LinuxCNC install for Debian 12.5 - Bookworm and Debian 13 Trixie

Base Installation

The installation process described below is now obsolete, just install the LinuxCNC ISO for 2.9.4. However, the later steps about the new kernel and Realtek dkms drivers are still relevant. Note that if you wish to install QtPyVCP, probe-basic or EtherCAT, use the ISO because it has the repositories for these already installed.

Also, you might like to follow this step by step illustrated tutorial on installing linuxcnc https://baxedm.com/linuxcnc-installation-for-edm-machine-on-linux-debian-12/

1. Download the debian bookworm 12.6 netinst ISO.
   https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/debian-12.6.0-amd64-netinst.iso
   As of July 2023, the latest 2.9.4 distribution is now in Debian 13 (Trixie). You may wish to use it instead of Debian 12.
   https://cdimage.debian.org/cdimage/weekly-builds/amd64/iso-cd/debian-testing-amd64-netinst.iso
   Note it is possible to use the Debian installer found at https://www.debian.org and follow these steps
2. Download Balena Etcher from https://www.balena.io/etcher.
3. Burn your image to a USB stick using Etcher.

4. Insert the USB stick into your PC and boot into it. You may need to enter the BIOS and change the boot order. If you have used the LinuxCNC installer, Jump ahead to step 10, otherwise continue.
5. **Do not add a root password!**
6. Give your PC a name.
7. Add a new user and allocate a password. I normally make both the username and password the same as the PC name.
8. When it asks to select the GUI, select XFCE and deselect gnome and Debian. Only Select XFCE or MATE.
   Take care here, if you don't select any options you will end up with a text based prompt only and as you won't have a root account if you followed these instructions,, you can't fix it without re-installing.
9. Complete the install and boot into debian!
   *If you end up at a text mode only console and do happen to have a root account, type **su** to enter root account and type:*
   /usr/sbin/usermod -aG sudo your-user-name
   to enable sudo. Then type
   sudo tasksel
   to add the GUI environment. Reboot to enable the graphical user interface.
10. Open a terminal window and Type the following lines:
    ```
    sudo apt update
    sudo dpkg -P raspi-firmware
    sudo apt dist-upgrade
    ```
    Note that the second line (sudo dpkg …) removes firmware for the raspberry pi incorrectly installed by Debian and is a bug out of our control
11. If you have not used the official LinuxCNC ISO, Install the PREEMPT_RT real time kernel otherwise jump ahead to step 14.
    Note using Debian 13, should install the RT kernel with linuxcnc-uspace so skip to step 13.
    a. Type:
       ```
       sudo apt-get install linux-image-rt-amd64
       ```
    b. Reboot into the new kernel.
    c. Type
       ```
       uname -v
       ```
       and check that PREEMPT_RT is shown in the prompt.
    d. Open Applications Menu > System > Synaptic Package Manager search for *linux-image* and right click on the original non-rt and select *Mark for Complete Removal*.
    e. Close and Reboot. This is to force the system to boot from the RT kernel. If you prefer to retain both kernels then the other kernels need not be

deleted, but grub boot configuration changes will be needed beyond the scope of this document.

12. On Debian 12, Install linuxcnc:
    a. Download the script by executing from the command line:
       `wget https://www.linuxcnc.org/linuxcnc-install.sh`
    b. Make it executable:
       `chmod +x linuxcnc-install.sh`
    c. Install linuxcnc and its environment:
       `sudo ./linuxcnc-install.sh`
    d. You should now find LinuxCNC in your menus.
13. On Debian 13 simply install from repositories
    a. `sudo apt install linuxcnc-uspace linuxcnc-uspace-dev`
    b. Go back to 11.d and remove the non-RT kernel if desired. Then continue to 14.
14. You are using mesa hardware, type:
    `sudo apt install mesaflash`
15. LinuxCNC should be in the menus or open a terminal and type:
    `linuxcnc`
    and run an axis sim to check its working
16. From the menu settings/Power manager set the power settings to suit your needs. You can turn off screen saver and screen lock here.
17. Install geany and grub-customizer:
    `sudo apt install geany grub-customizer`
18. Finally now geany is installed, enable auto login
    `sudo geany /etc/lightdm/lightdm.conf`
19. Scroll down to about line 126 and uncomment (remove #) both of the following lines and add YOUR login user name. An example for user matt:
    `autologin-user=matt`
    `autologin-user-timeout=0`

Optional steps:

Note that isolcpus can make a huge difference to latency on some systems because it isolates specific CPU cores so they are purely used by real time threads (e.g. the linuxcnc servo thread). The instructions below assume a 4 core CPU, e.g., Celeron, i3, i5, etc.) those with 2 cores or more than 4 cores need different isolcpus settings. Never isolate core 0 as it is used for system threads so it already includes a lot of running threads.

20. Now we need to isolate 2 cores for better RT performance:
    `sudo grub-customizer`

21. On the General Settings in the kernel parameters field, change it from:
    `quiet`
    to:
    `quiet isolcpus=2,3`
22. Disable energy efficient ethernet. Add this to the field above
    `igb.EEE=0` or `r8168.EEE=0` depending on your ethernet NIC
23. Disable Active State Power management. Add this to the field above
    `pcie_aspm=off`
24. Save the config, close grub-customiser and reboot for changes to take effect.
25. Check latency with:
    `latency-histogram --nobase --sbins=1000`
    It should be much improved.

Set fixed ip address - only for mesa card.

Usually we set up the mesa card to have the ip address 10.10.10.10. We need to set a fixed ip address of 10.10.10.1 to the network interface that connects to it.

26. Type
    `ip a`
    to determine the network interface name used for your mesa card. This is usually something like **eth0** or **enp2s0**
27. Type
    `sudo geany /etc/network/interfaces`
    to append the following at the end of the file:
    ```
    auto enp2s0
    iface enp2s0 inet static
    address 10.10.10.1
    hardware-irq-coalesce-rx-usecs 0
    ```
    Note, the last line is only required for Intel network cards. It seems to be ignored on non-applicable hardware.
28. Save and close geany.
29. Reboot to restart the network or type:
    `sudo systemctl restart networking`
30. Ping the mesa card to confirm it's all working
    `ping 10.10.10.10`

Review latency

31. Use latency-histogram instead of latency-test to review latency:
    `latency-histogram --nobase --sbins 1000`
    How to evaluate latency is covered in the linuxcnc documents
32. Among other things, latency is affected by:
    a. BIOS settings
    b. Isolcpus and other boot time settings
    c. Kernel version used

Review network latency

This section primarily is written for mesa ethernet connected devices but it also applies to other similar devices such as Remora, EC500, Ethercat etc. Note the `show param` settings only apply to devices connected using the hm2_eth component.

Network latency can also affect communication with ethercat slaves and all of the optimisations discussed here will be helpful. However, In our experience, the ethercat protocol is less affected https://drive.google.com/file/d/1LMwTa6jREfzxM2ZdunWz1DAwHz-TxWVZ/view?usp=sharing hm2_eth devices.

If network latency is excessive, the ethercat device may not have enough time to complete its work before the servo cycle ends. If this happens, you will receive an error finishing read error and the device will be disabled. This appears to be more prevalent with Realtek network devices. You can review network latency by looking at some linuxcnc parameters while linuxcnc is running.

33. In a new terminal console type:
    `halcmd show param hm2*tmax*`
    This will list the maximum time spent for network reads and writes in timer ticks. You can also display these parameters using halshow now you know the names.
34. You can convert timer ticks to usec (1000 usec per servo thread cycle) by dividing by the cpu frequency reported by:
    `latency-histogram --nobase --sbins 1000`
35. You can also use ping to get an idea of network latency without linuxcnc running. PCW says, To test ping times, you must run the test at a real time priority, and run the test for a long time, something like:
    `sudo :chrt 99 ping -i .001 -q 10.10.10.10`

36. Following these tests, we determined one PC was spending up to 80% of a servo cycle reading and writing on the network leaving little time to get real work done.
37. Improving network latency with some realtek cards is discussed under a later heading.

CPU Affinity and other optimisations

Linuxcnc's Real time servo thread runs on a single CPU core. The Interrupt (IRQ) that controls the network card (NIC) that talks to the Mesa card (or other Ethercat device) runs on a separate non-realtime core. This means that latency and delays can occur. The solution is to use CPU affinity settings that force the NIC IRQ to run on the same core as the servo thread where they can buddy up for better performance.

Refer to this forum thread. It contains a script in the first post to set the affinity and a number of other optimisations:
https://forum.linuxcnc.org/38-general-linuxcnc-questions/51647-latency-error-finishing-read-and-irq-affinity

Installing a later kernel

Since the release of Debian Bullseye (Linux kernel 5.10), Real time performance has been disappointing. In particular, network latency when communicating with a Mesa ethernet card (or similar device) has been generating ***Error Finishing Read*** *Errors. This means that the network latency left insufficient time for the servo thread cycle to complete in time.*
This appears to have been more prevalent with Realtek Network interfaces. Fortunately, each iteration of the Linux kernel has improved results, particularly since the release of 6.x kernels. Debian Bookworm (Debian 12) is using the 6.1 kernel which is not too bad. In our testing, we found that latency improved by 265% if we used the 6.3 kernel. We have compiled this version of the kernel for your convenience. Updated to the final 6.3 kernel on 1 May 2023.

Don't use this kernel with Trixie because it uses the much later 6.12 kernel

Only try installing it if you have exhausted all options by following the steps below:

38. Download the 2 deb files (image, source) from

https://drive.google.com/drive/folders/1NzQIHnf9M_cHzuZCqSldVFGschOOxaER?usp=sharing

The link above is to my latest kernel versions following the final release of 6.3 kernel and the matching preempt_rt patches. ***Note: This now is an older version of the kernel.***

39. Navigate to the Downloads folder and open a new Terminal session.
40. Install the debs as follows (pressing tab auto completes the command):

```
dpkg -i linux-source(tab)
dpkg -i linux-image(tab)
```

41. Reboot into the new kernel.
42. Check that

```
uname -v
```

shows the 6.3 kernel is installed.

43. If it isn't, use grub-customizer mentioned earlier to change the kernel boot order and reboot again.

Realtek network drivers

Some users have been reporting significant error finishing read issues with some Realtek NIC's.

There are two additional device drivers available in Debian for realtek cards;

r8125-dkms for 2.5 Gb network cards - RTL8125, RTL8125B(S)(G)

r8168-dkms for the following network cards RTL8111B/RTL8111C, RTL8111D/RTL8111E, RTL8111F/RTL8111G(S), RTL8111H(S), RTL8118(A)(S), RTL8119i, RTL8111L, RTL8111K, RTL8168B, RTL8168E, RTL8168H, RTL8111DP, RTL8111EP, RTL8111FP, RTL8411/RTL8411B, RTL8101E, RTL8102E, RTL8103E, RTL8105E, RTL8106E, RTL8107E, RTL8401, RTL8402

Installing the r8168-dkms driver improved network latency by 400% on our R8111 network card. Similar results were reported on other affected hardware.

For more information about this issue, read this forum sticky post: https://forum.linuxcnc.org/9-installing-linuxcnc/47696-installing-linuxcnc-and-debian-bookworm-on-problematic-hardware-eg-realtek-nic .

r8168-dkms and r8125-dkms drivers are in the non-free packages which are not included by default.

44. You can see your driver if you type the following to identify your NIC name:
```
ip a
```
45. Now display the NIC info, e.g.:
```
ethtool enps02
```
46. If it seems you could benefit from this driver, continue.
47. Type
```
sudo geany /etc/apt/sources.list
```
48. Add a space followed by **non-free** to each of the 4 lines that end with **firmware-non-free** as follows:
```
deb http://deb.debian.org/debian/ bookworm main non-free-firmware non-free
deb-src http://deb.debian.org/debian/ bookworm main non-free-firmware non-free
deb http://security.debian.org/debian-security bookworm-security main
non-free-firmware non-free

deb-src http://security.debian.org/debian-security bookworm-security main
non-free-firmware non-free
```
49. Save and close geany
50. Type:
```
sudo apt update
```
51. you need to install some utilities. Type:
```
sudo apt install build-essential dkms
```
52. If you have not installed a later kernel as described above install linux-headers. Type:
```
sudo apt install linux-headers-$(uname -r)
```
53. You can now install the r8168 or R8125 driver. Depending on your driver, either type
```
sudo apt install r8168-dkms
```
or
```
sudo apt install r8125-dkms
```
54. Reboot.
55. Check you still have a network driver:
```
ip a
```
56. Check you can still ping the mesa card:
```
ping 10.10.10.10
```
57. If you have to remove this driver, it needs to be purged completely or you will have no network. E.g.:
```
sudo apt purge r8168-dkms
```
58. Optional kernel settings (install using grub-customizer as per `isolcpus` setting)
```
r8168.aspm=0 r8168.eee_enable=0 pcie_aspm=off loglevel=3
```
I did not test them individually to see which helped but I suspect that disabling the "Energy Efficient Ethernet" mode might be a good culprit.

EtherCAT

EtherCAT is a fieldbus technology to control servo and stepper drives as well as other I/O boards etc. It interfaces with LinuxCNC but adds an additional layer of complexity to your configuration.
Note the optimisations for network latency discussed in this document are also helpful for an ethercat environment,

59. Install EtherCAT if required by following these instructions: https://forum.linuxcnc.org/ethercat/45336-ethercat-installation-from-repositories-how-to-step-by-step?start=0

Spindle Control via Modbus (RS485)

An alternative to controlling your spindle instead of a 0-10 volt potentiometer is to use a RS485 interface that exists on most variable frequency drives (VFDs). This requires a RS485 interface or usb dongle on your PC. There are several linuxcnc modules for controlling various VFD's. I use vfdmod and a BD600 vfd.

60. Install vfdmod to control spindle VFD with RS485 if required: https://forum.linuxcnc.org/24-hal-components/38733-vfdmod-an-easy-vfd-control-over-modbus-rtu
    Start by downloading and installing the debs with gdebi or dpkg -i then follow the wiki instructions
61. Settings for BD600 are here later in that thread: https://forum.linuxcnc.org/24-hal-components/38733-vfdmod-an-easy-vfd-control-over-modbus-rtu?start=80#236664

QtPyVCP, Probe Basic and EtherCAT

We no longer provide links to the QtPyVCP project as it is not part of LinuxCNC. However, the official LinuxCNC ISO installs the repositories for it and for EtherCAT. Some brief descriptions can be found in your home folder.

For a more integrated GUI with similar probing features, we recommend and prefer QtDragon which is included in the LinuxCNC distribution and has good (and improving) documentation. It has the advantage of supporting multiple screen sizes. QtDragon is a new GUI and may have bugs but they are quickly resolved if reported on the LinuxCNC QtVCP forum section.
Link to documentation: https://linuxcnc.org/docs/2.9/html/gui/qtdragon.html .

Building Linuxcnc 2.9.4 or master on Bookworm/Trixie from source

LinuxCNC V 2.9.2 is just a snapshot loaded from time to time. To upgrade to the latest version, you need to build from source. I have a video of how to do this: https://youtu.be/Oe4Hfq3G2_A . It's not that difficult and it is well documented here: https://linuxcnc.org/docs/2.9/html/code/building-linuxcnc.html .

62. Proceed as follows:
```
sudo apt install git dpkg-dev build-essentials
cd ~
git clone https://github.com/LinuxCNC/linuxcnc.git
linuxcnc-dev
cd ~/linuxcnc-dev
git checkout v2.9.4
```
Note that if you wish to build master branch, skip the last line above
63. We now have the source so we need to make a deb file Type::
```
cd ~/linuxcnc-dev/debian
./configure uspace no-docs
```
This will list all of the missing dependencies.

64. Copy this list into a text editor. Edit it to remove any brackets, version information etc. Copy the list to the clipboard.

65. Type the following and paste the list at the end of the line below
```
sudo apt install -y <<your list>>
cd ..
```

66. Once installed, repeat the configure command again and if all is well, continue otherwise install the missing ones until its clean.
67. So when done, continue (this will take a while). Type
```
dpkg-buildpackage -b -uc
```
68. Occasionally, a missing dependency is reported when this is running, install it with
```
sudo apt install -y
```
and repeat Step 2. At this point, your home folder should contain several .deb files. We can ignore linuxcnc-uspace-dev for now as it won't have any significant changes very often.
69. We now need to uninstall your existing deb back in step 11 and reinstall the new ones
```
sudo apt remove linuxcnc-uspace
cd ~
```

```
ls
dpkg -i linuxcnc-uspace_(tab)
```
That's it you have upgraded.

If you need to update LinuxCNC again you can just
```
cd ~/linuxcnc/src
git pull
```
And repeat the process starting again at Step 60 above.

No Microphone?

```
sudo apt install wireplumber
systemctl --user --now enable wireplumber.service
```

Ref: https://forums.debian.net/viewtopic.php?t=153430