

# lamaPLC Communication: USB

*Universal Serial Bus (USB)* is an industry standard that allows data exchange and power delivery between many types of electronic devices. It specifies its architecture, in particular its physical interface, and communication protocols for data transfer and power delivery to and from hosts, such as personal computers, to and from peripheral devices, e.g., displays, keyboards, and mass storage devices, and to and from intermediate hubs, which multiply the number of a host's ports.



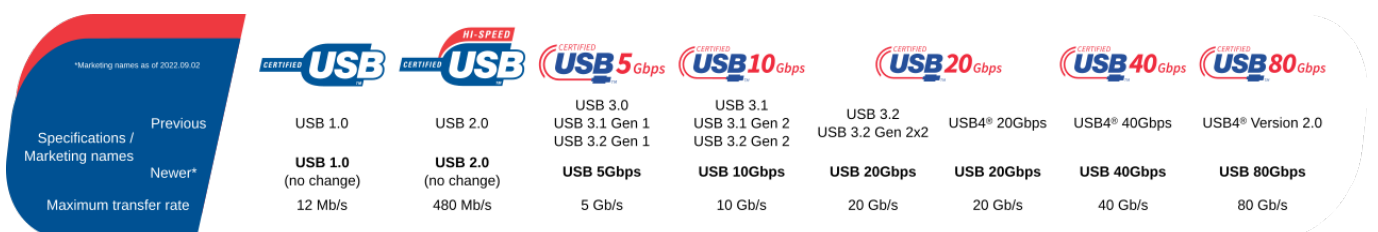
Introduced in 1996, USB was initially created to standardize the connection of peripherals to computers, replacing various interfaces such as serial, parallel, game, and ADB ports. Early versions of USB became common on many devices such as keyboards, mice, cameras, printers, scanners, flash drives, smartphones, game consoles, and power banks. Since then, USB has evolved into a standard that replaces nearly all common ports on computers, mobile devices, peripherals, power supplies, and many other small electronics.

The current standard uses the USB-C connector to replace many different types of power (up to 240 W), display (e.g., DisplayPort, HDMI), and other connectors, as well as all previous USB connectors.

As of 2024, USB consists of four generations of specifications: USB 1.x, USB 2.0, USB 3.x, and USB4.



## transfer rates of the USB interfaces



USB standard	USB speed
<b>USB 1.0</b>	12 Mbit/s (1,5 MByte/s)
<b>USB 2.0</b>	480 Mbit/s (60 MByte/s)
<b>USB 3.0</b>	5 Gbit/s (625 MByte/s)
<b>USB 3.1</b>	10 Gbit/s (1,25 GByte/s)
<b>USB 3.2</b>	20 Gbit/s (2,5 GByte/s)
<b>USB 4</b>	40 Gbit/s (5 GByte/s)

<b>USB 4 type 2</b>	80 Gbit/s (10 GByte/s)
---------------------	------------------------

## System design

USB device communication relies on **pipes** (*logical channels*). A pipe is a connection from the host controller to a logical entity within a device, called an endpoint. Since pipes correspond to endpoints, the terms are sometimes used interchangeably. Each USB device can have up to 32 endpoints (*16 in and 16 out*), although having so many is unusual. Endpoints are defined and numbered by the device during initialization (the period after physical connection called "*enumeration*"), making them relatively permanent, while pipes can be opened and closed. There are two types of pipe: stream and message.

A message pipe is bi-directional and is used for control transfers. Message pipes are typically used for short, simple commands to the device, and for status responses from the device, used, for example, by the bus control pipe number 0.

A stream pipe is a unidirectional pipe connected to a unidirectional endpoint that transfers data using isochronous, interrupt, or bulk transfer methods.

### Isochronous transfers

At a guaranteed data rate (for fixed-bandwidth streaming data) but with potential data loss (e.g., real-time audio or video)

### Interrupt transfers

Devices that require guaranteed quick responses (bounded latency), such as pointing devices, mice, and keyboards.

### Bulk transfers

Large sporadic transfers consuming all remaining bandwidth, but with no guarantees on bandwidth or latency (e.g., file transfers).

When a host initiates a data transfer, it sends a **TOKEN** packet containing an endpoint identified by a tuple of (*device\_address*, *endpoint\_number*). If the transfer goes from the host to the endpoint, the host sends an **OUT** packet (a specific type of **TOKEN packet**) with the target device address and endpoint number.

If data is transferred from the device to the host, the host sends an **IN** packet instead. If the destination endpoint is a unidirectional endpoint with a designated direction that does not match the **TOKEN** packet (for example, if the manufacturer's designated direction is **IN** but the **TOKEN** packet is an **OUT**), the **TOKEN** packet is ignored. Otherwise, it is accepted, and the data transaction can begin. A bidirectional endpoint, however, accepts both **IN** and **OUT** packets.

## Device classes

The functionality of a USB device is defined by a class code sent to a USB host. This allows the host to load software modules for the device and to support new devices from different manufacturers.

Device classes include:

Class	Usage	Description	Examples, or exception
00	Device	Unspecified	Device class is unspecified, interface descriptors are used to determine needed drivers
01	Interface	Audio	Speaker, microphone, sound card, USB, and FireWire
02	Both	Communications and CDC control	UART and RS-232 serial adapter, modem, Wi-Fi adapter, Ethernet adapter. Used together with class 0Ah (CDC-Data) below
03	Interface	USB human interface device (HID)	Keyboard, mouse, joystick
05	Interface	Physical interface device (PID)	Force feedback joystick
06	Interface	Media (Picture Transfer Protocol(PTP) / Media Transfer Protocol (MTP)	Scanner, Digital camera
07	Interface	Printer	Laser printer, inkjet printer, CNC machine
08	Interface	USB mass storage, USB Attached SCSI	USB flash drive, memory card reader, digital audio player, digital camera, external drive
09	Device	USB hub	High speed USB hub
0A	Interface	CDC-Data	Used together with class 02h (Communications and CDC Control) above
0B	Interface	Smart card	USB smart card reader
0D	Interface	Content security	Fingerprint reader
0E	Interface	Video	Webcam
0F	Interface	Personal healthcare device class (PHDC)	Pulse monitor (watch)
10	Interface	Audio/Video (AV)	Webcam, TV
11	Device	Billboard	Describes USB-C alternate modes supported by device
DC	Both	Diagnostic device	USB compliance testing device
E0	Interface	Wireless Controller	Bluetooth adapter
EF	Both	Miscellaneous	ActiveSync device
FE	Interface	Application-specific	IrDA Bridge, RNDIS, Test & Measurement Class (USBTMC), USB DFU (Device Firmware Upgrade)
FFh	Both	Vendor-specific	Indicates that a device needs vendor-specific drivers

## USB mass storage / USB drive

The USB *mass storage device class* (**MSC** or **UMS**) standardizes connections to storage devices. At first intended for magnetic and optical drives, it has been extended to support flash drives and SD card readers. The ability to boot a write-locked SD card with a USB adapter is particularly advantageous for maintaining the integrity and non-corruptible, pristine state of the booting medium.



Though most personal computers since early 2005 can boot from USB mass storage devices, USB is not intended as a primary bus for a computer's internal storage. However, USB supports hot swapping, making it useful for mobile peripherals, including various types of drives.

Several manufacturers offer external portable USB hard disk drives or empty enclosures for disk drives. These offer performance comparable to internal drives, limited by the number and types of attached USB devices and by the USB interface's upper limit. Other competing standards for external drive connectivity include eSATA, ExpressCard, FireWire (IEEE 1394), and, most recently, Thunderbolt.

## Media Transfer Protocol

*Media Transfer Protocol* (**MTP**) was designed by Microsoft to give higher-level access to a device's filesystem than USB mass storage, at the level of files rather than disk blocks. It also has optional DRM features. MTP was designed for use with portable media players. Still, it has since been adopted as the primary storage access protocol for the Android operating system, starting with version 4.1 Jelly Bean, as well as Windows Phone 8 (Windows Phone 7 devices had used the Zune protocol—an evolution of MTP).

The primary reason is that MTP does not require exclusive access to the storage device as UMS does, alleviating potential problems if an Android program requests access. At the same time, it is attached to a computer. The main drawback is that MTP is not as well-supported on non-Windows operating systems.

## Human interface devices

A USB mouse or keyboard can usually be used with older computers that have PS/2 ports with the aid of a small USB-to-PS/2 adapter. For mice and keyboards with dual-protocol support, a passive adapter that contains no logic circuitry may be used: the USB or PS/2 hardware in the keyboard or mouse is designed to detect whether it is connected to a USB or PS/2 port and to communicate using the appropriate protocol. Active converters that connect USB keyboards and mice (usually one of each) to PS/2 ports also exist.

## Device Firmware Upgrade mechanism

*Device Firmware Upgrade* (**DFU**) is a generic mechanism for upgrading the firmware of USB devices with improved versions provided by their manufacturers, offering (for example) a way to deploy firmware bug fixes. During the firmware upgrade, USB devices switch to a PROM-programming mode,

effectively acting as a PROM programmer. Any USB device class can implement this capability by following the official DFU specifications. Doing so allows the use of DFU-compatible host tools to update the device.

## Audio streaming

The USB Device Working Group has established specifications for audio streaming, and specific standards have been created and implemented for audio-class devices such as microphones, speakers, headsets, telephones, and musical instruments. The working group has released three versions of audio device specifications: USB Audio 1.0, 2.0, and 3.0, known as “**UAC**” and “**ADC**” respectively.

UAC 3.0 mainly introduces improvements for portable devices, such as reducing power consumption by bursting data and staying in low-power mode more often, along with power domains for various device components, enabling them to be shut down when not in use.

## USB OTG

USB OTG, or **USB On-The-Go**, is a specification that allows devices like smartphones and tablets to act as hosts and control other USB devices, enabling direct connection of peripherals such as USB flash drives, keyboards, and mice without a computer. It enables two-way data transfer and makes mobile devices more flexible and interactive by allowing them to switch between peripheral and host roles.

### How it works

- In a standard USB connection, a computer always acts as the host, and the peripheral device (such as a printer) is the peripheral.
- USB OTG enables this role to be reversed, allowing a phone to act as a host to control a USB flash drive or mouse, or to function as a peripheral when connected to a computer.
- To use it, you need an OTG-capable device and an OTG adapter or cable, which converts the device's charging port (like USB-C or Micro-USB) into a standard USB-A port for connecting peripherals.

### Examples of use

- Connect storage devices: Plug in a USB flash drive or external hard drive to a phone to transfer files or expand storage capacity, as shown in this YouTube video.
- Use a keyboard and mouse: Connect a keyboard and mouse for easier data entry or gaming on a tablet or phone, as explained by Coolblue and Anker.
- Connect other accessories: Attach peripherals like gamepads, cameras, or an external digital audio converter (DAC) for better audio quality, notes XDA Developers.

# Sources

Wikipedia (<https://en.wikipedia.org/wiki/USB>)

## USB topics on lamaPLC

Page	Date	Tags
<ul style="list-style-type: none"> <li>• <a href="#">lamaPLC Communication: USB</a></li> </ul>	2026/04/23 21:51	usb, usb otg, otg, msc, ums, mtp, dfu, uac, adc, communication, bus, basic
<ul style="list-style-type: none"> <li>• <a href="#">lamaPLC: ESP32 / ESP8266</a></li> </ul>	2025/11/22 00:07	esp8266, esp32, esp32-c2, esp32-c3, esp32-c5, esp32-c6, esp32-c61, esp32-h2, esp32-s2, esp32-s3, esp32-p4, espressif systems, communication, ethernet, ip, wi-fi, thread, zigbee, matter, homekit, bluetooth, mqtt, adc, spi, uart, i2c, i2s, rmt, pwm, usb, usb otg, twai

[USB](#), [USB OTG](#), [OTG](#), [MSC](#), [UMS](#), [MTP](#), [DFU](#), [UAC](#), [ADC](#), [communication](#), [bus](#), [basic](#)

This page has been accessed for: Today: 4, Until now: 169

From: <http://lamaplc.com/> - **lamaPLC**

Permanent link: [http://lamaplc.com/doku.php?id=com:basic\\_usb](http://lamaplc.com/doku.php?id=com:basic_usb)

Last update: **2026/04/21 20:46**

