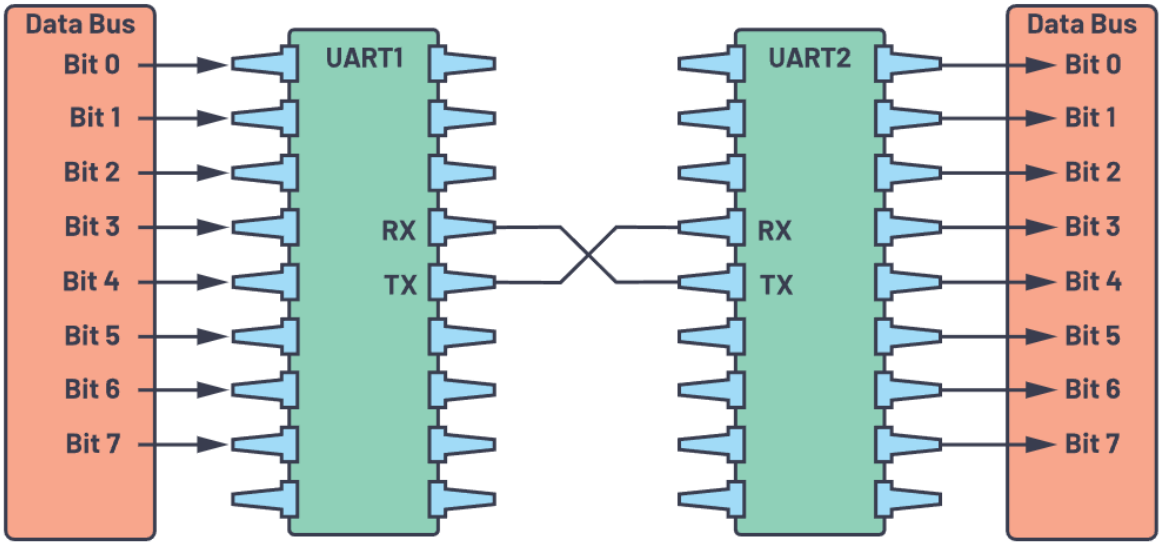
UART

Introduction:

- UART, or universal asynchronous receiver-transmitter, is one of the most used device-to-device communication protocol
- UART can work with many different types of serial protocols that involve transmitting and receiving serial data.
- In serial communication, data is transferred bit by bit using a single line or wire. In two-way communication, we use two wires for successful serial data transfer.
- Embedded systems, microcontrollers, and computers mostly use UART as a form of device-to-device hardware communication protocol. Among the available communication protocols, UART uses only two wires for its transmitting and receiving ends.
- UART is a hardware communication protocol that uses asynchronous serial communication with configurable speed. Asynchronous means there is no clock signal to synchronize the output bits from the transmitting device going to the receiving end.



- The transmitting UART is connected to a controlling data bus that sends data in a parallel form. From this, the data will now be transmitted on the transmission line (wire) serially, bit by bit, to the receiving UART. This, in turn, will convert the serial data into parallel for the receiving device.
- The UART lines serve as the communication medium to transmit and receive one data to another.
- The baud rate is the rate at which information is transferred to a communication channel. In the serial port context, the set baud rate will serve as the maximum number of bits per second to be transferred.

Table 1. UART Summary

Wires	2
Speed	9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600, 1000000, 1500000
Methods of Transmission	Asynchronous
Maximum Number of Masters	1
Maximum Number of Slaves	1

-

- The UART interface does not use a clock signal to synchronize the transmitter and receiver devices; it transmits data asynchronously. Instead of a clock signal, the transmitter generates a bitstream based on its clock signal while the receiver is using its internal clock signal to sample the incoming data.

Data Transmission

- In UART, the mode of transmission is in the form of a packet. The piece that connects the transmitter and receiver includes the creation of serial packets and controls those physical hardware lines. A packet consists of a start bit, data frame, a parity bit, and stop bits.

-

1. **Start Bit:** The UART data transmission line is normally held at a high voltage level when it's not transmitting data.
   To start the transfer of data, the transmitting UART pulls the transmission line from high to low for one clock cycle.
   When the receiving UART detects the high to low voltage transition, it begins reading the bits in the data frame at the frequency of the baud rate.
2. **Data Frame:** The data frame contains the actual data being transferred.
   It can be five bits up to eight bits long if a parity bit is used.
   If no parity bit is used, the data frame can be nine bits long.
   generally, data is sent with LSB first.
3. **Parity:** The parity bit is a way for the receiving UART to tell if any data has changed during transmission.
   bits can be changed by electromagnetic radiation, mismatched baud rates, or long-distance data transfers.
   If the parity bit is a 0 (even parity), the 1 or logic-high bit in the data frame should total to an even number. If the parity bit is a 1 (odd parity), the 1 bit or logic highs in the data frame should total to an odd number.
   When the parity bit matches the data, the UART knows that the transmission was

free of errors. But if the parity bit is a 0, and the total is odd, or the parity bit is a 1, and the total is even, the UART knows that bits in the data frame have changed.

4. **Stop Bits:** To signal the end of the data packet, the sending UART drives the data transmission line from a low voltage to a high voltage for one (1) to two (2) bit(s) duration.

UART Transmission

- The transmitting UART receives data in parallel from the data bus.
-

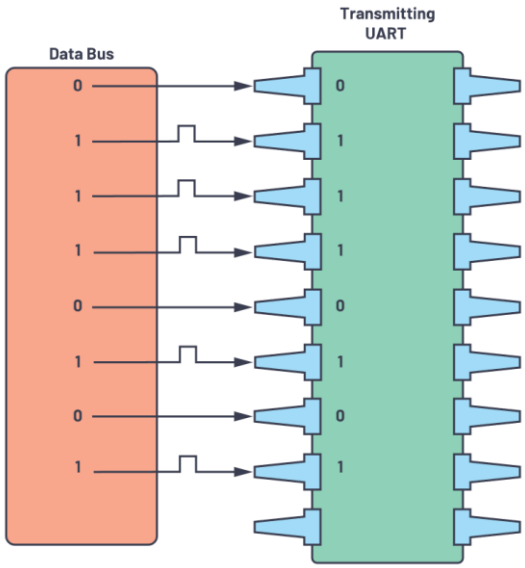


Figure 8. Data bus to the transmitting UART.

-
- The transmitting UART adds the start bit, parity bit, and the stop bit(s) to the data frame.

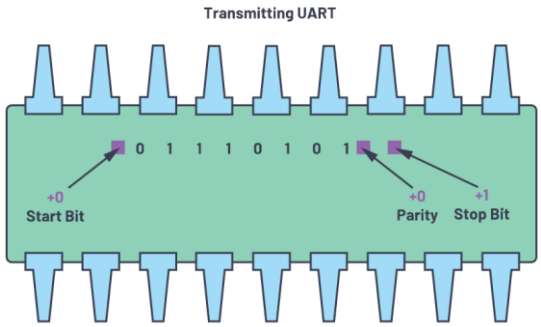
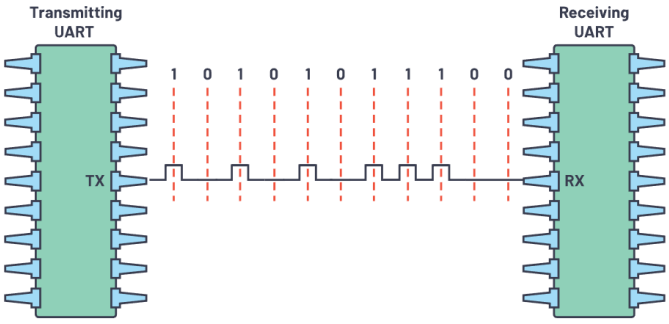
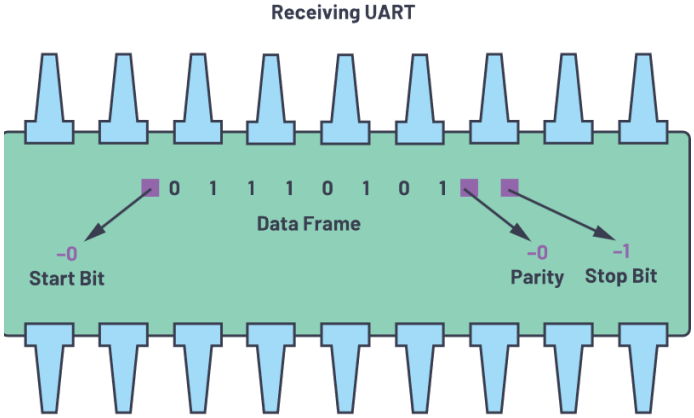


Figure 9. UART data frame at the Tx side.

-
- The entire packet is sent serially starting from start bit to stop bit from the transmitting UART to the receiving UART.
-
-



- *Figure 10. UART transmission.*
- The receiving UART discards the start bit, parity bit, and stop bit from the data frame.



- *Figure 11. The UART data frame at the Rx side.*
- The receiving UART converts the serial data back into parallel and transfers it to the data bus on the receiving end.