

# HOW TO INSTALL PIKVM on x86

by @srepac srepac@kvmnerds.com June 10, 2023

## KNOWLEDGE REQUIREMENTS:

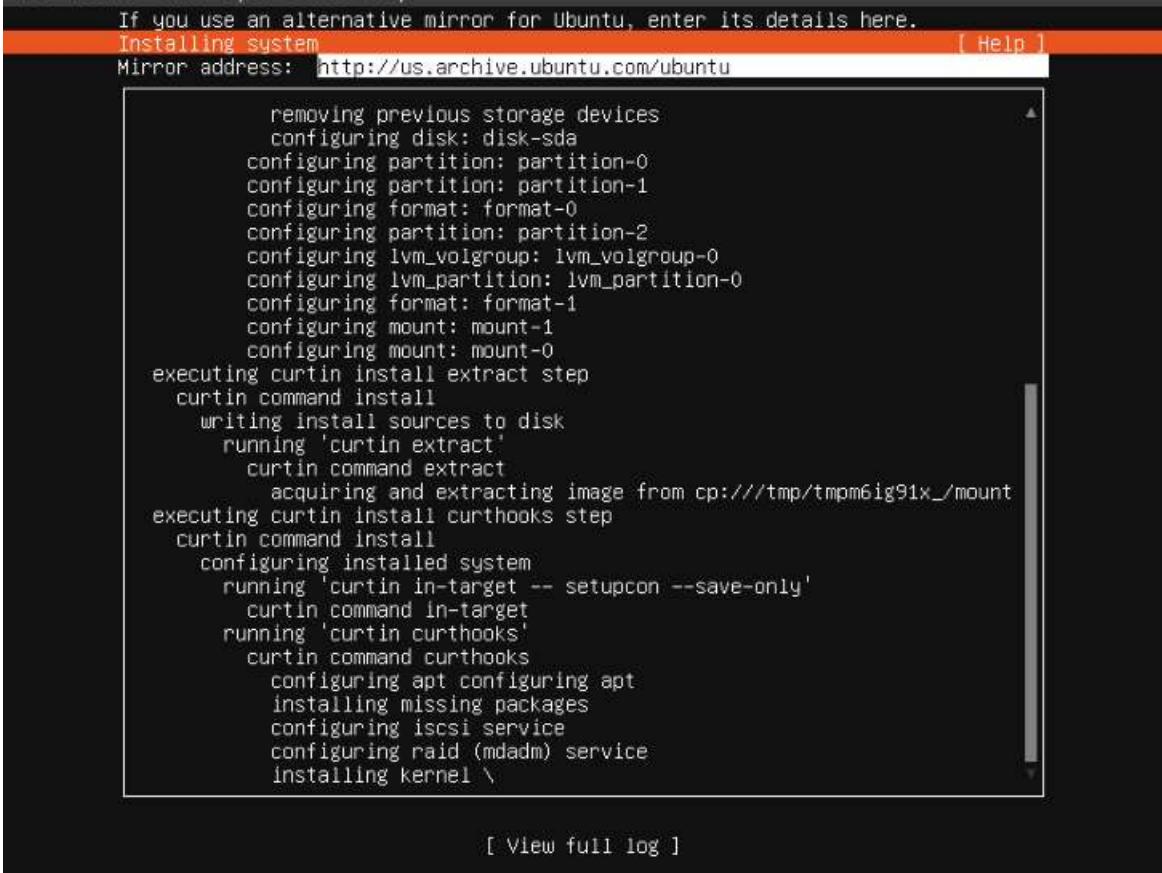
- User has basic Linux sysadmin knowledge on how to edit files
- User has basic networking knowledge to get the Ubuntu system's IP address

## PART REQUIREMENTS:

- Laptop or desktop PC running Ubuntu 22.04 LTS (jammy) server or desktop OS  
<https://mirror.fcix.net/ubuntu-releases/22.04.2/ubuntu-22.04.2-live-server-amd64.iso>
- USB UART cable (connect PC to serial HID) <https://a.co/d/4Nyq5Gs>
- ch9329 serial HID (for keyboard and mouse control) <https://a.co/d/9bVkJhd>
- HDMI-USB dongle or USB loop video capture) <https://www.amazon.com/dp/B0BZYN4T4C> (Loop capture is better)

## STEP-BY-STEP INSTRUCTIONS:

0. Install ubuntu 22.04 LTS (jammy) desktop or server OS on your PC. During the install, create new user/password: **opikvm/opikvm**  
The user will be used later to login via SSH and install pikvm x86. Please refer to ubuntu docs for details.



If you use an alternative mirror for Ubuntu, enter its details here.  
Installing system [ Help ]  
Mirror address: <http://us.archive.ubuntu.com/ubuntu>

```
removing previous storage devices
configuring disk: disk-sda
configuring partition: partition-0
configuring partition: partition-1
configuring format: format-0
configuring partition: partition-2
configuring lvm_volidgroup: lvm_volidgroup-0
configuring lvm_partition: lvm_partition-0
configuring format: format-1
configuring mount: mount-1
configuring mount: mount-0
executing curtin install extract step
curtin command install
    writing install sources to disk
        running 'curtin extract'
            curtin command extract
                acquiring and extracting image from cp:///tmp/tmpm6ig91x/_mount
executing curtin install curthooks step
curtin command install
    configuring installed system
        running 'curtin in-target -- setupcon --save-only'
            curtin command in-target
        running 'curtin curthooks'
            curtin command curthooks
                configuring apt
                    configuring apt
                installing missing packages
                configuring iscsi service
                configuring raid (mdadm) service
                installing kernel \
[ View full log ]
```

1. Wire up the USB UART and CH9329 board as follows. Pay close attention to the jumper setting on the CH9329 and the wiring connection from UART to the CH9329. **NOTE: Red wire (+5vcc) is not connected.**

**You can power the ch9329 board either from the pikvm -OR- from target PC NOT BOTH.** We will be powering the ch9329 board from the target system's USB port, which is the reason why the red wire is not connected. *Connect the UART cable side to the Ubuntu linux system.*

UART White/RX -> TX on ch9329  
UART Green/TX -> RX on ch9329  
UART Black/GND -> GND on ch9329





2. Login via SSH to your ubuntu PC with **opikvm/opikvm** username and password. Run the following commands (as root) to clone my kvmd-armbian git and the install-x86.sh script and run part 1 of the installer. Press ENTER when asked to reboot.

NOTE: During the install, whenever you see "Restart services..." while running install-x86.sh, please press ENTER key. If you don't Ubuntu will be waiting for you in order to go to the next command.

```
sudo su -  
apt update  
apt upgrade -y  
  
apt install -y git vim make python3-dev gcc  
git clone https://github.com/srepac/kvmd-armbian.git  
cd kvmd-armbian/  
chmod +x install-x86.sh  
./install-x86.sh
```

```
root@vmpikvm: ~/kvmd-armbian
root@vmpikvm:~/kvmd-armbian# ./install-x86.sh
Python 3.10 is supported.
./install-x86.sh: line 55: /proc/device-tree/model: No such file or directory

Running part 1 of PiKVM installer script for x86 by @srepac

-> Getting Pi-KVM packages from https://files.pikvm.org/repos/arch/rpi4
wget https://files.pikvm.org/repos/arch/rpi4 -o /var/cache/kvmd/packages.txt

wget https://files.pikvm.org/repos/arch/rpi4/janus-gateway-pikvm-0.13.3-1-armv7h.pkg.tar.xz -o /var/cache/kvmd/janus-gateway-pikvm-0.13.3-1-armv7h.pkg.tar.xz
wget https://files.pikvm.org/repos/arch/rpi4/kvmd-3.221-1-any.pkg.tar.xz -o /var/cache/kvmd/kvmd-3.221-1-any.pkg.tar.xz
wget https://files.pikvm.org/repos/arch/rpi4/kvmd-platform-v0-hdmi-rpi3-3.221-1-any.pkg.tar.xz -o /var/cache/kvmd/kvmd-platform-v0-hdmi-rpi3-3.221-1-any.pkg.tar.xz
wget https://files.pikvm.org/repos/arch/rpi4/kvmd-platform-v0-hdmiusb-rpi3-3.221-1-any.pkg.tar.xz -o /var/cache/kvmd/kvmd-platform-v0-hdmiusb-rpi3-3.221-1-any.pkg.tar.xz
wget https://files.pikvm.org/repos/arch/rpi4/kvmd-platform-v2-hdmi-rpi3-3.221-1-any.pkg.tar.xz -o /var/cache/kvmd/kvmd-platform-v2-hdmi-rpi3-3.221-1-any.pkg.tar.xz
wget https://files.pikvm.org/repos/arch/rpi4/kvmd-webterm-0.43-1-any.pkg.tar.xz -o /var/cache/kvmd/kvmd-webterm-0.43-1-any.pkg.tar.xz

ls -l /var/cache/kvmd
total 2164
drwxr-xr-x 2 root root 4096 May 30 18:30 ARCHIVE
-rw-r--r-- 1 root root 1054380 May 28 21:37 janus-gateway-pikvm-0.13.3-1-armv7h.pkg.tar.xz
-rw-r--r-- 1 root root 1100132 May 28 21:37 kvmd-3.221-1-any.pkg.tar.xz
-rw-r--r-- 1 root root 6168 May 28 21:37 kvmd-platform-v0-hdmi-rpi3-3.221-1-any.pkg.tar.xz
-rw-r--r-- 1 root root 6628 May 28 21:37 kvmd-platform-v0-hdmiusb-rpi3-3.221-1-any.pkg.tar.xz
-rw-r--r-- 1 root root 6204 May 28 21:37 kvmd-platform-v2-hdmi-rpi3-3.221-1-any.pkg.tar.xz
-rw-r--r-- 1 root root 6468 May 28 21:37 kvmd-webterm-0.43-1-any.pkg.tar.xz
-rw-r--r-- 1 root root 19603 May 30 18:30 packages.txt

Platform selected -> kvmd-platform-v0-hdmiusb-rpi3

/etc/modules
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.

dwc2
libcomposite
i2c-dev

# https://unix.stackexchange.com/questions/66901/how-to-bind-usb-device-under-a-static-name
# https://wiki.archlinux.org/index.php/Udev#Setting_static_device_names
KERNEL=="video[0-9]*", SUBSYSTEM=="video4linux", PROGRAM="/usr/bin/kvmd-udev-hdmiusb-check rpi3 1-1.4:1.0", ATTR{index}=="0", GROUP=="kvmd", SYMLINK+="kvmd-video"
KERNEL=="ttyUSB0", SYMLINK+="kvmd-hid"

-> Installing dependencies for pikvm
apt install -y nginx python3 net-tools bc expect v4l-utils iptables vim dos2unix screen tmate nfs-common gpiod ffmpeg dialog
iptables dnsmasq git python3-pip tesseract-ocr tesseract-ocr-eng libasound2-dev libsndfile-dev libspeexdsp-dev lm-sensors

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Extracting templates from packages: 100%
Scanning processes...
Scanning candidates...
Scanning linux images...
```

```
>>> Installing dependencies for pikvm
apt install -y nginx python3 net-tools bc expect v4l-utils iptables vim dos2unix screen tmate nfs-common gpiod ffmpeg dialog
iptables dnsmasq git python3-pip tesseract-ocr tesseract-ocr-eng libasound2-dev libsndfile-dev libspeexdsp-dev lm-sensors

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Extracting templates from packages: 100%
scanning processes...
scanning candidates...
scanning linux images...

Running kernel seems to be up-to-date.

Restarting services...
service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service
systemctl restart user@1000.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.

apt-get install python3-aiofiles -y
scanning processes...
scanning candidates...
scanning linux images...

Running kernel seems to be up-to-date.

Restarting services...
```

3. Log back in as opikvm/opikvm and run part 2 of installer after reboot (as root).

```
sudo su -
cd kvmd-armbian/
./install-x86.sh
```

```
root@vmpikvm:~/kvmd-armbian
opikvm@vmpikvm:~$ sudo su -
[sudo] password for opikvm:
root@vmpikvm:~# cd kvmd-armbian/
root@vmpikvm:~/kvmd-armbian# ./install-x86.sh
Python 3.10 is supported.
./install-x86.sh: line 55: /proc/device-tree/model: No such file or directory

Running part 2 of PIKVM installer script for x86 by @srepac
=> Ensuring KVMD users and groups ...

-> Disabling nginx service, so that we can use kvmd-nginx instead
Synchronizing state of nginx.service with sysv service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable nginx
Removed /etc/systemd/system/multi-user.target.wants/nginx.service.

-> Creating symlinks for use with kvmd python scripts

-> Creating kvmd-webterm homedir
drwxr-xr-x 2 kvmd-webterm root 4096 May 30 18:54 /home/kvmd-webterm
tar xvf ch9329.tar -c /usr/lib/python3/dist-packages
kvmd/plugins/hid/ch9329/
kvmd/plugins/hid/ch9329/_init__.py
kvmd/plugins/hid/ch9329/keyboard.py
kvmd/plugins/hid/ch9329/tty.py
kvmd/plugins/hid/ch9329/mouse.py
total 28
-rw-r--r-- 1 root root 10421 May 16 16:41 __init__.py
-rw-r--r-- 1 root root 3021 May 16 16:42 keyboard.py
-rw-r--r-- 1 root root 4145 May 16 16:42 mouse.py
-rw-r--r-- 1 root root 2718 May 16 16:42 tty.py
-> Making backup of files that require modification
cp /usr/lib/python3/dist-packages/kvmd/plugins/ugpio/gpio.py /usr/lib/python3/dist-packages/kvmd/plugins/ugpio/gpio.py.orig
cp /usr/lib/python3/dist-packages/kvmd/apps/kvmd/info/hw.py /usr/lib/python3/dist-packages/kvmd/apps/kvmd/info/hw.py.orig
cp /usr/lib/python3/dist-packages/kvmd/apps/kvmd/info/base.py /usr/lib/python3/dist-packages/kvmd/apps/kvmd/info/base.py.orig
-rw-r--r-- root/root 4291 2023-05-29 01:27 kvmd/plugins/ugpio/gpio.py
-rw-r--r-- root/root 5567 2023-05-29 01:27 kvmd/apps/kvmd/info/hw.py
-rw-r--r-- root/root 1713 2023-05-29 01:27 kvmd/apps/kvmd/info/base.py
tar xvf x86-mods.tar -c /usr/lib/python3/dist-packages
kvmd/plugins/ugpio/gpio.py
kvmd/apps/kvmd/info/hw.py
kvmd/apps/kvmd/info/base.py
-rw-r--r-- 1 root root 4291 May 29 01:27 /usr/lib/python3/dist-packages/kvmd/plugins/ugpio/gpio.py
-rw-r--r-- 1 root root 5567 May 29 01:27 /usr/lib/python3/dist-packages/kvmd/apps/kvmd/info/hw.py
-rw-r--r-- 1 root root 1713 May 29 01:27 /usr/lib/python3/dist-packages/kvmd/apps/kvmd/info/base.py
-> Checking kvmd -m works before continuing
ipmi:
    auth:
        file: /etc/kvmd/ipmipasswd

    kvmd:
        timeout: 5.0
        unix: /run/kvmd/kvmd.sock

    server:
        host: '::'
        port: 623
        timeout: 10.0

    sol:
        device: ''
        proxy_port: 0
        select_timeout: 0.1
        speed: 115200
```

```
[root@vmpikvm: ~/kvmd-armbian
timeout: 300
Did kvmd -m run properly? [y/n] y
-> Enabling kvmd-nginx kvmd-webterm and kvmd services, but do not start them.
Created symlink /etc/systemd/system/multi-user.target.wants/kvmd-nginx.service -> /lib/systemd/system/kvmd-nginx.service.
Created symlink /etc/systemd/system/multi-user.target.wants/kvmd-webterm.service -> /lib/systemd/system/kvmd-webterm.service.
Created symlink /etc/systemd/system/multi-user.target.wants/kvmd.service -> /lib/systemd/system/kvmd.service.
Created symlink /etc/systemd/system/multi-user.target.wants/kvmd-fix.service -> /lib/systemd/system/kvmd-fix.service.

Check kvmd devices
1rwxrwxrwx 1 root root 7 May 30 18:51 /dev/kvmd-hid -> ttyUSB0
1rwxrwxrwx 1 root root 6 May 30 18:51 /dev/kvmd-video -> video0

You should see devices for keyboard, mouse, and video.

Point a browser to https://vmpikvm
If it doesn't work, then reboot one last time.
Please make sure kvmd services are running after reboot.
    Loaded: loaded (/lib/systemd/system/kvmd-nginx.service; enabled; vendor preset: enabled)
    Loaded: loaded (/lib/systemd/system/kvmd-webterm.service; enabled; vendor preset: enabled)
    Loaded: loaded (/lib/systemd/system/kvmd.service; enabled; vendor preset: enabled)
    Loaded: loaded (/lib/systemd/system/kvmd-fix.service; enabled; vendor preset: enabled)
-> Extracting aiofiles.tar into /usr/lib/python3.11/site-packages
aiofiles/
aiofiles/__pycache__/
aiofiles/__pycache__/base.cpython-310.pyc
aiofiles/__pycache__/__init__.cpython-310.opt-1.pyc
aiofiles/__pycache__/os.cpython-310.opt-1.pyc
aiofiles/__pycache__/base.cpython-310.opt-1.pyc
aiofiles/__pycache__/__init__.cpython-310.pyc
aiofiles/__pycache__/ospath.cpython-310.pyc
aiofiles/__pycache__/ospath.cpython-310.opt-1.pyc
aiofiles/__pycache__/os.cpython-310.pyc
aiofiles/os.py
aiofiles/threadpool/
aiofiles/threadpool/__pycache__/
aiofiles/threadpool/__pycache__/__init__.cpython-310.opt-1.pyc
aiofiles/threadpool/__pycache__/utils.cpython-310.opt-1.pyc
aiofiles/threadpool/__pycache__/binary.cpython-310.opt-1.pyc
aiofiles/threadpool/__pycache__/utils.cpython-310.pyc
aiofiles/threadpool/__pycache__/__init__.cpython-310.pyc
aiofiles/threadpool/__pycache__/text.cpython-310.opt-1.pyc
aiofiles/threadpool/__pycache__/text.cpython-310.pyc
aiofiles/threadpool/__pycache__/binary.cpython-310.pyc
aiofiles/threadpool/__init__.py
aiofiles/threadpool/binary.py
aiofiles/threadpool/utils.py
aiofiles/threadpool/text.py
aiofiles/__init__.py
aiofiles/tempfile/
aiofiles/tempfile/__pycache__/
aiofiles/tempfile/__pycache__/__init__.cpython-310.opt-1.pyc
aiofiles/tempfile/__pycache__/tempytypes.cpython-310.opt-1.pyc
aiofiles/tempfile/__pycache__/tempytypes.cpython-310.pyc
aiofiles/tempfile/__pycache__/__init__.cpython-310.pyc
aiofiles/tempfile/__init__.py
aiofiles/tempfile/tempytypes.py
aiofiles/base.py
aiofiles/ospath.py
-> Renaming original aiofiles and creating symlink to correct aiofiles
1rwxrwxrwx 1 root root 42 May 30 18:54 aiofiles -> /usr/lib/python3.11/site-packages/aiofiles
drwxr-xr-x 2 root root 4096 May 30 18:43 aiofiles-0.8.0.dist-info
drwxr-xr-x 5 root root 4096 May 30 18:43 aiofiles.20230530.1854
root@vmpikvm:~/kvmd-armbian#
```

4. In case you haven't already done so as per Step 2, connect the USB UART cable to your linux x86 pikvm now. Make a note which ttyUSB showed up after connecting the USB UART to your linux x86 pikvm by running the dmesg command below.

```
dmesg | grep ttyUSB
```

Example output:

```
root@toughbook:~# dmesg | grep ttyUSB
[ 28.570919] usb 1-3: Qualcomm USB modem converter now attached to ttyUSB0
[290551.818150] usb 5-2: p12303 converter now attached to ttyUSB1
[290668.340266] p12303 ttyUSB1: p12303 converter now disconnected from ttyUSB1
```

In this example, **ttyUSB1** was created for the p12303 UART cable since another built-in USB device to the toughbook laptop already took **ttyUSB0**. NOTE: for 99% of cases out there, the default will be **ttyUSB0** but I'm showing this just in case.

5. Edit /etc/kvmd/udev/rules.d/99-kvmd.rules and update the KERNEL== line with the **ttyUSB?** device from previous step.

```
nano /etc/kvmd/udev/rules.d/99-kvmd.rules
```

```
root@toughbook:# nano /etc/kvmd/udev/rules.d/99-kvmd.rules
# https://unix.stackexchange.com/questions/66901/how-to-bind-usb-device-under-a-static-name
# https://wiki.archlinux.org/index.php/Udev#Setting\_static\_device\_names
KERNEL=="video[0-9]*", SUBSYSTEM=="video4linux", PROGRAM="/usr/bin/kvmd-udev-hdmiusb-check rpi3 %b", ATTR{index}=="0",
GROUP="kvmd", SYMLINK+="kvmd-video"
KERNEL=="ttyUSB1", SYMLINK+="kvmd-hid"
```

6. Run the following to create the symlink and restart kvmd services.

```
udevadm trigger; sleep 5; ls -l /dev/kvmd*
systemctl restart kvmd
```

Example output:

```
root@toughbook:~# udevadm trigger; sleep 5; ls -l /dev/kvmd*
lrwxrwxrwx 1 root root 7 Jun 10 18:10 /dev/kvmd-hid -> ttyUSB1
lrwxrwxrwx 1 root root 6 Jun 10 18:10 /dev/kvmd-video -> video0
root@toughbook:~# systemctl restart kvmd
```

7. Connect the ch9329 and HDMI to your target system and enjoy!

**PiKVM** The Open Source IP-KVM  
Copyright © 2018-2023 Maxim Devaev

Server: toughbook

 KVM     Terminal     Logout

Please note that when you are working with a KVM session or another application that captures the keyboard, you can't use some keyboard shortcuts such as Ctrl+Alt+Del (which will be caught by your OS) or Ctrl+W (caught by your browser).

To override this limitation you can use [Google Chrome](#) or [Chromium](#) in application mode.

```
# On Linux using Chromium/Chrome via any terminal:  
$ `which chromium 2>/dev/null || which chrome 2>/dev/null || which google-chrome` --app="https://192.168.0.130"  
  
# On MacOS using Terminal application:  
$ /Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome --app="https://192.168.0.130/"  
  
# On Windows via cmd.exe:  
C:\> start chrome --app="https://192.168.0.130/"/>
```

[PiKVM Project](#) | [Documentation](#) | [Support](#)