

# USB over IP UDC

## Introduction

In USB protocol we have two different sides of communication: USB host and USB device. USB host is usually a computer which we would like to extend with some additional functionality. For example we would like to extend it with printing functionality with our USB printer or with some additional memory using our pendrive. USB device can be not only as simple as pendrive or printer but may be also quite complex with a lot of additional functionalities like mobile phone. To be accurate each device which has a USB Device Controller can become a USB device. If such device is running Linux it needs two types of dedicated drives: UDC driver and USB gadget driver. USB gadget driver is responsible for providing implementation of upper layers of USB protocol and protocols build on the top of it (for example mass storage or Ethernet). This driver doesn't access hardware but only communicates with UDC driver and pass messages which should be sent. UDC driver is responsible for passing messages (called USB requests) from gadget driver to USB physical link (usb cable).

USB over IP (<http://usbip.sourceforge.net/>) is a project which aims to develop a general USB device sharing system over IP network. To fully share a USB device USBIP encapsulates USB requests into TCP/IP traffic and transmits them between computers.

Currently Linux kernel supports device sharing using USBIP. This means that using mainline kernel you may connect USB device to one computer and share it with another computer using network (<https://www.kernel.org/doc/readme/tools-usb-usbip-README>).

## Problem statement

Current Linux kernel allows you to use a device connected to one computer on the other computer system.

As mentioned before Linux supports also USB device mode using USB gadget subsystem. Thanks to this it is possible to make your computer a USB device (choosing which protocols it will support). But software implementation of USB device is not enough. We need some hardware to communicate with USB host. This is where your task starts!

We would like you to write USB UDC driver which emulates that your computer has a hardware suitable for USB communication. This driver should take USB requests from gadget driver (there is an API defined in kernel) and instead of talking with hardware, send those requests using USB over IP protocol.

Whole project consist of two parts: UDC driver in kernel and USBIP daemon modification. UDC driver should take the requests from gadget and pass them to

userspace. Second part is a modification of USBIP daemon to allow it to make a use of UDC driver and pass those requests to remote computer.

At this moment you may ask yourself why we would like to do such things. So let me explain. It's a common situation that developer is using smartphone emulator to develop application. Sometimes he would like to connect this virtual smartphone as a USB device to his computer just like a physical one. Currently he cannot do this. To connect using ADB or Kies we need to use some hacks. We would like to eliminate them and just simply allow emulating this connection using USB over IP.

### **Other requirements:**

Our target is to work in open source and with open source. That's why we would like you to develop this project on github or any other opened platform and share it with community. We would like also to include your changes to mainline kernel and USBIP project.

This all means that your code should have a good quality and be written according to kernel rules. We would like you also to work with community. This means discussion with USBIP project community and maintainers and also with linux-usb mailing list.

### **Time estimation:**

We would like to have the solution ready till the **30<sup>th</sup> May 2015**.

The number of people needed: approximately **4**.

### **Prerequisites:**

C, Kernel programming, Sockets, Interest in USB, USB over IP