Universal Asynchronous Receiver Transmitter (UART) Protocol

Last Updated : 29 Dec, 2022

UART means Universal Asynchronous Receiver Transmitter Protocol. UART is used for serial communication from the name itself we can understand the functions of UART, where **U** stands for Universal which means this protocol can be applied to any transmitter and receiver, and **A** is for Asynchronous which means one cannot use clock signal for communication of data and **R** and **T** refers to Receiver and Transmitter hence UART refers to a protocol in which serial data communication will happen without clock signal.

UART is established for serial communication. In this article, we will discuss how parallel communication is established with respect to serial communication using UART as well as how to configure UART and what is the data format in UART. Later on, we will discuss the Pros and Cons of the UART.

UART Basics

UART is a Universal Asynchronous Receiver Transmitter protocol that is used for serial communication. Two wires are established here in which only one wire is used for transmission whereas the second wire is used for reception. Data format and transmission speeds can be configured here. So, before starting with the communication define the data format and transmission speed. Data format and transmission speed for communication will be defined here and we do not have a clock over here that's why it is referred to as asynchronous communication with UART protocol. Here we will see how this protocol is designed physically.

UART
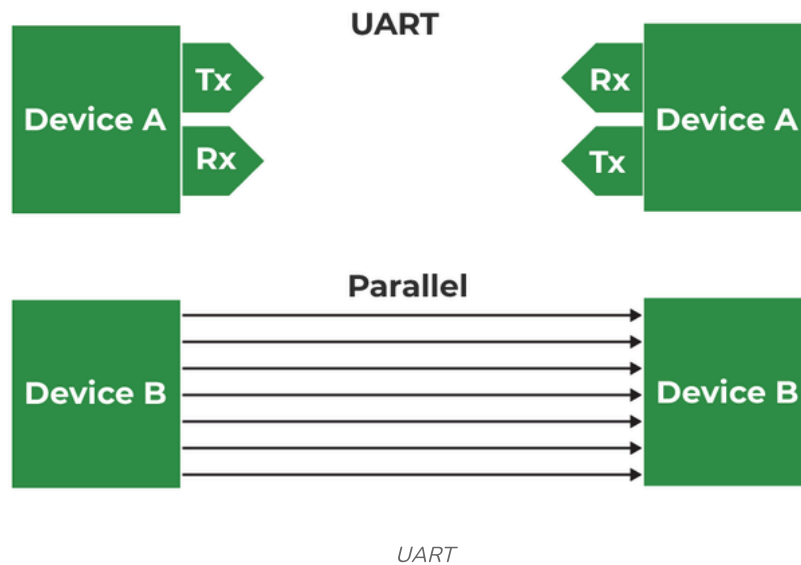
Here, DEVICE A that is having transmitter PIN and a receiver pin; DEVICE B is having a receiver and transmission pin. The Transmitter of DEVICE A is one that should be connected with the receiver pin of DEVICE B and the transmitter pin of DEVICE B should be connected with the receiver pin of DEVICE A we just need to connect two wires for communication.

If DEVICE A wants to send data, then it will be sending data on the transmitter's pin and here receiver of this DEVICE B will receive it over and if DEVICE A wants to receive the data, then that is possible on the RX line that will be forwarded by TX of DEVICE B. On comparing this serial communication of UART with parallel then it can be observed that in parallel multiple buses are required. Based on the number of lines bus complexity of UART is better but parallel communication is good in terms of speed.

So, when speed is required at that time we should select parallel communication and for a low-speed application, UART must be used and the bus complexity will be less.

The configuration of UART is done before transmission, both of these devices are connected with protocol and should know the speed of data transmission. First, define the speed of both devices. Now, configure the speed of DEVICE A and B for data transmission which is referred to as Baud Rate so here Baud Rate will be the same for DEVICE A and B otherwise both of these devices cannot understand at what speed and at what rate data is coming. After that, configure the data length so here DEVICE A and DEVICE B both are configured at fixed data length if DEVICE A is transmitting data, then it is configured with fixed data. Like if DEVICE A is configured with the eight-bit size of data then DEVICE B should also be configured at the same size of data which is eight bits. After this, check data transmission or receiving time, forward start bits, and stop bits.

Now we will see the data format and when the communication is according to UART protocol. We are using NRZ encoding for data communication.

UART Data Format

UART Data Format

Suppose DEVICE A is sending data to DEVICE B, and the transmitter of DEVICE A will send data to the receiver of Device B then it will be logic high. Now, send the start bit that will be logic 0 and once we have the start bit, DEVICE B will know that somebody is communicating. Now there is the same speed configuration with both devices. So, after the start bit, DEVICE A can forward data.

Consider 8 bits of data length, so we will be forwarding 8 bits and those 8 bits will be received by DEVICE B a parity bit can also be used which is optional, but this is quite effective. By using the parity bit, it can be identified whether the received data is correct or not. Suppose we are sending 1 1 1 0 0 0 1 0. Now, we have 4 ones; an even number of ones are there hence the parity is even and for that, logic 0 will be assigned. Suppose we are receiving data with some error, say zero is converted into one; Now incorrect data that is 1 1 1 1 0 0 1 0 for this incorrect data

parity will be 0 as there are 5 ones, here is a mismatch in the parity bit and hence it is confirmed that the received data has some error.

Pros of UART Protocol

- It is having less physical interfacing based on only two lines.
- Simple to configure data and data size. Speed is also configurable. In the majority of cases, this baud rate is 9600 for the UART protocol. Full duplex mode configuration is possible by using two wires so that is the major advantage of UART.
- Error detention is possible

Cons of UART Protocol

- UART is having serial communication, therefore, it has less speed.
- Start bit, stop bit, and the parity bit is other overhead.
- Since this is asynchronous communication so here there are many things that we need to do in configuration, for instance, we should configure both devices at the same speed because the clock signal is absent.

A code...   11

Similar Reads

Difference Between USART and UART

USART and UART are two basic serial communication interfaces that are popular in embedded systems. Though both are used to share data between two devices, they...

🕐 5 min read

Ethernet Transmitter Algorithm

Prerequisite - Ethernet and Ethernet Frame Format Ethernet Adapter or Network Interface Card (NIC) is hardware device/component, that enables you to send and...

🕐 6 min read

Rake Receiver

The dictionary meaning of rake is to gather or collect together something and actually is a garden tool to collect leaves. But in the terms of computer network, it is for the...

🕐 5 min read

Hot Standby Router Protocol (HSRP) and Virtual Router Redundancy Protocol...

Hot Standby Router Protocol (HSRP) is a CISCO proprietary protocol used to provide redundancy in a network. Only one router is the active router while others will be in...

🕐 3 min read

Cisco Discovery Protocol (CDP) and Link Layer Discovery Protocol (LLDP) in Dat...

Layer 2 or the Datalink layer provides physical addressing and access to media. It defines how data is to be formatted for transmission and how access to the network...

🕐 2 min read

Difference between Stop and Wait protocol and Sliding Window protocol

Both the Stop and Wait protocol and the Sliding Window protocol are the techniques to the solution of flow control handling. The main difference between the Stop-and-...

🕐 5 min read

Difference between Border Gateway Protocol (BGP) and Routing Information...

When dealing with network routing, it's crucial to understand the various protocols that help direct traffic across complex networks. Two significant protocols in this...

🕐 5 min read

Difference between File Transfer Protocol (FTP) and Secure File Transfer Proto...

Data transfer is the most common task in network and system communications, and they involve transfer of files using two means i.e the File Transfer Protocol (FTP) and...

🕐 5 min read

Difference between Spanning Tree Protocol (STP) and Rapid Spanning Tree...

Spanning Tree Protocol (STP) and Rapid Spanning Tree Protocol (RSTP), are designed to avoid network loops in Ethernet. It provide loop-free logical topology. They were...

🕐 5 min read

Difference between Serial Line Internet Protocol (SLIP) and Point-to-Point...

SLIP and PPP, which are the widely used serial connection techniques, help to connect the distant devices of the network. The fullform of SLIP is Session Layer...

🕐 6 min read

Article Tags :　　Computer Networks　　Computer Subject

Company	Languages
About Us	Python
Legal	Java
In Media	C++
Contact Us	PHP
Advertise with us	GoLang
GFG Corporate Solution	SQL
Placement Training Program	R Language

GeeksforGeeks Community

Android Tutorial

Tutorials Archive

DSA

Data Structures

Algorithms

DSA for Beginners

Basic DSA Problems

DSA Roadmap

Top 100 DSA Interview Problems

DSA Roadmap by Sandeep Jain

All Cheat Sheets

Data Science & ML

Data Science With Python

Data Science For Beginner

Machine Learning

ML Maths

Data Visualisation

Pandas

NumPy

NLP

Deep Learning

Web Technologies

HTML

CSS

JavaScript

TypeScript

ReactJS

NextJS

Bootstrap

Web Design

Python Tutorial

Python Programming Examples

Python Projects

Python Tkinter

Web Scraping

OpenCV Tutorial

Python Interview Question

Django

Computer Science

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Software Development

Software Testing

DevOps

Git

Linux

AWS

Docker

Kubernetes

Azure

GCP

DevOps Roadmap

System Design

High Level Design

Low Level Design

UML Diagrams

Interview Guide

Design Patterns

OOAD

System Design Bootcamp

Interview Questions

Inteview Preparation

Competitive Programming

Top DS or Algo for CP

Company-Wise Recruitment Process

Company-Wise Preparation

Aptitude Preparation

Puzzles

School Subjects

Mathematics

Physics

Chemistry

Biology

Social Science

GeeksforGeeks Videos

DSA

Python

Java

C++

Web Development

English Grammar

Data Science

Commerce

CS Subjects

World GK

English Grammar

Data Science

Commerce

CS Subjects

World GK