

UART PROTOCOL

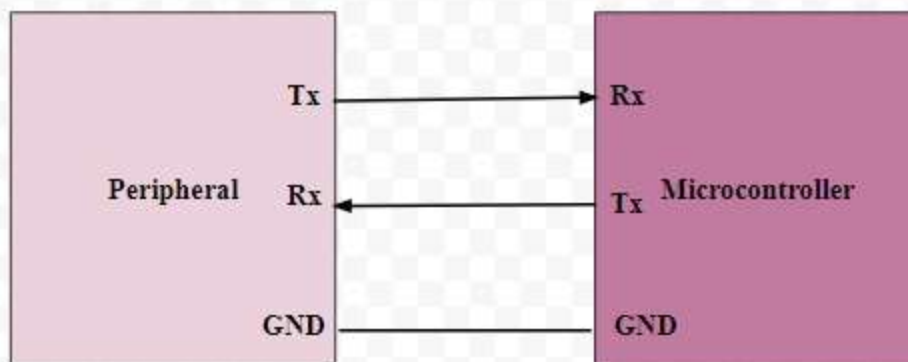
Ezrah Ogori
EMBEDEDXPRESS

UART Communication : Block Diagram and Its Applications

If we remember the old computer parts like printer, mouse, the keyboard is associated with the help of connectors. The communication process between the computer and these parts could be done using the UART. Universal Serial Bus (USB) has changed all kinds of communication principles on computers. But, UART is still used in the above-declared applications. Approximately all all types of microcontrollers architectures have built-in UART hardware due to serial communication and use only two cables for communication. This article discusses what UART, **How UART Works**, the difference between serial and parallel communication, **UART block diagram**, UART communication, UART interfacing, Applications, Advantages, and Disadvantages.

What is UART?

The **UART full form** is “Universal Asynchronous Receiver/Transmitter”, and it is an inbuilt IC within a microcontroller but not like a communication protocol (I2C & SPI). The main function of UART is to serial data communication. In UART, the communication between two devices can be done in two ways namely serial data communication and parallel data communication.



UART Communication

UART

Serial and Parallel Communication

In serial data communication, the data can be transferred through a single cable or line in a bit- by- bit form and it requires just two cables. Serial data communication is not expensive when we compared with parallel communication. It requires very less circuitry as well as wires. Thus, this communication is very useful in compound circuits compared with parallel communication.

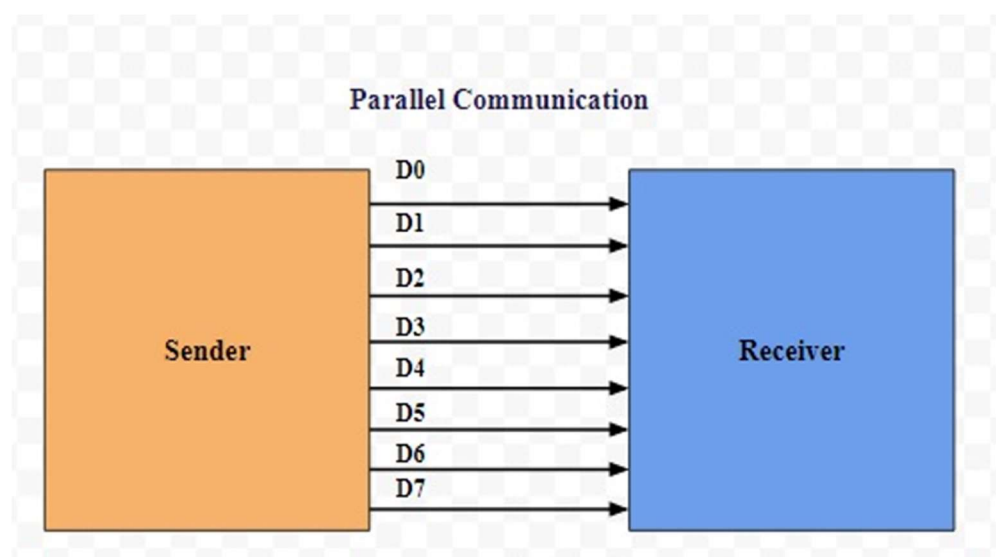
In parallel data communication, the data can be transferred through multiple cables at once. Parallel data communication is expensive as well as very fast, as its requires

additional hardware and cables. The best examples for this communication are old printers, PCI, RAM, etc.

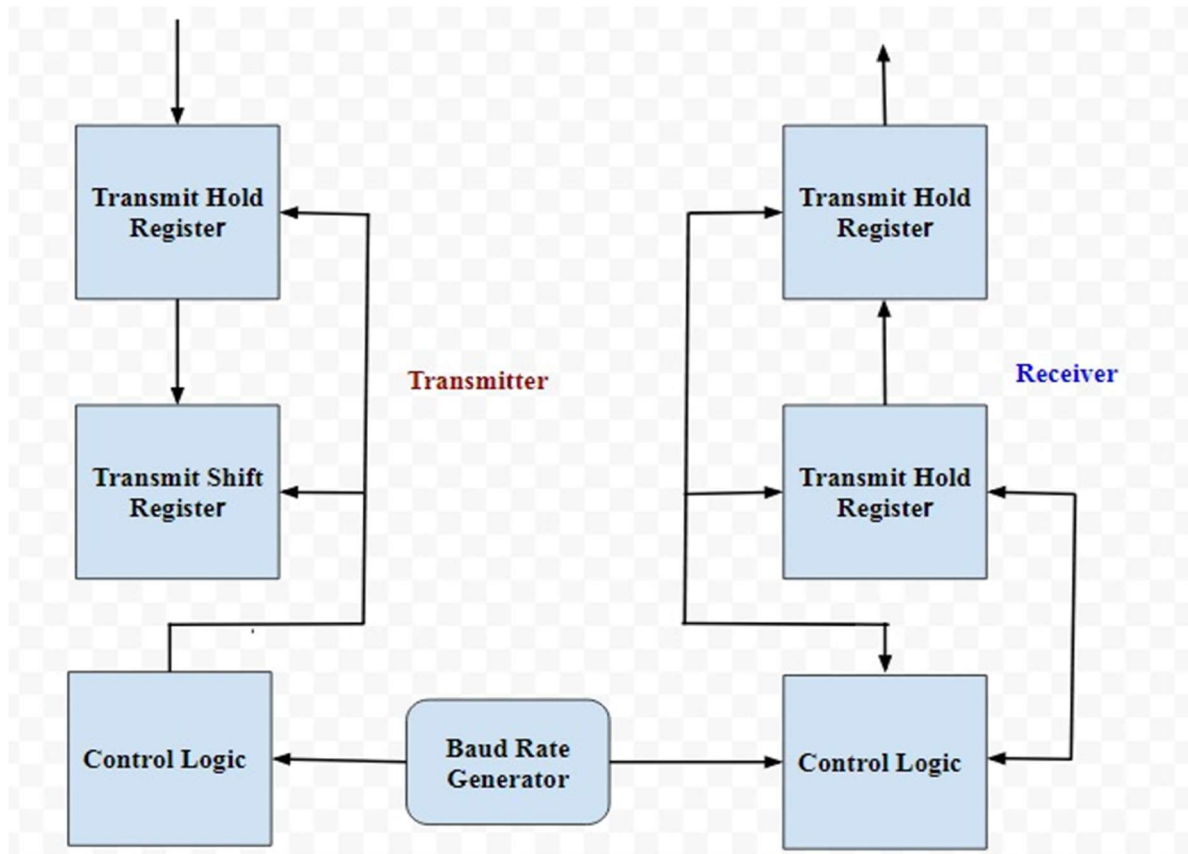
UART Block Diagram

The UART block diagram consists of two components namely the transmitter & receiver that is shown below. The transmitter section includes three blocks namely transmit hold register, shift register and also control logic. Likewise, the receiver section includes a receive hold register, shift register, and control logic. These two sections are commonly provided by a baud-rate-generator. This generator is used for generating the speed when the transmitter section & receiver section has to transmit or receive the data.

The hold register in the transmitter comprises the data-byte to be transmitted. The shift registers in transmitter and receiver move the bits to the right or left till a byte of data is transmitted or received. A read (or) write control logic is used for telling when to read or write.



The baud-rate-generator among the transmitter and the receiver generates the speed that ranges from 110 bps to 230400 bps. Typically, the baud rates of microcontrollers are 9600 to 115200.

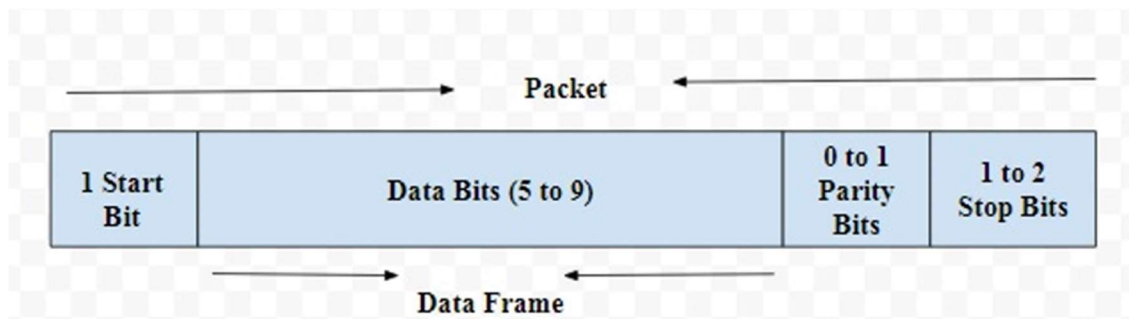


UART Block Diagram

UART Communication

In this communication, there are two types of UARTs available namely transmitting UART and receiving UART, and the communication between these two can be done directly by each other. For this, simply two cables are required to communicate between two UARTs. The flow of data will be from both the transmitting (Tx) & receiving (Rx) pins of the UARTs. In UART, the data transmission from Tx UART to Rx UART can be done asynchronously (there is no CLK signal for synchronizing the o/p bits).

The data transmission of a UART can be done by using a data bus in the form of parallel by other devices like a microcontroller, memory, CPU, etc. After receiving the parallel data from the bus, it forms a data packet by adding three bits like start, stop and parity. It reads the data packet bit by bit and converts the received data into the parallel form to eliminate the three bits of the data packet. In conclusion, the data packet received by the UART transfers in parallel toward the data bus at the receiving end.



UART Communication

Start Bit

Start-bit is also known as a synchronization bit that is placed before the actual data. Generally, an inactive data transmission line is controlled at a high-voltage level. In order to begin the data transmission, the UART transmission drags the data-line from a high voltage level (1) to a low voltage level (0). The obtaining UART notices this transform from the high level to low level over the data line as well as starts understanding the real data. Generally, there is just a single start bit.

Stop Bit

The Stop Bit is placed at the ending of the data packet. Usually, this bit is 2-bits lengthy but frequently on bit only utilized. In order to stop the broadcast, the UART keeps the data-line on high voltage.

Parity Bit

Parity bit lets the receiver to ensure whether the collected data is right or not. It is a low-level fault checking system & parity bit is available in two ranges such as Even Parity as well as Odd Parity. Actually, this bit is not widely used so it is not compulsory.

Data Bits or Data Frame

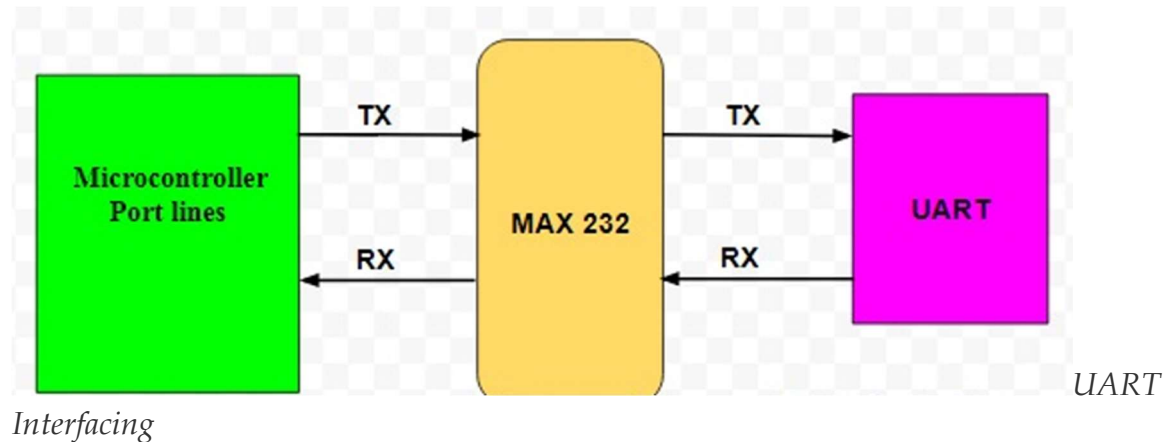
The data bits include the real data being conveyed from the sender to receiver. The data frame length could be between 5 & 8. If the parity bit is not used when the data frame length could be 9-bit long. Generally, the LSB of the data to be transmitted first then it is very useful for transmitting.

UART Interfacing

The following figure shows UART interfacing with a microcontroller. The UART communication can be done using three signals like TXD, RXD, and GND.

By using this, we can exhibit a text in personal computer from 8051 microcontroller board as well as the UART module. In 8051 board, there are two serial interfaces such as UART0

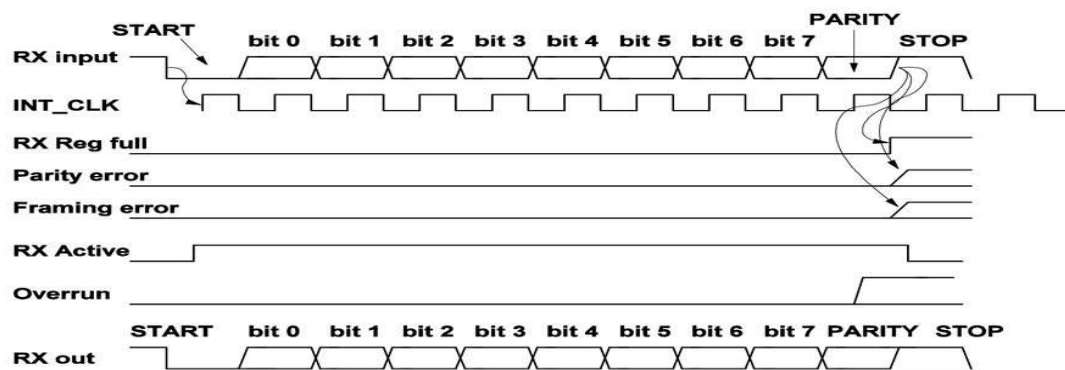
and UART1. Here, UART0 interfacing is used. The Tx pin transmits the information to PC & Rx pin receives the information from PC. Baud rate can be used to denote the speeds of both the microcontroller and PC. The data transmission and reception can be done properly when the baud rates of both microcontroller & PC are similar.



TIMING DIAGRAM FOR UART RECEIVE PROTOCOL

UART Implementation

- **UART Receive Timing diagram:**



©Alex Doboli 2006

Applications of UART

UART is normally used in microcontrollers for exact requirements, and these are also available in various communication devices like wireless communication, GPS units, Bluetooth, and many other applications.

The communication standards like RS422 & TIA are used in UART except for RS232. Usually, a UART is a separate IC used in UART serial communications.

Advantages and Disadvantages of UART

The pros and cons of UART include the following

- It requires only two wires for data communication
- CLK signal is not required.
- It includes a parity bit for allowing to check the errors
- The data packet arrangement can be modified because both surfaces are arranged for it
- The data frame size is a max of 9 bits
- It doesn't hold several slave (or) master systems
- The every UART baud rate should be in 10% of each other

Thus, this is all about an overview of **Universal Asynchronous Receiver Transmitter** (UART) is one of the fundamental interfaces which gives a simple, cost-effective & consistent communication among microcontroller as well as PC.