single transfer mode, while HDLC provides three choices. Both use Normal Response Mode (NRM) in which a secondary node is precluded from communicating with a primary node until the primary gives permission. The two additional HDLC modes are Asynchronous Response Mode (ARM) and Asynchronous Balanced Mode (ABM). In ARM mode, any secondary can initiate communication without receiving permission from the primary. ABM mode requires that all devices be configured as combination nodes that, depending on the situation, can assume the role of primary or secondary in the network. In such a system, any device can initiate communication at any time without permission.