

**Getting Started**  
**V2.9.0~pre1+git20230208.f1270d6ed7, 28**  
**Jun 2026**

# Índice general

<b>1. About LinuxCNC</b>	<b>1</b>
1.1. The Software	1
1.2. The Operating System	2
1.3. Getting Help	2
1.3.1. IRC	2
1.3.2. Mailing List	2
1.3.3. Web Forum	3
1.3.4. LinuxCNC Wiki	3
1.3.5. Bug Reports	3
<b>2. System Requirements</b>	<b>4</b>
2.1. Minimum Requirements	4
2.2. Kernel and Version requirements	4
2.2.1. Preempt-RT with <i>linuxcnc-ospace</i> package	5
2.2.2. RTAI with <i>linuxcnc</i> package	5
2.2.3. Xenomai with <i>linuxcnc-ospace</i> package	5
2.2.4. RTAI with <i>linuxcnc-ospace</i> package	5
2.3. Problematic Hardware	5
2.3.1. Laptops	5
2.3.2. Video Cards	5
<b>3. Getting LinuxCNC</b>	<b>6</b>
3.1. Download the image	6
3.1.1. Normal Download	6
3.1.2. Download using <i>zsync</i>	7
3.1.3. Verify the image	7
3.2. Write the image to a bootable device	8
3.3. Testing LinuxCNC	9
3.4. Installing LinuxCNC	9
3.5. Updates to LinuxCNC	9

3.6. Install Problems . . . . .	10
3.7. Alternate Install Methods . . . . .	10
3.7.1. Installing on Debian Buster (with Preempt-RT kernel) . . . . .	11
3.7.2. Installing on Debian Buster (with experimental RTAI kernel) . . . . .	12
3.7.3. Installing on Raspbian 10 . . . . .	13
3.7.4. Installing on Ubuntu Precise . . . . .	13
<b>4. Running LinuxCNC</b>	<b>15</b>
4.1. Invoking LinuxCNC . . . . .	15
4.2. Configuration Launcher . . . . .	15
4.3. Next steps in configuration . . . . .	18
4.4. Simulator Configurations . . . . .	18
4.5. Configuration Resources . . . . .	18
<b>5. Updating LinuxCNC</b>	<b>19</b>
5.1. Upgrade to the new version . . . . .	19
5.1.1. Apt Sources Configuration . . . . .	20
5.1.2. Upgrading to the new version . . . . .	21
5.1.2.1. Debian Wheezy & Stretch and Ubuntu Lucid . . . . .	21
5.1.3. Ubuntu Precise . . . . .	21
5.2. Updating without Network . . . . .	22
5.3. Updating Configuration Files for 2.9 . . . . .	22
5.3.1. Stricter handling of pluggable interpreters . . . . .	22
5.3.2. Canterp . . . . .	22
5.4. Updating Configuration Files (for 2.8.x) . . . . .	22
5.4.1. Distribution Configurations (updates for joints_axes) . . . . .	23
5.4.2. Automatic updates (update_ini script for joints_axes) . . . . .	23
5.4.3. Multiple Spindle Support . . . . .	23
5.4.4. TRAJ velocities, accelerations names . . . . .	24
5.4.5. Kinematics modules . . . . .	24
5.4.6. Lathe Configurations . . . . .	25
5.4.7. Consistent Joints/Axes specifications . . . . .	25
5.4.8. Home sequences . . . . .	26
5.4.9. Locking rotary indexer (updates for joints_axes) . . . . .	26
5.4.10 Stricter INI file syntax . . . . .	27
5.4.11 [TRAJ] settings . . . . .	28
5.5. HAL Changes (updates for joints_axes 2.8.x) . . . . .	28
5.5.1. Wheel or MPG (manual pulse generator) jogging . . . . .	28
5.5.2. INI HAL pins . . . . .	29

5.6. HAL Changes (Other 2.8.x)	29
5.6.1. halcompile	29
5.6.2. Parameter to Pin changes	30
5.7. Interface changes for joints_axes 2.8.x	30
5.7.1. Python LinuxCNC module	30
5.8. GUIs (updates for joints_axes 2.8.x)	30
5.8.1. Notes on joint/axis jogging, homing, and kinematics	30
5.8.2. Halui	31
5.8.2.1. TELEOP jogging (also called axis or world jogging)	31
5.8.2.2. Joint jogging	31
5.8.2.3. Additional pin renames	32
5.8.3. AXIS GUI	32
5.8.3.1. Identity Kinematics	32
5.8.3.2. Special case kinematics	32
5.8.3.3. Non-identity kinematics	32
5.8.3.4. Home icons	33
5.8.3.5. Limit icons	33
5.8.3.6. Key bindings for a fourth axis	33
5.8.4. TkLinuxCNC	33
5.8.4.1. emcsh commands	33
5.8.5. Touchy	34
5.8.6. Gscreen	34
5.8.7. GMOCCAPY	34
5.8.8. shuttlexpress driver renamed to shuttle	34
5.8.9. linuxcncrsh	34
5.9. Tools	35
5.9.1. Calibration (emccalib.tcl)	35
5.10 Obsolete Guis (removed for 2.8.x)	35
5.11 Deprecated Guis (marked at 2.8.x)	35
5.12 Simulator configurations (updates for joints axes 2.8.x)	35
5.12.1 Pre-joints_axes	35
5.12.2 Post-joints_axes	36
5.12.2.1 Equivalent HAL commands file	36
5.12.2.2 Notes	37
5.13 Miscellaneous updates for 2.8.x	37
5.13.1 Motion pins	37
5.13.2 HAL pins	38
5.13.3 HAL component updates	38
5.13.4 XHC-HB04 Pendant Support	38

5.13.4.1xhc_hb04_util.comp (helper component) . . . . .	38
5.13.4.2xhc_hb04.tcl (optional LIB configuration HAL file) . . . . .	38
5.13.5XHC-WHB04B-6 pendant support . . . . .	38
5.13.6bldc3_hall . . . . .	38
5.13.7[JOINT_n]HOME_SEQUENCE Starting values . . . . .	38
5.13.8[JOINT_n]HOME_SEQUENCE Negative values . . . . .	39
5.13.9TWOPASS support for complex loadrt config= items . . . . .	39
5.14Changes beyond 2.8.x (master branch development) . . . . .	39
5.14.1Python3 and GTK3 . . . . .	39
5.14.2LinuxCNC Startup . . . . .	39
5.14.3G-code changes . . . . .	39
5.14.4Configuration Updates . . . . .	39
5.14.4.1INI file Settings . . . . .	39
5.14.5Code Updates . . . . .	40
5.14.5.1Reverse Run . . . . .	40
5.14.5.2Number of Joints . . . . .	40
5.14.5.3Extra Joints . . . . .	40
5.14.5.4Homing . . . . .	40
5.14.5.5Motion . . . . .	40
5.14.5.6Switchkins Kinematics Modules . . . . .	41
5.14.5.7Trajectory Planner . . . . .	41
5.14.5.8Homing . . . . .	41
5.14.5.9Other . . . . .	41
5.14.6HAL . . . . .	41
5.14.6.1Components . . . . .	41
5.14.6.2Motion module pins . . . . .	42
5.14.7Configs . . . . .	42
5.14.7.1INI file . . . . .	42
5.14.7.2Simulation Configs . . . . .	42
5.15Changes after 2.8.x . . . . .	42
<b>6. Glossary</b>	<b>43</b>
<b>7. Legal Section</b>	<b>49</b>
7.1. Copyright Terms . . . . .	49
7.2. GNU Free Documentation License . . . . .	49
<b>8. Índice alfabético</b>	<b>54</b>

The LinuxCNC Team



This handbook is a work in progress. If you are able to help with writing, editing, or graphic preparation please contact any member of the writing team or join and send an email to [emc-users@lists.sourceforge.net](mailto:emc-users@lists.sourceforge.net)

Copyright © 2000-2020 LinuxCNC.org

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you do not find the license you may order a copy from:

Free Software Foundation, Inc.  
51 Franklin Street  
Fifth Floor  
Boston, MA 02110-1301 USA.

(The English language version is authoritative)

LINUX® is the registered trademark of Linus Torvalds in the U.S. and other countries. The registered trademark Linux® is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

The LinuxCNC project is not affiliated with Debian®. *Debian* is a registered trademark owned by Software in the Public Interest, Inc.

The LinuxCNC project is not affiliated with UBUNTU®. *UBUNTU* is a registered trademark owned by Canonical Limited.

---

# Capítulo 1

## About LinuxCNC

### 1.1. The Software

- LinuxCNC (the Enhanced Machine Control) is a software system for computer control of machine tools such as milling machines and lathes, robots such as puma and scara and other computer controlled machines up to 9 axes.
  - LinuxCNC is free software with open source code. Current versions of LinuxCNC are entirely licensed under the GNU General Public License and Lesser GNU General Public License (GPL and LGPL)
  - LinuxCNC provides:
    - easy discovery and testing without installation with the LiveCD
    - easy installation from the Live CD
    - easy to use graphical configuration wizards to rapidly create a configuration specific to the machine
    - directly available as regular packages of recent releases of Debian (since Bookworm) and Ubuntu (since Kinetic Kudu)
    - a graphical user interface (actually several interfaces to choose from)
    - a graphical interface creation tool (Glade)
    - an interpreter for *G-code* (the RS-274 machine tool programming language)
    - a realtime motion planning system with look-ahead
    - operation of low-level machine electronics such as sensors and motor drives
    - an easy to use *breadboard* layer for quickly creating a unique configuration for your machine
    - a software PLC programmable with ladder diagrams
  - It does not provide drawing (CAD - Computer Aided Design) or G-code generation from the drawing (CAM - Computer Automated Manufacturing) functions.
  - It can simultaneously move up to 9 axes and supports a variety of interfaces.
  - The control can operate true servos (analog or PWM) with the feedback loop closed by the LinuxCNC software at the computer, or open loop with step-servos or stepper motors.
  - Motion control features include: cutter radius and length compensation, path deviation limited to a specified tolerance, lathe threading, synchronized axis motion, adaptive feedrate, operator feed override, and constant velocity control.
-

- Support for non-Cartesian motion systems is provided via custom kinematics modules. Available architectures include hexapods (Stewart platforms and similar concepts) and systems with rotary joints to provide motion such as PUMA or SCARA robots.
- LinuxCNC runs on Linux using real time extensions.

## 1.2. The Operating System

LinuxCNC is available as ready-to-use packages for the Ubuntu and Debian distributions.

## 1.3. Getting Help

### 1.3.1. IRC

IRC stands for Internet Relay Chat. It is a live connection to other LinuxCNC users. The LinuxCNC IRC channel is #linuxcnc on libera.chat.

The simplest way to get on the IRC is to use the embedded client on this [page](#).

#### Some IRC etiquette

- Ask specific questions... Avoid questions like "Can someone help me?".
- If you're really new to all this, think a bit about your question before typing it. Make sure you give enough information so someone can solve your question.
- Have some patience when waiting for an answer, sometimes it takes a while to formulate an answer or everyone might be busy working or something.
- Set up your IRC account with your unique name so people will know who you are. If you use the java client, use the same name every time you log in. This helps people remember who you are and if you have been on before many will remember the past discussions which saves time on both ends.

#### Sharing Files

The most common way to share files on the IRC is to upload the file to one of the following or a similar service and paste the link:

- *For text:* <https://pastebin.com/>, <http://pastie.org/>, <https://gist.github.com/>
- *For pictures:* <https://imagebin.org/>, <https://imgur.com/>, <https://bayimg.com/>
- *For files:* <https://filedropper.com/>, <https://filefactory.com/>, <https://1fichier.com/>

### 1.3.2. Mailing List

An Internet Mailing List is a way to put questions out for everyone on that list to see and answer at their convenience. You get better exposure to your questions on a mailing list than on the IRC but answers take longer. In a nutshell you e-mail a message to the list and either get daily digests or individual replies back depending on how you set up your account.

You can subscribe to the emc-users mailing list at: <https://lists.sourceforge.net/lists/listinfo/emc-users>

---



### **1.3.3. Web Forum**

A web forum can be found at <https://forum.linuxcnc.org> or by following the link at the top of the linuxcnc.org home page.

This is quite active but the demographic is more user-biased than the mailing list. If you want to be sure that your message is seen by the developers then the mailing list is to be preferred.

### **1.3.4. LinuxCNC Wiki**

A Wiki site is a user maintained web site that anyone can add to or edit.

The user maintained LinuxCNC Wiki site contains a wealth of information and tips at: <https://wiki.linuxcnc.org>

### **1.3.5. Bug Reports**

Report bugs to the LinuxCNC [github bug tracker](#).

---

## Capítulo 2

# System Requirements

### 2.1. Minimum Requirements

The minimum system to run LinuxCNC and Debian / Ubuntu may vary depending on the exact usage. Stepper systems in general require faster threads to generate step pulses than servo systems. You can use the Live-CD to test the software before committing to a permanent installation on a computer. Keep in mind that the Latency Test numbers are more important than the processor speed for software step generation. More information on the Latency Test is [here](#). In addition LinuxCNC needs to be run on an operating system that uses a specially modified kernel, see [Kernel and Version Requirements](#).

Additional information is on the LinuxCNC Wiki site: [Hardware Requirements](#)

LinuxCNC and Debian Linux should run reasonably well on a computer with the following minimum hardware specification. These numbers are not the absolute minimum but will give reasonable performance for most stepper systems.

- 700 MHz x86 processor (1.2 GHz x86 processor recommended) or Raspberry Pi 4 or better.
- To run LinuxCNC 2.8 and Debian Buster from the LiveCD the system should be 64-bit capable.
- 512 MB or more of RAM
- 8 GB hard disk
- Graphics card capable of at least 1024x768 resolution, which is not using the NVidia or ATI fglrx proprietary drivers. Modern onboard graphic chipsets seem to generally be OK.
- A network or Internet connection (not strictly needed, but very useful for updates and for communicating with the LinuxCNC community)

Minimum hardware requirements change as Linux distributions evolve so check the [Debian](#) web site for details on the LiveCD you're using. Older hardware may benefit from selecting an older version of the LiveCD when available.

### 2.2. Kernel and Version requirements

LinuxCNC requires a kernel modified for realtime use to control real machine hardware. It can, however run on a standard kernel in simulation mode for purposes such as checking G-code, testing config files and learning the system. To work with these kernel versions there are two versions of LinuxCNC distributed. The package names are "linuxcnc" and "linuxcnc-uspace".

The realtime kernel options are preempt-rt, RTAI and Xenomai.

You can discover the kernel version of your system with the command:

```
uname -a
```

If you see (as above) `-rt-` in the kernel name then you are running the `preempt-rt` kernel and should install the `"uspace"` version of LinuxCNC. You should also install `uspace` for `"sim"` configs on non-realtime kernels

If you see `-rtai-` in the kernel name then you are running RTAI realtime. See below for the LinuxCNC version to install.

### 2.2.1. Preempt-RT with *linuxcnc-uspace* package

Preempt-RT is the newest of the realtime systems, and is also the version that is closest to a mainline kernel. Preempt-RT kernels are available as precompiled packages from the main repositories. The search term `"PREEMPT_RT"` will find them, and one can be downloaded and installed just like any other package. Preempt-RT will generally have the best driver support and is the only option for systems using the Mesa ethernet-connected hardware driver cards. In general `preempt-rt` has the worst latency of the available systems, but there are exceptions.

### 2.2.2. RTAI with *linuxcnc* package

RTAI has been the mainstay of LinuxCNC distributions for many years. It will generally give the best realtime performance in terms of low latency, but might have poorer peripheral support and not as many screen resolutions. An RTAI kernel is available from the LinuxCNC package repository. If you installed from the Live/Install image then switching kernel and LinuxCNC flavour is described in [Installing-RTAI].

### 2.2.3. Xenomai with *linuxcnc-uspace* package

Xenomai is also supported, but you will have to find or build the kernel and compile LinuxCNC from source to utilise it.

### 2.2.4. RTAI with *linuxcnc-uspace* package

It is also possible to run LinuxCNC with RTAI in user-space mode. As with Xenomai you will need to compile from source to do this.

## 2.3. Problematic Hardware

### 2.3.1. Laptops

Laptops are not generally suited to real time software step generation. Again a Latency Test run for an extended time will give you the info you need to determine suitability.

### 2.3.2. Video Cards

If your installation pops up with 800 x 600 screen resolution then most likely Debian does not recognize your video card or monitor. This can sometimes be worked-around by installing drivers or creating / editing `Xorg.conf` files.

---

## Capítulo 3

# Getting LinuxCNC

This section describes the recommended way to download and make a fresh install of LinuxCNC. There are also [Alternate Install Methods](#) for the adventurous. If you have an existing install that you want to upgrade, go to the [Updating LinuxCNC](#) section instead.

---

**nota**

LinuxCNC requires a special kernel with real-time extensions. There are three possibilities here: preempt-rt, RTAI or Xenomai. In addition there are two versions of LinuxCNC which work with these kernels. See the table below for details.

---

Fresh installs of LinuxCNC are most easily created using the Live/Install Image. This is a hybrid ISO filesystem image that can be written to a USB storage device or a DVD and used to boot a computer. At boot time you will be given a choice of booting the "Live" system (to run LinuxCNC without making any permanent changes to your computer) or booting the Installer (to install LinuxCNC and its operating system onto your computer's hard drive).

The outline of the process looks like this:

1. Download the Live/Install Image.
2. Write the image to a USB storage device or DVD.
3. Boot the Live system to test out LinuxCNC.
4. Boot the Installer to install LinuxCNC.

### 3.1. Download the image

This section describes some methods for downloading the Live/Install Image.

#### 3.1.1. Normal Download

For x86 PCs Download the Live/Install CD by clicking here:

<https://linuxcnc.org/iso/linuxcnc-2.8.4-buster.iso>

For the Raspberry Pi a complete SD card image is available here:

<https://linuxcnc.org/iso/linuxcnc-2.8.1-pi4.zip> (this will auto-update to 2.8.4)

---

This can be installed using the normal Pi [install process](#) including with the Raspberry Pi Imager app.

This SD image is reported not to work with the Raspberry Pi4 8GB model. Note also that this version of the SD image limits available memory to 3GB as this is necessary to persuade it to run with both WiFi and USB working on some versions of the Pi. You can experiment with removing this limit by editing the config-rt.txt file in the boot directory. If you can't boot after the change then the file can be edited back by mounting the SD card in a different computer (maybe even a Pi with a USB card reader).

### 3.1.2. Download using zsync

zsync is a download application that efficiently resumes interrupted downloads and efficiently transfers large files with small modifications (if you have an older local copy). Use zsync if you have trouble downloading the image using the [Normal Download](#) method.

zsync in Linux

- Install zsync using Synaptic or, by running the following in a [terminal](#)

```
sudo apt-get install zsync
```

- Then run this command to download the iso to your computer

```
zsync https://linuxcnc.org/iso/linuxcnc-2.8.4-buster.iso
```

or

```
zsync https://linuxcnc.org/iso/linuxcnc-2.8.1-pi4.zip.zsync
```

**zsync in Windows** There is a Windows port of zsync. It works as a console application. It can be downloaded from:

<https://www.assembla.com/spaces/zsync-windows/documents>

### 3.1.3. Verify the image

(This step is unnecessary if you used zsync)

1. After downloading, verify the checksum of the image to ensure integrity.

```
md5sum linuxcnc-2.8.4-buster.iso
```

or

```
sha256sum linuxcnc-2.8.4-buster.iso
```

2. Then compare to these checksums

```
md5sum: 8a6e6abd2c792c3e06fbee0ed049ed41
sha256sum: 0bfeac3ddfe1bdbf5ca4dad84eeec165741d3f253a16b75e4405c06b7b489700
```

**Verify md5sum on Windows or Mac** Windows and Mac OS X do not come with an md5sum program, but there are alternatives. More information can be found at: [How To MD5SUM](#)

## 3.2. Write the image to a bootable device

The Raspberry Pi image is a complete SD card image and should be written to an SD card in [the normal way](#).

The LinuxCNC Live/Install ISO Image is a hybrid ISO image which can be written directly to a USB storage device (flash drive) or a DVD and used to boot a computer. The image is too large to fit on a CD.

Writing the image to a USB storage device in Linux

1. Connect a USB storage device (for example a flash drive or thumb drive type device).
2. Determine the device file corresponding to the USB flash drive. This information can be found in the output of `dmesg` after connecting the device. `/proc/partitions` may also be helpful.
3. Use the `dd` command to write the image to your USB storage device. For example, if your storage device showed up as `/dev/sde`, then use this command:

```
dd if=linuxcnc-2.8.4-buster.iso of=/dev/sde
```

Writing the image to a USB storage device in Mac OSX

1. Open a terminal and type

```
diskutil list
```

2. Insert the USB and note the name of the new disk that appears, eg `/dev/disk5`
3. unmount the USB. The number found above should be substituted in place of the N

```
diskutil unmountDisk /dev/diskN
```

4. Transfer the data with `dd`, as for Linux above. Note that the disk name has an added "r" at the beginning

```
sudo dd if=/path-to.iso of=/dev/rdiskN bs=1m
```

5. Note that this may take a long time to complete and there will be no feedback during the process.

Writing the image to a DVD in Linux

1. Insert a blank DVD into your burner. A *CD/DVD Creator* or *Choose Disc Type* window will pop up. Close this, as we will not be using it.
2. Browse to the downloaded image in the file browser.
3. Right click on the ISO image file and choose Write to Disc.
4. Select the write speed. It is recommended that you write at the lowest possible speed.
5. Start the burning process.
6. If a *choose a file name for the disc image* window pops up, just pick OK.

Writing the image to a DVD in Windows

1. Download and install Infra Recorder, a free and open source image burning program: <http://infrarecorder.org>

2. Insert a blank CD in the drive and select Do nothing or Cancel if an auto-run dialog pops up.
3. Open Infra Recorder, and select the *Actions* menu, then *Burn image*.

Writing the image to a DVD in Mac OSX

1. Download the .iso file
2. Right-click on the file in the Finder window and select "Burn to disc" (The option to burn to disc will only appear if the machine has an optical drive fitted or connected)

### 3.3. Testing LinuxCNC

With the USB storage device plugged in or the DVD in the DVD drive, shut down the computer then turn the computer back on. This will boot the computer from the Live/Install Image and choose the Live boot option.

---

**nota**

If the system does not boot from the DVD or USB stick it might be necessary to change the boot order in the PC BIOS.

---

Once the computer has booted up you can try out LinuxCNC without installing it. You can not create custom configurations or modify most system settings in a Live session, but you can (and should) run the latency test.

To try out LinuxCNC: from the Applications/CNC menu pick LinuxCNC. A dialog box will open from which you can choose one of many sample configurations. At this point it only really makes sense to pick a "sim" configuration. Some of the sample configurations include onscreen 3D simulated machines, look for "Vismach" to see these.

To see if your computer is suitable for software step pulse generation run the Latency Test as shown [here](#).

At the time of writing the Live-Image is only available with the preempt-rt kernel and a matching LinuxCNC. On some hardware this might not offer good enough latency. There is an experimental version available using the RTAI realtime kernel which will often give better latency.

### 3.4. Installing LinuxCNC

To install LinuxCNC from the LiveCD select *Install (Graphical)* at bootup.

### 3.5. Updates to LinuxCNC

With the normal install the Update Manager will notify you of updates to LinuxCNC when you go online and allow you to easily upgrade with no Linux knowledge needed. It is OK to upgrade everything except the operating system when asked to.

**aviso**

Do not upgrade the operating system if prompted to do so. You should accept OS *updates* however, especially security updates.

---

## 3.6. Install Problems

In rare cases you might have to reset the BIOS to default settings if during the Live CD install it cannot recognize the hard drive during the boot up.

## 3.7. Alternate Install Methods

The easiest, preferred way to install LinuxCNC is to use the Live/Install Image as described above. That method is as simple and reliable as we can make it, and is suitable for novice users and experienced users alike. However, this will typically replace any existing operating system.

In addition, for experienced users who are familiar with Debian system administration (finding install images, manipulating apt sources, changing kernel flavors, etc), new installs are supported on following platforms: ("amd64" means "64-bit", and is not specific to AMD processors, it will run on any 64-bit x86 system)

Distribution	Architecture	Kernel	Package name	Typical use
Debian Buster	amd64 & i386	Stock	linuxcnc-ospace	simulation only
Debian Buster	amd64 & armhf	preemp-rt	linuxcnc-ospace	machine control & simulation
Debian Buster	amd64	RTAI	linuxcnc	machine control (known issues)
Debian Jessie	amd64 & i386	Stock	linuxcnc-ospace	simulation only
Debian Wheezy	i386	RTAI	linuxcnc	machine control & simulation
Debian Wheezy	amd64 & i386	Preempt-RT	linuxcnc-ospace	machine control & simulation
Debian Wheezy	amd64 & i386	Stock	linuxcnc-ospace	simulation only
Ubuntu Precise	i386	RTAI	linuxcnc	machine control & simulation
Ubuntu Precise	amd64 & i386	Stock	linuxcnc-ospace	simulation only

### nota

LinuxCNC v2.8 is not supported on Ubuntu Lucid or older.

**Preempt-RT kernels** The Preempt-rt kernels are available for Debian from the regular debian.org archive. The preempt-rt kernel for RaspBerry Pi is available from the LinuxCNC repository. The package is called `linux-image-rt-*`. Simply install the package in the same way as any other package from the Synaptic Package manager or with `apt-get` at the command-line.

**RTAI Kernels** The RTAI kernels are available for download from the linuxcnc.org debian archive. The apt source is:

- Debian Buster: deb <https://linuxcnc.org> buster base
- Debian Wheezy: deb <https://linuxcnc.org> wheezy base
- Ubuntu Precise: deb <https://linuxcnc.org> precise base

### nota

Debian Wheezy and Ubuntu Precise are both extremely old, and are out of their support period. It is strongly advised not to use either for a new install, and to seriously consider upgrading an existing installation.



The Buster / RTAI package is only available on amd64, but there are very few surviving systems that can not run a 64-bit OS.

**aviso**

There are known issues with the 64-bit RTAI 5.2 kernel with this version of LinuxCNC. The system will occasionally lock solid. However, this has, so far, been seen only during system exit. While running the system appears to be stable. But should nevertheless be considered experimental at this point.

---

**nota**

If you decide to use the RTAI 5.2 kernel and see a problem outside the circumstances described above, then please report it immediately to the project developers.

---

### 3.7.1. Installing on Debian Buster (with Preempt-RT kernel)

1. Install Debian Buster (Debian 10), amd64 version. You can download the installer here: <https://www.debian.org/releases/buster/>.
2. After burning the iso and booting up, if you don't want Gnome desktop, select *Advanced Options* > *Alternative desktop environments* and pick the one you like.
3. Then select *Install* or *Graphical Install*.

**aviso**

Do not enter a root password, if you do sudo is disabled and you won't be able to complete the following steps.

---

1. Run the following in a [terminal](#) to bring the machine up to date with the latest packages.

```
sudo apt-get update
sudo apt-get dist-upgrade
```

2. Install the Preempt-RT kernel and modules

```
sudo apt-get install linux-image-rt-amd64
```

3. Re-boot, and select the Linux 4.19.0-9-rt-amd64 kernel (the exact kernel version might be different, look for the "-rt" suffix). This might be hidden in the "Advanced options for Debian Buster" sub-menu in Grub. When you log in, verify that `PREEMPT RT` is reported by the following command.

```
uname -v
```

4. Open Applications Menu > System > Synaptic Package Manager, search for *linux-image*, right click on the original non-rt and select *Mark for Complete Removal*. 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.
  5. Add the LinuxCNC Archive Signing Key to your apt keyring by running
-

```
sudo apt-key adv --keyserver hkp://keys.openpgp.org --recv-key 3cb9fd148f374fef
Alternate keyserver: keyserver.ubuntu.com
```

6. Add the apt repository:

```
echo deb https://linuxcnc.org/ buster base 2.8-rtpreempt | sudo tee -a /etc/apt/sources ↵
.list.d/linuxcnc.list
echo deb-src https://linuxcnc.org/ buster base 2.8-rtpreempt | sudo tee -a /etc/apt/ ↵
sources.list.d/linuxcnc.list
```

7. Update the package list from linuxcnc.org

```
sudo apt-get update
```

8. Install uspace (a reboot may be required prior to installing uspace)

```
sudo apt-get install linuxcnc-uspace
```

9. Optionally you can install mesaflash if you are using a Mesa card.

```
sudo apt install mesaflash
```

### 3.7.2. Installing on Debian Buster (with experimental RTAI kernel)



#### aviso

This kernel has known stability problems. It appears to run reliably once LinuxCNC is loaded. However, kernel panics have been seen at system shut-down.

1. This kernel and LinuxCNC version can be installed on top of the LiveDVD install, or alternatively on a fresh Install of Debian Buster 64-bit as described above.
2. Add the LinuxCNC Archive Signing Key to your apt keyring (Not necessary if switching the real-time mode of a LinuxCNC Live-CD image).

```
# Alternate keyserver: keyserver.ubuntu.com
sudo apt-key adv --keyserver hkp://keys.openpgp.org --recv-key 3cb9fd148f374fef
```

3. Add the apt repository:

```
echo deb https://linuxcnc.org/ buster base 2.8-rt | sudo tee /etc/apt/sources.list.d/ ↵
linuxcnc.list
echo deb-src https://linuxcnc.org/ buster base 2.8-rt | sudo tee -a /etc/apt/sources. ↵
list.d/linuxcnc.list
```

4. Update the package list from linuxcnc.org

```
sudo apt-get update
```

5. Install the new realtime kernel, RTAI and the rtai version of linuxcnc

```
sudo apt-get install linux-image-4.19.195-rtai-amd64
```

6. Install the RTAI application layer

```
sudo apt-get install rtai-modules-4.19.195
```

7. Install LinuxCNC (may be necessary to reboot before installing)

```
sudo apt-get install linuxcnc
```

Reboot the machine, ensuring that the system boots from the new 4.19.195-rtai kernel.

### 3.7.3. Installing on Raspbian 10

1. Download a stock Raspbian image to an SD card and install in the [usual way](#).
2. Boot the Pi and open a terminal.
3. Add the LinuxCNC Archive Signing Key to your apt keyring.

```
# Alternate keyserver: keyserver.ubuntu.com  
sudo apt-key adv --keyserver hkp://keys.openpgp.org --recv-key 3cb9fd148f374fef
```

4. Add the apt repository:

```
echo deb https://linuxcnc.org/ buster base 2.8-rtpreempt | sudo tee /etc/apt/sources. ↵  
list.d/linuxcnc.list
```

5. Update the package list from linuxcnc.org

```
sudo apt-get update
```

6. install the realtime kernel

```
sudo apt-get install linux-image-4.19.71-rt24-v7l+
```

7. Install linuxcnc (a reboot may be required prior to installing)

```
sudo apt-get install linuxcnc-ospace
```

### 3.7.4. Installing on Ubuntu Precise

1. Install Ubuntu Precise 12.04 x86 (32-bit). Any flavor should work (regular Ubuntu, Xubuntu, Lubuntu, etc). 64-bit (AMD64) is currently not supported. You can download the installer here: <http://releases.ubuntu.com/precise/> Note the warnings that this release is out of support. But it is a way to install LinuxCNC with a well-tested RTAI kernel.

2. Run the following to bring the machine up to date with the latest packages in Ubuntu Precise.

```
sudo apt-get update  
sudo apt-get dist-upgrade
```

3. Add the LinuxCNC Archive Signing Key to your apt keyring by running

```
# Alternate keyserver: keyserver.ubuntu.com  
sudo apt-key adv --keyserver hkp://keys.openpgp.org --recv-key 3cb9fd148f374fef
```

4. Add a new apt source

```
sudo add-apt-repository "deb https://linuxcnc.org/ precise base 2.8-rt"
```

5. Fetch the package list from linuxcnc.org.

```
sudo apt-get update
```

6. Install the RTAI kernel and modules by running

```
sudo apt-get install linux-image-3.4-9-rtai-686-pae rtai-modules-3.4-9-rtai-686-pae
```

7. If you want to be able to build LinuxCNC from source using the git repo, also run

```
sudo apt-get install linux-headers-3.4-9-rtai-686-pae
```

8. Reboot, and make sure you boot into the rtai kernel. When you log in, verify that the kernel name is 3.4-9-rtai-686-pae.

```
uname -r
```

9. Run

```
sudo apt-get install linuxcnc
```

---

## Capítulo 4

# Running LinuxCNC

### 4.1. Invoking LinuxCNC

After installation, LinuxCNC starts just like any other Linux program: run it from the [terminal](#) by issuing the command `linuxcnc`, or select it in the *Applications -> CNC* menu.

### 4.2. Configuration Launcher

When starting LinuxCNC (from the CNC menu or from the command line without specifying an INI file) the Configuration Selector dialog starts.

The Configuration Selector dialog allows the user to pick one of their existing configurations (My Configurations) or select a new one (from the Sample Configurations) to be copied to their home directory. Copied configurations will appear under My Configurations on the next invocation of the Configuration Selector.

The Configuration Selector offers a selection of configurations organized:

- *My Configurations* - User configurations located in `linuxcnc/configs` in your home directory.
- *Sample Configurations* - Sample configurations, when selected, are copied to `linuxcnc/configs`. Once a sample configuration was copied to your local directory, the launcher will offer it as *My Configurations*. The names under which these local configurations are presented correspond to the names of the directories within the `configs/` directory:
  - *sim* - Configurations that include simulated hardware. These can be used for testing or learning how LinuxCNC works.
  - *by\_interface* - Configurations organized by GUI.
  - *by\_machine* - Configurations organized by machine.
  - *apps* - Applications that do not require starting `linuxcnc` but may be useful for testing or trying applications like [PyVCP](#) or [GladeVCP](#).
  - *attic* - Obsolete or historical configurations.

The *sim* configurations are often the most useful starting point for new users and are organized around supported GUIs:

- *axis* - Keyboard and Mouse GUI
  - *gmoccapy* - Touch Screen GUI
-

- *gscreen* - Touch Screen GUI
- *low\_graphics* - Keyboard GUI
- *pyvcp\_demo* - Paneles de Control Virtuales Python
- *qtvcp\_screens* - Guis diseñadas con Qt5 y Python
- *tklinuxcnc* - Keyboard and Mouse Gui(no longer maintained)
- *touchy* - Touch Screen GUI

A GUI configuration directory may contain subdirectories with configurations that illustrate special situations or the embedding of other applications.

The *by\_interface* configurations are organized around common, supported interfaces like:

- general mechatronics
- mesa
- parport
- pico
- pluto
- servotogo
- vigilant
- vitalsystems

Related hardware may be required to use these configurations as starting points for a system.

The *by\_machine* configurations are organized around complete, known systems like:

- boss
- cooltool
- plasmac
- scortbot erIII
- sherline
- smithy
- tormach

A complete system may be required to use these configurations.

The *apps items* are typically 1) utilities that don't require starting linuxcnc or 2) demonstrations of applications that can be used with linuxcnc:

- info - creates a file with system information that may be useful for problem diagnosis.
  - gladevcp - Example GladeVCP applications.
  - halrun - Starts halrun in an [terminal](#).
  - latency - Applications to investigate latency
    - latency-test - standard test
-

- latency-plot - stripchart
- latency-histogram - histogram
- parport - Applications to test parport.
- pyvcp - Example pyvcp applications.
- xhc-hb04 - Applications to test an xhc-hb04 USB wireless MPG

**nota**

Under the Apps directory, only applications that are usefully modified by the user are offered for copying to the user's directory.

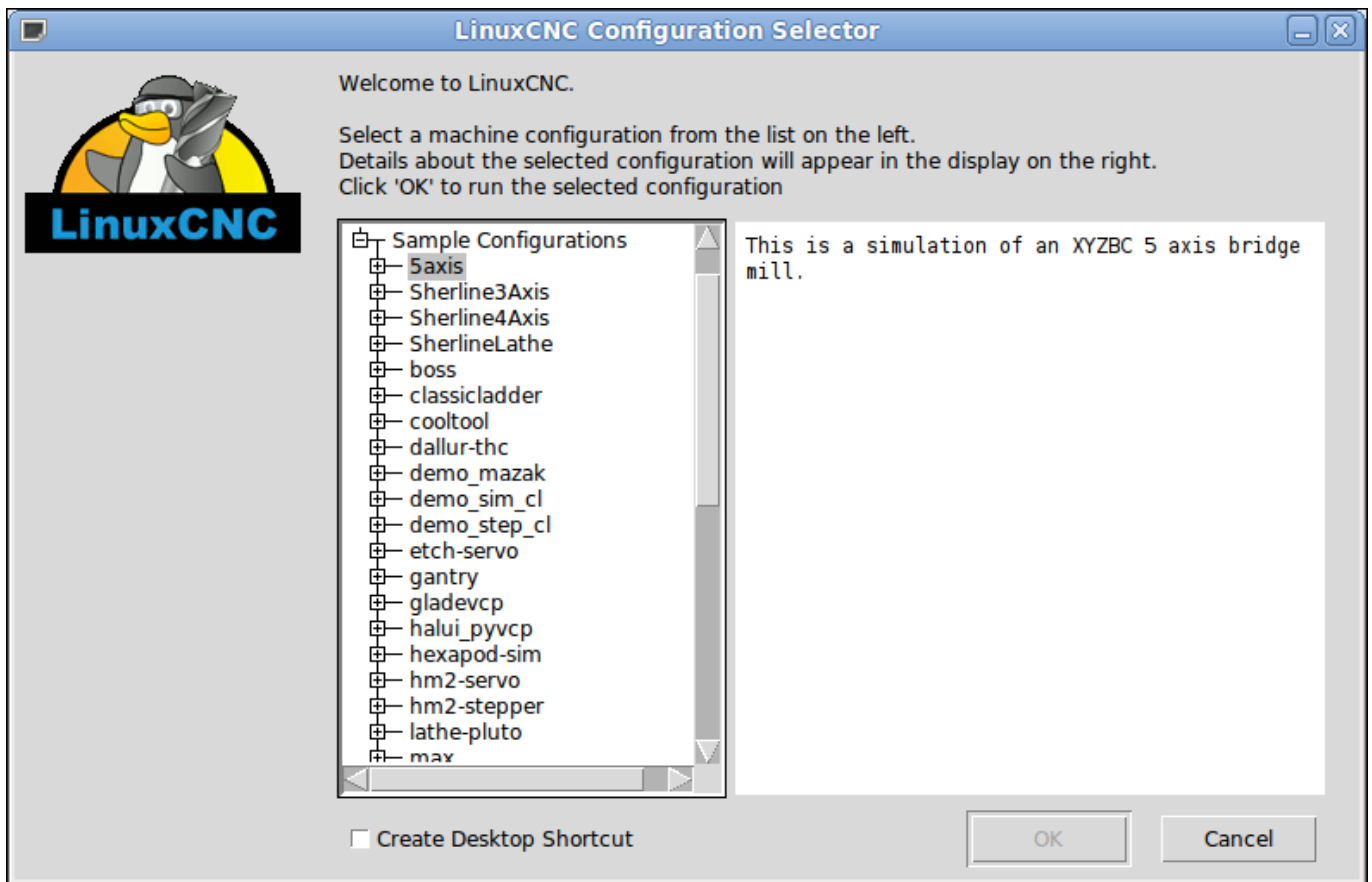


Figura 4.1: Selector de Configuración de LinuxCNC

Click any of the listed configurations to display specific information about it. Double-click a configuration or click OK to start the configuration.

Select *Create Desktop Shortcut* and then click *OK* to add an icon on the Ubuntu desktop to directly launch this configuration without showing the Configuration Selector screen.

When you select a configuration from the Sample Configurations section, it will automatically place a copy of that config in the `~/linuxcnc/configs` directory.

### 4.3. Next steps in configuration

After finding the sample configuration that uses the same interface hardware as your machine (or a simulator configuration), and saving a copy to your home directory, you can customize it according to the details of your machine. Refer to the Integrator Manual for topics on configuration.

### 4.4. Simulator Configurations

All configurations listed under Sample Configurations/sim are intended to run on any computer. No specific hardware is required and real-time support is not needed.

These configurations are useful for studying individual capabilities or options. The sim configurations are organized according to the graphical user interface used in the demonstration. The directory for axis contains the most choices and subdirectories because it is the most tested GUI. The capabilities demonstrated with any specific GUI may be available in other GUIs as well.

### 4.5. Configuration Resources

The Configuration Selector copies all files needed for a configuration to a new subdirectory of `~/linuxcnc/configs` (equivalently: `/home/username/linuxcnc/configs`). Each created directory will include at least one INI file (`inifilename.ini`) that is used to describe a specific configuration.

File resources within the copied directory will typically include one or more INI file (`filename.ini`) for related configurations and a tool table file (`toolfilename.tbl`). Additionally, resources may include HAL files (`filename.hal`, `filename.tcl`), a README file for describing the directory, and configuration specific information in a text file named after a specific configuration (`inifilename.txt`). That latter two files are displayed when using the Configuration Selector.

The supplied sample configurations may specify the parameter `HALFILE` (`filename.hal`) in the configuration INI file that are not present in the copied directory because they are found in the system HAL file library. These files can be copied to the user configuration directory and altered as required by the user for modification or test. Since the user configuration directory is searched first when finding HAL files, local modifications will then prevail.

The Configuration selector makes a symbolic link in the user configuration directory (named `hallib`) that points to the system HAL file library. This link simplifies copying a library file. For example, to copy the library `core_sim.hal` file in order to make local modifications:

```
cd ~/linuxcnc/configs/name_of_configuration
cp hallib/core_sim.hal core_sim.hal
```



## Capítulo 5

# Updating LinuxCNC

Updating LinuxCNC to a new minor release (ie to a new version in the same stable series, for example from 2.7.0 to 2.7.1) is an automatic process if your PC is connected to the internet. You will see an update prompt after a minor release along with other software updates. If you don't have an internet connection to your PC see [Updating without Network](#).

### 5.1. Upgrade to the new version

This section describes how to upgrade LinuxCNC from version 2.8 to the new 2.9.0 version. It assumes that you have an existing 2.8 install that you want to update.

To upgrade LinuxCNC from a version older than 2.8, you have to first [upgrade your old install to 2.8](#), then follow these instructions to upgrade to the new version.

If you do not have an old version of LinuxCNC to upgrade, then you're best off making a fresh install of the new version as described in the section [Getting LinuxCNC](#).

Furthermore, if you are running Ubuntu Precise or Debian Wheezy it is well worth considering making a backup of the "linuxcnc" directory on removable media and performing a [clean install of a newer OS and LinuxCNC version](#) as these releases were EOL in 2017 and 2018 respectively. If you are running on Ubuntu Lucid then you will have to do this, as Lucid is no longer supported by LinuxCNC (it was EOL in 2013).

To upgrade major versions like 2.8 to 2.9 when you have a network connection at the machine you need to disable the old linuxcnc.org apt sources in the file /etc/apt/sources.list and add a new linuxcnc.org apt source for 2.9, then upgrade LinuxCNC.

The details will depend on which platform you're running on. Open a [terminal](#) then type `lsb_release -ic` to find this information out:

```
lsb_release -ic
Distributor ID: Debian
Codename:      wheezy
```

You should be running on Debian Stretch or Wheezy (as above), or Ubuntu Precise. Packages are also available for Debian Jessie or Debian Buster if you happen to already be running one of those.

You will also need to check which realtime kernel is being used:

```
uname -r
4.19.0-9-rt-amd64
```

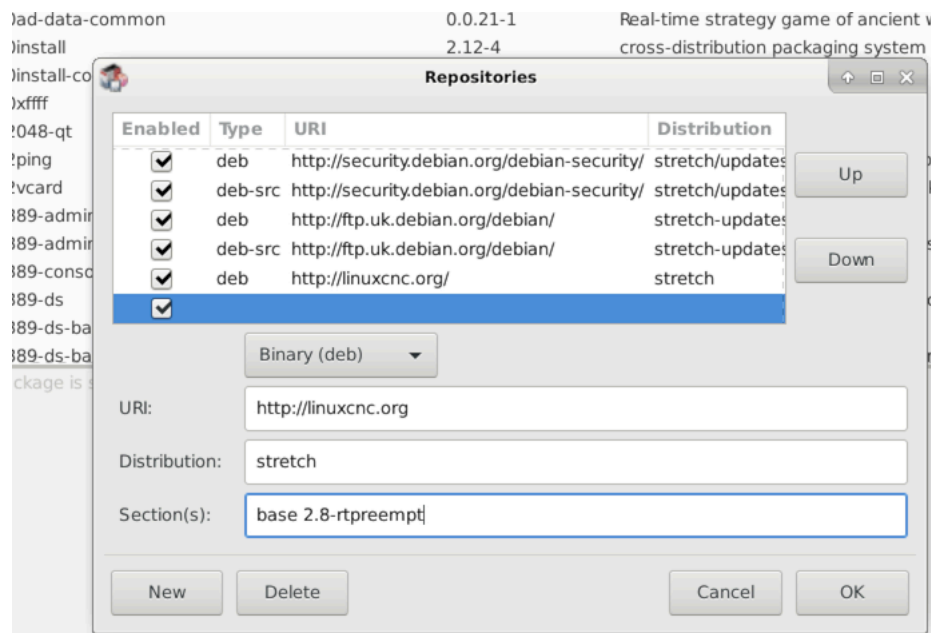
If you see (as above) -rt- in the kernel name then you are running the preempt-rt kernel and should install the "uspace" version of LinuxCNC. You should also install uspace for "sim" configs on non-realtime kernels

If you see -rtai- in the kernel name then you are running RTAI realtime. See below for the LinuxCNC version to install.

### 5.1.1. Apt Sources Configuration

- Open the Software Sources window. The process for doing this differs slightly on the three supported platforms:
  - Debian:
    - Click on Applications Menu, then System, then Synaptic Package Manager.
    - In Synaptic, click on the Settings menu, then click Repositories to open the Software Sources window.
  - Ubuntu Precise:
    - Click on the Dash Home icon in the top left.
    - In the Search field, type "software", then click on the Ubuntu Software Center icon.
    - In the Ubuntu Software Center window, click on the Edit menu, then click on Software Sources... to open the Software Sources window.
  - Ubuntu Lucid:
    - Click the System menu, then Administration, then Synaptic Package Manager.
    - In Synaptic, click on the Settings menu, then click on Repositories to open the Software Sources window.
- In the Software Sources window, select the Other Software tab.
- Delete or un-check all the old linuxcnc.org entries (leave all non-linuxcnc.org lines as they are).
- Click the Add button and add a new apt line. The line will be slightly different on the different platforms:

Platform	apt source line
Debian Stretch	deb https://linuxcnc.org stretch base 2.8-rtpreempt
Debian Wheezy	deb https://linuxcnc.org wheezy base 2.8-rt
Ubuntu Precise	deb https://linuxcnc.org precise base 2.8-rt
Debian Jessie - preempt	deb https://linuxcnc.org jessie base 2.8-rtpreempt
Debian Jessie - RTAI	deb https://linuxcnc.org jessie base 2.8-rt
Debian Buster - preempt	deb https://linuxcnc.org buster base 2.8-rtpreempt
Debian Buster - RTAI	deb https://linuxcnc.org buster base 2.8-rt



- Click Add Source, then Close in the Software Sources window. If it pops up a window informing you that the information about available software is out-of-date, click the Reload button.

### 5.1.2. Upgrading to the new version

Now your computer knows where to get the new version of the software, next we need to install it. The process again differs depending on your platform.

#### 5.1.2.1. Debian Wheezy & Stretch and Ubuntu Lucid

Debian Wheezy and Stretch both use the Synaptic Package Manager.

- Open Synaptic using the instructions in [Setting apt sources](#) above.
- Click the Reload button.
- Use the Search function to search for linuxcnc.
- The package is called "linuxcnc" for RTAI kernels and "linuxcnc-ospace" for preempt-rt.
- Click the check box to mark the new linuxcnc and linuxcnc-doc-\* packages for upgrade. The package manager may select a number of additional packages to be installed, to satisfy dependencies that the new linuxcnc package has.
- Click the Apply button, and let your computer install the new package. The old linuxcnc package will be automatically upgraded to the new one.

### 5.1.3. Ubuntu Precise

- Click on the Dash Home icon in the top left.
- In the Search field, type "update", then click on the Update Manager icon.
- Click the Check button to fetch the list of packages available.
- Click the Install Updates button to install the new versions of all packages.

## 5.2. Updating without Network

To update without a network connection you need to download the .deb then install it with dpkg. The .debs can be found in <https://linuxcnc.org/dists/>.

You have to drill down from the above link to find the correct deb for your installation. Open a [terminal](#) and type in `lsb_release -ic` to find the release name of your OS.

```
> lsb_release -ic
Distributor ID: Debian
Codename:      buster
```

Pick the OS from the list then pick the major version you want like 2.8-rt for RTAI or 2.8-rtpreempt for preempt-rt.

Next pick the type of computer you have: binary-amd64 for any 64-bit x86, binary-i386 for 32 bit, binary-armhf (32bit) or binary-arm64 (64bit) for Raspberry Pi.

Next pick the version you want from the bottom of the list like *linuxcnc-uspace\_2.8.0\_amd64.deb* (choose the latest by date). Download the deb and copy it to your home directory. You can rename the file to something a bit shorter with the file manager like *linuxcnc\_2.8.0.deb* then open a terminal and install it with the package manager with this command:

```
sudo dpkg -i linuxcnc_2.8.0.deb
```

## 5.3. Updating Configuration Files for 2.9

### 5.3.1. Stricter handling of pluggable interpreters

If you just run regular G-code and you don't know what a pluggable interpreter is, then this section does not affect you.

A seldom-used feature of LinuxCNC is support for pluggable interpreters, controlled by the undocumented [TASK]INTERPRETER INI setting.

Versions of LinuxCNC before 2.9.0 used to handle an incorrect [TASK]INTERPRETER setting by automatically falling back to using the default G-code interpreter.

As of 2.9.0, an incorrect [TASK]INTERPRETER value will cause LinuxCNC to refuse to start up. Fix this condition by deleting the [TASK]INTERPRETER setting from your INI file, so that LinuxCNC will use the default G-code interpreter.

### 5.3.2. Canterp

If you just run regular G-code and you don't use the canterp pluggable interpreter, then this section does not affect you.

In the extremely unlikely event that you are using canterp, know that the module has moved from /usr/lib/libcanterp.so to /usr/lib/linuxcnc/canterp.so, and the [TASK]INTERPRETER setting correspondingly needs to change from libcanterp.so to canterp.so.

## 5.4. Updating Configuration Files (for 2.8.x)

The new version of LinuxCNC differs from version 2.7 in some ways that may require changes to your machine configuration.

---

The main difference is that LinuxCNC no longer makes any assumptions about which joint controls which axis. This change is generally termed "joints-axes" after the name of the development branch where the changes started. This change has been in development since at least 2010, and has finally been merged.

#### 5.4.1. Distribution Configurations (updates for joints\_axes)

The LinuxCNC distribution includes many example configurations organized in directory hierarchies named: by\_machine, by\_interface, and sim (simulated machines). These configurations are often used as starting points for making a new configuration, as examples for study, or as complete simulated machines that can run without special hardware or real-time kernels.

The configuration files in these directory trees have been updated for the changes required for the joints\_axes updates.

#### 5.4.2. Automatic updates (update\_ini script for joints\_axes)

Since the joints\_axes updates require a number of changes to user INI files and their related HAL files, a script named update\_ini is provided to automatically convert user configurations.

This script is invoked when a user starts an existing configuration for the first time after updating LinuxCNC. The script searches the user INI file for a [EMC]VERSION item. If this item 1) does not exist, or 2) exists and is set to the historical CVS value "\$Revision\$", or is a numerical value less than 1.1, then the update\_ini script will popup a dialog to offer to edit the user files to create an updated configuration. If the user accepts, the configuration will be updated.

For example, if the user configuration is named bigmill.ini, the bigmill.ini file and its local associated HAL files will be edited to incorporate joints\_axes changes. All files of the initial configuration will be saved in a new directory named after the original configuration with a ".old" suffix (bigmill.old in the example).

The update\_ini script handles all common user items that are found in basic machines employing identity kinematics. Less common items used in more complex machines may not be converted automatically. Examples of complex machine configurations include:

- gantries with two joints for an axis
- machines with jogwheels
- robots with non-identity kinematics
- configurations using haltcl files

The following subsections and the section for *HAL Changes* list items that may require additional user edits to INI or HAL files.

#### 5.4.3. Multiple Spindle Support

LinuxCNC now supports up to 8 spindles (and can be recompiled for more). Existing G-code will run without modification and most configurations will default to single spindles. To specify more than one spindle set the [TRAJ]SPINDLES= entry in the INI file **and** include the num\_spindles= parameter for the motion module (set with either [EMCMOT]EMCMOT = motmod num\_spindles= or included in a HAL file loadrt entry for motmod).

The motion module num\_spindles= parameter and the [TRAJ]SPINDLES= settings **must** match.

The spindle control pin names have been changed to make spindles look more like axes and joints. motion.spindle-speed-out is now spindle.0.speed-out for example. The automatic update script will

take care of these changes. To control extra spindles the G and M-codes which control spindle speed now accept an additional "\$" argument, for example M3 \$2 to start the third spindle. "\$" was chosen to avoid clashes with any existing code letters. It should be possible to create custom G-codes to match any other multi-spindle controller. See the G-code and M-code manuals for code changes, and man motion for the HAL pin changes.

#### 5.4.4. TRAJ velocities, accelerations names

With incorporation of joints\_axes functionality, some names were changed to clarify available functionality.

```
was: [TRAJ]MAX_VELOCITY      is: [TRAJ]MAX_LINEAR_VELOCITY
was: [TRAJ]DEFAULT_VELOCITY is: [TRAJ]DEFAULT_LINEAR_VELOCITY

was: [TRAJ]MAX_ACCELERATION is: [TRAJ]MAX_LINEAR_ACCELERATION
was: [TRAJ]DEFAULT_ACCELERATION is: [TRAJ]DEFAULT_LINEAR_ACCELERATION
```

#### 5.4.5. Kinematics modules

The gentrivkins and gantrykins kinematics modules have been removed as their functionality is now available in the updated trivkins module.

The gentrivkins module has only been available in prior joints\_axes branches. To convert, it is necessary to change the name.

HAL file examples:

```
was: loadrt gentrivkins
is: loadrt trivkins

was: loadrt gentrivkins coordinates=xyyz
is: loadrt trivkins coordinates=xyyz
```

Configurations using gantrykins should be updated to use trivkins with the kinstype= parameter set to BOTH (for KINEMATICS\_BOTH).

HAL file example:

```
was: loadrt gantrykins coordinates=xyyz
is: loadrt trivkins coordinates=xyyz kinstype=BOTH
```

See the trivkins man page for additional information (*\$ man trivkins*)

Note: the most supported usage for specifying kinematics in joints\_axes is to set values in the configuration INI file [KINS] section and then reference them within the specified [HAL]HALFILES ( .hal .tcl files). For example:

```
INI file:  [KINS]
           KINEMATICS = trivkins
           JOINTS = 3
           ...

HAL file:  loadrt [KINS]KINEMATICS

haltcl file: loadrt $::KINS(KINEMATICS)
```

### 5.4.6. Lathe Configurations

Prior to `joints_axes` incorporation, lathes were often configured as if they were three axis (XYZ) machines with an unused axis (Y). This was convenient for sharing HAL files (especially for simulation configs) but required specification of `[TRAJ]AXES =3`, a *dummy* `AXIS_Y` section, and provisions for homing the unused Y coordinate. These arrangements are no longer required or recommended.

Historical lathe configurations used the default options for the `trivkins` kinematics module. These default options configure all axis letters (XYZABCUVW). With `joints_axes` incorporation, a more appropriate kinematics specification sets the coordinates to the exact ones used (XZ) and sets the number of joints accordingly to 2. There is no need for an INI file `[AXIS_Y]` section and only two `[JOINT_N]` sections need be defined.

Example INI file items for a lathe (only sections relevant to kinematics are shown):

```
[KINS]
KINEMATICS = trivkins coordinates=xz
JOINTS = 2

[TRAJ]
COORDINATES = XZ
...

[AXIS_X]
...

[AXIS_Z]
...

[JOINT_0]
...

[JOINT_1]
...
```

Note that some simulation configurations may still use the historical lathe configuration precedents.

### 5.4.7. Consistent Joints/Axes specifications

INI file items that affect joints and axes usage must be consistent.

The motion kinematics module typically loaded with `[KINS]KINEMATICS=` must use a number of joints equal to the number specified with `[KINS]JOINTS=`.

The kinematics module must implement axis letters that are consistent with the specification used by the task module item `[TRAJ]COORDINATES=`.

Examples:

Three axis Cartesian machine using `trivkins` (`KINEMATICS_IDENTITY`):

```
[KINS]KINEMATICS = trivkins
[KINS]JOINTS      = 3
[TRAJ]COORDINATES = XYZ
```

Two axis lathe using `trivkins` (`KINEMATICS_IDENTITY`) with non-consecutive axis letters:

```
[KINS]KINEMATICS = trivkins coordinates=XZ
[KINS]JOINTS      = 2
[TRAJ]COORDINATES = XZ
```

Gantry using trivkins with duplicated axis letters and KINEMATICS\_BOTH to allow individual joint positioning (for homing):

```
[KINS]KINEMATICS = trivkins coordinates=XYZZ kinstype=BOTH
[KINS]JOINTS      = 4
[TRAJ]COORDINATES = XYZZ
```

Gantry using trivkins (KINEMATICS\_BOTH) with duplicated axis letters and a rotary axis with skipped axis letters (A,B skipped):

```
[KINS]KINEMATICS = trivkins coordinates=XYZZC kinstype=BOTH
[KINS]JOINTS      = 5
[TRAJ]COORDINATES = XYZZC
```

Linear Delta Robot with non-identity kins (KINEMATICS\_BOTH) working in Cartesian frame with an additional rotary coordinate:

```
[KINS]KINEMATICS = lineardeltakins
[KINS]JOINTS      = 4
[TRAJ]COORDINATES = XYZA
```

Note: Some general-purpose kinematics modules (like trivkins) implement identity kinematics with support for coordinate specification (axis letters). Axis letters may be omitted. Axis letters may be duplicated. Joints are assigned to axis letters in a defined manner (*\$ man trivkins*).

Note: For trivkins module loading, do not include spaces about the = sign or letters:

```
This:      [KINS]KINEMATICS = trivkins coordinates=XZ
NOT This:  [KINS]KINEMATICS = trivkins coordinates = X Z
```

Note: Custom kinematics modules that implement non-identity kinematics (like lineardeltakins) define machine-specific relationships between a set of coordinates and a set of joints. Typically, custom kinematics modules compute the joints-axes relationships within the custom module but it is important to use consistent settings for the related INI items: *[KINS]JOINTS* and *[TRAJ]COORDINATES*. The details will usually be explained in the module man page (for example, *\$ man lineardeltakins*).

### 5.4.8. Home sequences

**Negative** values may be used for the INI file items named *[JOINT\_n]HOME\_SEQUENCE*. Prior to joints\_axes incorporation a value of -1 or the omission of the item indicated no sequence was applicable. Now, only omission of the item is used for that purpose. See the chapter: [Homing Configuration](#) for more information.

### 5.4.9. Locking rotary indexer (updates for joints\_axes)

With joints\_axes, an indexer is a joint that can be homed (joint mode) but must also be unlocked from G-code. This requires a one-to-one correspondence between a single joint and an axis.

Specify the joint number that corresponds to a rotary axis (*L* = A,B, or C) with an INI file setting for the axis:

```
[AXIS_L]LOCKING_INDEXER_JOINT = joint_number_for_indexer
```

Specify that the joint is a locking indexer with an INI file setting for the joint (*N* is the joint\_number\_for\_indexer):

```
[JOINT_N]LOCKING_INDEXER = 1
```

HAL pins can be created to coordinate use of a locking indicator joint:



```
joint.N.unlock      (BIT output from HAL)
joint.N.is-unlocked (BIT input to HAL)
```

To create these HAL pins for locking joints, specify all joints that are used as locking indexers with the `unlock_joints_mask` parameter for the motmod module. (bit0(LSB)==>joint0, bit1==>joint1, etc.)

```
[EMCMOT]
EMCMOT = motmod unlock_joints_mask=BITMASK
```

As an example, consider a machine using trivkins kinematics with coordinates XYZB where B is a locking indexer. For trivkins, joint numbers (starting with 0) are assigned consecutively to the coordinates specified (axis coordinate letters may be omitted). For this example, X==>joint0, Y==>joint1, Z==>joint2, B==>joint3. The mask to specify joint 3 is 000001000 (binary) == 0x08 (hexadecimal)

The required INI file entries for this trivkins XYZB example are:

```
[KINS]
JOINTS = 4
KINEMATICS = trivkins coordinates=XYZB
...

[TRAJ]
COORDINATES = XYZB
...

[EMCMOT]
EMCMOT = motmod unlock_joints_mask=0x08
...

[AXIS_B]
LOCKING_INDEXER_JOINT = 3
...

[JOINT_3]
LOCKING_INDEXER = 1
...
```

For more complex kinematics, select the joint number as required — there must be a one-to-one correspondence between the rotary axis and the joint number.

(See the motion man page (*\$ man motion*) for more information on motmod)

#### 5.4.10. Stricter INI file syntax

Lines with numeric INI variables are no longer allowed to have trailing text. In earlier versions of LinuxCNC any text after the number was silently ignored, but as of this version such text is totally disallowed. This includes hash characters ("*#*"), which in this position are a part of the value, not a comment character.

For example, lines like this will no longer be accepted:

```
MAX_VELOCITY = 7.5 # This is the max velocity of the axis.
```

They could be transformed into pairs of lines like this:

```
# This is the max velocity of the axis.
MAX_VELOCITY = 7.5
```

### 5.4.11. [TRAJ] settings

In 2.7.x versions, trajectory planning ([TRAJ]) settings included:

```
[TRAJ]
DEFAULT_ACCELERATION
MAX_ACCELERATION
```

Interim work prepared for distinct linear and angular items by renaming these items as:

```
[TRAJ]
DEFAULT_LINEAR_ACCEL
MAX_LINEAR_ACCEL
```

As these abbreviated names were inconsistent with other name conventions and the implementation of the update\_ini script, the interim naming has been corrected to use:

```
[TRAJ]
DEFAULT_LINEAR_ACCELERATION
MAX_LINEAR_ACCELERATION
```

---

#### nota

Support for specifying trajectory planning angular default and maximum accelerations has not been implemented.

---

## 5.5. HAL Changes (updates for joints\_axes 2.8.x)

### 5.5.1. Wheel or MPG (manual pulse generator) jogging

Prior to incorporation of joints\_axes updates, wheel jogging was supported in joint mode only and controlled with HAL pins:

```
bit   IN   axis.M.jog-enable
float IN   axis.M.jog-scale
s32   IN   axis.M.jog-counts
bit   IN   axis.M.jog-vel-mode
```

where *M* is a number corresponding to an axis letter (0==>X, 1==>Y, etc.)

With incorporation of joints\_axes updates, wheel jogging is available for joints in joint mode and for each axis coordinate in teleop mode. The controlling HAL pins provided are:

```
bit   IN   joint.N.jog-enable
float IN   joint.N.jog-scale
s32   IN   joint.N.jog-counts
bit   IN   joint.N.jog-vel-mode

bit   IN   axis.L.jog-enable
float IN   axis.L.jog-scale
s32   IN   axis.L.jog-counts
bit   IN   axis.L.jog-vel-mode
```

where *N* is a joint number and *L* is an axis letter.

To use an MPG in identity kins configurations where there is a one-to-one correspondence of a joint number and an axis letter, it may be convenient to connect the corresponding HAL pins. For example, if joint 1 corresponds exactly to axis letter y:

---

```
net jora_1_y_enable    => joint.1.jog-enable => axis.y.jog-enable
net jora_1_y_scale     => joint.1.jog-scale  => axis.y.jog-scale
net jora_1_y_counts    => joint.1.jog-counts => axis.y.jog-counts
net jora_1_y_vel-mode  => joint.1.jog-counts => axis.y.jog-vel-mode
```

(The signal names `jora_1_y_*` are examples, names prior to conversion for `joints_axes` will depend upon the specific configuration details.)

Configurations with non-identity kinematics and configurations that use duplicated axis letters (for example, gantries using more than one joint for an axis coordinate) will require appropriate independent control logic to support both joint and teleop (world) jogging.

## 5.5.2. INI HAL pins

HAL pins are created for INI file items for both joints (`[JOINT_N]`) and axes (`[AXIS_L]`):

For  $N = 0 \dots [\text{KINS}](\text{JOINTS}-1)$ :

INI file item	HAL pin name
<code>[JOINT_N]BACKLASH</code>	<code>ini.N.backlash</code>
<code>[JOINT_N]FERROR</code>	<code>ini.N.ferror</code>
<code>[JOINT_N]MIN_FERROR</code>	<code>ini.N.min_ferror</code>
<code>[JOINT_N]MIN_LIMIT</code>	<code>ini.N.min_limit</code>
<code>[JOINT_N]MAX_LIMIT</code>	<code>ini.N.max_limit</code>
<code>[JOINT_N]MAX_VELOCITY</code>	<code>ini.N.max_velocity</code>
<code>[JOINT_N]MAX_ACCELERATION</code>	<code>ini.N.max_acceleration</code>
<code>[JOINT_N]HOME</code>	<code>ini.N.home</code>
<code>[JOINT_N]HOME_OFFSET</code>	<code>ini.N.home_offset</code>

For  $L = x y z a b c u v w$ :

INI file item	HAL pin name
<code>[AXIS_L]MIN_LIMIT</code>	<code>ini.L.min_limit</code>
<code>[AXIS_L]MAX_LIMIT</code>	<code>ini.L.max_limit</code>
<code>[AXIS_L]MAX_VELOCITY</code>	<code>ini.L.max_velocity</code>
<code>[AXIS_L]MAX_ACCELERATION</code>	<code>ini.L.max_acceleration</code>

---

### nota

In prior versions of LinuxCNC (before `joints_axes` updates), the HAL pin names `ini._N_.*` referred to axes with  $0 \Rightarrow x$ ,  $1 \Rightarrow y$ , etc. (pins were created for all 9 axes). See the man page of *milltask* for more information.

---

## 5.6. HAL Changes (Other 2.8.x)

### 5.6.1. halcompile

The number of names= instances was formerly limited to 16. Now, for realtime components (loadrt) the instances are assigned dynamically with no built-in limit. The limit of 16 still applies to names= items for userspace (loadusr) components.

For components using *personality*, the maximum number is now settable by a command line option `-P|--personalities`.

---

### 5.6.2. Parameter to Pin changes

The following HAL output pins were changed from parameters to pins so that they can be connected to signals:

```
motion.servo.last-period    (servo last period in clks)
motion.servo.last-period_ns (kernel-dependent availability)
```

## 5.7. Interface changes for joints\_axes 2.8.x

### 5.7.1. Python LinuxCNC module

The jog() interface includes a *joint-flag* to specify joint (True) or teleop (False) jogging:

```
jog(command, joint-flag, axis-or-joint-number, velocity[, distance])

jog(linuxcnc.JOG_STOP, joint-flag, axis-or-joint-number)
jog(linuxcnc.JOG_CONTINUOUS, joint-flag, joint-flag, velocity)
jog(linuxcnc.JOG_INCREMENT, joint-flag, axis-or-joint-number, velocity, distance)
```

## 5.8. GUIs (updates for joints\_axes 2.8.x)

### 5.8.1. Notes on joint/axis jogging, homing, and kinematics

With incorporation of joints\_axes updates, LinuxCNC enforces the distinctions of joints and axes (coordinate letters) — but some GUIs (like the AXIS GUI) may hide some of the distinctions for simple machines.

In most cases, you can think of joints as *motors*.

The relationships between joints and axis coordinates are determined by the mathematical kinematics functions that describe a machine's motion.

World coordinates (X,Y,Z,A,B,C,U,V,W) are determined by applying *FORWARD* kinematics operations to joint (motor) positions.

When moving in world space (e.g., G-code movements) the required joint (motor) positions are determined by applying *INVERSE* kinematics operations to the coordinates requested for motion in world space.

Moving in world space is possible only *after* homing.

For simple machines (like mills and lathes) there is a one-to-one equivalence of joints and axis coordinate letters. For example, on an XYZ mill, the relationships are typically: axisX==joint0, axisY==joint1, axisZ==joint2. This correspondence is characterized as *IDENTITY* kinematics and the kinematics module used is usually trivkins (trivial kinematics). (See the trivkins man page *\$ man trivkins*)

Joint jogging (by joint number 0,1,...) is used in joint mode (usually used only *BEFORE* homing). When homing is completed, the jogging mode is *AUTOMATICALLY* switched from joint mode to world mode and axis jogging (coordinate letter X,Y,...) is used. This is appropriate for all G-code moves requested by MDI commands or by G-code programs.

Although jogging in joint mode is often never required after homing, some GUIs (like AXIS) provide a keyboard shortcut (\$) to allow toggling between joint and world (teleop) modes for machines that use *non-IDENTITY* kinematics.

In many common situations, joint jogging is never needed since homing is accomplished using home switches and/or the various homing methods provided by LinuxCNC. One simply turns on the machine, issues the Home-All command, the machine homes and changes to world mode automatically. See [Homing Configuration](#).

Machines that do not use home switches may require manual jogging in joint mode before homing each and every joint. It is also possible to use immediate homing (see the homing docs) for joints that do not require homing to a fixed position.

Although a GUI may hide joints/axes distinctions for *IDENTITY* kinematics machines, it is usually important to complete homing in order to run programs or use features provided by a GUI.

By default, the trivkins module declares itself as having *IDENTITY* kinematics. The distinctions of joint/world operations can be made visible in the AXIS GUI when using trivkins by setting the kinematics type to a *non-IDENTITY* type using *kinstype=both*. The *both* setting indicates that both forward and inverse kinematics functions are available and GUI provisions that hide the distinctions of joints and axis letters should not be employed. For example, for an xyz configuration, specify:

```
[KINS]
KINEMATICS = trivkins coordinates=xyz kinstype=both
```

With this setting, identity kinematics will be used but the AXIS GUI will:

1. show joint numbers prior to homing
2. show axis letters after successful homing
3. support toggling between joint and teleop modes with the \$ key

### 5.8.2. Halui

Halui now supports teleop jogging resulting in some changed pin names and numerous new names for jogging-related pins.

See the man page (`$ man halui`) for all pin names.

#### 5.8.2.1. TELEOP jogging (also called axis or world jogging)

New pins for teleop jogging are:

```
new: halui.axis.jog-speed
new: halui.axis.jog-deadband

new: halui.axis.L.plus
new: halui.axis.L.minus
... etc.
```

where *L* is a letter corresponding to one of the axis letters specified by [TRAJ]COORDINATES or *selected* for the axis selected by the halui.axis.*L*.select pins.

#### 5.8.2.2. Joint jogging

All pins for joint jogging were renamed for specificity:

```
was: halui.jog-speed           is: halui.joint.jog-speed
was: halui.jog-deadband        is: halui.joint.jog-deadband

was: halui.jog.N.plus          is: halui.joint.N.plus
was: halui.jog.N.minus         is: halui.joint.N.minus
... etc.                      ... etc.
```

where  $N$  is a joint number (0 ...  $num\_oints-1$ ) or *selected* for the joint selected by the `halui.joint.N.select` pins.

### 5.8.2.3. Additional pin renames

The HAL pins for *selected* joints were renamed for consistency with related pins.

```
was: halui.joint.selected.is_homed
is: halui.joint.selected.is-homed

was: halui.joint.selected.on-soft-limit
is: halui.joint.selected.on-soft-min-limit
```

## 5.8.3. AXIS GUI

### 5.8.3.1. Identity Kinematics

The AXIS GUI continues to support identity kinematics configurations. This GUI hides the distinctions of axes and joints in order to simplify the display and usage of simple machines.

### 5.8.3.2. Special case kinematics

Some machines, typically gantrys, may use a configuration with more than one joint assigned to an axis letter. This can be done with the `trivkins` kinematics module using repeated coordinate letters. For example, a machine configured with INI settings:

```
[KINS]
KINEMATICS = trivkins coordinates=XYZZ kintype=BOTH
...
[TRAJ]
COORDINATES = XYZZ
...
```

This machine, after homing, has a one-to-one correspondence between a single axis letter (Y) and a pair of joints (1,2). Using *kinematics=BOTH* allows control of individual joints in joint mode *if/when required*.

### 5.8.3.3. Non-identity kinematics

The AXIS GUI supports configurations using non-identity kinematics with:

1. Key binding (\$) to toggle joint or teleop mode
2. Preview Tab display of joints or axes according to joint or teleop mode
3. Preview Tab display of *Home* and *Limit* icons in joint mode
4. Preview Tab display of *All-homed* and *Any-limit* icons in teleop mode
5. DRO Tab display of joint or axes according to joint or teleop mode
6. Jogging is supported in both joint and teleop motion modes.
7. External changes to the joint/teleop motion mode are detected.

#### 5.8.3.4. Home icons

For identity kinematics, *Home* icons are shown for the corresponding (one-to-one) axis letter when a joint is homed.

For non-identity kinematics, *Home* icons are shown for individual joints when a joint is homed in joint display mode. An *All-homed* icon is displayed for all axis letters when ALL joints are homed in world display mode.

#### 5.8.3.5. Limit icons

For identity kinematics, *Limit* icons are shown for the corresponding (one-to-one) axis letter when a joint limit is active.

For non-identity kinematics, *Limit* icons are shown for individual joints when the joint limit is active in joint display mode. An *Any-Limit* icon is displayed if any joint is at a limit in teleop display mode.

#### 5.8.3.6. Key bindings for a fourth axis

In the AXIS GUI, jogging keys are assigned to axes in a configurable fashion. For 3-axis machines, XYZA machines, and lathes the default is the same as in 2.7. For other machines, the 4 pairs of jogging keys are assigned to the first 4 axes that exist in the order XYZ ABC UVW. These assignments can be controlled by new INI file directives in the [\[DISPLAY\] section of the INI file](#).

Note that the parameters used for jogging may not be appropriate for both modes for machines with non-identity kinematics.

### 5.8.4. TkLinuxCNC

The TkkLinuxCNC GUI supports both identity and non-identity kinematics, includes GUI radiobuttons and a key binding (\$) for toggling joint and teleop modes. External changes to joint or teleop motion mode are detected. Jogging is supported in both joint and teleop motion modes. Note that the parameters used for jogging may not be appropriate for both modes for machines with non-identity kinematics.

OpenGL is not used by TkLinuxCNC so it may be used to isolate problems and system dependencies that are exposed with more modern GUIs like AXIS.

The rudimentary backplot GUI provided is available for use with identity kinematics (xyz) machine configurations.

#### 5.8.4.1. emcsh commands

The code of emcsh.cc provides the set of Tcl commands used by TkLinuxCNC. The commands are available to Tcl applications as the Tcl package named *Linuxcnc*. A number of commands previously required the use of a numeric argument to specify an axis coordinate (0-->X, 1-->Y, ..., 8-->W). These commands have been simplified to use an argument that is just the coordinate letter.

Commands now using a coordinate letter argument are:

1. emc\_pos\_offset
2. emc\_abs\_cmd\_pos
3. emc\_abs\_act\_pos
4. emc\_rel\_cmd\_pos

5. emc\_rel\_act\_pos
6. emc\_tool\_offset
7. emc\_probed\_pos

### 5.8.5. Touchy

The Touchy GUI continues to support the identity kinematics configurations that it supported prior to joints\_axes incorporation. Jogging is done in teleop mode.

### 5.8.6. Gscreen

The Gscreen GUI continues to support the identity kinematics configurations that it supported prior to joints\_axes incorporation. Jogging is done in teleop mode.

### 5.8.7. GMOCCAPY

The GMOCCAPY GUI continues to support the identity kinematics configurations that it supported prior to joints\_axes incorporation. Jogging is done in teleop mode.

### 5.8.8. shuttlexpress driver renamed to shuttle

The HAL driver for the Contour Designs ShuttleXpress device has been renamed from "shuttlexpress" to just "shuttle". If your HAL files include some variant of "loadusr shuttlexpress", replace "shuttlexpress" with "shuttle".

Support has been added for the ShuttlePRO, a bigger version of the ShuttleXpress, so the old driver name is no longer accurate.

### 5.8.9. linuxcncrsh

"Home All" is now supported with the set home subcommand by using -1 for the joint number.

The jogging commands have been altered to accommodate both joint (free) and teleop (world) jogging.

```
was: set jog      joint_number      speed
is: set jog      joint_number|axis_letter speed

was: set jog_incr joint_number      speed increment
is: set jog_incr joint_number|axis_letter speed increment

was: set jog_stop
is: set jog_stop joint_number|axis_letter
```

---

#### nota

Test for teleop mode using command: get teleop\_enable

If TELEOP\_ENABLE=YES, use axis\_letter;

Else use joint\_number

---



**nota**

Formerly, the command `set jog 0 1.234` would jog the zeroth axis (X) with requested speed=1.234 in any mode (free or teleop). This command now attempts to jog the zeroth joint (Joint0) provided the mode is free (not teleop). To jog the X axis, the mode must be teleop and the corresponding command is: `set jog x 1.234`.

## 5.9. Tools

### 5.9.1. Calibration (emccalib.tcl)

The calibration/tuning tool now supports stanzas:

[JOINT\_N], [AXIS\_L], [SPINDLE\_S], [TUNE]

where *N* is a joint number (0 .. ([KINS]JOINTS-1) ), *L* is an axis coordinate letter (X,Y,Z,A,B,C,U,V,W), and *S* is a spindle number (0 .. 9).

**nota**

The number of allowed spindles is 8 but legacy configurations may include a stanza [SPINDLE\_9] unrelated to an actual spindle number.

**nota**

The [TUNE] stanza may be used for specifying tunable items not relevant to the other supported stanzas.

## 5.10. Obsolete Guis (removed for 2.8.x)

The GUIs *mini*, *keystick*, and *xlinuxcnc* have been removed in conjunction with updates for joints\_axes. All related source code, examples, and documentation are available in the git repository.

## 5.11. Deprecated Guis (marked at 2.8.x)

The *linuxcnclcd* GUI is a candidate for removal. Should this component be removed, all related source code, examples, and documentation will be available in the git repository.

## 5.12. Simulator configurations (updates for joints axes 2.8.x)

### 5.12.1. Pre-joints\_axes

Prior to joints\_axes incorporation, the HAL files used in sim configs typically supported a common milling machine — a Cartesian system with trivial kinematics and three axes named X Y Z. Typical HAL file entries:

```
[HAL]
HALFILE = core_sim.hal
HALFILE = sim_spindle_encoder.hal
HALFILE = axis_manualtoolchange.hal
HALFILE = simulated_home.hal
```

Lathe configs often shared the same HAL files and used the expedient method of specifying 3 axes with Y unused. More complex sim configs provided specific sets of HAL files according to the configuration purpose.

### 5.12.2. Post-joints\_axes

With the incorporation of joints\_axes functionality, many sims provided in the distribution now take advantage of a general purpose HAL file that supports numerous configurations automatically. A typical sim config HALFILE specification is:

```
[HAL]
HALFILE = LIB:basic_sim.tcl
```

The basic\_sim.tcl HALFILE supports a number of commonly required functions for any number of joints as specified by:

```
[KINS]
...
JOINTS = number_of_joints
...
```

Functions supported include:

1. *ddts* — differentiator HAL components are loaded and connected for each joint (and xy, xyz for trivkins machines)
2. *simulated\_home* — a sim\_home\_switch HAL component is loaded and connected for each joint. The homing conditions are specified by the usual [JOINT\_n]HOME\_\* INI file items.
3. *use\_hal\_manualtoolchange* — the user space hal\_manualtoolchange component is loaded and connected.
4. *sim\_spindle* — the sim\_spindle component is loaded and connected to additional loaded HAL components to simulate the inertia of a rotating spindle mass.

The functions are activated by default but can be excluded using options: *-no\_make\_ddts*, *-no\_simulated\_home*, *-no\_use\_hal\_manualtoolchange*, *-no\_sim\_spindle*.

For example, to omit creation of ddts:

```
HALFILE = LIB:basic_sim.tcl -no_make_ddts
```

Omitting one or more of the core functions allows testing without the function or addition of new HALFILES to implement or expand on the functionality.

#### 5.12.2.1. Equivalent HAL commands file

When LIB:basic\_sim.tcl is used, an equivalent HAL file is created (in the configuration directory) to show the halcmd commands issued. The file name is based on the name of the INI file with *\_cmds* appended to the basename and a conventional *.hal* file extension. Example:

```
inifilename:          example.ini
equivalent_halfilename: example_cmds.hal
```

The equivalent HAL file supersedes previous instances of files with the same filename. INI file variables substitutions specified in the INI file and interpreted by halcmd are automatically substituted in the created HAL file. If there are [HAL]HALFILES specified before LIB:basic\_sim.tcl, their halcmd commands are included too.

The equivalent HAL file can be used to create a new configuration based on the original configuration made with LIB:basic\_sim.tcl with the following steps:

1. Run the simulator configuration to create a new equivalent HAL file, for example: *example\_cmds.hal*.
2. To use this new equivalent HAL file in the original simulator configuration INI file (or a copy of it), edit to change:

```
[HAL]
HALFILE = LIB:basic_sim.tcl other_parameters
```

to:

```
[HAL]
HALFILE = ./example_cmds.hal
```

#### 5.12.2.2. Notes

All components and connections made by LIB:basic\_sim.tcl can be viewed using halcmd. The entire HAL configuration (except for userspace components loaded with loadusr) can be saved to a file using:

```
$ halcmd save > hal.save
```

Use of LIB:basic\_sim.tcl reduces the effort needed to make a simulation config since it handles most of the required component loading and HAL connections.

The sim config *Sample Configurations/sim/axis/minimal\_xyz.ini* demonstrates a working xyz configuration that uses LIB:basic\_sim.tcl with a minimal number of INI file settings.

## 5.13. Miscellaneous updates for 2.8.x

Commits to unreleased branches may make changes that affect testers and early-adopters of the unreleased software.

### 5.13.1. Motion pins

New pins (see the motion man page for more info):

```
--- axis.L.jog-accel-fraction joint.N.jog-accel-fraction ---
```

### 5.13.2. HAL pins

Name changes:

```
was: axis.L.vel-cmd
is:  axis.l.teleop-vel-cmd
```

New pins:

```
motion.homing-inhibit (see motion manpage)
```

### 5.13.3. HAL component updates

1. siggen: new pin *reset* to set output signal values to predefined state
2. biquad: pins *type,f0,Q,s1,s2* were formerly params
3. userkins: template for user-built kinematics modules using *halcompile*

### 5.13.4. XHC-HB04 Pendant Support

#### 5.13.4.1. xhc\_hb04\_util.comp (helper component)

Remove unused pin *jogenable-off*.

Add pin *amux-enable* so that the multiplexed acceleration reductions are now enabled by the ANDing the pins: *is-manual* and *amux-enable*. These two pins are typically connected to *halui.mode.is-manual* and *halui.mode.is-teleop* respectively.

#### 5.13.4.2. xhc\_hb04.tcl (optional LIB configuration HAL file)

Remove signal pendant:jogenable-off for removed pin *pendant\_util.jogenable-off*.

Support new motion pins for reduced accelerations (*axis.L.jog-accel-fraction*, *joint.N.jog-accel-fraction*) for wheel jogging. The use of [APPLICATIONS]APP=xhc-hb04-accel is no longer supported. Reduced accels are applied for wheel jogging only (not for nml commands issued by GUIs).

### 5.13.5. XHC-WHB04B-6 pendant support

See the documentation for the xhc-whb04b-6 component.

### 5.13.6. bldc3\_hall

The *bldc\_hall3* component has been removed. The **bldc** component is more flexible and better tested.

### 5.13.7. [JOINT\_n]HOME\_SEQUENCE Starting values

Starting sequence values may be 0, 1 (or -1) only. See the "Homing Configuration" documentation for more information.

### 5.13.8. [JOINT\_n]HOME\_SEQUENCE Negative values

Joints using a negative HOME\_SEQUENCE are not allowed to jog in joint mode in order to prevent misalignment (racking) in common gantry configurations. As always, machines with any kinematics type must be homed prior to enabling conventional world mode jogging.

### 5.13.9. TWOPASS support for complex loadrt config= items

Added twopass support for loadrt config modparams with multiple settings separated by blanks and enclosed with quotes. Example:

```
loadrt hm2_eth board_ip=10.10.10.10 config="num_encoders=2 num_pwmgens=2 num_stepgens=3"
```

## 5.14. Changes beyond 2.8.x (master branch development)

The master branch is version-tagged with prerelease notation, typically 2.9~pre\*

### 5.14.1. Python3 and GTK3

2.9 changed to Python3 and GTK3. This only affects you if you have custom glade or Python handlers in your config.

1. Run py3clean in your config directory to remove any temporary files.
2. Run py3clean in your LinuxCNC source directory if you compile from source.
3. Run 2to3 -w on any .py files you have written.
4. Make sure the interpreter in the first line of the script is Python 3, not Python 2.
5. Open the ui file in glade, and save it. It should convert everything that can be converted automatically, and give you warnings.

### 5.14.2. LinuxCNC Startup

The main script, **linuxcnc**, supports a new option (-H dirname) to specify an additional user-specified directory for HAL files. This directory is searched before the usual search of 1) the INI directory and 2) the system HAL file library directory.

### 5.14.3. G-code changes

G43.2 (additional offsets) now accepts transient offsets to be added by axis words as well as from the tool table.

### 5.14.4. Configuration Updates

#### 5.14.4.1. INI file Settings

New: [JOINT\_n]HOME\_INDEX\_NO\_ENCODER\_RESET—support encoder with index that does not reset upon receipt of index pulse following assertion of index\_enable.

axis.py default for [DISPLAY]GEOMETRY was:"XYZBCUVW",is:"XYZABCUVW"

### 5.14.5. Code Updates

Management of the internal storage of tool data and the communication of same between EMCIO and TASK has been refactored to use memory mapped storage. Legacy use of nml messages for tooldata is deprecated and may be removed before a new release.

Code references to the sequential indexes for internal tooldata have been clarified but legacy variable names persist for **selected\_pocket** and **current\_pocket**. Variables with these names refer to the sequential index for internal tooldata not an actual pocket number. These variable names may be renamed in the future and require changes to user-fielded Python remap applications that modify tool handling.

A new optional interface is provided to support management of tool data by an external database application.

The ioControl\_v2.cc file providing the userspace program iov2 has no maintainer and its use is deprecated — it may be removed before the next release.

#### 5.14.5.1. Reverse Run

Support added for reverse run in the trajectory planner, the task, and motion modules, the Python interface, the AXIS GUI, and the test suite.

#### 5.14.5.2. Number of Joints

The maximum number of joints (EMCMOT\_MAX\_JOINTS) increased from 9 to 16. The AXIS GUI now supports display of up to 16 joints.

#### 5.14.5.3. Extra Joints

A new motmod parameter (num\_extrajoints) specifies joints that are homed by conventional joint homing methods but controlled by new HAL pins (joint.N.posthome-cmd) after homing. Such joints may be managed by independent motion planner/controllers in HAL and manipulated from G-code using custom M-codes. See the motion man page for more info.

#### 5.14.5.4. Homing

A homing api is provided by src/emc/motion/homing.h to support users' custom homing code that replaces src/emc/motion/homing.c with a user-customized homing.c file.

#### 5.14.5.5. Motion

The motion module supports kinematics modules that define new functions kinematicsSwitchable() and kinematicsSwitch() to switch their kinematics type. A HAL pin, motion.switchkins-type, is provided for use of such kinematics modules.

Provided kinematics modules that implement kinematics switching use the switchkins.o object to supply the required rtapi\_main() and related functions. Kinematics modules that do not support kinematics switching use the macro "KINS\_NOT\_SWITCHABLE" provided by kinematics.h.

---

#### 5.14.5.6. Switchkins Kinematics Modules

Several kinematics modules are now switchable between their eponymous kinematics and an alternate identity kinematics mode.

Kinematic modules supporting switchkins:

1. xyzac-trt-kins table-rotary-tilting (supersedes xyzac-trt-kins)
2. xyzbc-trt-kins table-rotary-tilting (supersedes xyzbc-trt-kins)
3. genserkins generalized serial-link kinematics
4. genhexkins generalized hexapod parallel kinematics
5. scarakins scara robot
6. pumakins puma robot
7. 5axiskins bridgemill (xyzbcw 6axes)

The switchkins modules above (and trivkins) support a `coordinates=` parameter that optionally specifies an ordered set of coordinate letters that are sequentially assigned to joint numbers (beginning with joint0).

The switchkins modules above include provisions for compile-time support of an additional user-specified kinematics type identified on the make command line by the `userkfuncs` environmental variable. (See `src/Makefile`)

The userspace test program `bin/genserkins` has been isolated to a single file (`ugenserkins.c`) since its original source file (`genserkins.c`) has been refactored for switchkins support. The userspace test program has not been actively maintained and its use is deprecated. The `ugenserkins.c` file may be removed in the future.

#### 5.14.5.7. Trajectory Planner

The trajectory planner is now implemented as a loadable module (default: `tpmod`). An alternate (user-built) planner can be loaded using INI setting `[TRAJ]TPMOD=modulename` or the `linuxcnc -t modulename` option. The example file `src/hal/components/tpcomp.comp` illustrates a method for creating a module using `halcompile`.

#### 5.14.5.8. Homing

Homing functions are now implemented by a loadable module (default: `homemod`). An alternate (user-built) planner can be loaded using INI setting `[EMCMOT]HOMEMOD=modulename` or the `linuxcnc -m modulename` option. The example file `src/hal/components/homecomp.comp` is a minimal example of a homing module that can be built with `halcompile`.

#### 5.14.5.9. Other

`lib/hallib/sim_lib.tcl`: simulate encoder index if `[JOINT_n]HOME_USE_INDEX` is specified.

`lib/python/vismach.py`: new HAL pin `vismach.plotclear`

### 5.14.6. HAL

#### 5.14.6.1. Components

`sim_home_switch`: added I/O pin for index-enable

#### **5.14.6.2. Motion module pins**

motion.feed-upm — current feed in units per minute

### **5.14.7. Configs**

#### **5.14.7.1. INI file**

[DISPLAY]GEOMETRY settings that include the ! character specify that displayed rotations respect G5x,G92 offsets.

#### **5.14.7.2. Simulation Configs**

sim/configs/axis/axis\_9axis: demonstrate simulated encoder index

## **5.15. Changes after 2.8.x**

Future versions of this document will take into account changes made to the development branch after the latest 2.8.x release.



## Capítulo 6

# Glossary

A listing of terms and what they mean. Some terms have a general meaning and several additional meanings for users, installers, and developers.

### **Acme Screw**

A type of lead-screw that uses an Acme thread form. Acme threads have somewhat lower friction and wear than simple triangular threads, but ball-screws are lower yet. Most manual machine tools use acme lead-screws.

### **Axis**

One of the computer controlled movable parts of the machine. For a typical vertical mill, the table is the X axis, the saddle is the Y axis, and the quill or knee is the Z axis. Angular axes like rotary tables are referred to as A, B, and C. Additional linear axes relative to the tool are called U, V, and W respectively.

### **AXIS(GUI)**

One of the Graphical User Interfaces available to users of LinuxCNC. It features the modern use of menus and mouse buttons while automating and hiding some of the more traditional LinuxCNC controls. It is the only open-source interface that displays the entire tool path as soon as a file is opened.

### **GMOCCAPY (GUI)**

A Graphical User Interfaces available to users of LinuxCNC. It features the use and feel of an industrial control and can be used with touch screen, mouse and keyboard. It support embedded tabs and hal driven user messages, it offers a lot of hal beens to be controlled with hardware. GMOCCAPY is highly customizable.

### **Backlash**

The amount of "play" or lost motion that occurs when direction is reversed in a lead screw. or other mechanical motion driving system. It can result from nuts that are loose on leadscrews, slippage in belts, cable slack, "wind-up" in rotary couplings, and other places where the mechanical system is not "tight". Backlash will result in inaccurate motion, or in the case of motion caused by external forces (think cutting tool pulling on the work piece) the result can be broken cutting tools. This can happen because of the sudden increase in chip load on the cutter as the work piece is pulled across the backlash distance by the cutting tool.

### **Backlash Compensation**

Any technique that attempts to reduce the effect of backlash without actually removing it from the mechanical system. This is typically done in software in the controller. This can correct the final resting place of the part in motion but fails to solve problems related to direction changes while in motion (think circular interpolation) and motion that is caused when external forces (think cutting tool pulling on the work piece) are the source of the motion.

**Ball Screw**

A type of lead-screw that uses small hardened steel balls between the nut and screw to reduce friction. Ball-screws have very low friction and backlash, but are usually quite expensive.

**Ball Nut**

A special nut designed for use with a ball-screw. It contains an internal passage to re-circulate the balls from one end of the screw to the other.

**CNC**

Computer Numerical Control. The general term used to refer to computer control of machinery. Instead of a human operator turning cranks to move a cutting tool, CNC uses a computer and motors to move the tool, based on a part program.

**Comp**

A tool used to build, compile and install LinuxCNC HAL components.

**Configuration(n)**

A directory containing a set of configuration files. Custom configurations are normally saved in the users home/linuxcnc/configs directory. These files include LinuxCNC's traditional INI file and HAL files. A configuration may also contain several general files that describe tools, parameters, and NML connections.

**Configuration(v)**

The task of setting up LinuxCNC so that it matches the hardware on a machine tool.

**Coordinate Measuring Machine**

A Coordinate Measuring Machine is used to make many accurate measurements on parts. These machines can be used to create CAD data for parts where no drawings can be found, when a hand-made prototype needs to be digitized for moldmaking, or to check the accuracy of machined or molded parts.

**Display units**

The linear and angular units used for onscreen display.

**DRO**

A Digital Read Out is a system of position-measuring devices attached to the slides of a machine tool, which are connected to a numeric display showing the current location of the tool with respect to some reference position. DROs are very popular on hand-operated machine tools because they measure the true tool position without backlash, even if the machine has very loose Acme screws. Some DROs use linear quadrature encoders to pick up position information from the machine, and some use methods similar to a resolver which keeps rolling over.

**EDM**

EDM is a method of removing metal in hard or difficult to machine or tough metals, or where rotating tools would not be able to produce the desired shape in a cost-effective manner. An excellent example is rectangular punch dies, where sharp internal corners are desired. Milling operations can not give sharp internal corners with finite diameter tools. A *wire* EDM machine can make internal corners with a radius only slightly larger than the wire's radius. A *sinker* EDM can make internal corners with a radius only slightly larger than the radius on the corner of the sinking electrode.

**EMC**

The Enhanced Machine Controller. Initially a NIST project. Renamed to LinuxCNC in 2012.

**EMCIO**

The module within LinuxCNC that handles general purpose I/O, unrelated to the actual motion of the axes.

**EMCMOT**

The module within LinuxCNC that handles the actual motion of the cutting tool. It runs as a real-time program and directly controls the motors.

---

**Encoder**

A device to measure position. Usually a mechanical-optical device, which outputs a quadrature signal. The signal can be counted by special hardware, or directly by the parport with LinuxCNC.

**Feed**

Relatively slow, controlled motion of the tool used when making a cut.

**Feed rate**

The speed at which a cutting motion occurs. In auto or MDI mode, feed rate is commanded using an F word. F10 would mean ten machine units per minute.

**Feedback**

A method (e.g., quadrature encoder signals) by which LinuxCNC receives information about the position of motors.

**Feedrate Override**

A manual, operator controlled change in the rate at which the tool moves while cutting. Often used to allow the operator to adjust for tools that are a little dull, or anything else that requires the feed rate to be "tweaked".

**Floating Point Number**

A number that has a decimal point. (12.300) In HAL it is known as float.

**G-code**

The generic term used to refer to the most common part programming language. There are several dialects of G-code, LinuxCNC uses RS274/NGC.

**GUI**

Graphical User Interface.

**General**

A type of interface that allows communications between a computer and a human (in most cases) via the manipulation of icons and other elements (widgets) on a computer screen.

**LinuxCNC**

An application that presents a graphical screen to the machine operator allowing manipulation of the machine and the corresponding controlling program.

**HAL**

Hardware Abstraction Layer. At the highest level, it is simply a way to allow a number of building blocks to be loaded and interconnected to assemble a complex system. Many of the building blocks are drivers for hardware devices. However, HAL can do more than just configure hardware drivers.

**Home**

A specific location in the machine's work envelope that is used to make sure the computer and the actual machine both agree on the tool position.

**INI file**

A text file that contains most of the information that configures LinuxCNC for a particular machine.

**Instance**

One can have an instance of a class or a particular object. The instance is the actual object created at runtime. In programmer jargon, the "Lassie" object is an instance of the "Dog" class.

**Joint Coordinates**

These specify the angles between the individual joints of the machine. See also Kinematics

**Jog**

Manually moving an axis of a machine. Jogging either moves the axis a fixed amount for each key-press, or moves the axis at a constant speed as long as you hold down the key. In manual mode, jog speed can be set from the graphical interface.

---

**kernel-space**

See real-time.

**Kinematics**

The position relationship between world coordinates and joint coordinates of a machine. There are two types of kinematics. Forward kinematics is used to calculate world coordinates from joint coordinates. Inverse kinematics is used for exactly the opposite purpose. Note that kinematics does not take into account, the forces, moments etc. on the machine. It is for positioning only.

**Lead-screw**

An screw that is rotated by a motor to move a table or other part of a machine. Lead-screws are usually either ball-screws or acme screws, although conventional triangular threaded screws may be used where accuracy and long life are not as important as low cost.

**Machine units**

The linear and angular units used for machine configuration. These units are specified and used in the INI file. HAL pins and parameters are also generally in machine units.

**MDI**

Manual Data Input. This is a mode of operation where the controller executes single lines of G-code as they are typed by the operator.

**NIST**

National Institute of Standards and Technology. An agency of the Department of Commerce in the United States.

**NML**

Neutral Message Language provides a mechanism for handling multiple types of messages in the same buffer as well as simplifying the interface for encoding and decoding buffers in neutral format and the configuration mechanism.

**Offsets**

An arbitrary amount, added to the value of something to make it equal to some desired value. For example, G-code programs are often written around some convenient point, such as X0, Y0. Fixture offsets can be used to shift the actual execution point of that G-code program to properly fit the true location of the vice and jaws. Tool offsets can be used to shift the "uncorrected" length of a tool to equal that tool's actual length.

**Part Program**

A description of a part, in a language that the controller can understand. For LinuxCNC, that language is RS-274/NGC, commonly known as G-code.

**Program Units**

The linear and angular units used in a part program. The linear program units do not have to be the same as the linear machine units. See G20 and G21 for more information. The angular program units are always measured in degrees.

**Python**

General-purpose, very high-level programming language. Used in LinuxCNC for the Axis GUI, the Stepconf configuration tool, and several G-code programming scripts.

**Rapid**

Fast, possibly less precise motion of the tool, commonly used to move between cuts. If the tool meets the workpiece or the fixturing during a rapid, it is probably a bad thing!

**Rapid rate**

The speed at which a rapid motion occurs. In auto or MDI mode, rapid rate is usually the maximum speed of the machine. It is often desirable to limit the rapid rate when testing a G-code program for the first time.

---

**Real-time**

Software that is intended to meet very strict timing deadlines. Under Linux, in order to meet these requirements it is necessary to install a realtime kernel such as RTAI and build the software to run in the special real-time environment. For this reason real-time software runs in kernel-space.

**RTAI**

Real Time Application Interface, see <https://www.rtai.org/>, the real-time extensions for Linux that LinuxCNC can use to achieve real-time performance.

**RTLINUX**

See <https://en.wikipedia.org/wiki/RTLinux>, an older real-time extension for Linux that LinuxCNC used to use to achieve real-time performance. Obsolete, replaced by RTAI.

**RTAPI**

A portable interface to real-time operating systems including RTAI and POSIX pthreads with realtime extensions.

**RS-274/NGC**

The formal name for the language used by LinuxCNC part programs.

**Servo Motor**

Generally, any motor that is used with error-sensing feedback to correct the position of an actuator. Also, a motor which is specially-designed to provide improved performance in such applications.

**Servo Loop**

A control loop used to control position or velocity of an motor equipped with a feedback device.

**Signed Integer**

A whole number that can have a positive or negative sign. In HAL it is known as s32. (A signed 32-bit integer has a usable range of -2,147,483,647 to +2,147,483,647.)

**Spindle**

The part of a machine tool that spins to do the cutting. On a mill or drill, the spindle holds the cutting tool. On a lathe, the spindle holds the workpiece.

**Spindle Speed Override**

A manual, operator controlled change in the rate at which the tool rotates while cutting. Often used to allow the operator to adjust for chatter caused by the cutter's teeth. Spindle Speed Override assumes that the LinuxCNC software has been configured to control spindle speed.

**Stepconf**

An LinuxCNC configuration wizard. It is able to handle many step-and-direction motion command based machines. It writes a full configuration after the user answers a few questions about the computer and machine that LinuxCNC is to run on.

**Stepper Motor**

A type of motor that turns in fixed steps. By counting steps, it is possible to determine how far the motor has turned. If the load exceeds the torque capability of the motor, it will skip one or more steps, causing position errors.

**TASK**

The module within LinuxCNC that coordinates the overall execution and interprets the part program.

**Tcl/Tk**

A scripting language and graphical widget toolkit with which several of LinuxCNCs GUIs and selection wizards were written.

**Traverse Move**

A move in a straight line from the start point to the end point.

---

**Units**

See "Machine Units", "Display Units", or "Program Units".

**Unsigned Integer**

A whole number that has no sign. In HAL it is known as u32. (An unsigned 32-bit integer has a usable range of zero to 4,294,967,296.)

**World Coordinates**

This is the absolute frame of reference. It gives coordinates in terms of a fixed reference frame that is attached to some point (generally the base) of the machine tool.

## Capítulo 7

# Legal Section

Translations of this file provided in the source tree are not legally binding.

### 7.1. Copyright Terms

**Copyright (c) 2000-2022 LinuxCNC.org**

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

### 7.2. GNU Free Documentation License

**GNU Free Documentation License Version 1.1, March 2000**

Copyright © 2000 Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

#### 1. APPLICABILITY AND DEFINITIONS

---

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

## **2. VERBATIM COPYING**

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## **3. COPYING IN QUANTITY**

If you publish printed copies of the Document numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

---



If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

#### 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled “History” in the various original documents, forming one section entitled “History”; likewise combine any sections entitled “Acknowledgements”, and any sections entitled “Dedications”. You must delete all sections entitled “Endorsements.”

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an “aggregate”, and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document’s Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this

License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <https://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

### **ADDENDUM:** How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

---

## Capítulo 8

# Índice alfabético

### A

About LinuxCNC, [1](#)  
acme screw, [43](#)  
axis, [43](#)

### B

backlash, [43](#)  
backlash compensation, [43](#)  
ball nut, [44](#)  
ball screw, [44](#)

### C

CNC, [44](#)  
comp, [44](#)  
Configuration Launcher, [15](#)  
coordinate measuring machine, [44](#)

### D

display units, [44](#)  
DRO, [44](#)

### E

EDM, [44](#)  
EMC, [44](#)  
EMCIO, [44](#)  
EMCMOT, [44](#)  
encoder, [45](#)

### F

feed, [45](#)  
feed rate, [45](#)  
feedback, [45](#)  
feedrate override, [45](#)

### G

G-code, [45](#)  
Getting Help, [2](#)  
Getting LinuxCNC, [6](#)  
GUI, [43](#), [45](#)

### H

HAL, [45](#)  
home, [45](#)

### I

INI, [45](#)

Installation:Alternate Methods, [10](#)

Installation:Problems, [10](#)

Instance, [45](#)

### J

jog, [45](#)  
joint coordinates, [45](#)

### K

kinematics, [46](#)

### L

lead screw, [46](#)  
LinuxCNC:Alternate Install Methods, [10](#)  
LinuxCNC:Installation Problems, [10](#)  
loop, [47](#)

### M

machine units, [46](#)  
MDI, [46](#)

### N

NIST, [46](#)  
NML, [46](#)

### O

offsets, [46](#)

### P

part Program, [46](#)  
program units, [46](#)

### R

rapid, [46](#)  
rapid rate, [46](#)  
real-time, [47](#)  
RS274NGC, [47](#)  
RTAI, [47](#)  
RTAPI, [47](#)  
RTLINUX, [47](#)  
Running LinuxCNC, [15](#)

### S

servo motor, [47](#)  
Signed Integer, [47](#)  
spindle, [47](#)  
stepper motor, [47](#)  
System Requirements, [4](#)

## **T**

TASK, [47](#)  
Tk, [47](#)  
Traverse Move, [47](#)

## **U**

units, [48](#)  
Unsigned Integer, [48](#)  
Updates to LinuxCNC, [9](#)  
Updating LinuxCNC, [19](#)

## **W**

world coordinates, [48](#)

---