DIY KVM over IP

That’s a lot of letters

What is a KVM?

In this context, we are talking the venerable mainstay of the datacentre, the Keyboard, Video, and Mouse (KVM). The stalwart companion of the person maintaining the servers in the racks.

This was a major step up from the crash cart

Diversion One: Crash Carts

These things are still sold.

Diversion One: Crash Carts

A more modern version

The modern KVM

Today this usually refers to a rack mountable 1U device that integrates the keyboard, monitor and mouse into a slide out unit that has a selector for multiple devices. In the image below, it is a 8 port device and the current price is $3782.99

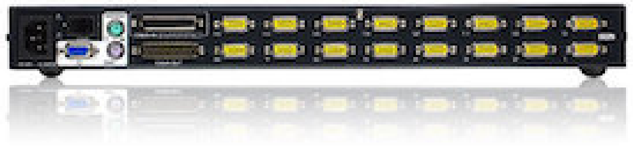
The KVM evolved slowly

As requirements grew, these units were attached to larger and larger switches. I think the largest I have seen personally was a 128 unit.

It was a pure analog solution, with proprietary cabling to bundle the keyboard, video and mouse signals into a single cable that would have non-standard wiring,

Example: ATEN KH 116

- A single PS/2 console controls up to 16 computers
- Remote Console Operation - control your system from the remote PS/2 keyboard, mouse, and monitor consoles
- Cat 5e cable to connect the KH0116 and remote units (KA9250)
- Dedicated chain ports - daisy chain up to 31 additional units - control up to 512 computers from a single console



Put it on the network!

Since we have “remote”, why not make it network aware and we can access it from anywhere on our network with proprietary software.

Maybe browser and java based…

After all, we have network serial ports to talk to switches and such devices, so this should be easy.

KVM over IP = Expensive...

These devices are not cheap. A single port unit goes for roughly $500 and heads north rapidly.

Do we still use these?

In most current use cases, the rack mount server manufacturer provides this functionality via an iLO card. It has many names, rilo, drac, etc. Basically it is a daughter card that is added to the server (usually comes bundled but mostly disabled). A license is required to fully unlock it.

Sometimes the older solution is cheaper.

Additional Functionality

Now a modern server management card includes a LAN interface, sensors, a serial port, the ability to power on and off the server, upgrade the firmware, as well as a suite of other functionality. The main ones have already been covered.

Pi-KVM

Which finally brings us to tonight’s topic, building a DIY KVM based on a Raspberry Pi.

- Why would we want a KVM?
- How much does this one cost?
- How much work is it?

Why would we want one?

The use cases will vary widely.

In my case:

- My lab system(s) are remote from me.
    - They are somewhere where the noise will not bother me or my wife
- Convenience

How much does this cost?

As specified later, assuming you need to buy everything, roughly $150.00

How much work is it?

Well, that depends…

- Get the parts
- Assemble the parts
- Burn the image
- Test

Hardware decisions

Before we get started, you can do this on pretty much any Pi B (not v1) or Pi Zero. It is recommended that you use a Pi 4 or a Pi Zero W as they have OTG built in which reduces parts and makes it easier to do.

Version 0

- Raspberry Pi 2 or 3
- MicroSD card (8 GB is enough)
- USB-A 3A charger (female socket) or power supply
- For keyboard & mouse emulator (HID):
    - Arduino Pro Micro (based on an ATMega32u4)
    - Logic level shifter
    - 1x NPN transistor (almost any NPN transistor: 2n2222 or similar)
    - 1x 390 Ohm resistor
    - A breadboard and wires
- 2x USB A-to-micro cables (male-male, for power and HID)
- HDMI capture device

Version 1

There is no version 1

Version 2

- Raspberry Pi 4 2 GB model Raspberry Pi Zero W
- MicroSD card (min 16 GB recommended)
- USB-A 3A charger (female socket) or power supply
- Video capture device, either HDMI to CSI-2 bridge based on TC358743 or a HDMI to USB dongle
- USB-micro splitter or USB-C splitter
- USB-A-to-micro cables or USB-A-to-USB-C cables (male-male, for power and keyboard & mouse emulator)

Version 3 (available soon)

- A fully assembled PiHat format device
- HDMI capture based on the TC358743
- OTG Keyboard & mouse; Mass Storage Drive emulation
- Ability to simulate “removal and insertion” for USB
- Onboard ATX power control
- Onboard fan controller
- A real-time clock
- CISCO-style and USB serial console port
- Optional AVR-based HID
- Optional OLED screen to display information
- Continued use of Pi-KVM OS
- It will cost about $100 (or less)

My hardware

Item	Name
1	Raspberry Pi Model 4/2GB
2	USB-C/PWR Splitter
3	USB-C to USB-C 3.1 Gen 2 Cable
4	USB C Cable Short, Type C to USB 2.0
5	HDMI Video Capture Card HD
6	Aluminum Raspberry Pi 4 Model B Case
7	Ultra Thin HDMI Cable
8	Raspberry Pi 4 Power Supply, 5V 3A

Hardware Assembly

Basic Assembly

I won’t go into great detail here but I will provide photos for the connectivity.

Since I don’t know what case you will use, just put the Pi in the case and make sure any cooling is set up. The Pi4 runs a bit hot. Passive cooling will be fine if there is some airflow, but you can play it safe and get a case with a fan. The one I have pictured does not have a fan, however my other Pi4 cases do.

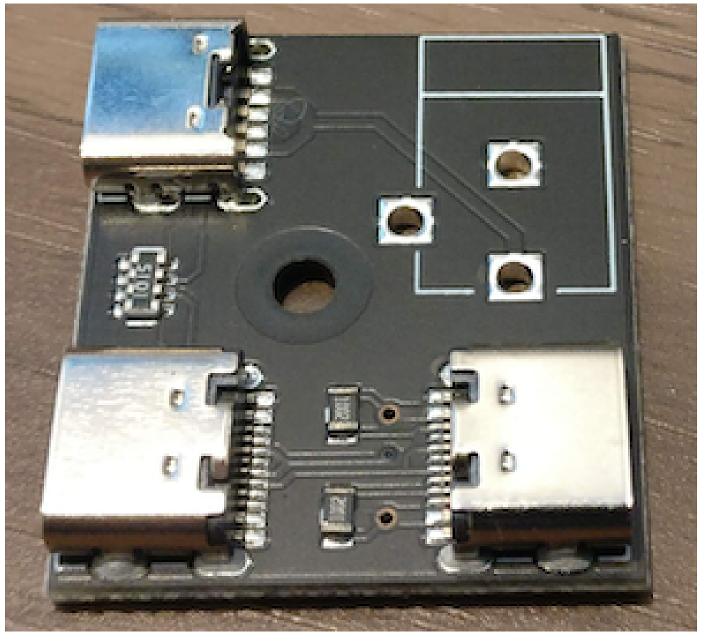
Power splitter

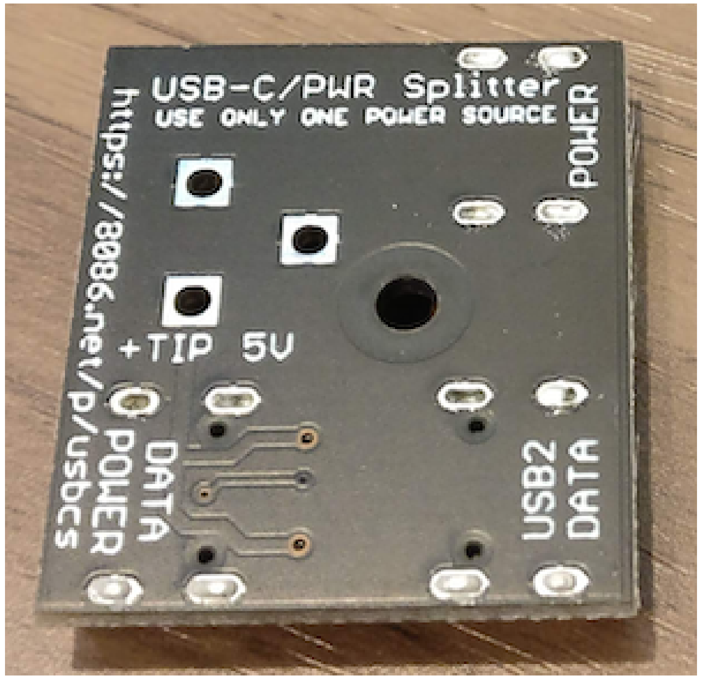
We are isolating the power from the target USB to our Pi and providing a power adepter. Mostly because if you reboot, you may lose power to the USB ports and that will crash the Pi. Not good.

- Wall wart to Splitter Power (switched off)
- USB-C to USB-A goes to the target computer
- USB-C to USB-C goes to the Pi

Power splitter pics

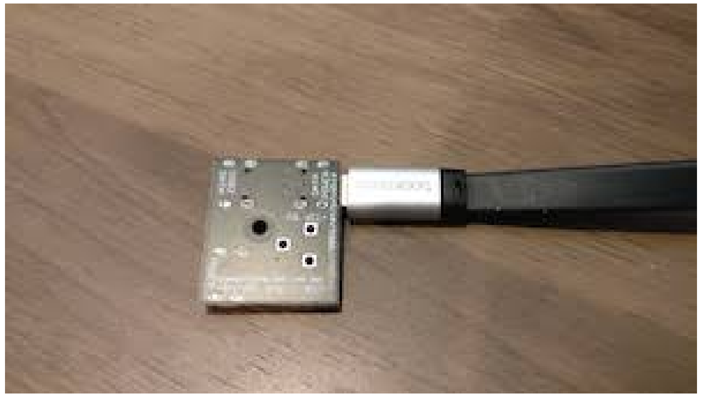
Just so you can see what it actually looks like





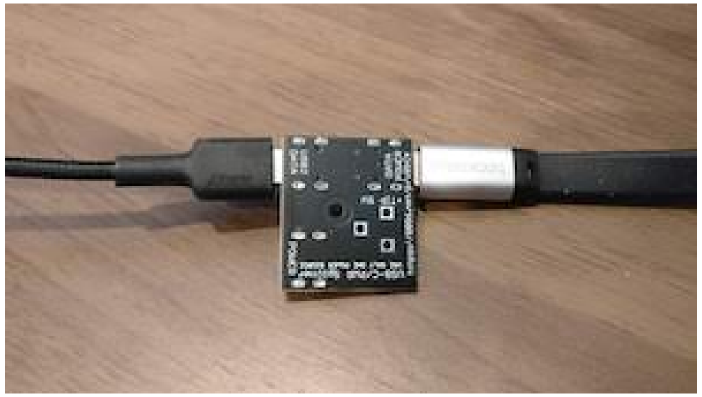
First USB connection

Connect the short USB-C to USB-C cable to the data & power connector.



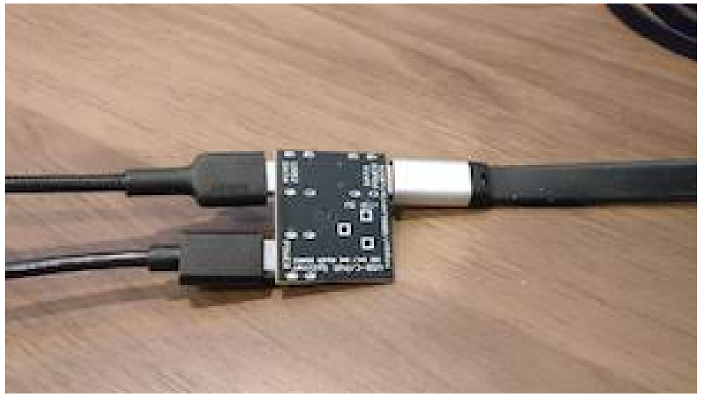
Second USB connection

Connect the USB-C to USB-A to the data connector.



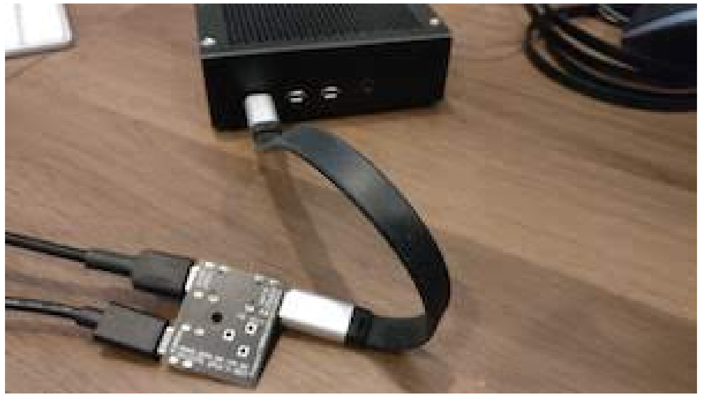
Power connection

Connect the external power supply USB-C connector to the power connector. Leave the power turned off.



UBC-C to Pi4

Now connect the free USB-C connector to the USB-C port on the Raspberry Pi.

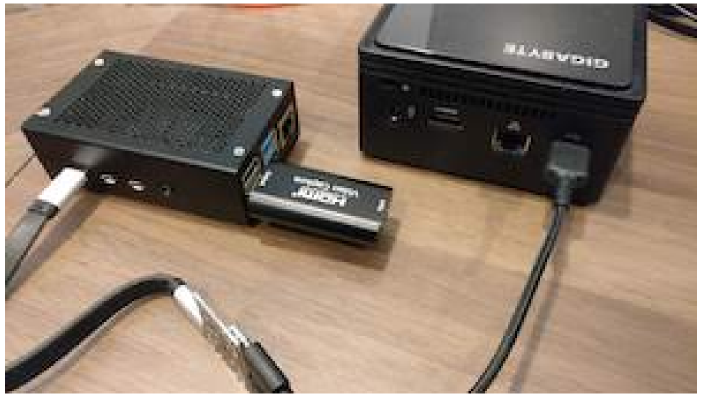


USB-A to target computer

Connect the free USB-A connector to a USB port on the target computer.

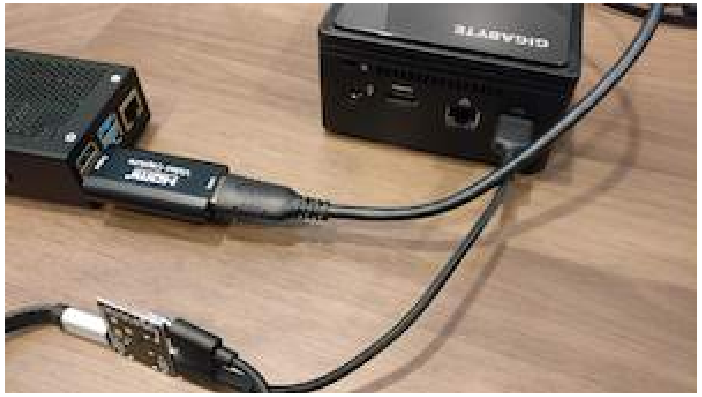
Video Capture

Plug the video capture device into the bottom left USB 2 port as that is what the software has been pre-configured to use. It can be changed.



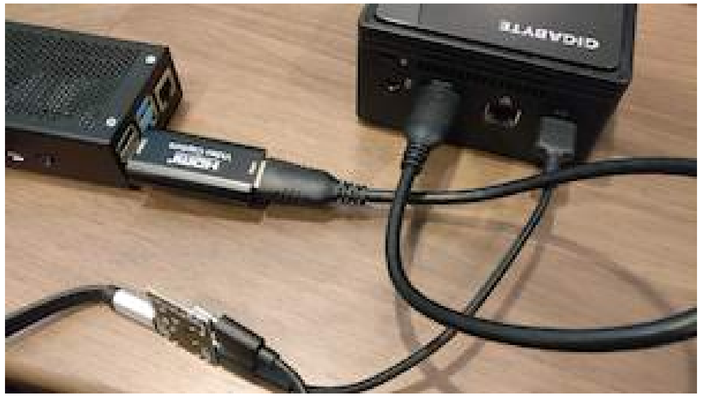
Connect video cable to Pi.

Connect the HDMI cable the HDMI input of the capture device.



Connect video cable to target

Connect the other end of the HDMI cable to the target system video out. If you have other interfaces, you will need an adapter.



All done

Software

Download

Download the software from:

The project download page

Specifically HDMI-to-USB dongle

The project site has multiple instructions on how to create the micro SD card image.

Other things

When booting the device, you may want a monitor and keyboard for the first boot so you can get the IP address it obtained from DHCP or set a proper address.

Demo

No demo tonight, we had one last month.

Parts List

- Raspberry Pi Model 4/2GB
- USB-C/PWR Splitter
- USB-C to USB-C 3.1 Gen 2 Cable
- USB C Cable Short, Type C to USB 2.0
- HDMI Video Capture Card HD
- Aluminum Raspberry Pi 4 Model B Case
- Ultra Thin HDMI Cable
- Raspberry Pi 4 Power Supply, 5V 3A

References

- π-kvm Project Page
- Pi-KVM Github
- Pi-KVM: an inexpensive KVM over IP