QtPlasmaC

Table of Contents

1. Preamble
2. License
3. Introduction
4. Installing LinuxCNC
    - 4.1. If The User Does Not Have Linux Installed
    - 4.2. Package Installation (Buildbot) If The User Has Linux on Debian 12 (Bookworm)
    - 4.3. Package Installation (Buildbot) If The User Has Linux on Debian 11 (Bullseye) or Debian 10 (Buster)
    - 4.4. Run In Place Installation If The User Has Linux Installed
5. Creating A QtPlasmaC Configuration
    - 5.1. Modes
    - 5.2. Available I/Os
    - 5.3. Recommended Settings:
    - 5.4. Configuring
    - 5.5. Qt Dependency Errors
    - 5.6. Initial Setup
6. Migrating to QtPlasmac From PlasmaC (AXIS or GMOCCAPY)
    - 6.1. Quick Method
    - 6.2. New Base Config Method
7. Other QtPlasmaC Setup Considerations
    - 7.1. Lowpass Filter
    - 7.2. Contact Bounce
    - 7.3. Contact Load
    - 7.4. Desktop Launcher
    - 7.5. QtPlasmaC Files
    - 7.6. INI File
8. QtPlasmaC GUI Overview
    - 8.1. Exiting QtPlasmaC
    - 8.2. MAIN Tab
    - 8.3. Preview Views
    - 8.4. CONVERSATIONAL Tab
    - 8.5. PARAMETERS Tab
    - 8.6. SETTINGS Tab
    - 8.7. STATISTICS Tab
9. Using QtPlasmaC
    - 9.1. Units Systems
    - 9.2. Preamble and Postamble Codes
    - 9.3. Mandatory Codes
    - 9.4. Coordinates
    - 9.5. Cut Feed Rate
    - 9.6. Material File
    - 9.7. Manual Material Handling
    - 9.8. Automatic Material Handling
    - 9.9. Material Addition Via Magic Comments In G-code
    - 9.10. Material Converter
    - 9.11. LASER
    - 9.12. CAMERA
    - 9.13. Path Tolerance
    - 9.14. Paused Motion
    - 9.15. Pause At End Of Cut
    - 9.16. Multiple Tools
    - 9.17. Velocity Reduction
    - 9.18. THC (Torch Height Controller)
    - 9.19. Cutter Compensation
    - 9.20. Initial Height Sense (IHS) Skip
    - 9.21. Probing
    - 9.22. Offset Probing
    - 9.23. Cut Types
    - 9.24. Hole Cutting - Intro
    - 9.25. Hole Cutting
    - 9.26. Hole Cutting - Automatic
    - 9.27. Single Cut
    - 9.28. Thick Materials
    - 9.29. Mesh Mode (Expanded Metal Cutting)
    - 9.30. Ignore Arc OK
    - 9.31. Cut Recovery
    - 9.32. Run From Line
    - 9.33. Scribe
    - 9.34. Spotting
    - 9.35. Virtual Keyboard Custom Layouts
    - 9.36. Keyboard Shortcuts

9.37. MDI
10. Conversational Shape Library
10.1. Conversational Settings
10.2. Conversational Lines And Arcs
10.3. Conversational Single Shape
10.4. Conversational Group Of Shapes
10.5. Conversational Block
10.6. Conversational Saving A Job
11. Error Messages
11.1. Error Logging
11.2. Error Message Display
11.3. Critical Errors
11.4. Warning Messages
12. Updating QtPlasmaC
12.1. Standard Update
12.2. Continuous Update
13. Modify An Existing QtPlasmaC Configuration
14. Customizing QtPlasmaC GUI
14.1. Add A Custom Style
14.2. Create A New Style
14.3. Returning To The Default Styling
14.4. Custom Python Code
14.5. Custom G-code Filter
15. QtPlasmaC Advanced Topics
15.1. Custom User Buttons
15.2. Peripheral Offsets (Laser, Camera, Scribe, Offset Probe)
15.3. Keep Z Motion
15.4. External HAL Pins
15.5. Hide Program Buttons
15.6. Tuning Mode o Arc OK
15.7. Lost Arc Delay
15.8. Zero Window
15.9. Tuning Void Sensing
15.10. Max Offset
15.11. Enable Tabs During Automated Motion
15.12. Override Jog Inhibit Via Z+ Jog
15.13. QtPlasmaC State Outputs
15.14. QtPlasmaC Debug Print
15.15. Hypertherm PowerMax Communications
16. Internationalisation
17. Appendix
17.1. Example Configurations
17.2. NGC Samples
17.3. QtPlasmaC Specific G-codes
17.4. QtPlasmaC G-code Examples
17.5. Mesa THCAD
17.6. RS485 Connections
17.7. Arc OK With A Reed Relay
17.8. Contact Load Schematics
18. Known Issues
18.1. Keyboard Jogging
19. Support

1. Preamble

Except where noted, this guide assumes the user is using the latest version of QtPlasmaC. Version history can be seen by visiting this link which will show the latest available version. The installed QtPlasmaC version is displayed in the title bar. See Update QtPlasmaC for information on updating QtPlasmaC.

2. License

QtPlasmaC and all of its related software are released under GPLv2.

3. Introduction

QtPlasmaC is a GUI for plasma cutting which utilises the plasmac component for controlling a plasma table from LinuxCNC v2.9 or later using the Debian Buster or similar distribution.

The QtPlasmaC GUI supports up to five axes and uses the QtVCP infrastructure provided with LinuxCNC.

The standard theme is based on a design by user "pinder" on the LinuxCNC Forum and the colors are able to be changed by the user.

The QtPlasmaC GUI will run on any hardware that is supported by LinuxCNC v2.9 or later provided there are enough hardware I/O pins to fulfill the requirements of a plasma configuration.

There are three available formats:

- 16:9 with a minimum resolution of 1366 x 768
- 9:16 with a minimum resolution of 768 x 1366
- 4:3 with a minimum resolution of 1024 x 768

Screenshot examples of QtPlasmaC are below:

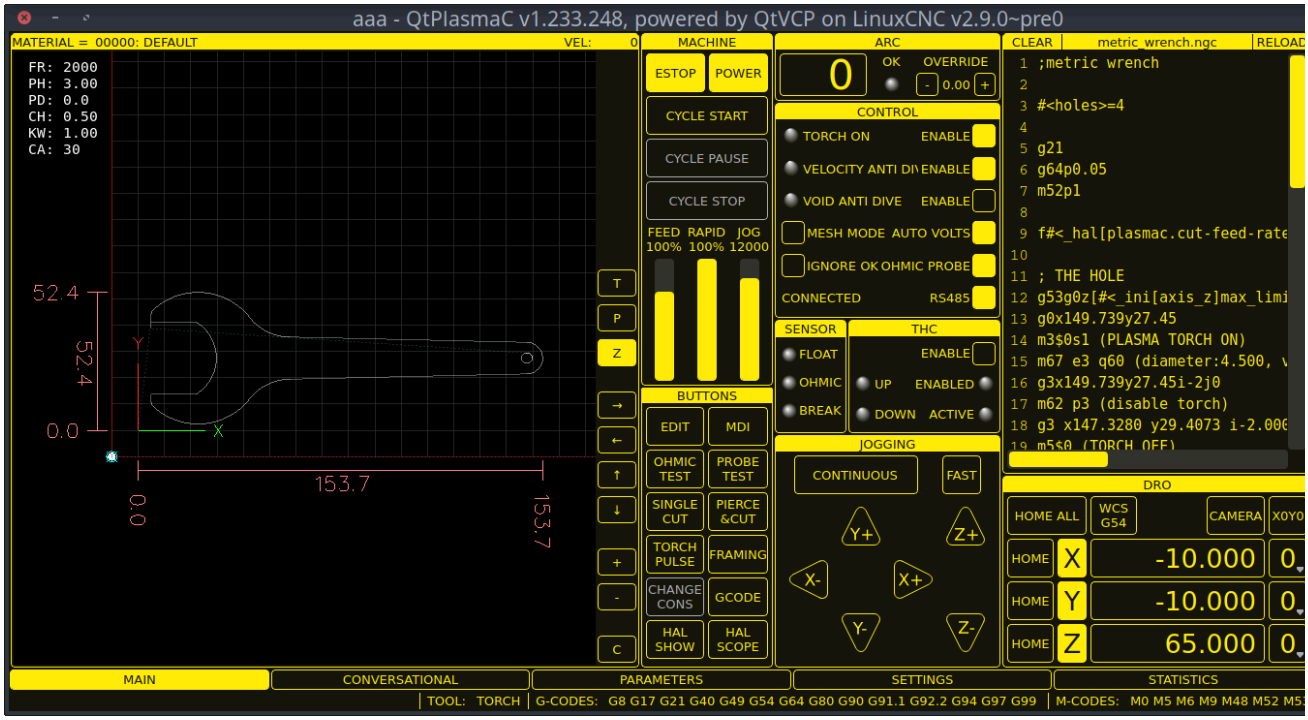


Figure 1. **16:9**

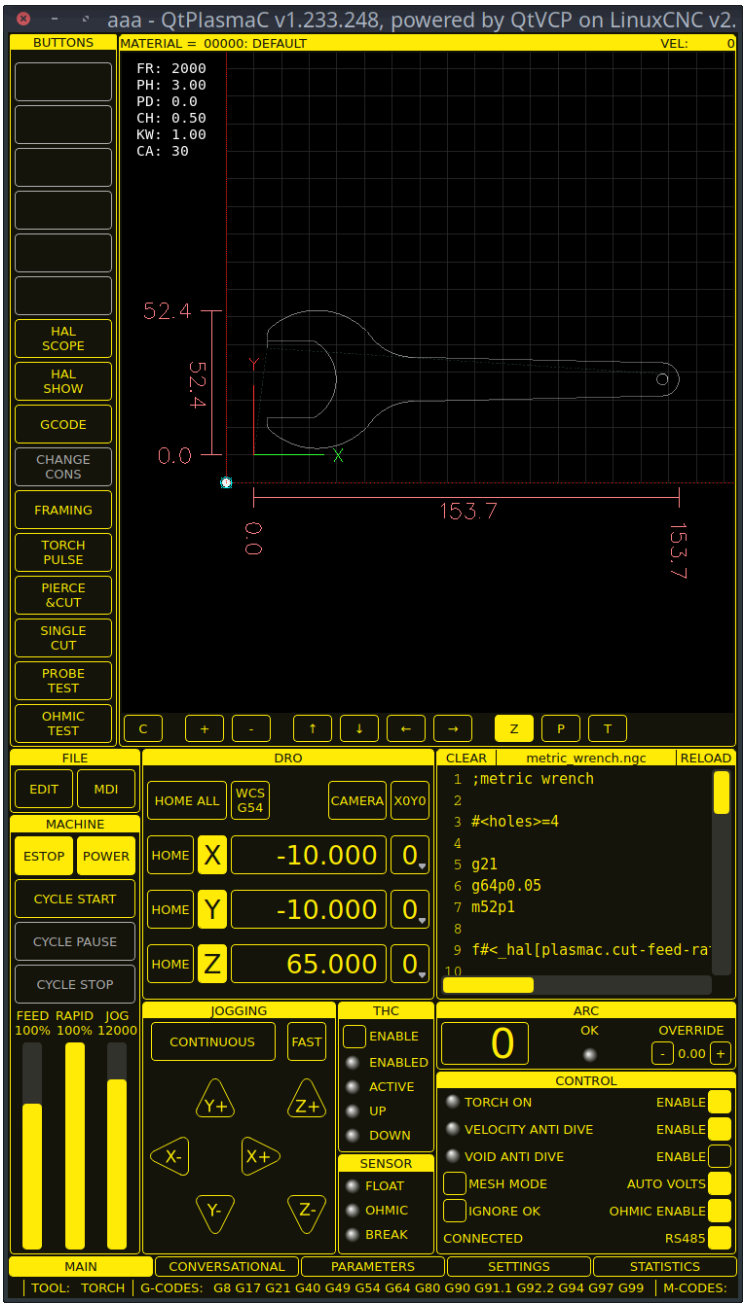

aaa - QtPlasmaC v1.233.248, powered by QtVCP on LinuxCNC v2.
BUTTONS
MATERIAL = 00000: DEFAULT
VEL: 0
FR: 2000
PH: 3.00
PD: 0.0
CH: 0.50
KW: 1.00
CA: 30
52.4
52.4
0.0
153.7
0.0
153.7
HAL SCOPE
HAL SHOW
GCODE
CHANGE CONS
FRAMING
TORCH PULSE
PIERCE &CUT
SINGLE CUT
PROBE TEST
OHMIC TEST
C
+
-
↑
↓
←
→
Z
P
T
FILE
EDIT
MDI
MACHINE
ESTOP
POWER
CYCLE START
CYCLE PAUSE
CYCLE STOP
FEED RAPID JOG
100% 100% 12000
DRO
HOME ALL
WCS G54
CAMERA
X0Y0
HOME X -10.000 0
HOME Y -10.000 0
HOME Z 65.000 0
CLEAR
metric_wrench.ngc
RELOAD
1 ;metric wrench
2
3 #<holes>=4
4
5 g21
6 g64p0.05
7 m52p1
8
9 f#<_hal[plasmac.cut-feed-ra
10
JOGGING
CONTINUOUS
FAST
Y+
Z+
X-
X+
Y-
Z-
THC
ENABLE
ENABLED
ACTIVE
UP
DOWN
SENSOR
FLOAT
OHMIC
BREAK
ARC
0
OK
OVERRIDE
- 0.00 +
CONTROL
TORCH ON
ENABLE
VELOCITY ANTI DIVE
ENABLE
VOID ANTI DIVE
ENABLE
MESH MODE
AUTO VOLTS
IGNORE OK
OHMIC ENABLE
CONNECTED
RS485
MAIN
CONVERSATIONAL
PARAMETERS
SETTINGS
STATISTICS
TOOL: TORCH
G-CODES: G8 G17 G21 G40 G49 G54 G64 G80 G90 G91.1 G92.2 G94 G97 G99
M-CODES:

Figure 2. **9:16**

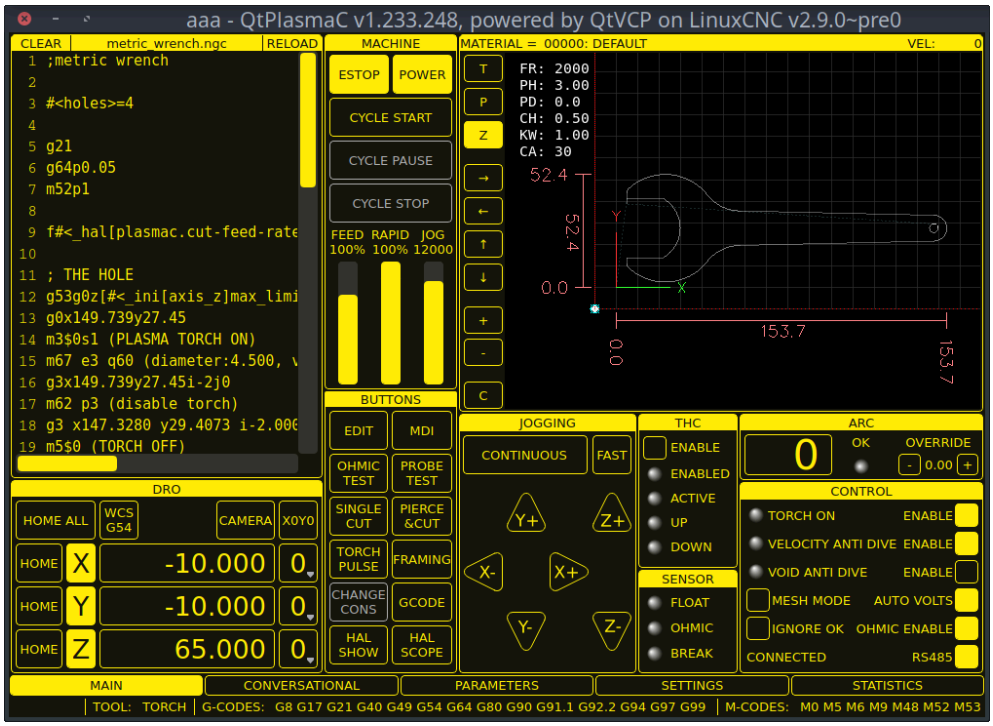


Figure 3. 4:3

4. Installing LinuxCNC

The preferred method for installing LinuxCNC is via an ISO image as described below.

> **Note** It is possible to install and run LinuxCNC on a variety of Linux distributions however that is beyond the scope of this User Guide. If the user wishes to install a Linux distribution other than those recommended, they will first need to install their preferred Linux distribution and then install LinuxCNC v2.9 or later along with any required dependencies.

4.1. If The User Does Not Have Linux Installed

Installation instructions are available from here.

Following these instructions will yield a machine with the current stable branch of LinuxCNC (v2.9) on Debian 12 (Bookworm).

4.2. Package Installation (Buildbot) If The User Has Linux on Debian 12 (Bookworm)

Follow the instructions from the Updating LinuxCNC on Debian Bookworm section from here.

4.3. Package Installation (Buildbot) If The User Has Linux on Debian 11 (Bullseye) or Debian 10 (Buster)

A package installation (Buildbot) uses prebuilt packages from the LinuxCNC Buildbot.

Add the GPG keys and add the repository to the sources list to suit the Debian version.

The below stanza would add the 2.9 Bullseye repository.

```
deb     http://buildbot2.highlab.com/ bookworm 2.9-uspace
```

4.4. Run In Place Installation If The User Has Linux Installed

A run in place installation runs LinuxCNC from a locally compiled version usually located at ~/linuxcnc-dev, instructions for building a run in place installation are available from here.

5. Creating A QtPlasmaC Configuration

Prior to creating a QtPlasmaC configuration, it is important that the user has a firm understanding of the operating modes available, as well as the I/O's that are required for successful plasma operation.

5.1. Modes

QtPlasmaC requires the selection of one of following three operating modes:

Mode	Description
0	Uses an external arc voltage input to calculate both Arc Voltage (for Torch Height Control) and Arc OK.
1	Uses an external arc voltage input to calculate Arc Voltage (for Torch Height Control). Uses an external Arc OK input for Arc OK.
2	Uses an external Arc OK input for Arc OK. Use external up/down signals for Torch Height Control.

Important If the plasma power source has an Arc OK (Transfer) output then it is recommended to use that for Arc OK rather than the soft (calculated) Arc OK provided by mode 0. It may also be possible to use a reed relay as an alternative method to establish an Arc OK signal when the power source does not provide one.

Note For fine tuning of Mode 0 Ark OK see Tuning Mode 0 Arc OK in the Advanced Topics section of the manual.

5.2. Available I/Os

Note This section only touches on the hardware I/O's required for QtPlasmaC. Base machine requirements such as limit switches, home switches, etc. are in addition to these.

Name	Modes	Description
Arc Voltage	0, 1	Analog input; **optional.** HAL pin name `plasmac.arc-voltage-in` Connected to the velocity output of an encoder equipped breakout board. This signal is used to read the arc voltage to determine the necessary corrections to maintain the torch distance from the work piece during cutting.
Arc OK	1, 2	Digital input; **optional.** HAL pin name `plasmac.arc-ok-in` Connected from the Arc OK output of the plasma power source to an input on the breakout board. This signal is used to determine if the cutting arc has been established and it is ok for the machine to move (sometimes called arc transfer).
Float Switch	0, 1, 2	Digital input; **optional, see info below table:** HAL pin name `plasmac.float-switch` Connected from a breakout board input to a switch on the floating head. This signal is used to mechanically probe the work piece with the torch and set Z zero at the top of the work piece. If used and no ohmic probe is configured, this is the probing method. If used and an ohmic probe is configured, this is the fallback probing method.
Ohmic Probe	0, 1, 2	Digital input; **optional, see info below table:** HAL pin name `plasmac.ohmic-probe` Connected from to the ohmic probe's output to a breakout board input. This

Name	Modes	Description
		signal is used to probe electronically by completing a circuit using the work piece and the torch consumables and set Z zero at the top of the work piece. If used, this is the primary probing method. If an ohmic probe fails to locate the work piece, and there is no float switch is present, probing will continue until the torch breaks away or the minimum Z limit is reached.
Ohmic Probe Enable	0, 1, 2	Digital output; **optional, see info below table:** HAL pin name `plasmac.ohmic-enable` Connected from a breakout board output to an input to control the ohmic probe's power.
Breakaway Switch	0, 1, 2	Digital input; **optional, see info below table:** HAL pin name `plasmac.breakaway` Connected from a breakout board input to a torch breakaway detection switch. This signal senses if the torch has broken away from its cradle.
Torch On	0, 1, 2	Digital output; **required.** HAL pin name `plasmac.torch-on` Connected from a breakout board output to the torch-on input of the plasma power supply. This signal is used to control the plasma power supply and start the arc.
Move Up	2	Digital input; **optional.** HAL pin name `plasmac.move-up` Connected from the up output of the external THC control to a break out board input. This signal is used to control the Z axis in an upward motion and make necessary corrections to maintain the torch distance from the work piece during cutting.
Move Down	2	Digital input; **optional.** HAL pin name `plasmac.move-down` Connected from the down output of the external THC control to a break out board input. This signal is used to control the Z axis in a downward motion and make necessary corrections to maintain the torch distance from the work piece during cutting.
Scribe Arming	0, 1, 2	Digital output; **optional.** HAL pin name `plasmac.scribe-arm` Connected from a breakout board output to the scribe arming circuit. This signal is used to place the scribe into position on the work piece .
Scribe On	0, 1, 2	Digital output; **optional.** HAL pin name `plasmac.scribe-on` Connected from a breakout board output to the scribe-on circuit. This signal is used to turn the scribing device on.
Laser On	0, 1, 2	Digital output; **optional.** HAL pin name `qtplasmac.laser_on` This signal is used to turn the alignment laser on.

Only one of either **Float Switch** or **Ohmic Probe** is required. If both are used then **Float Switch** will be a fallback if **Ohmic Probe** is not sensed.

If **Ohmic Probe** is used then **Ohmic Probe Enable** is required to be checked on the QtPlasmaC GUI.

Breakaway Switch is not mandatory because the **Float Switch** is treated the same as a breakaway when not probing. If they are two separate switches, and there are not enough inputs on the breakout board, they could be combined and connected as a **Float Switch**.

Note The minimum I/O requirement for a QtPlasmaC configuration to function are: **Arc Voltage** input OR **Arc OK** input, **Float Switch** input, and **Torch On** output. To reiterate, in this case QtPlasmaC will treat the float switch as a breakaway switch when it is not probing.

5.3. Recommended Settings:

Refer to the Heights Diagram diagram for a visual representation of the terms below.

- **[AXIS_Z] MIN_LIMIT** should be just below top of the slats with allowances for float_switch_travel and over travel tolerance. For example, if the user's float switch takes 4â□□mm (0.157") to activate then set the Z minimum to 5â□□mm (0.2") plus an allowance for overrun (either calculated using the equation below or allow 5â□□mm (0.2") below the lowest slat).
- **[AXIS_Z] MAX_LIMIT** should be the highest the user wants the Z axis to travel (it must not be lower than Z HOME_OFFSET).
- **[AXIS_Z] HOME** should be set to be approximately 5â□□mm-10â□□mm (0.2"-0.4") below the maximum limit.
- **Floating Head** - it is recommended that a floating head be used and that it has enough movement to allow for overrun during probing. Overrun can be calculated using the following formula:

```
o = 0.5 * a * (v / a)^2
```

where: o = overrun, a = acceleration in units/s^2 and v = velocity in units/s.

Metric example: given a Z axis MAX_ACCELERATION of 600â□□mm/s^2 and MAX_VELOCITY of 60â□□mm/s, the overrun would be 3â□□mm.

Imperial example: given a Z axis MAX_ACCELERATION of 24â□□in/s^2 and MAX_VELOCITY of 2.4â□□in/s, the overrun would be 0.12â□□in.

On machines that will utilize an ohmic probe as the primary method of probing, it is highly recommended to install a switch on the floating head as a backup means of stopping Z motion in the event of ohmic probe failure due to dirty surfaces.

5.4. Configuring

LinuxCNC provides two configuration wizards which can be used to build a machine configuration. The choice of these wizards is dependent on the hardware used to control the machine.

If the user wishes to use a Run In Place installation then prior to running one of the following commands they will need to run the following command from a terminal:

```
source ~/linuxcnc-dev/scripts/rip-environment
```

If using a Package installation then no additional action is required.

If using a parallel port, use the StepConf wizard by running the `stepconf` command in a terminal window or launching it using the **Application -> CNC -> StepConf Wizard** desktop menu entry.

If using a Mesa Electronics board, use the PnCconf wizard by running the `pncconf` command in a terminal window or launching it using the **Application -> CNC -> PnCConf Wizard** desktop menu entry.

If using a Pico Systems board, this LinuxCNC forum thread may be helpful.

The machine specific settings are not described here, refer to the documentation for the particular configuration wizard that is being used.

There are LinuxCNC forum sections available for these wizards:

StepConf Wizard

PnCconf Wizard

Fill in the required entries to suit the machine wiring/breakout board configuration.

QtPlasmaC adds two pages to the LinuxCNC configuration wizards for QtPlasmaC specific parameters, the two pages are QtPlasmaC options and User Buttons. Complete each of the wizards QtPlasmaC page to suit the machine that is being configured and the user button requirements.

Note that PnCconf options allow user selection of Feed Override, Linear Velocity, and Jog Increments, whereas in StepConf these are automatically calculated and set.



Figure 4. PnCConf QtPlasmaC Options

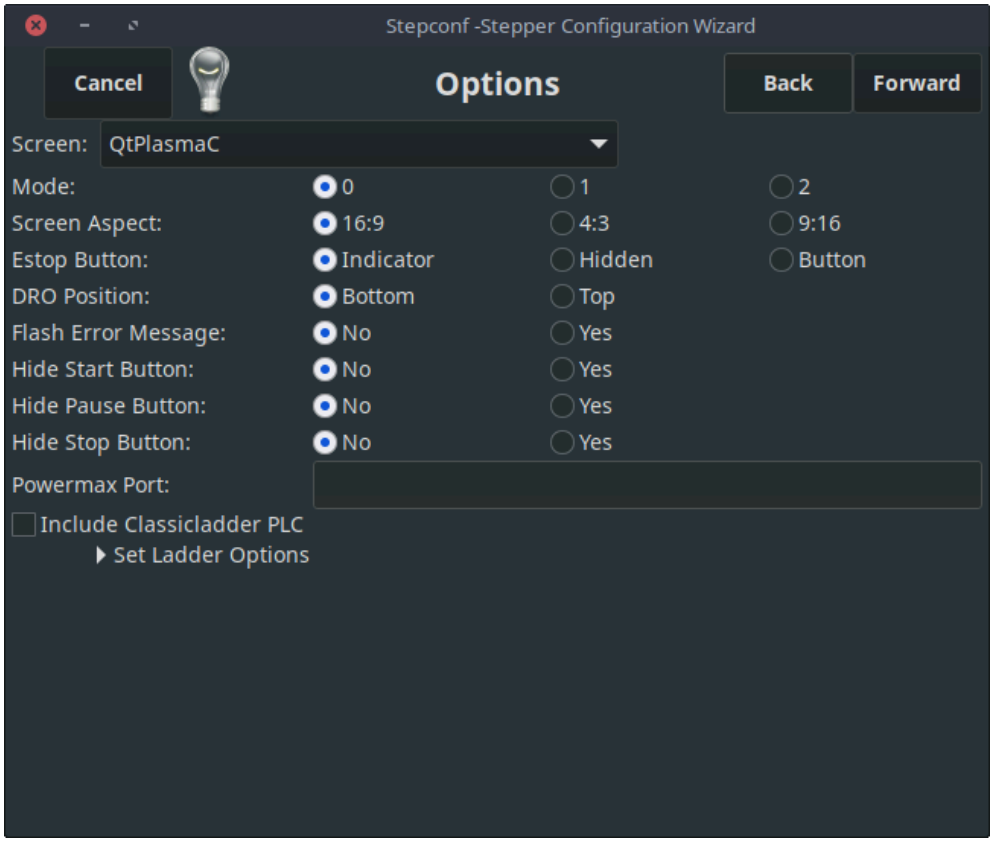


Figure 5. StepConf QtPlasmaC Options

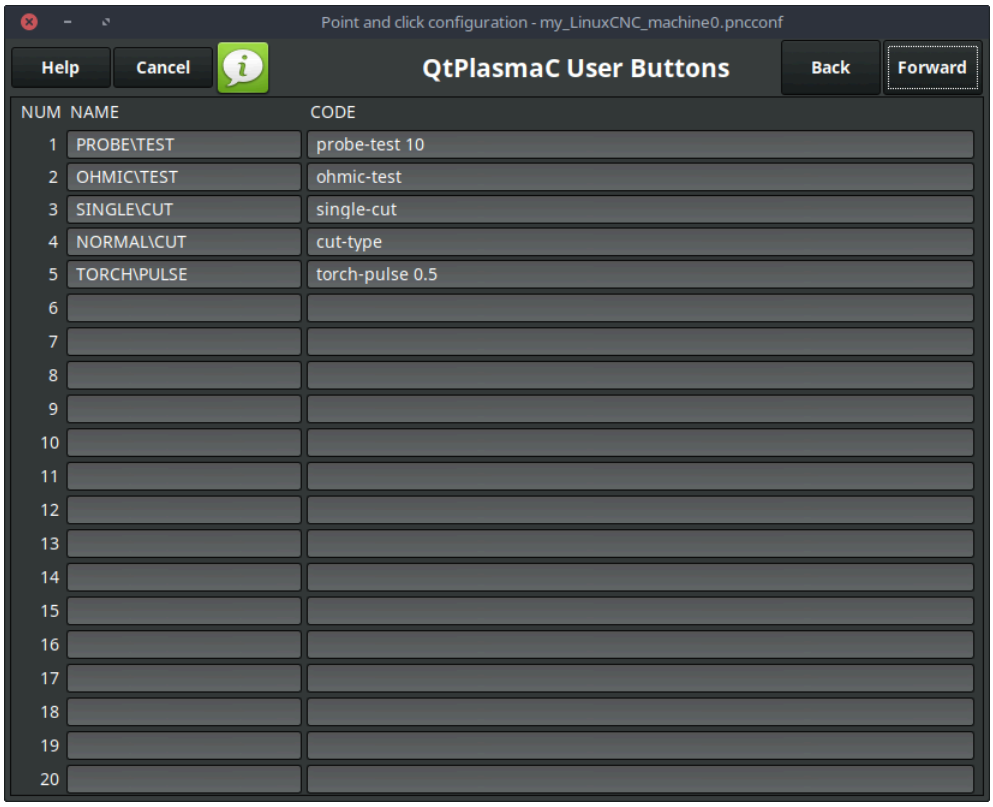


Figure 6. QtPlasmaC User Buttons

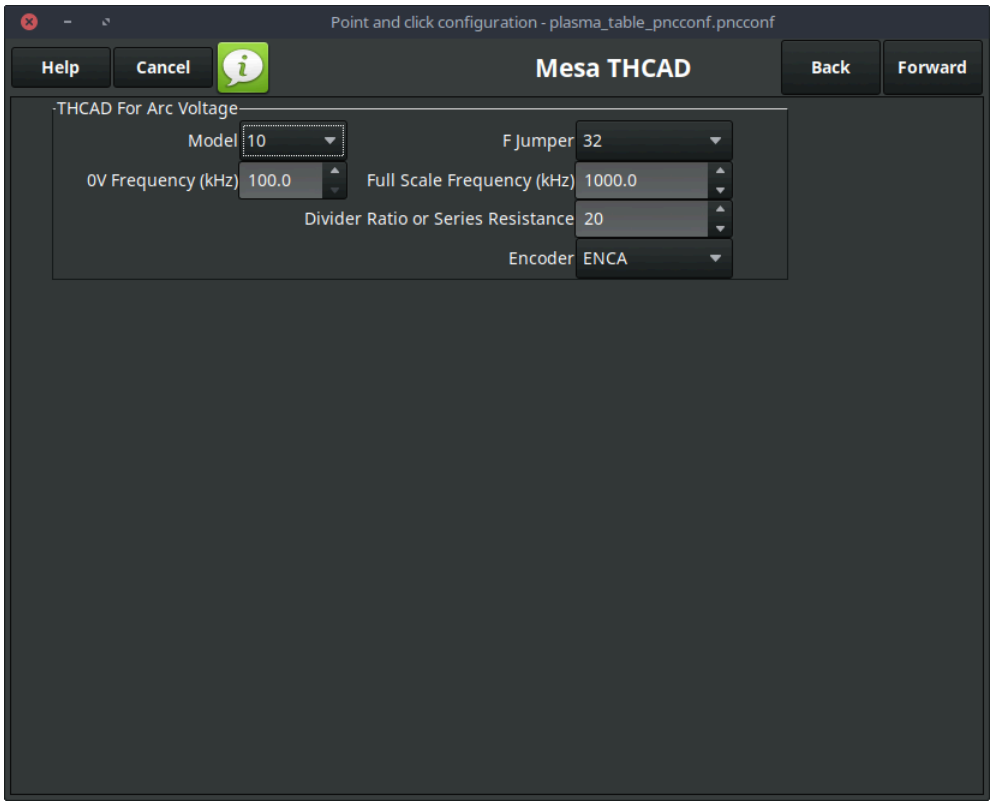


Figure 7. QtPlasmaC THCAD

The THCAD screen will only appear if a Plasma Encoder is selected in the card screen. The the dedicated section on Mesa THCAD for more information.

When the configuration is complete, the wizard will save a copy of the configuration that may be loaded and edited at a later time, a working QtPlasmaC configuration will be created in the following directory: ~/linuxcnc/configs/*<machine_name>*.

The way the newly created QtPlasmaC configuration can be run from the terminal command line slightly differs depending the way LinuxCNC was installed:

For a package installation (Buildbot):

```
linuxcnc ~/linuxcnc/configs/_<machine_name>_/_<machine_name>_.ini
```

For a run in place installation:

```
~/linuxcnc-dev/scripts/linuxcnc
~/linuxcnc/configs/_<machine_name>_/_<machine_name>_.ini
```

After running the above command LinuxCNC should be running with the QtPlasmaC GUI visible.

Important BEFORE PROCEEDING, THE USER SHOULD BE ABLE TO HOME THE MACHINE, ZERO EACH AXIS, JOG ALL AXES TO SOFT LIMITS WITHOUT CRASHING, AND RUN TEST G-CODE PROGRAMS WITHOUT ANY ERRORS.

ONLY WHEN this criteria is met should the user proceed with the QtPlasmaC initial setup.

Note It is possible to create a sim configuration using StepConf but it is not possible to have tandem joints in the sim configuration.

5.5. Qt Dependency Errors

If any Qt dependency errors are encountered while attempting to run the QtPlasmaC configuration, the user may need to run the QtVCP installation script to resolve these issues.

For a package installation (Buildbot) enter the following command in a terminal window:

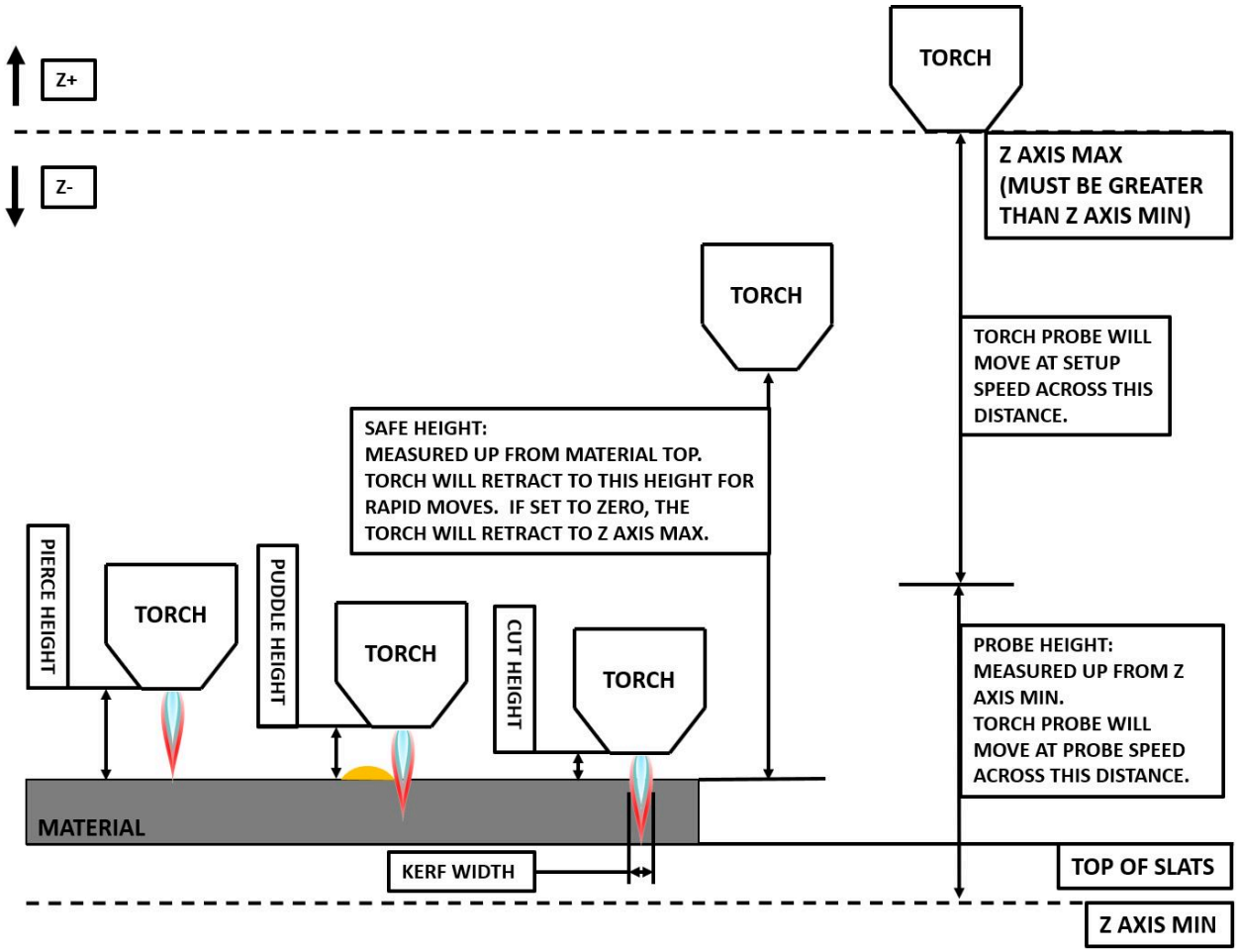
```
/usr/lib/python3/dist-packages/qtvcp/designer/install_script
```

For a run in place installation enter the following command in a terminal window:

```
~/linuxcnc-dev/lib/python/qtvcp/designer/install_script
```

5.6. Initial Setup

The following heights diagram will help the user visualize the different heights involved in plasma cutting and how they are measured:



Click on the Parameters Tab to view the **CONFIGURATION** section which shows the user settable parameters. It is necessary to ensure every one of these settings is tailored to the machine.

To set the Z axis DRO relative to the Z axis MINIMUM_LIMIT, the user should perform the following steps. It is important to understand that in QtPlasmaC, touching off the Z axis DRO has no effect on the Z axis position while running a G-code program. These steps simply allow the user to more easily set the probe height as after performing the steps, the displayed Z axis DRO value will be relative to Z axis MINIMUM_LIMIT.

Note The user should be familiar with the recommended Z Axis Settings.

1. Home the Z axis.
2. Ensure there is nothing below the torch then jog the Z axis down until it stops at the Z axis MINIMUM_LIMIT then click the 0 next to the Z axis DRO to **Touch Off** with the Z axis selected to set the Z axis at zero offset. This step only serves to allow the user to more easily visualize and adjust **Probe Height** this value is measured from the Z axis MINIMUM_LIMIT up.
3. Home the Z axis again.

Probe Test

If the machine is equipped with a float switch then the user will need to set the offset in the **CONFIGURATION** section of the **PARAMETERS** tab. This will be done by running a "Probe Test" cycle.

1. Check that the Probe Speed and the Probe Height in the **CONFIGURATION** section of the **PARAMETERS** tab are correct. QtPlasmaC can probe at the full Z axis velocity so long as the machine has enough movement in the float switch to absorb any overrun. If the machine is suitable, the user could set the Probe Height to a value near the Z axis minimum and do all probing at full speed.
2. If the machine is not already homed and in the home position, home the machine.
3. Place some material on the slats under the torch.
4. Press the **PROBE TEST** button.
5. The Z axis will probe down, find the material then move up to the specified **Pierce Height** as set by the currently selected material. The torch will wait in this position for the time set in the

<machine_name>.prefs file. The default probe test hold time is 10 seconds, this value may be edited in the *<machine_name>*.prefs file. After this the torch will return to the starting height.

6. Measure the distance between the material and the tip of the torch while the torch is waiting at **Pierce Height**.
7. If the measurement is greater than the **Pierce Height** of the currently selected material, then reduce the "Float Travel" in the **CONFIGURATION** section of the **PARAMETERS** tab by the difference between the measured value and the specified value. If the measurement is less than **Pierce Height** of the currently selected material, then increase the "Float Travel" in the **CONFIGURATION** section of the **PARAMETERS** tab by the difference between the specified value and the measured value.
8. After the adjustments to the "Float Travel" have been made, repeat the process from #4 above until the measured distance between the material and the torch tip matches the **Pierce Height** of the currently selected material.
9. If the table has a laser or camera for sheet alignment, a scribe, or uses offset probing then the required offsets need to be applied by following the procedure described in Peripheral Offsets.
10. CONGRATULATIONS! The user should now have a working QtPlasmaC Configuration.

Note If the amount of time between the torch contacting the material and when the torch moves up and comes to rest at the Pierce Height seems excessive, see the probing section for a possible solution.

Important IF USING A **Mesa Electronics THCAD** THEN THE **Voltage Scale** VALUE WAS OBTAINED MATHEMATICALLY. IF THE USER INTENDS TO USE CUT VOLTAGES FROM A MANUFACTURE'S CUT CHART THEN IT WOULD BE ADVISABLE TO DO MEASUREMENTS OF ACTUAL VOLTAGES AND FINE TUNE THE **Voltage Scale** AND **Voltage Offset**.

Warning PLASMA CUTTING VOLTAGES CAN BE LETHAL, IF THE USER IS NOT EXPERIENCED IN DOING THESE MEASUREMENTS GET SOME QUALIFIED HELP.

6. Migrating to QtPlasmac From PlasmaC (AXIS or GMOCCAPY)

There are two methods available to get from a working PlasmaC configuration to a new QtPlasmaC configuration. These methods assume the user is on LinuxCNC v2.9 or later, QtVCP is installed, and all dependency requirements are satisfied.

If there are Qt dependency errors, the user should run the QtVCP install script.

6.1. Quick Method

A quick method to move to QtPlasmaC from PlasmaC (loaded on top of either AXIS or GMOCCAPY) is to use the plasmac2qt conversion program, which will attempt to create a new QtPlasmaC configuration from an existing PlasmaC INI file. This program will convert the user's parameters, settings, and materials from the previous PlasmaC configuration and create a new QtPlasmaC configuration directory in the ~/linuxcnc/configs directory.

This methods will keep the original PlasmaC config as a backup with _plasmac and a time stamp appended to the directory name.

To run the plasmac2qt conversion program, use the following instructions:

For a package installation (Buildbot) enter the following line in a terminal window:

```
qtplasmac-plasmac2qt
```

For a run in place installation enter the following lines in terminal window:

```
source ~/linuxcnc-dev/scripts/rip-environment
qtplasmac-plasmac2qt
```

The following screen will be displayed:

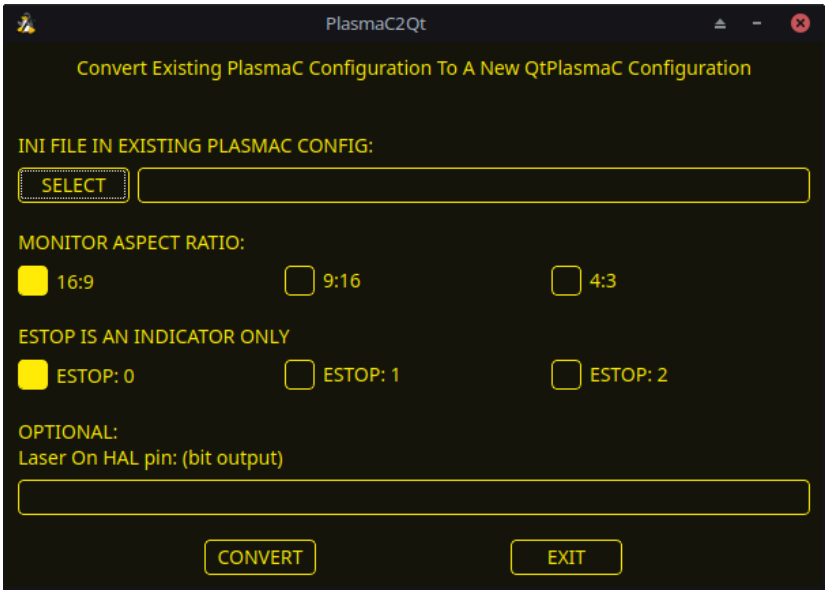


Table 1. Mandatory Settings

Field	Description	Examples
INI FILE IN EXISTING PLASMAC CONFIG	This is the INI file of the PlasmaC config that requires migrating.	*<machine_name>*.ini
MONITOR ASPECT RATIO	This is the aspect ratio format for the GUI.	16:9
ESTOP	Selects the required E-stop type based on the following criteria: 0 - Estop is an indicator only. 1 - Estop indicator is hidden. 2 - Estop is a button.	ESTOP:1

Optional Setting - This setting is not required unless the machine has a laser for sheet alignment. Leave this blank if it is not used/required.

Leave this blank if it is not used/required.

Field	Description	Examples
Laser On HAL Pin	Power on a laser crosshair for sheet alignment.	**Parallel Port Example:** parport.0.pin-16-out **Mesa 7i96 Example:** hm2_7i96.0.ssr.00.out-00

After filling in the appropriate entries, press **CONVERT**.

Note This method will not change any existing debounce components to the new dbounce component. If the user wishes to change to the new dbounce component then the New Base Config method should be used for migration.

6.2. New Base Config Method

This method to move to QtPlasmaC from PlasmaC (loaded on top of either AXIS or GMOCCAPY) is to use a configuration wizard to create a new configuration. This method then allows changing of the base machine configuration at a later date via the configuration wizard, provided that the base INI and base HAL files have not been edited.

This method requires that the user take note of all HAL pins used in the existing config so they can be entered into the configuration wizard. Any custom HAL commands will also need to be noted and added manually to either the custom.hal file or the custom_postgui.hal file which will be created by the configuration wizard.

After using the wizard, the user can then run a conversion program (cfg2prefs) to convert the parameters, settings, and materials from the previous PlasmaC configuration to the new QtPlasmaC configuration. This tool should be used immediately after the user has created a new QtPlasmaC configuration.

Prior to running this conversion program, it is mandatory that the user have both an existing PlasmaC configuration and a new QtPlasmaC configuration. This program **will overwrite** the existing QtPlasmaC preferences and materials files, and should be used with caution if it is not being run on a new QtPlasmaC configuration.

The program will create a time-stamped backup of the original preferences file and the existing materials file (if it exists).

It will read the existing *<machine_name>config.cfg, <machine_name>_run.cfg, <machine_name>_wizards.cfg, and plasmac_stats.var files and write them to an existing _<machine_name>*.prefs file. It will also copy the *<machine_name>*_material.cfg file to the existing QtPlasmaC configuration.

To run the cfg2prefs conversion program, use the following instructions:

For a package installation (Buildbot) enter the following line in a terminal window:

```
qtplasmac-cfg2prefs
```

For a run in place installation enter the following lines in terminal window:

```
source ~/linuxcnc-dev/scripts/rip-environment
qtplasmac-cfg2prefs
```

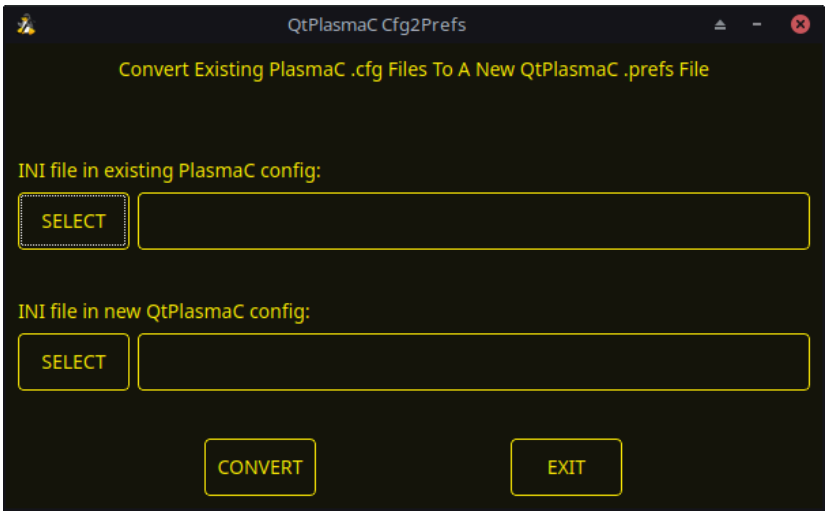


Figure 8. `qtplasmac-cfg2prefs`

Select the INI file of the old PlasmaC configuration, select the INI file of the new QtPlasmaC configuration, then press **CONVERT**.

7. Other QtPlasmaC Setup Considerations

7.1. Lowpass Filter

The plasmac HAL component has a built in lowpass filter that if used is applied to the **plasmac.arc-voltage-in** input pin to filter any noise that could cause erroneous voltage readings. The lowpass filter should only be used after using Halscope to determine the required frequency and whether the amplitude of the noise is large enough to cause any issues. For most plasma machines lowpass is not required and should not be used unless it is required.

The HAL pin assigned to this filter is **plasmac.lowpass-frequency** and is set to 0 (disabled) by default. To apply a lowpass filter to the arc-voltage, the user would edit the following entry in the custom.hal file in the machine's configuration directory to add the appropriate cutoff frequency as measured in Hertz (Hz).

For example:

```
setp plasmac.lowpass-frequency 100
```

The above example would give a cutoff frequency of 100Hz.

7.2. Contact Bounce

Contact bounce from mechanical relays, switches, or external interference may cause some inconsistent behavior of the following switches:

- Float Switch
- Ohmic Probe
- Breakaway Switch
- Arc OK (for modes 1 & 2)

Due to the fact that the software is capable of sampling rates faster than the contact bounce period, it is possible that the software may see contact bounce as several changes in input states occurring in a very small time period, and incorrectly interpret this as a very quick on-off of the input. One method of mitigating contact bounce is to "debounce" the input. To summarize debounce, it requires the input state to be stable at the opposite state of the output state for consecutive delay periods before changing the state of the output.

Debounce delay periods can be changed by editing the appropriate debounce value in the custom.hal file in the *<machine_name>* config directory.

Each increment of delay adds one servo thread cycle to the debounce time. For example: given a servo thread period of 1000000 (measured in nano seconds), a debounce delay of 5 would equate to 5000000â□□ns, or 5â□□ms.

For the Float and Ohmic switches this equates to a 0.001â□□mm (0.00004") increase in the probed height result.

It is recommended to keep the debounce values as low as possible while still achieving consistent results. Using Halscope to plot the inputs is a good way to establish the correct value.

For QtPlasmaC installations, debounce is achieved by using the HAL dbounce component which is a later alternative to the original debounce component. This new version allows for the loading and naming of individual debounce instances and is compatible with Twopass HAL file processing.

All four signals above have an individual debounce component so the debounce periods can be catered individually to each input. Any changes made to these values in the custom.hal file will not be overwritten by later updates of QtPlasmaC.

The default delay for all four inputs is five servo thread periods. In most cases this value will work quite well. If any of the inputs do not use mechanical switches, it may be possible to either reduce or remove the delay for those inputs.

If debounce is required for other equipment like home or limit switches etc. then more dbounce components may added in any of the HAL files without any regard to the signals listed here.

7.3. Contact Load

Mechanical relays and switches usually require a minimum current passing through the contacts for reliable operation. This current varies with the material that the contacts in the device are made from.

Depending on the specified minimum contact current and the current drawn by the input device there may be a need to provide a method to increase the current through the contacts.

Most relays using gold contacts will not require any additional current for reliable operation.

There are two different methods available to provide this minimum current if it is required:

1. A 0.1â□□Î¼F film capacitor placed across the contacts.
2. A 1200â□□Î© 1â□□W resistor across the load (see calculations below).

Schematics are shown at contact load schematics.

More information on contact switching load can be seen on page VI of the finder General Technical Information document.

Calculations:

If using a Mesa card, the input resistance of a 7I96 is 4700â□□Î© (symbol R)(always consult the product manual associated with the revision being used as these values sometimes vary between revisions), giving a contact current of 5.1â□□mA (symbol I) assuming a supply voltage (symbol U) of 24â□□V (I = U/R)[1].

As an example, the typical relay used in a Hypertherm Powermax 65 plasma cutter (TE T77S1D10-24) requires a minimum contact load of 100â□□mA @ 5â□□VDC which will dissipate 0.5â□□W (P = I * V). If using a 24â□□VDC power supply this would then equate to a minimum current of 20.8â□□mA. Because there is less current drawn by the Mesa input than is required by the relay there needs to be an increase in the current.

The resistance can be calculated using $R = U_s / (I_m - I_i)$ where:

- R = calculated resistance
- U_s = supply voltage
- I_m = minimum current required
- I_i = input current

Using a 7I96 with an input current of 5.1â□□mA gives a calculated value of 1529â□□Î© (= 24â□□V / (.0208 - .0051)â□□A). This could then be rounded down to a commonly available 1500â□□Î© resistor, giving a small safety margin.

The power dissipation can by calculated using $P = U_s^2 / R_s$ where:

- P = power
- U_s = supply voltage
- R_s = selected resistance

This gives a value of 0.38â□□W. This could then be rounded up to 1â□□W, giving a good safety margin. The final selection would be a 1500â□□Î© 1â□□W resistor.

7.4. Desktop Launcher

If a link to the launch the configuration was not created when creating the config, the user could create a desktop launcher to the config by right clicking on the desktop and selecting Create Launcher or similar. This will bring up a dialog box to create a launcher. Give the icon a nice short name, enter anything for the command and click OK.

After the launcher appears on the desktop, right click on it and then edit it with the user's editor of choice. Edit the file so it looks similar to:

```
[Desktop Entry]
Comment=
Terminal=false
Name=LinuxCNC
Exec=sh -c "linuxcnc
$HOME/linuxcnc/configs/<machine_name>/<machine_name>.ini"
Type=Application
Icon=/usr/share/pixmaps/linuxcncicon.png
```

If the user would like a terminal window to open behind the GUI window then change the Terminal line to:

```
Terminal=true
```

Displaying a terminal can be handy for error and information messages.

7.5. QtPlasmaC Files

After a successful QtPlasmaC installation, the following files are created in the configuration directory:

Filename	Function
<machine_name>.ini	Configuration file for the machine.
<machine_name>.hal	HAL for the machine.
<machine_name>.prefs	Configuration file for QtPlasmaC specific parameters and preferences.
custom.hal	HAL file for user customization.
custom_postgui.hal	HAL file for user customization which is run after the GUI has initialized.
shutdown.hal	HAL file which is run during the shutdown sequence.
tool.tbl	Tool table used to store offset information for additional tools (scribe, etc.) used by the QtPlasmaC configuration.
qtplasmac	Link to the directory containing common qtplasmac support files.
backup	Directory for backups of config files.

Note *<machine_name>* is whatever name the user entered into the "Machine Name" field of the configuration wizard program.

Note Custom commands are allowed in custom.hal and the custom_postgui.hal files as they are not overwritten during updates.

After running a new configuration for the first time the following files will be created in the configuration directory:

Filename	Function
<machine_name>_material.cfg	File for storing the material settings from the MATERIAL section of the PARAMETERS Tab.
update_log.txt	File for storing log of major updates. Major updates are those that make any modification to a user's configuration.
qtvcp.prefs	File containing the QtVCP preferences.
qtplasmac.qss	File storing the stylesheet for the currently loaded session of QtPlasmaC.

Note The configuration files (*<machine_name>*.ini and *<machine_name>*.hal) that are created by configuration wizard are notated to explain the requirements to aid in manual manipulation of these configurations. They may be edited with any text editor.

Note The *<machine_name>*.prefs file is plain text and may be edited with any text editor.

7.6. INI File

QtPlasmaC has some specific *<machine_name>*.ini file variables as follows:

[FILTER] Section

These variables are mandatory.

```
PROGRAM_EXTENSION = .ngc,.nc,.tap G-code File (*.ngc, *.nc, *.tap)
ngc               = qtplasmac_gcode
```

```
nc                = qtplasmac_gcode
tap               = qtplasmac_gcode
```

[RS274NGC] Section

These variables are mandatory.

```
RS274NGC_STARTUP_CODE = G21 G40 G49 G80 G90 G92.1 G94 G97 M52P1
SUBROUTINE_PATH       = ./:../../nc_files
USER_M_PATH           = ./:../../nc_files
```

Note for a imperial config replace G21 above with G20.

Note both the above paths show the minimum requirements.

Important SEE PATH TOLERANCE FOR RS274NGC_STARTUP_CODE INFORMATION RELATED TO G64.

[HAL] Section

These variables are mandatory.

```
HALUI           = halui (required)
HALFILE         = _<machine_name>_.hal (the machine HAL file)
HALFILE         = plasmac.tcl (the standard QtPlasmaC HAL file )
HALFILE         = custom.hal (Users custom HAL commands)
POSTGUI_HALFILE = postgui_call_list.hal (required)
SHUTDOWN        = shutdown.hal (shutdown HAL commands)
```

Note The user could place custom HAL commands in the custom.hal file as this file is not overwritten by QtPlasmaC updates.

[DISPLAY] Section

This variable is mandatory.

```
DISPLAY = qtvcp qtplasmac      (use 16:9 resolution)
        = qtvcp qtplasmac_9x16 (use 9:16 resolution)
        = qtvcp qtplasmac_4x3  (use 4:3 resolution)
```

There are multiple QtVCP options that are described here: QtVCP INI Settings

For example the following would start a 16:9 resolution QtPlasmaC screen in full screen mode:

```
DISPLAY = qtvcp -f qtplasmac
```

[TRAJ] Section

This variable is mandatory.

```
SPINDLES = 3
```

[AXIS_X] Section

These variables are mandatory.

```
MAX_VELOCITY     = double the value in the corresponding joint
MAX_ACCELERATION = double the value in the corresponding joint
OFFSET_AV_RATIO  = 0.5
```

[AXIS_Y] Section

These variables are mandatory.

```
MAX_VELOCITY     = double the value in the corresponding joint
MAX_ACCELERATION = double the value in the corresponding joint
OFFSET_AV_RATIO  = 0.5
```

[AXIS_Z] Section

These variables are mandatory.

```
MIN_LIMIT        = just below the top of the table's slats
MAX_VELOCITY     = double the value in the corresponding joint
MAX_ACCELERATION = double the value in the corresponding joint
OFFSET_AV_RATIO  = 0.5
```

Note QtPlasmaC uses the LinuxCNC External Offsets feature for all Z axis motion, and for moving the X and/or Y axis for a consumable change while paused. For more information on this feature, please read External Axis Offsets in the LinuxCNC documentation.

8. QtPlasmaC GUI Overview

The following sections will give a general overview of the QtPlasmaC layout.

8.1. Exiting QtPlasmaC

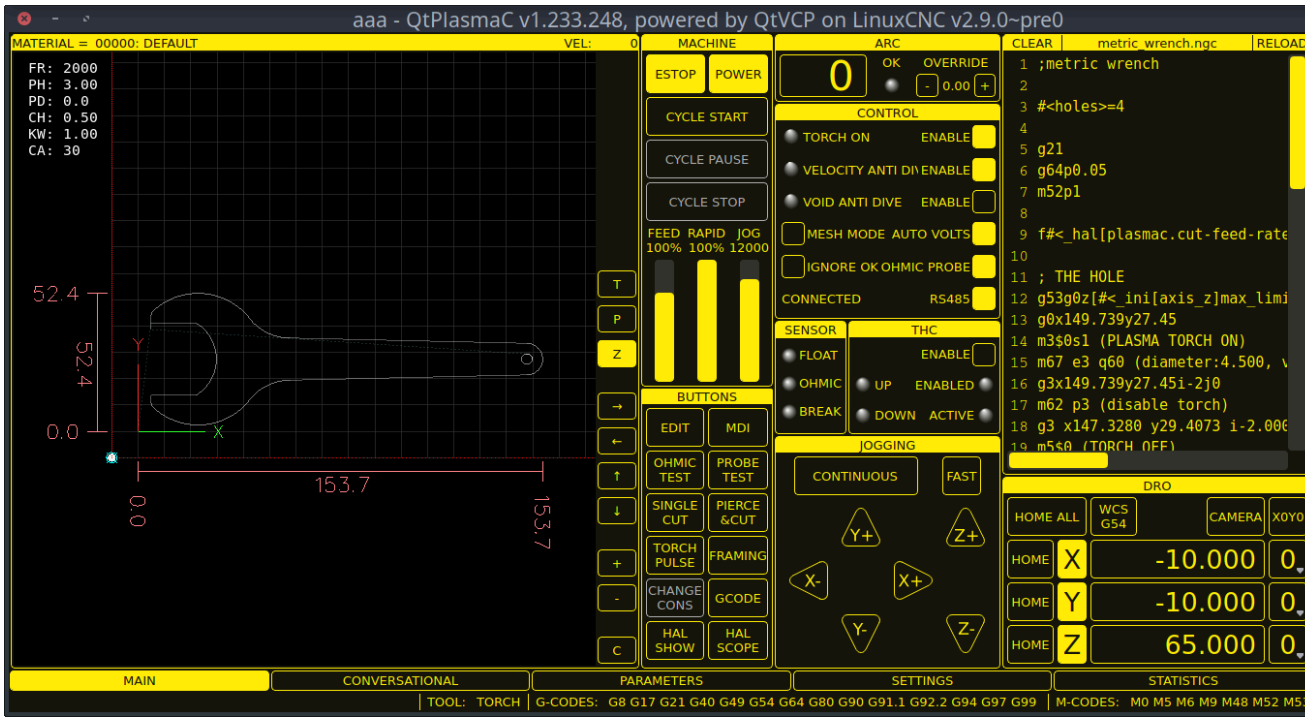
Exiting or shutting down QtPlasmaC is done by either:

1. Click the window shutdown button on the window title bar
2. Long press the **POWER** button on the MAIN Tab.

A shutdown warning can be displayed on every shutdown by checking the **Exit Warning** checkbox on the SETTINGS Tab.

8.2. MAIN Tab

Screenshot example of the QtPlasmaC MAIN Tab in **16:9** aspect ratio:



Some functions/features are only used for particular modes and are not displayed if they are not required by the chosen QtPlasmaC mode.

Table 2. Features of the PREVIEW WINDOW

Name	Description
Material	The top header is clickable in this area to reveal a drop down menu. It is used to manually select the current material cut parameters. If there are no materials in the material file then only the default material will be displayed.
VEL:	This displays the actual cut feed rate the table is moving at.
FR:	If "View Material" is selected on the SETTINGS Tab, this displays the currently selected material's Feed Rate.
PH:	If "View Material" is selected on the SETTINGS Tab, this displays the currently selected material's Pierce Height.
PD:	If "View Material" is selected on the SETTINGS Tab, this displays the currently selected material's Pierce Delay.
CH:	If "View Material" is selected on the SETTINGS Tab, this displays the currently selected material's Cut Height.
CA:	If "View Material" is selected on the SETTINGS Tab, and RS485 communications are enabled, this displays the currently selected material's Cut Amperage.
T	This button changes the preview to a top down full table view.
P	This button changes the preview to an isometric view.
Z	This button changes the preview to a top down view.
â☐☐	This button pans the preview right.
â☐☐	This button pans the preview left.
â☐☐	This button pans the preview up.
â☐☐	This button pans the preview down.
+	This button zooms the preview.

Name	Description
-	This button zooms the preview.
C	This button clears the live plot.

Table 3. **MACHINE** representation

Name	Description
ESTOP	Setting Estop type = 0 in the **[GUI_OPTIONS]** section of the *<machine_name>*.prefs file, will change this button to an indicator of the hardware E-stop's status only. This is the default behavior. Setting the option Estop type = 1 in the **[GUI_OPTIONS]** section of the *<machine_name>*.prefs file, will hide this button. Setting the option Estop type = 2 in the **[GUI_OPTIONS]** section of the *<machine_name>*.prefs file, will enable this button to act as a GUI E-stop.
POWER	This button turns the GUI on and allows QtPlasmaC/LinuxCNC to control the hardware. Pressing and holding the **POWER** button for longer than two seconds will bring up a dialog to exit the QtPlasmaC application.
CYCLE START	This button starts the cycle for any loaded G-code file.
CYCLE PAUSE	This button pauses the cycle for any loaded G-code file. If a cycle is paused, this button will display **CYCLE RESUME** and flash. Pressing **CYCLE RESUME** will resume the cycle.
CYCLE STOP	This button stops any actively running or paused cycle. This includes: - G-code Programs - Torch pulse if the pulse was started during **CYCLE PAUSE** (this will cancel the paused G-code program execution as well) - Probe Test - Framing - Manual Cut
FEED	This slider overrides the feed rate for all feed moves. Any value other than 100% will cause the label to flash. Clicking the label will return the slider to 100%.
RAPID	This slider overrides the rapid rate for all rapid moves. Any value other than 100% will cause the label to flash. Clicking the label will return the slider to 100%.
JOG	This slider sets the jog rate. Clicking the label will return the slider to the default linear velocity as set in the *<machine_name>*.ini file.

BUTTONS

The Button Panel contains buttons useful for the operation of the machine.

The **EDIT** and **MDI** buttons are permanent, all other buttons are user programmable in the *<machine_name>*.prefs file.

See custom user buttons for detailed information on custom user buttons.

Name	Description
EDIT	This button opens a G-code editor for the currently loaded program.

Name	Description
MDI	This button places QtPlasmaC into Manual Data Input (MDI) mode which will display the MDI HISTORY and an entry box over top of the G-code window. Once pressed, this button will display "MDI CLOSE". Pressing **MDI CLOSE** will close the MDI. Please see the MDI section for additional MDI information.
OHMIC TEST	This button will enable the Ohmic Probe Enable output signal and if the Ohmic Probe input is sensed, the LED indicator in the SENSOR Panel will light. The main purpose of this is to allow a quick test for a shorted torch tip.
PROBE TEST	This button will initiate a Probe Test.
SINGLE CUT	This button will show the dialog box to start an automatic Single Cut.
NORMAL CUT	This button will toggle between Cut Types (NORMAL CUT and PIERCE ONLY).
TORCH PULSE	This button will initiate a Torch Pulse.

Table 4. ARC

Name	Modes	Description
Arc Voltage	0, 1	Displays the actual arc voltage.
OK	0, 1, 2	Indicates the status of the Arc OK signal.
+	0, 1	Each press of this button will raise the target voltage by the THC Threshold voltage (The distance changed will be Height Per Volt * THC Threshold voltage).
-	0, 1	Each press of this button will lower the target voltage by the THC Threshold voltage (The distance changed will be Height Per Volt * THC Threshold voltage).
OVERRIDE	0, 1	Clicking this label will return any voltage override to 0.00.

Table 5. CONTROL

Name	Modes	Description
TORCH ON	0, 1, 2	Indicates the status of the Torch On output signal.
TORCH ON ENABLE	0, 1, 2	This box toggles between Enabling and Disabling the torch. This box defaults to unfilled (disabled) when QtPlasmaC is first run. This box must be filled to change it to "Torch Enabled" before material cutting can commence. If this box is not filled, then running a loaded program will cause the machine to run the cycle without firing the torch. This is sometimes referred to as a "dry run".
VELOCITY ANTI DIVE	0, 1, 2	Indicates that the THC is locked at the current height due to the cut velocity falling below the Velocity Anti Dive (VAD) Threshold percentage set on the PARAMETERS Tab.
VELOCITY ANTI DIVE ENABLE	0, 1, 2	This box toggles between Enabling and Disabling VELOCITY ANTI DIVE.
VOID ANTI DIVE	0, 1	Indicates that the THC is locked due to a void being sensed.
VOID ANTI DIVE ENABLE	0, 1	This box toggles between Enabling and Disabling VOID ANTI DIVE.
MESH MODE	0, 1, 2	This box will enable or disable Mesh Mode for the cutting of expanded metal. This check box may be enabled or disabled at any time during normal cutting.

Name	Modes	Description
		Mesh mode: - Will require an Arc OK signal to start machine motion. - Will disable the THC. - Will not stop machine motion if the Arc OK signal is lost. - Will automatically select CPA mode if PowerMax communications are being used. For more information see Mesh Mode (expanded metal).
AUTO VOLTS	0, 1	This box will enable or disable Auto Volts.
IGNORE OK	0, 1, 2	This box will determine if QtPlasmaC ignores the Arc OK signal. This check box may be enabled or disabled at any time during normal cutting. Additionally this mode may be enabled or disabled via proper M codes in a running program. Ignore Arc OK mode: - Will not require an Arc OK signal be received before starting machine motion after the "Torch On" signal is given. - Will disable the THC. - Will not stop machine motion if the Arc OK signal is lost. For more information see Ignore Arc Ok.
OHMIC PROBE	0, 1, 2	This box enables or disables the ohmic probe input. If the Ohmic Probe input is disabled, the Ohmic Probe LED will still show the status of the probe input, but the Ohmic Probe results will be ignored.
RS485	0, 1, 2	This box will enable or disable the communications to a PowerMax. This button is only visible if a PM_PORT option is configured in the `[POWERMAX]` section of the *<machine_name>*.prefs file.
Status	0, 1, 2	When PowerMax communications are enabled, this will display one of the following: **CONNECTING**, **CONNECTED**, **COMMS ERROR**, or a **Fault Code**. For more information, see the PowerMax Communications section.

Table 6. SENSOR

Name	Description
FLOAT	Indicates that the float switch is activated.
OHMIC	Indicates that the probe has sensed the material.
BREAK	Indicates that the torch breakaway sensor is activated.

Table 7. THC

Name	Description
ENABLE	This box determines whether the THC will be enabled or disabled during a cut.
ENABLED	This LED indicates whether the THC is enabled or disabled.
ACTIVE	This LED indicates that the THC is actively controlling the Z axis.
UP	This LED indicates that the THC is commanding the Z axis to raise.
DOWN	This LED indicates that the THC is commanding the Z axis to lower.

Note **JOGGING.**

During Paused Motion, this section will become CUT RECOVERY

Name	Description
CONTINUOUS	This drop down button will change the jog increment. Options are determined by the values in the **[DISPLAY]** section of the *<machine_name>*.ini file and begin with the label "INCREMENTS =".
FAST	This button will toggle between FAST which is the default linear velocity in the *<machine_name>*.ini file or SLOW which is 10% of the default value.
Y+	This button moves the Y axis in the positive direction.
Y-	This button moves the Y axis in the negative direction.
X+	This button moves the X axis in the positive direction.
X-	This button moves the X axis in the negative direction.
Z+	This button moves the Z axis in the positive direction.
Z-	This button moves the Z axis in the negative direction.

Note

CUT RECOVERY

During Paused Motion, this section will be shown on top of the JOGGING panel. The following section will cover each button encountered in this panel. Please see CUT RECOVERY for a detailed description of the cut recovery functionality.

Name	Description
PAUSED MOTION FEED SLIDER	In the event of a paused program, this interface allows X/Y motion to follow the programmed path in the reverse or forward direction. This slider's range is from 1%-100% of the Cut Feed Rate for the currently selected material.
FEED	This displays the paused motion feed rate.
REV	In the event of a paused program, this button will move the machine in reverse along the programmed path until it reaches the last M3 command that was either executed or that QtPlasmaC was attempting to execute before the program became paused.
FWD	In the event of a paused program, this button will move the machine forward along the programmed path indefinitely until the program's end, skipping over M3 commands.
CANCEL MOVE	This button will cancel any Cut Recovery movement that was made, and return the torch to the position the Cut Recovery movement was initiated. Note that if FWD or REV were used to move the torch, CANCEL will not return to the position of the torch when the pause occurred.
MOVE x.xxx	This displays the amount of travel that will be incurred with each press of an arrow key, in the direction the arrow key was pressed. This value displayed below MOVE represents the Kerf Width of the currently selected material.
DIRECTIONAL ARROWS	These buttons will move the torch in the direction indicated by a distance of one Kerf Width (of the currently selected material) per press.

Table 8. **G-CODE WINDOW**

Name	Description
CLEAR	This button will clear the currently opened program. If a file is open, the default material will be selected. If no file is open, the preview will be reset to a top down full table view. The torch (T0) will be selected if it was not the active tool.

Name	Description
OPEN	This button will open a FILE OPEN panel over the PREVIEW WINDOW.
RELOAD	This button will reload the currently loaded G-code File.

Table 9. **DRO**

Name	Description
HOME ALL	This button will home all of the axes in the order set by HOME_SEQUENCE in the *<machine_name>*.ini file.
WCS G54	This drop down button will change the current work offset.
CAMERA	This button will display a CAMVIEW panel on top of the PREVIEW WINDOW and will allow the user to set an origin with or without rotation. See the CAMERA section for detailed instructions.
LASER	This button will allow the user to use a laser to set an origin with or without rotation. See the LASER section for detailed instructions.
X0 Y0	This button will set the current position to X0 Y0.
HOME [AXIS]	This button will home the corresponding axis.
0 [AXIS]	This drop down button will display the following options: **Zero** - zeros the axis. **Set** - launches a dialog box to manually input the axis' coordinate. **Divide By 2** - divides the currently displayed coordinate in the DRO by two. **Set To Last** - sets the axis to the previously set coordinate.

8.3. Preview Views

The QtPlasmaC preview screen has the ability to be switched between different views and displays, as well as zooming in and out, and panning horizontally and vertically.

When QtPlasmaC is first started, the Z (top down) view will be selected as the default view for a loaded G-code file, but the full table view will be displayed.

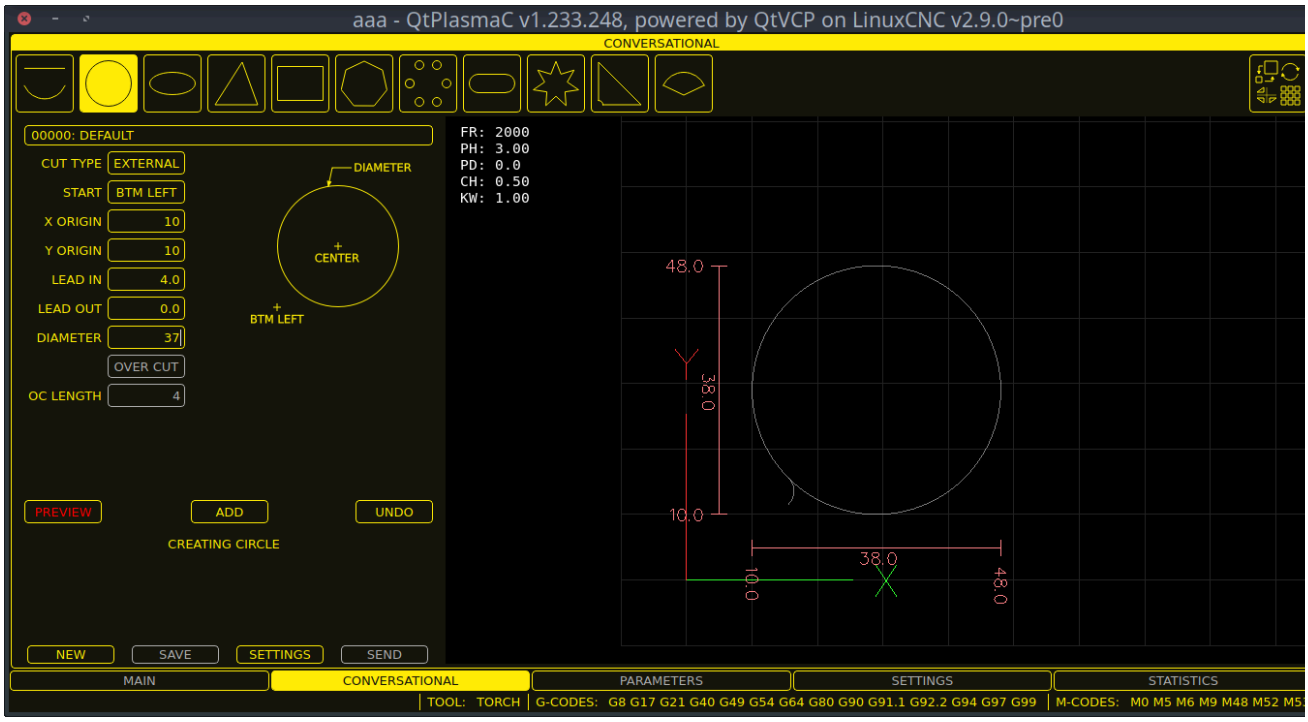
When a G-code file is loaded, the display will change to the selected view.

Whenever there is no G-code file loaded, the full table will automatically be displayed irrespective of which view is currently selected (the highlighted button representing the currently selected view will not change).

If a full table is displayed due to no G-code file being loaded and the user wishes to change the view orientation, then pressing either Z or P will change the display to the newly selected view. If the user then wishes to display the full table while maintaining the currently selected view as the default view for a loaded G-code file, then pressing CLEAR will achieve this and allow the selected view orientation to prevail the next time a G-code file is loaded.

8.4. CONVERSATIONAL Tab

Screenshot example of the QtPlasmaC CONVERSATIONAL Tab in **16:9** aspect ratio:



The CONVERSATIONAL Tab enables the user to quickly program various simple shapes for quick cutting without the need for CAM software.

See Conversational Shape Library for detailed information on the Conversational feature.

It is possible to hide this tab so the conversational feature cannot be used by an operator. This may be achieved either by wiring the pin to a physical key-switch or similar or it may also be set in a HAL file using the following command:

```
setp qtplasmac.conv_disable 1
```

8.5. PARAMETERS Tab

Screenshot example of the QtPlasmaC PARAMETERS Tab in **16:9** aspect ratio:



Some functions/features are only used for particular modes and are not displayed if they are not required by the chosen QtPlasmaC mode.

This tab is used to display configuration parameters that are modified infrequently.

It is possible to hide this tab so machine settings cannot be modified by unauthorized personnel. This may be achieved either by wiring the pin to a physical key-switch or similar or it may also be set in a HAL file using the following command:

```
setp qtplasmac.param_disable 1
```

Table 10. CONFIGURATION - ARC

Name	Modes	Description
Start Fail Timer	0, 1, 2	This sets the amount of time (in seconds) QtPlasmaC will wait between commanding a "Torch On" and receiving an Arc OK signal before timing out and displaying an error message.
Max Starts	0, 1, 2	This sets the number of times QtPlasmaC will attempt to start the arc.
Retry Delay	0, 1, 2	This sets the time (in seconds) between an arc failure and another arc start attempt.
Voltage Scale	0, 1	This sets the arc voltage input scale and is used to display the correct arc voltage. For initial setup, see Calibration Values.
Voltage Offset	0, 1	This sets the arc voltage offset and is used to display zero volts when there is zero arc voltage input. For initial setup, see Calibration Values.
Height Per Volt	0, 1, 2	This sets the distance the torch would need to move to change the arc voltage by one volt. Used for manual height manipulation only.
OK High Volts	0	This sets the voltage threshold below which Arc OK signal is valid.
OK Low Volts	0	This sets the voltage threshold above which the Arc OK signal is valid.

Note When setting the OK Low Volts and OK High Volts in Mode 0, the cut voltage of a stable arc must be greater than the OK Low Volts value but lower than the OK High Volts value for QtPlasmaC to receive a valid Arc OK signal. To further clarify, to have a valid Arc OK, the arc voltage must fall between the two limits.

Table 11. CONFIGURATION - PROBING

Name	Description
Float Travel	This sets the amount of travel the float switch moves before completing the float switch circuit. This distance can be measured by using the Probe Test button, and the method described in Initial Setup.
Probe Speed	This sets the speed at which the torch will probe to find the material after it moves to the Probe Height.
Probe Height	This sets the height above the Z axis minimum limit that Probe Speed begins. Refer to the Heights Diagram diagram for a visual representation.
Ohmic Offset	This sets the distance above the material the torch will should go after a successful ohmic probe. It is mainly used to compensate for high probing speeds.
Ohmic Retries	This sets the number of times QtPlasmaC will retry a failed ohmic probe before falling back to the float switch for material detection.
Skip IHS	This sets the distance threshold used to determine if an Initial Height Sense (probe) can be skipped for the current cut, see IHS Skip.

Note If the amount of time between the torch contacting the material and when the torch moves up and comes to rest at the Pierce Height seems excessive, see the probing section for a possible solution.

Table 12. CONFIGURATION - SAFETY

Name	Description
Safe Height	This sets the height above the material that the torch will retract to before executing rapid moves. If set to Zero then Z axis maximum height will be used for the safe height. Refer to the Heights Diagram diagram for a visual representation.

Table 13. CONFIGURATION - SCRIBING

Name	Description
Arm Delay	This sets the delay (in seconds) from the time the scribe command is received to the activation of the scribe. This allows the scribe to reach surface of the material before activating the scribe.
On Delay	This sets the delay (in seconds) to allow the scribe mechanism to start before beginning motion.

Table 14. CONFIGURATION - SPOTTING

Name	Description
Threshold	This sets the arc voltage at which the delay timer will begin. oâ☐☐V starts the delay when the torch on signal is activated.
Time On	This sets the length of time (in milliseconds) the torch is on after threshold voltage is reached.

Table 15. CONFIGURATION - MOTION

Name	Description
Max. Speed	Displays the maximum velocity the Z axis is capable of (this is controlled by the *<machine_name>*.ini file).
Setup Speed	The Z axis velocity for setup moves (movements to Probe Height, Pierce Height, Cut Height, etc.).

Note Setup Speed has no effect on THC speed which is capable of the velocity displayed in the Max. Speed field.

CONFIGURATION - THC

Two methods of THC activation are available and are selected with the **Auto Activation** checkbutton. Both methods begin their calculations when the current velocity of the torch matches the cut feed rate specified for the selected material:

1. Delay Activation (the default) is selected when **Auto Activation** is unchecked. This method uses a time delay set with the **Delay** parameter.
2. Auto Activation is selected when **Auto Activation** is checked. This method determines that the arc voltage is stable by using the **Sample Counts** and **Sample Threshold** parameters.

Name	Modes	Description
Delay	0, 1, 2	This sets the delay (in seconds) measured from the time the Arc OK signal is received until Torch Height Controller (THC) activates. This is only

Name	Modes	Description
		available when Auto THC is not enabled.
Sample Counts	0, 1	This sets the number of consecutive arc voltage readings within THC Sample Threshold required to activate the Torch Height Controller (THC). This is only available when Auto THC is enabled.
Sample Threshold	0, 1	This sets the maximum voltage deviation allowed for THC Sample Counts. This is only available when Auto THC is enabled.
Threshold	0, 1	This sets the voltage variation allowed from the target voltage before for THC makes movements to correct the torch height.
Speed (PID-P)	0, 1, 2	This sets the Proportional gain for the THC PID loop. This roughly equates to how quickly the THC attempts to correct changes in height.
VAD Threshold	0, 1, 2	(Velocity Anti Dive) This sets the percentage of the current cut feed rate the machine can slow to before locking the THC to prevent torch dive.
Void Slope	0, 1	(Void Anti Dive) This sets the size of the change in cut voltage per seconds necessary to lock the THC to prevent torch dive (higher values need greater voltage change to lock THC).
PID-I	0, 1	This sets the Integral gain for the THC PID loop. Integral gain is associated with the sum of errors in the system over time and is not always needed.
PID-D	0, 1	This sets the Derivative gain for the THC PID loop. Derivative gain works to dampen the system and reduce over correction oscillations and is not always needed.

Note PID loop tuning is a complicated process and is outside the scope of this User Guide. There are many sources of information available to assist with understanding and tuning PID loops. If the THC is not making corrections fast enough, it is recommended to increase the P gain in small increments until the system operates favorably. Large P gain adjustments can result in over correction and oscillations.

SAVE & RELOAD Buttons

The **SAVE** button will save the currently displayed parameters to the *<machine_name>*.prefs file.

The **RELOAD** button will reload all the parameters from the *<machine_name>*.prefs file.

Table 16. MATERIAL - The parameters which are active for the current cut.

Name	Description
Material	The top drop down menu is used to manually select the current material cut parameters. If there are no materials in the material file then only the default material will be displayed.
Kerf Width	This sets the kerf width for the currently selected material. Refer to the Heights Diagram diagram for a visual representation.
Pierce Height	This sets the pierce height for the currently selected material. Refer to the Heights Diagram diagram for a visual representation.
Pierce Delay	This sets the pierce delay (in seconds) for the currently selected material.
Cut Height	This sets the cut height for the currently selected material. Refer to the Heights Diagram diagram for a visual representation.
Cut Feed Rate	This sets the cut feed rate for the currently selected material.
Cut Amps	This sets the cut amperage for the currently selected material. This is a visual indicator to the operator only, unless PowerMax communications are being used.

Name	Description
Cut Volts	This sets the cut voltage for the currently selected material.
Puddle Height	Expressed as a percentage of Pierce Height, this sets the Puddle Jump height for the currently selected material. Typically used for thicker materials, Puddle Jump allows the torch to have an intermediate step between Pierce Height and Cut Height. If set, the torch will proceed from Pierce Height to P-Jump Height for a period of time (P-Jump Delay) before proceeding to Cut Height to effectively "jump" over the molten puddle. Refer to the Heights Diagram diagram for a visual representation.
Puddle Delay	This sets the amount of time (in seconds) the torch will stay at the P-Jump Height before proceeding to Cut Height.
Pause At End	This sets the amount of time (in seconds) the torch will stay on at the end of the cut before proceeding with the M5 command to turn off and raise the torch. For more information see Pause At End Of Cut.
Gas Pressure	This sets the gas pressure for the currently selected material. This setting is only valid if PowerMax communications are being used. 0 = Use the PowerMax's automatic pressure mode.
Cut Mode	This sets the cut mode for the currently selected material. This setting is only valid if PowerMax communications are being used. 1 = Normal 2 = CPA (Constant Pilot Arc) 3 = Gouge/Mark

Note See the thick materials section for more information on puddle jump.

SAVE, RELOAD, NEW, & DELETE Buttons

The **SAVE** button will save the current material set to the *<machine_name>*_material.cfg file.

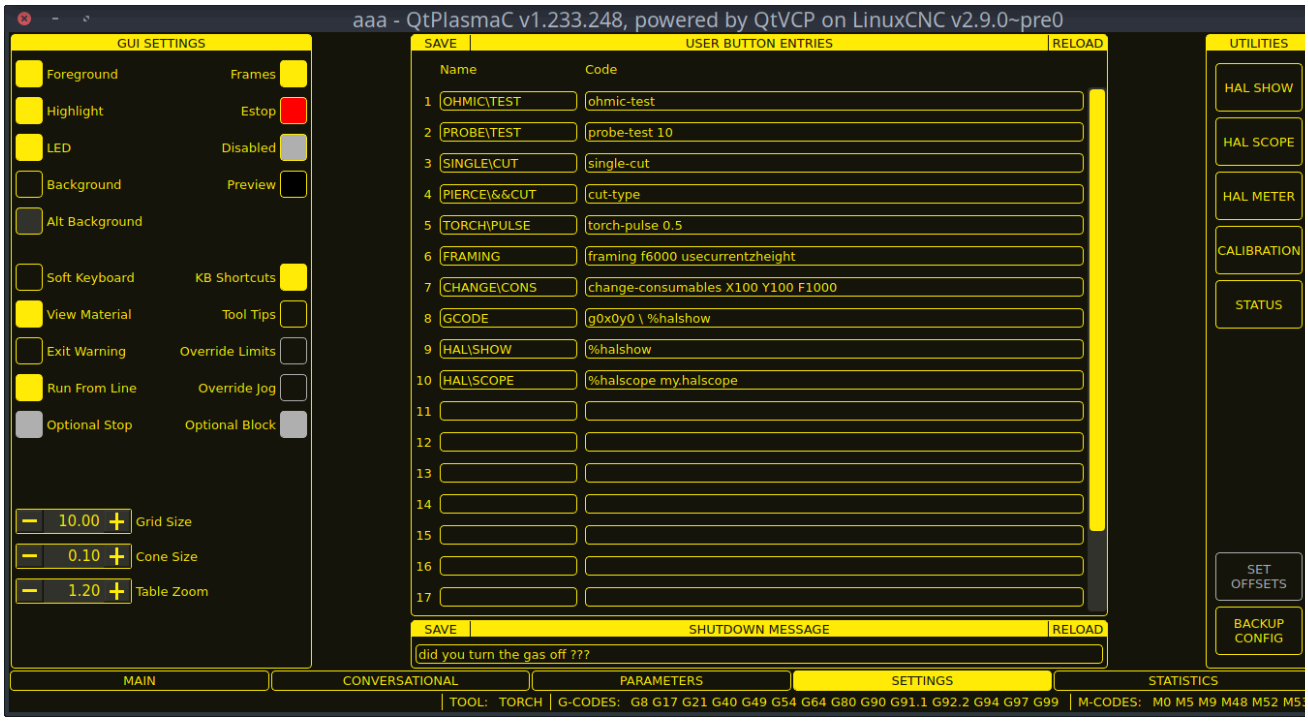
The **RELOAD** button will reload the material set from the *<machine_name>*_material.cfg file.

The **NEW** button will allow a new material to be added to the material file. The user will be prompted for a material number and a material name, all other parameters will be read from the currently selected material. Once entered, QtPlasmaC will reload the material file and display the new material. The Cut Parameters for the new material will then need to be adjusted and saved.

The **DELETE** this button is used to delete a material. After pressing it, the user will be prompted for a material number to be deleted, and prompted again to ensure the user is sure. After deletion, the material file will be reloaded and the drop down list will display the default material.

8.6. SETTINGS Tab

Screenshot example of the QtPlasmaC SETTINGS Tab in **16:9** aspect ratio:



This tab is used to display GUI configuration parameters, button text, and shutdown text that are modified infrequently as well as some utility buttons.

It is possible to hide this tab so machine settings cannot be modified by unauthorized personnel. This may be achieved either by wiring the pin to a physical key-switch or similar or it may also be set in a HAL file using the following command:

```
setp qtplasmac.settings_disable 1
```

GUI SETTINGS

This section shows parameters that effect the GUI appearance and GUI behaviors.

To return any of the color changes to their default values, see the Returning To The Default Styling section.

Table 17. GUI SETTINGS Parameters that effect the GUI appearance and GUI behaviors.

Name	Description
Foreground	This button allows the user to change the color of the GUI Foreground.
Highlight	This button allows the user to change the color of the GUI Highlight.
LED	This button allows the user to change the color of the GUI LED.
Background	This button allows the user to change the color of the GUI Background.
Alt Background	This button allows the user to change the color of the GUI Alternate Background.
Frames	This button allows the user to change the color of the GUI Frames.
Estop	This button allows the user to change the color of the GUI Estop.
Disabled	This button allows the user to change the color of the GUI's Disabled features.
Preview	This button allows the user to change the color of the GUI Preview Window Background.
Soft Keyboard	This radio button allows the user to enable or disable the soft touchscreen keyboard. If the "onboard" virtual keyboard is installed then the custom layouts will be enabled.
KB Shortcuts	This radio button allows the user to enable or disable Keyboard Shortcuts within the GUI (such as keyboard jogging).

Name	Description
	In addition to the standard jog keys, a list of the additional shortcuts is available in the keyboard shortcuts section.
View Material	This radio button allows the user to enable or disable the addition of a visual reference showing key material cut settings to the Preview Windows of the MAIN and CONVERSATIONAL tabs. Examples are: Feed Rate, Pierce Height, Pierce Delay, and Cut Height. Cut Amps will be shown if PowerMax communications are enabled.
Exit Warning	This radio button allows the user to enable or disable whether a warning will always be displayed during shutdown. It is possible to add a custom message to the warning by editing the EXIT WARNING MESSAGE option in the **[GUI_OPTIONS]** section of the *<machine_name>*.prefs file. The custom message can be made multi-line by adding a "\" between lines.
Optional Stop	This radio button allows the user to enable or disable whether or not a running program will pause at an **M1** command.
Run From Line	This radio button allows the user to enable or disable Run From Line. If enabled, the user can click on a line of G-code and have the program start from that line.
Override Limits	This radio button allows the user to temporarily Override the input from a Limit Switch in the event the limit switch becomes tripped during operation. This button can only be clicked when a limit switch is tripped.
Override Jog	This radio button will also allow jogging while jogging is inhibited due to a float switch, breakaway switch, or ohmic probe activation. This button can only be clicked when a jog is inhibited.
Optional Block	This radio button allows the user to enable or disable whether or not lines starting with "/" will be skipped if present in a running program.
Grid Size	This allows a user to change the size of the grid in the Preview Window on the MAIN Tab. Grid size of 0.0 will disable the grid.
Cone Size	This allows a user to change the size of the cone (which represents the current tool) in the Preview Window on the MAIN Tab.
Table Zoom	This allows a user to change the default zoom level for the top down full table view in the Preview Window on the MAIN Tab.

USER BUTTON ENTRIES USERBUTTON

This section shows the text that appears on the Custom User Buttons as well as the code associated with the user button. User buttons may be changed and the new settings used without restarting LinuxCNC.

The text and/or code may be edited at any time and will be loaded ready for use if the **SAVE** button is clicked.

Deleting the **Name** and **Code** text will cause that user button to be hidden if the **SAVE** button is clicked.

To return all the **Name** and **Code** text to their last saved values press the **RELOAD** button.

Name	Code
The text that is displayed on the button	The code that is run when the button is pressed.

Note There are 20 user buttons available but not all may be displayed depending on the window size.

EXIT WARNING MESSAGE

This section shows the text that appears on the shutdown dialog if the **Exit Warning** is enabled .

The text may be edited at any time and will be loaded ready for use if the **SAVE** button is clicked.

To return the **EXIT WARNING MESSAGE** text to the last saved value press the **RELOAD** button.

UTILITIES

Some standard LinuxCNC utilities are provided as an aid in the diagnosis of issues that may arise:

- Halshow
- Halscope
- Halmeter
- Calibration
- Status

In addition the following two QtPlasmaC specific utilities are provided:

The **SET OFFSETS** button is used if the table has a laser or camera for sheet alignment, a scribe, or uses offset probing. The required offsets for these peripherals need to be applied by following the procedure described in Peripheral Offsets.

The **BACKUP CONFIG** button will create a complete machine configuration backup for archiving or to aid in fault diagnosis. A compressed backup of the machine configuration will be saved in the user's Linux home directory. The file name will be *<machine_name><version>*<date>_<time>.tar.gz, where *<machine_name>* is the machine name entered in the configuration wizard, *<version>* is the current QtPlasmaC version the user is on, *<date>* is the current date (YY-MM-DD), and *<time>* is the current time (HH-MM-SS).

Prior to the backup being made, the machine log will be saved to a file in the configuration directory named machine_log_<date>_<time>.txt where *<date>* and *<time>* are formatted as described above. This file along with up to five previous machine logs will also be included in the backup.

These files are not required by QtPlasmaC and are safe to delete at any time.

8.7. STATISTICS Tab

The STATISTICS Tab provides statistics to allow for the tracking of consumable wear and job run times. These statistics are shown for the current job as well as the running total. Previous job statistics are reset once the next program is run. The total values may be reset either individually by clicking the corresponding "RESET" button, or they may all be reset together by clicking "RESET ALL".

The **RS485 PMX STATISTICS** panel will be only be displayed if the user has Hypertherm PowerMax communications and a valid RS485 connection to the PowerMax is established. This panel will show the **ARC ON TIME** for the PowerMax in hh:mm:ss format.

The **MACHINE LOG** is also displayed on the STATISTICS Tab, this log will display any errors and/or important information that occurs during the current LinuxCNC session. If the user makes a backup of the configuration from the SETTINGS Tab then the machine log is also included in the backup.



9. Using QtPlasmaC

Once QtPlasmaC is successfully installed, no Z axis motion is required to be part of the G-code cut program. In fact, if any Z axis references are present in the cut program, the standard QtPlasmaC configuration will remove them during the program loading process.

For reliable use of QtPlasmaC the user should **NOT** use any Z axis offsets other than the coordinate system offsets (G54-G59.3).

QtPlasmaC will automatically add a line of G-code to move the Z axis to the correct height at the beginning of every G-code program.

Version Information - QtPlasmaC will display versioning information in the title of the main window. The information will be displayed as followed "QtPlasmaC v*N.XXX.YYY* - powered by QtVCP on LinuxCNC vZ.Z.Z" where *N* is the version of QtPlasmaC, *XXX* is the version of the HAL component (PlasmaC.comp), *YYY* is the GUI version, and *Z.Z.Z* is the version of LinuxCNC.

9.1. Units Systems

All settings and parameters in QtPlasmaC are required to be in the same units as specified in the INI file, being either metric or imperial.

If the user is attempting to run a G-code file that is in the "other" units system then all parameters including the material file parameters are still required to be in the native machines units. Any further conversions necessary to run the G-code file will be handled automatically by the G-code filter program.

For example: If a user had a metric machine and wished to run a G-code file that was set up to cut 1/4" thick material using imperial units (inch - G20) then the user with the metric machine would need to ensure that either the material number in the G-code file was set to the corresponding metric material to be cut, or that a new material is created with the correct metric parameters for the metric material to be cut. If the metric user wanted to cut the G-code file using imperial material, then the new material parameters would need to be converted from imperial units to metric when they are entered.

9.2. Preamble and Postamble Codes

The following stanzas are the minimum recommended codes to include in the preamble and postamble of any G-code file to be run by QtPlasmaC:

Metric:

```
G21 G40 G49 G64p0.1 G80 G90 G92.1 G94 G97
```

Imperial:

```
G20 G40 G49 G64p0.004 G80 G90 G92.1 G94 G97
```

A detailed explanation of each G-code can be found in the docs here.

Note that throughout this user guide there are several additional recommendations for codes that are prudent to add to both the preamble and postamble depending on the features the user wishes to utilize.

9.3. Mandatory Codes

Aside from the preamble code, postamble code, and X/Y motion code, the only mandatory G-code syntax for QtPlasmaC to run a G-code program using a torch for cutting is `M3 $0 S1` to begin a cut and `M5 $0` to end a cut.

For backwards compatibility it is permissible to use `M3 S1` in lieu of `M3 $0 S1` to begin a cutting job and `M5` in lieu of `M5 $0` to end a cutting job. Note, that this applies to cutting jobs only, for scribe and spotting jobs the `$n` tool identifier is mandatory.

9.4. Coordinates

See recommended Z axis settings.

Each time LinuxCNC (QtPlasmaC) is started Joint homing is required. This allows LinuxCNC (QtPlasmaC) to establish the known coordinates of each axis and set the soft limits to the values specified in the *<machine_name>*.ini file in order to prevent the machine from crashing into a hard stop during normal use.

If the machine does not have home switches then the user needs to ensure that all axes are at the home coordinates specified in the *<machine_name>*.ini file before homing.

If the machine has home switches then it will move to the specified home coordinates when the Joints are homed.

Depending on the machine's configuration there will either be a **Home All** button or each axis will need to be homed individually. Use the appropriate button/buttons to home the machine.

As mentioned in the Initial Setup section, it is recommended that the first time QtPlasmaC is used that the user ensure there is nothing below the torch then jog the Z axis down until it stops at the Z axis MINIMUM_LIMIT then click the 0 next to the Z axis DRO to **Touch Off** with the Z axis selected to set the Z axis at zero offset. This should not need to be done again.

If the user intends to place the material in the exact same place on the table every time, the user could jog the X and Y axes to the machine to the corresponding X0 Y0 position as established by the CAM software and then **Touch Off** both axes with a zero offset.

If the user intends to place the material randomly on the table then the user must **Touch Off** the X and Y axes at the appropriate position before starting the program.

9.5. Cut Feed Rate

QtPlasmaC is able to read a material file to load all the required cut parameters. To enable to G-code file to use the cut feed rate setting from the cut parameters use the following code in the G-code file:

```
F#<_hal[plasmac.cut-feed-rate]>
```

It is possible to use the standard G-code **F** word to set the cut feed rate as follows:

```
F 1000
```

If the **F** word is used and the **F** word value does not match the cut feed rate of the selected material then a warning dialog will indicate this during loading of the G-code file.

9.6. Material File

Material handling uses a material file that was created for the machine configuration when the configuration wizard was ran and allows the user to conveniently store known material settings for easy recall either manually or automatically via G-code. The resulting material file is named ***<machine_name>*_material.cfg**.

QtPlasmaC does not require the use of a material file. Instead, the user could change the cut parameters manually from the MATERIAL section of the PARAMETERS Tab. It is also not required to use the automatic material changes. If the user does not wish to use this feature they can simply omit the material change codes from the G-code file.

It is also possible to not use the material file and automatically load materials from within the G-code file.

Material numbers in the materials file do not need to be consecutive nor do they need to be in numerical order.

The following variables are mandatory and an error message will appear if any are not found when the material file is loaded.

- PIERCE_HEIGHT
- PIERCE_DELAY
- CUT_HEIGHT
- CUT_SPEED

The following variables are optional. If they are not detected or have no value assigned, they will be assigned a value of 0 and no error message will appear.

- NAME
- KERF_WIDTH
- THC
- PUDDLE_JUMP_HEIGHT
- PUDDLE_JUMP_DELAY

- CUT_AMPS
- CUT_VOLTS
- PAUSE_AT_END
- GAS_PRESSURE
- CUT_MODE

Note Material numbers 1000000 and above are reserved for temporary materials.

Warning It is the responsibility of the operator to ensure that the variables are included if they are a requirement for the G-code to be run.

The material file uses the following format:

```
[MATERIAL_NUMBER_1]
NAME                = name
KERF_WIDTH          = value
THC                 = value (0 = off, 1 = on)
PIERCE_HEIGHT       = value
PIERCE_DELAY        = value
PUDDLE_JUMP_HEIGHT  = value
PUDDLE_JUMP_DELAY   = value
CUT_HEIGHT          = value
CUT_SPEED           = value
CUT_AMPS            = value (for info only unless PowerMax communications is
enabled)
CUT_VOLTS           = value (modes 0 & 1 only, if not using auto voltage
sampling)
PAUSE_AT_END        = value
GAS_PRESSURE        = value (only used for PowerMax communications)
CUT_MODE            = value (only used for PowerMax communications)
```

It is possible to add new material, delete material, or edit existing material from the PARAMETERS tab.. It is also possible to achieve this by using magic comments in a G-code file.

The material file may be edited with a text editor while LinuxCNC is running. After any changes have been saved, press **Reload** in the MATERIAL section of the PARAMETERS Tab to reload the material file.

9.7. Manual Material Handling

For manual material handling, the user would manually select the material from the materials list in the MATERIAL section of the PARAMETERS Tab before starting the G-code program. In addition to selecting materials with materials list in the MATERIAL section of the PARAMETERS Tab, the user could use the MDI to change materials with the following command:

```
M190 Pn
```

The following code is the minimum code necessary to have a successful cut using the manual material selection method:

```
F#<_hal[plasmac.cut-feed-rate]>
M3 $0 S1
.
.
M5 $0
```

Note Manual material handling will restrict the user to only one material for the entire job.

9.8. Automatic Material Handling

For automatic material handling, the user would add commands to their G-code file which will enable QtPlasmaC to change the material automatically.

The following codes may be used to allow QtPlasmaC to automatically change materials:

- **M190 P*n*** - Changes the currently displayed material to material number *n*.
- **M66 P3 L3 Q1** - Adds a small delay (1 second in this example) to wait for QtPlasmaC to confirm that it successfully changed materials.
- **F#<_hal[plasmac.cut-feed-rate]>** - Sets the cut feed rate to the feed rate shown in the MATERIAL section of the PARAMETERS Tab.

For automatic material handling, the codes MUST be applied in the order shown. If a G-code program is loaded which contains one or more material change commands then the first material will be displayed in the top header of the PREVIEW WINDOW on the MAIN Tab as the program is loading.

Minimum code necessary to have a successful cut using the automatic material selection method:

```
M190 Pn
M66 P3 L3 Q1
F#<_hal[plasmac.cut-feed-rate]>
M3 $0 S1
.
.
M5 $0
```

Note Returning to the default material prior to the end of the program is possible with the code **M190 P-1**.

9.9. Material Addition Via Magic Comments In G-code

By using "magic comments" in a G-code file it is possible to do the following:

- Add new materials to the *<machine_name>*_material.cfg file.
- Edit existing materials in the *<machine_name>*_material.cfg file.
- Use one or more temporary materials.

Temporary materials are numbered automatically by QtPlasmaC and the material change will also be done by QtPlasmaC and should not be added to the G-code file by CAM software or otherwise. The material numbers begin at 1000000 and are incremented for each temporary material. It is not possible to save a temporary material, however the user could create a new material while a temporary material is displayed and it will use the settings from the temporary material as the defaults.

Tip It is possible to use temporary materials only and have an empty *<machine_name>*_material.cfg file. This negates the need to keep the QtPlasmaC materials file updated with the CAM tool file.

- The entire comment must be in parentheses.
- The beginning of the magic comment must be: **(o=**
- The equals sign must immediately follow each parameter with no space.
- The mandatory parameters must be in the magic comment (for option 0, **na** is optional and **nu** is not used).
- There can be any number and type of magic comments in a G-code file.
- If option 0 is to be used in addition to option 1 and/or option 2 then all option 0 must appear after all option 1 or all option 2 in the G-code file.

The options are:

Option	Description
0	Creates a temporary default material. Material information added with this option will be discarded by a LinuxCNC restart or materials reload. They may also be overwritten by a new G-code file that has temporary materials.
1	Adds a new material if the number specified does not exist.
2	Overwrites an existing material if the number specified exists. Adds a new material if the number specified does not exist.

Mandatory parameters are:

Name	Description
o	Selects the option to be used.
nu	Sets the material number (not used for option 0).
na	Sets the material name (optional for option 0).
ph	Sets the pierce height.
pd	Sets the pierce delay.
ch	Sets the cut height.
fr	Sets the feed rate.

Optional parameters are:

Name	Description
kw	Sets the kerf width.
th	Sets the THC status (0=disabled, 1=enabled).
ca	Sets the cut amps.
cv	Sets the cut voltage.
pe	Sets the pause at end delay.
gp	Sets the gas pressure (PowerMax).
cm	Sets the cut mode (PowerMax).
jh	Sets the puddle jump height.
jd	Sets the puddle jump delay.

A complete example:

```
(o=0, nu=2, na=5mm Mild Steel 40A, ph=3.1, pd=0.1, ch=0.75, fr=3000, kw=0.5,
th=1, ca=45, cv=110, pe=0.1, gp=5, cm=1, jh=0, jd=0)
```

If a temporary material has been specified in a G-code file then the material change line (M190...) and wait for change line (M66...) will be added by the G-code filter and are not required in the G-code file.

9.10. Material Converter

This application is used to convert existing tool tables into QtPlasmaC material files. It can also create a material file from manual user input to entry fields.

At this stage the only conversions available are for tool tables exported from either SheetCam or Fusion 360.

SheetCam tool tables are complete and the conversion is fully automatic. The SheetCam tool file must be in the SheetCam .tools format.

Fusion 360 tool tables do not have all of the required fields so the user will be prompted for missing parameters. The Fusion 360 tool file must be in the JSON format of Fusion 360.

If the user has a format from a different CAM software they would like converted, create a **New Topic** in the PlasmaC forum section of the LinuxCNC forum to request this addition.

Material Converter may be run from a terminal using one of the two following methods.

For a package installation (Buildbot) enter the following command in a terminal window:

```
qtplasmac-materials
```

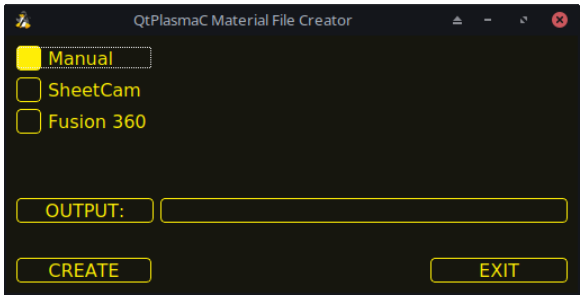
For a run in place installation enter the following two commands in a terminal window:

```
source ~/linuxcnc-dev/scripts/rip-environment
qtplasmac-materials
```

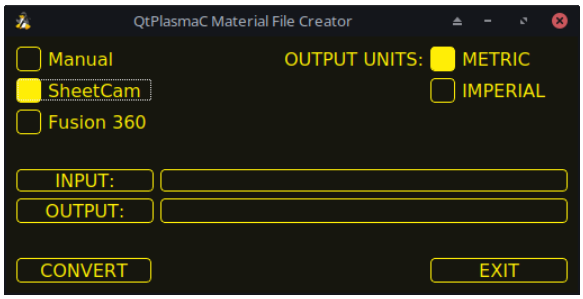
This will bring up the Material Converter Main dialog box with Manual selected as the default.

Select one of:

- **Manual** - to manually create a new material file.

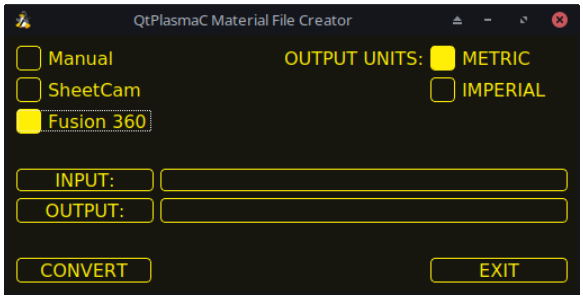


- **SheetCam** - to convert a SheetCam tool file.



For SheetCam only, select whether the user requires a metric or imperial output file.

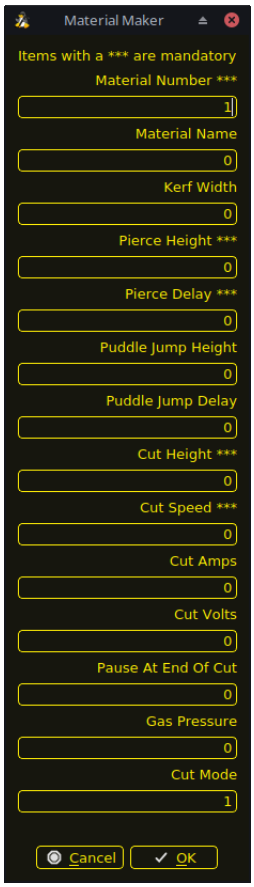
- **Fusion 360** - to convert a Fusion 360 tool file.



To convert:

1. Select the Input File to be converted, press **INPUT** to bring up a file selector or directly enter the file in the entry box.
2. Select the Output File to write to, press **OUTPUT** to bring up a file selector or directly enter the file in the entry box. This would normally be ~/linuxcnc/configs/*<machine_name>*_material.cfg. If necessary, the user could select a different file and hand edit the *<machine_name>*_material.cfg file.
3. Click **CREATE/CONVERT** and the new material file will be created.

For both a Manual creation or a Fusion 360 conversion, a dialog box will show with all available parameters displayed for input. Any entry marked with *** is mandatory and all other entries are optional depending on the user's configuration needs.



Note If the user selects ~/linuxcnc/configs/*<machine_name>*_material.cfg and the file already exists, it will be overwritten.

9.11. LASER

QtPlasmaC has the ability to use a laser to set the origin with or without rotation compensation. Rotation compensation can be used to align the work offset to a sheet of material with edge(s) that are not parallel to the machine's X/Y axes. The LASER button will be enabled after the machine is homed.

To use this feature, the user must set the laser's offset from the torch center by following the procedure described in Peripheral Offsets.

To modify the offsets manually, the user could edit either or both the following options in the **[LASER_OFFSET]** section of the *<machine_name>*.prefs file:

```
X axis = n.n
Y axis = n.n
```

where *n.n* is distance from the center line of the torch to the laser's cross hairs.

Additionally, the laser can be tied to any available output to turn the laser on and off via a HAL pin with the following name:

```
qtplasmac.laser_on
```

To set the origin with zero rotation:

1. Click the **LASER** button.
2. **LASER** button label will change to **MARK EDGE** and the HAL pin named qtplasmac.laser_on will be turned on.
3. Jog until the laser cross hairs are on top of the desired origin point.
4. Press **MARK EDGE**. The **MARK EDGE** button label will change to **SET ORIGIN**.
5. Press **SET ORIGIN**. The **SET ORIGIN** button label will change to **MARK EDGE** and the HAL pin named qtplasmac.laser_on will be turned off.
6. The torch will now move to the X0 Y0 position.
7. The offset is now successful set.

To set the origin with rotation:

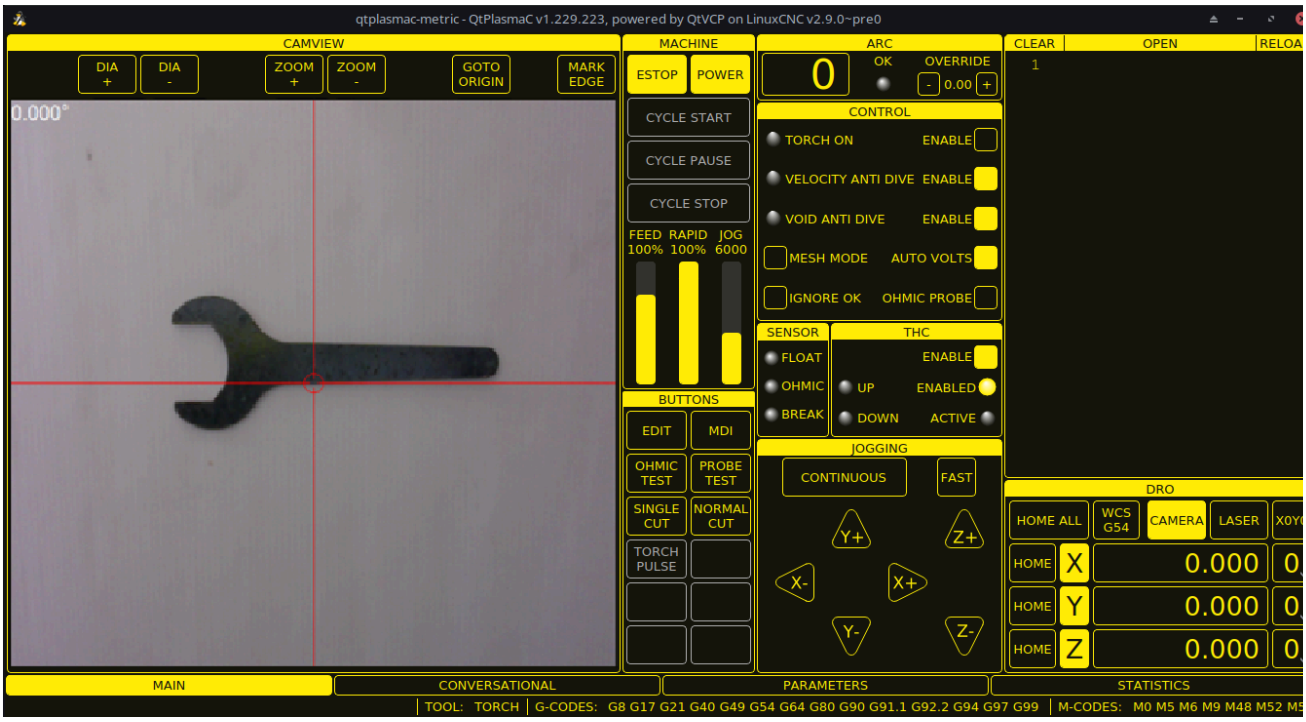
1. Click the **LASER** button.
2. **LASER** button label will change to **MARK EDGE** and the HAL pin named qtplasmac.laser_on will be turned on.
3. Jog until the laser cross hairs are at the edge of the material a suitable distance away from the desired origin point.
4. Press **MARK EDGE**. The **MARK EDGE** button label will change to **SET ORIGIN**.
5. Jog until the laser cross hairs are at the origin point of the material.
6. Press **SET ORIGIN**. The **SET ORIGIN** button label will change to **MARK EDGE** and the HAL pin named qtplasmac.laser_on will be turned off.
7. The torch will now move to the X0 Y0 position.
8. The offset is now successfully set.

To turn the laser off and cancel an alignment:

1. Press the **LASER** button and hold for longer than 750 mSec.
2. **LASER** button label will change to **LASER** and the HAL pin named qtplasmac.laser_on will be turned off.
3. Release the **LASER** button.

If an alignment laser has been set up then it is possible to use the laser during CUT RECOVERY for accurate positioning of the new start coordinates.

9.12. CAMERA



QtPlasmaC has the ability to use a USB camera to set the origin with or without rotation compensation. Rotation compensation can be used to align the work offset to a sheet of material with edge(s) that are not parallel to the machine's X/Y axes. The CAMERA button will be enabled after the machine is homed.

To use this feature, the user must set the camera's offset from the torch center by following the procedure described in Peripheral Offsets.

To modify the offsets manually, the user could edit either or both the following axes options in the **[CAMERA_OFFSET]** section of the *<machine_name>*.prefs file:

```
X axis = n.n
Y axis = n.n
Camera port = 0
```

where *n.n* is distance from the center line of the torch to the camera's cross hairs.

To set the origin with zero rotation:

1. Jog until the cross hairs are on top of the desired origin point.
2. Press **MARK EDGE**. The **MARK EDGE** button label will change to **SET ORIGIN** and the **GOTO ORIGIN** button will be disabled.
3. Press **SET ORIGIN**. The **SET ORIGIN** button label will change to **MARK EDGE** and the **GOTO ORIGIN** button will be enabled.
4. The torch will now move to the X0 Y0 position.
5. The offset is now successful set.

To set the origin with rotation:

1. Jog until the cross hairs are at the edge of the material a suitable distance away from the desired origin point.
2. Press **MARK EDGE**. The **MARK EDGE** button label will change to **SET ORIGIN** and the **GOTO ORIGIN** button will be disabled.
3. Jog until the cross hairs are at the origin point of the material.
4. Press **SET ORIGIN**. The **SET ORIGIN** button label will change to **MARK EDGE** and the **GOTO ORIGIN** button will be enabled.
5. The torch will now move to the X0 Y0 position.
6. The offset is now successfully set.

In the CAMVIEW panel, the mouse can affect the cross hairs and the zoom level as follows:

- Mouse Wheel Scroll - Change cross hair diameter.

- Mouse Wheel Button Double Click - Restores cross hair diameter to default.
- Mouse Left Button Clicked + Wheel Scroll - Changes camera zoom level.
- Mouse Left Button Clicked + Wheel Button Double Click - Restores default camera zoom level.

9.13. Path Tolerance

Path tolerance is set with a G64 command and a following P value. The P value corresponds to the amount that the actual cut path followed by the machine may deviate from the programmed cut path.

The default LinuxCNC path tolerance is set for maximum speed which will severely round corners when used with normal plasma cutting speeds.

It is recommended that the path tolerance is set by placing the appropriate G64 command and P value in the header of each G-code file.

The provided G-code filter program will test for the existence of a `G64 P__n__` command prior to the first motion command. If no G64 command is found it will insert a `G64 P0.1` command which sets the path tolerance to 0.1â□□mm. For a imperial config the command will be `G64 P0.004`.

For Metric:

```
G64 P0.1
```

For Imperial:

```
G64 P0.004
```

9.14. Paused Motion

QtPlasmaC has the ability to allow the repositioning of the X and Y axes along the current cut path while the G-code program is paused.

In order to use this feature, LinuxCNC's Adaptive Feed Control (M52) must be turned on (P1).

To enable **Paused Motion** The preamble of the G-code must contain the following line:

```
M52 P1
```

To turn off **Paused Motion** at any point, use the following command:

```
M52 P0
```

9.15. Pause At End Of Cut

This feature can be used to allow the arc to "catch up" to the torch position to fully finish the cut. It is usually required for thicker materials and is especially useful when cutting stainless steel.

Using this feature will cause all motion to pause at the end of the cut while the torch is still on. After the dwell time (in seconds) set by the **Pause At End** parameter in the MATERIAL section of the PARAMETERS Tab has expired, QtPlasmaC will proceed with the M5 command to turn off and raise the torch.

9.16. Multiple Tools

QtPlasmaC has the ability to allow the use of more than one type of plasma tool by utilizing LinuxCNC spindles as a plasma tool when running a G-code program.

Valid plasma tools for use are:

Name	TOOL #	Description
Plasma Torch	0	Used for normal Plasma cutting.
Scribe	1	Used for material engraving.
Plasma Torch	2	Used for spotting (creating dimples to aid in drilling).

A LinuxCNC spindle number (designated by $*n*) is required to be in the starting command and also the end command to be able to start and stop the correct plasma tool. Examples:

- `M3 $0 S1` will select and start the plasma cutting tool.
- `M3 $1 S1` will select and start the scribe.
- `M3 $2 S1` will select and start the plasma spotting tool.
- `M5 $0` will stop the plasma cutting tool.
- `M5 $1` will stop the scribe.
- `M5 $2` will stop the plasma spotting tool.

It is permissible to use **M5 $-1** in lieu of the M5 $*n* codes above to stop all tools.

In order to use a scribe, it is necessary for the user to add the X and Y axis offsets to the LinuxCNC tool table. Tool 0 is assigned to the Plasma Torch and Tool 1 is assigned to the scribe. Tools are selected with a **T*n* M6** command, and then a **G43 H0** command is required to apply the offsets for the selected tool. It is important to note that the LinuxCNC tool table and tool commands only come into play if the user is using a scribe in addition to a plasma torch. For more information, see scribe.

9.17. Velocity Reduction

There is a HAL pin available named **motion.analog-out-03** that can be changed in G-code with the **M67 (Synchronized with Motion)/M68 (Immediate)** commands. This pin will reduce the velocity to the percentage specified in the command.

It is important to thoroughly understand the difference between **Synchronized with Motion** and **Immediate**:

- `M67` (Synchronized with Motion) - The actual change of the specified output (P2 (THC) for example) will happen at the beginning of the next motion command. If there is no subsequent motion command, the output changes will not occur. It is best practice to program a motion code (G0 or G1 for example) right after a M67.
- `M68` (Immediate) - These commands happen immediately as they are received by the motion controller. Since these are not synchronized with motion, they will break blending. This means if these codes are used in the middle of active motion codes, the motion will pause to activate these commands.

Examples:

- `M67 E3 Q0` would set the velocity to 100% of **CutFeedRate**.
- `M67 E3 Q40` would set the velocity to 40% of **CutFeedRate**.
- `M67 E3 Q60` would set the velocity to 60% of **CutFeedRate**.
- `M67 E3 Q100` would set the velocity to 100% of **CutFeedRate**.

The minimum percentage allowed is 10%, values below this will be set to 10%.

The maximum percentage allowed is 100%, values above this will be set to 100%.

If the user intends to use this feature it would be prudent to add `M68 E3 Q0` to both the preamble and postamble of the G-code program so the machine starts and ends in a known state.

Important **G-CODE THC** AND **VELOCITY BASED THC** ARE NOT ABLE TO BE USED IF **CUTTER COMPENSATION** IS IN EFFECT; AN ERROR MESSAGE WILL BE DISPLAYED.

Warning If Cut Feed Rate in the MATERIAL section of the PARAMETERS Tab is set to Zero then QtPlasmaC will use **motion.requested-velocity** (as set by a standard Feedrate

call in the G-code) for the THC calculations. This is not recommended as it is not a reliable way of implementing velocity based THC.

Note All references to CutFeedRate refer to the **Cut Feed Rate** value displayed in the MATERIAL section of the PARAMETERS Tab.

9.18. THC (Torch Height Controller)

The THC can be enabled or disabled from the THC frame of the MAIN Tab.

The THC can also be enabled or disabled directly from the G-code program.

The THC does not become active until the velocity reaches 99.9% of the **CutFeedRate** and then the THC **Delay** time if any in the THC section of the PARAMETERS Tab has timed out. This is to allow the arc voltage to stabilize.

QtPlasmaC uses a control voltage which is dependent on the state of the **AUTO VOLTS** checkbox on the MAIN Tab:

1. If **Use Auto Volts** is checked then the actual cut voltage is sampled at the end of the THC **Delay** time and this is used as the target voltage to adjust the height of the torch.
2. If **Use Auto Volts** is not checked then the voltage displayed as Cut Volts in the MATERIAL section of the PARAMETERS Tab is used as the target voltage to adjust the height of the torch.

G-code THC

THC may be disabled and enabled directly from G-code, provided the THC is not disabled in the THC Section of the MAIN Tab, by setting or resetting the **motion.digital-out-02** pin with the M-Codes M62-M65:

- `M62 P2` will disable THC (Synchronized with Motion)
- `M63 P2` will enable THC (Synchronized with Motion)
- `M64 P2` will disable THC (Immediately)
- `M65 P2` will enable THC (Immediately)

It is important to thoroughly understand the difference between **Synchronized with Motion** and **Immediate**:

- `M62` and `M63` (Synchronized with Motion) - The actual change of the specified output (P2 (THC) for example) will happen at the beginning of the next motion command. If there is no subsequent motion command, the output changes will not occur. It is best practice to program a motion code (G0 or G1 for example) right after a M62 or M63.
- `M64` and `M65` (Immediate) - These commands happen immediately as they are received by the motion controller. Since these are not synchronized with motion, they will break blending. This means if these codes are used in the middle of active motion codes, the motion will pause to activate these commands.

Velocity Based THC

If the cut velocity falls below a percentage of **CutFeedRate** (as defined by the VAD Threshold % value in the THC frame of the CONFIGURATION section of the PARAMETERS Tab) the THC will be locked until the cut velocity returns to at least 99.9% of **CutFeedRate**. This will be made apparent by the **VELOCITY ANTI DIVE** indicator illuminating in the CONTROL Panel on the MAIN Tab.

Velocity based THC prevents the torch height being changed when velocity is reduced for a sharp corner or a small hole.

It is important to note that Velocity Reduction affects the Velocity Based THC in the following ways:

1. If Velocity Reduction is invoked in the middle of the cut, the THC will be locked.
2. The THC will remain locked until the velocity reduction is canceled by returning it to a value that is above the **VAD Threshold**, and the torch actually reaches 99.9% of the **CutFeedRate**.

9.19. Cutter Compensation

LinuxCNC (QtPlasmaC) has the ability to automatically adjust the cut path of the current program by the amount specified in Kerf Width of the selected material's Cut Parameters. This is helpful if the G-code is programmed to the nominal cut path and the user will be running the program on different thickness materials to help ensure consistently sized parts.

To use cutter compensation the user will need to use G41.1, G42.1 and G40 with the kerf width HAL pin:

- `G41.1 D#<_hal[plasmac.kerf-width]>` : offsets torch to the left of the programmed path
- `G42.1 D#<_hal[plasmac.kerf-width]>` : offsets torch to the right of the programmed path
- `G40` turns the cutter compensation off

Important IF **CUTTER COMPENSATION** IS IN EFFECT **G-CODE THC**, **VELOCITY BASED THC** AND **OVER CUT** ARE NOT ABLE TO BE USED; AN ERROR MESSAGE WILL BE DISPLAYED.

9.20. Initial Height Sense (IHS) Skip

Initial Height Sense may be skipped in one of two different ways:

1. If the THC is disabled, or the THC is enabled but not active, then the IHS skip will occur if the start of the cut is less than **Skip IHS** distance from the last successful probe.
2. If the THC is enabled and active, then the IHS skip will occur if the start of the cut is less than **Skip IHS** distance from the end of the last cut.

A value of zero for **Skip IHS** will disable all IHS skipping.

Any errors encountered during a cut will disable IHS skipping for the next cut if **Skip IHS** is enabled.

9.21. Probing

Probing may be done with either ohmic sensing or a float switch. It is also possible to combine the two methods, in which case the float switch will provide a fallback to ohmic probing. An alternative to ohmic probing is Offset Probing

If the machine's torch does not support ohmic probing, the user could have a separate probe next to the torch. In this case the user would extend the probe below the torch. The probe must NOT extend more than the minimum Cut Height below the torch and the Z axis offset distance needs to be entered as the **Ohmic Offset** in the PROBING frame of the CONFIGURATION section of the PARAMETERS Tab.

Probing setup is done in the PROBING frame of the CONFIGURATION section of the PARAMETERS Tab.

QtPlasmaC can probe at the full Z axis velocity so long as the machine has enough movement in the float switch to absorb any overrun. If the machine's float switch travel is suitable, the user could set the Probe Height to near the Z axis MINIMUM_LIMIT and do all probing at full speed.

Some float switches can exhibit a large switching hysteresis which shows up in the probing sequence as an excessive time to complete the final probe up.

- This time may be decreased by changing the speed of the final probe up.
- This speed defaults to 0.001â□□mm (0.000039") per servo cycle.
- It is possible to increase this speed by up to a factor of 10 by adding the following line to the custom.hal file:

```
setp plasmac.probe-final-speed n
```

where *n* is a value from 1-10. It is recommended to keep this value as low as possible.

Using this feature will change the final height slightly and will require thorough probe testing to confirm the final height.

This speed value affects ALL probing so if the user uses ohmic probing and the user changes this speed value then the user will need to probe test to set the require offset to compensate for this speed change as well as the float travel.

The reliability of this feature will only be as good as the repeatability of the float switch.

Note Probe Height refers to the height above the Z axis MINIMUM_LIMIT.

9.22. Offset Probing

Offset Probing is the use of a probe that is offset from the torch. This method is an alternative to Ohmic Probing and uses the `plasmac.ohmic-enable` output pin to operate a solenoid for extending and retracting the probe. The `plasmac.ohmic-probe` input pin is used to detect the material and the **Ohmic Offset** in the PROBING frame of the CONFIGURATION section of the PARAMETERS Tab is used to set the correct measured height.

The probe could be a mechanically deployed probe, a permanently mounted proximity sensor or even simply a stiff piece of wire extending about 0.5â□□mm (0.2") below the torch tip. If the probe is mechanically deployed then it needs to extend/retract rather quickly to avoid excessive probing times and would commonly be pneumatically operated.

To use this feature, the user must set the probe's offset from the torch center by following the procedure described in Peripheral Offsets.

To modify the offsets manually, the user could edit either or both the following options in the **[OFFSET_PROBING]** section of the *<machine_name>*.prefs file:

```
X axis = n.n
Y axis = n.n
Delay = t.t
```

where *n.n* is the offset of the probe from the torch center in machine units for the X and Y axes and *t.t* is the time in seconds to allow for any mechanical deployment of the probe if required.

Each of these parameters is optional and also may appear in any order. If a parameter is not detected then the default is 0.0. There can be no space after the X or Z, lower case is permissible.

When this variable appears in the INI file with either X or Y not equal to zero then QtPlasmaC will do **all** Ohmic Probing as Offset Probing. When a probe sequence has begun, the `plasmac.ohmic-enable` pin will be set True causing the probe to extend. When the material is detected the `plasmac.ohmic-enable` pin will be reset to false causing the probe to retract.

The probe will begin moving to the offset position simultaneously with the Z axis moving down to the Probe Height, probing will not commence unless the deployment timer has completed. It is required that the **Probe Height** in the PROBING frame of the CONFIGURATION section of the PARAMETERS Tab is above the top of the material to ensure that the probe is fully offset to the correct X/Y position before the final vertical probe down movement.

> **Important** PROBE HEIGHT NEEDS TO BE SET ABOVE THE TOP OF THE MATERIAL FOR OFFSET PROBING.

9.23. Cut Types

QtPlasmaC allows two different cut modes:

1. **NORMAL CUT** - runs the loaded G-code program to pierce then cut.
2. **PIERCE ONLY** - only pierces the material at each cut start position, useful prior to a **NORMAL CUT** on thick materials

There are two ways of enabling this feature:

1. Utilize the default custom user button to toggle between the cut types.
2. Adding the following line to the G-code program before the first cut to enable **Pierce Only** mode for the current file:

```
#<pierce-only> = 1
```

If using a custom user button is utilized then QtPlasmaC will automatically reload the file when the cut type is toggled.

9.24. Hole Cutting - Intro

It is recommended that any holes to be cut have a diameter no less than one and a half times the thickness of the material to be cut.

It is also recommended that holes with a diameter of less than 32â☐☐mm (1.26") are cut at 60% of the feed rate used for profile cuts. This should also lock out THC due to velocity constraints.

QtPlasmaC can utilize G-code commands usually set by a CAM Post Processor (PP) to aid in hole cutting or if the user does not have a PP or the user's PP does not support these methods then QtPlasmaC can automatically adapt the G-code to suit. This automatic mode is disabled by default.

There are three methods available for improving the quality of small holes:

1. **Velocity Reduction** - Reducing the velocity to approximately 60% of the **CutFeedRate**.
2. **Arc Dwell (Pause At End)** - Keeping the torch on for a short time at the end of the hole while motion is stopped to allow the arc to catch up.
3. **Over cut** - Turning the torch off at the end of the hole then continue along the path.

> **Note** If both **Arc Dwell** and **Over cut** are active at the same time then **Over cut** will take precedence.

> **Important** **OVER CUT** IS NOT ABLE TO BE USED IF CUTTER COMPENSATION IS IN EFFECT; AN ERROR MESSAGE WILL BE DISPLAYED.

9.25. Hole Cutting

G-code commands can be set up by either by a CAM Post Processor (PP) or by hand coding.

Hole Cutting Velocity Reduction

If cutting a hole requires a reduced velocity then the user would use the following command to set the velocity: `M67 E3 Qnn` where `nn` is the percentage of the velocity desired. For example, `M67 E3 Q60` would set the velocity to 60% of the current material's **CutFeedRate**.

See the Velocity Based THC section.

Sample code for hole cutting with reduced velocity.

```
G21 (metric)
G64 P0.005
M52 P1 (allow paused motion)
F#<_hal[plasmac.cut-feed-rate]> (feed rate from cut parameters)
G0 X10 Y10
M3 $0 S1 (start cut)
G1 X0
M67 E3 Q60 (reduce feed rate to 60%)
G3 I10 (the hole)
M67 E3 Q0 (restore feed rate to 100%)
M5 $0 (end cut)
G0 X0 Y0
M2 (end job)
```

Arc Dwell (Pause At End)

This method can be invoked by setting the Pause At End parameter in the MATERIAL frame of the PARAMETERS Tab.

Over cut

The torch can be turned off at the end of the hole by setting the `motion.digital-out-03` pin with the M-Codes `M62` (Synchronized with Motion)* or `M64` (Immediate). After turning the torch off it is necessary to allow the torch to be turned on again before beginning the next cut by resetting the `motion.digital-out-03` pin with the M-Codes `M63` or `M65`, this will be done automatically by the QtPlasmaC G-code parser if it reaches an M5 command without seeing a `M63 P3` or `M65 P3`.

After the torch is turned off the hole path will be followed for a default length of 4â□□mm (0.157"). This distance may be specified by adding `#<oclength> = n` to the G-code file.

- `M62 P3` will turn the torch off (Synchronized with Motion)
- `M63 P3` will allow the torch to be turned on (Synchronized with Motion)
- `M64 P3` will turn the torch off (Immediately)
- `M65 P3` will allow the torch to be turned on (Immediately)

It is important to thoroughly understand the difference between **Synchronized with motion** and **Immediate**:

- `M62 and M63` (Synchronized with Motion) - The actual change of the specified output (P2 (THC) for example) will happen at the beginning of the next motion command. If there is no subsequent motion command, the output changes will not occur. It is best practice to program a motion code (G0 or G1 for example) right after a M62 or M63.
- `M64 and M65` (Immediate) - These commands happen immediately as they are received by the motion controller. Since these are not synchronized with motion, they will break blending. This means if these codes are used in the middle of active motion codes, the motion will pause to activate these commands.

Sample code:

```
G21 (metric)
G64 P0.005
M52 P1 (allow paused motion)
F#<_hal[plasmac.cut-feed-rate]> (feed rate from cut parameters)
G0 X10 Y10
M3 $0 S1 (start cut)
G1 X0
M67 E3 Q60 (reduce feed rate to 60%)
G3 I10 (the hole)
M62 P3 (turn torch off)
G3 X0.8 Y6.081 I10 (continue motion for 4mm)
M63 P3 (allow torch to be turned on)
M67 E3 Q0 (restore feed rate to 100%)
M5 $0 (end cut)
G0 X0 Y0
M2 (end job)
```

9.26. Hole Cutting - Automatic

QtPlasmaC has the ability to automatically modify the G-code to reduce the velocity and/or apply **Over cut** which can be useful when cutting holes.

For valid hole sensing it is required that all values in the G2 or G3 G-code line are explicit, an error dialog will be displayed if any values are mathematically calculated.

QtPlasmaC Hole Sensing is disabled by default. It can be enabled/disabled by using the following G-code parameters to select the desired hole sensing mode:

- `#<holes> = 0` - Causes QtPlasmaC to disable hole sensing if it was previously enabled.
- `#<holes> = 1` - Causes QtPlasmaC to reduce the speed of holes less than 32â□□mm (1.26") to 60% of **CutFeedRate**.
- `#<holes> = 2` - Causes QtPlasmaC to Over cut the hole in addition to the velocity changes in setting 1.
- `#<holes> = 3` - Causes QtPlasmaC to reduce the speed of holes less than 32â□□mm (1.26") and arcs less than 16â□□mm (0.63") to 60% of **CutFeedRate**.
- `#<holes> = 4` - Causes QtPlasmaC to Over cut the hole in addition to the velocity change in setting 3.

The default hole size for QtPlasmaC hole sensing is 32â□□mm (1.26"). It is possible to change this value with the following command in a G-code file:

- `#<h_diameter> = nn` - To set a diameter (*nn*) in the same units system as the rest of the G-code file.

The default velocity for QtPlasmaC small holes is 60% of the current feed rate. It is possible to change this value with the following command in a G-code file:

- `#<h_velocity>` = *nn* - to set the percentage (*nn*) of the current feed rate required.

Over cut

If Hole Sensing modes 2 or 4 are active, QtPlasmaC will over cut the hole in addition to the velocity changes associated with modes 1 and 3.

The default over cut length for QtPlasmaC hole sensing is 4â□□mm (0.157"). It is possible to change this value with the following command in a G-code file:

- `#<oclength>` = *nn* to specify an over cut length (nn) in the same units system as the rest of the G-code file.

Arc Dwell (Pause At End)

This feature can be used in addition to setting the desired hole sensing mode via the appropriate G-code parameter by setting the Pause At End parameter in the MATERIAL frame of the PARAMETERS Tab.

Sample code:

```
G21 (metric)
G64 P0.005
M52 P1 (allow paused motion)
F#<_hal[plasmac.cut-feed-rate]> (feed rate from cut parameters)
#<holes> = 2 (over cut for holes)
#<oclength> = 6.5 (optional, 6.5mm over cut length)
G0 X10 Y10
M3 $0 S1 (start cut)
G1 X0
G3 I10 (the hole)
M5 $0 (end cut)
G0 X0 Y0
M2 (end job)
```

Note It is OK to have multiple and mixed hole commands in a single G-code file.

9.27. Single Cut

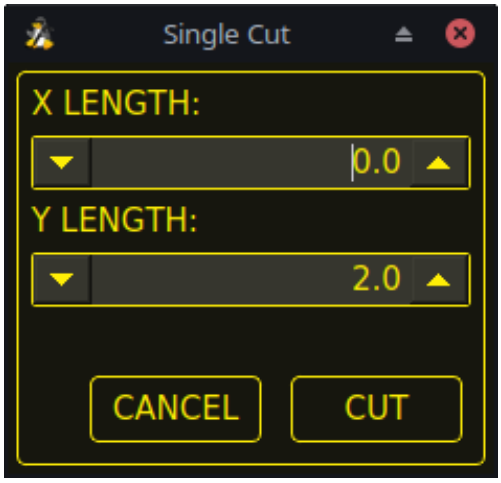
A single cut is a single unidirectional cutting move often used to cut a sheet into smaller pieces prior to running a G-code program.

The machine needs to be homed before commencing a single cut.

A single cut will commence from the machine's current X/Y position.

Automatic Single Cut

This is the preferred method. The parameters for this method are entered in the following dialog box that is displayed after pressing a user button which has been coded to run single cut:



1. Jog to the required X/Y start position.

2. Set required appropriate material, or edit the Feed Rate for the default material in the PARAMETERS Tab.
3. Press the assigned single cut user button.
4. Enter the length of the cut along the X and/or Y axes.
5. Press the **CUT** button and the cut will commence.

Pendant Single Cut

If the machine is equipped with a pendant that can start and stop the spindle plus jog the X and Y axes, the user can manually perform a single cut.

1. Jog to the required X/Y start position.
2. Set the required feed rate with the Jog Speed slider.
3. Start the cut process by starting the spindle.
4. After probing the torch will fire.
5. When the Arc OK is received the machine can be jogged along the cut line using the jog buttons.
6. When the cut is complete stop the spindle.
7. The torch will turn off and the Z axis will return to the starting position.

Manual Single Cut

Manual single cut requires that either keyboard shortcuts are enabled in the GUI SETTINGS section of the SETTINGS Tab, or a custom user button is specified as a manual cut button.

If the user is using a custom user button then substitute **F9** with **User Button** in the following description.

1. Jog to the required X/Y start position.
2. Start the procedure by pressing **F9**. The jog speed will be automatically set to the feed rate of the currently selected material. The jog label will blink to indicate that the jog speed is temporarily being overridden (jog speed manipulation will be disabled while a manual cut is active). **CYCLE START** will change to **MANUAL CUT** and blink.
3. After probing the torch will fire.
4. When the Arc OK is received the machine can be jogged along the cut line using the jog keys.
5. The Z height will remain locked at the cut height for the duration of the manual cut, regardless of the Torch Height Controller **ENABLE** status.
6. When the cut is complete press **F9** or **Esc** or the **CYCLE STOP** button.
7. The torch will turn off and the Z axis will return to the starting position.
8. The jog speed will automatically be returned to the value it was prior to initiating the manual cut process, the label will stop blinking and the jog speed manipulation will be enabled. **MANUAL CUT** will stop blinking and revert to **CYCLE START**.

> **Note** If the torch flames out during cutting, the user must still press **F9** or **Esc** or the **CYCLE STOP** button to end the cut. This clears the Z offsets and returns the torch to the starting position.

9.28. Thick Materials

Cutting thick materials can be problematic in that the large amount of molten metal caused by piercing can shorten the life of consumables and also may cause a puddle high enough that the torch may hit the puddle while moving to cut height.

The are two functions built into QtPlasmaC to help alleviate these issues.

Pierce Only

Pierce Only mode converts the loaded G-code program and then runs the program to pierce the material at the start position of each cut. Scribe and Spotting commands will be ignored and no pierce will take place in those locations.

This mode is useful for thick materials which may produce enough dross on the material surface from piercing to interfere with the torch while cutting. The entire sheet can pierced and then cleaned off prior to cutting.

It is possible to use near-end-of-life consumables for piercing and then they can be swapped out for good consumables to be used while cutting.

Pierce Only is one of two different cut types

Puddle Jump

Puddle Jump is the height that the torch will move to after piercing and prior to moving to **Cut Height** and is expressed as a percentage of **Pierce Height**. This allows the torch to clear any puddle of molten material that may be caused by piercing. The maximum allowable height is 200% of the **Pierce Height**

Setting for **Puddle Jump** are described in cut parameters

The recommended option is to use **Pierce Only** due to it being able to utilise near end of life consumables.

> **Important** | **PUDDLE JUMP** IS DISABLED DURING CUT RECOVERY

9.29. Mesh Mode (Expanded Metal Cutting)

QtPlasmaC is capable of cutting of expand (mesh) metal provided the machine has a pilot arc torch and it is capable of Constant Pilot Arc (CPA) mode.

Mesh Mode disables the THC and also ignores a lost Arc OK signal during a cut. It can be selected by checking the **Mesh Mode** check button in the CONTROL section of the MAIN Tab.

If the machine has RS485 communications enabled with a Hypertherm PowerMax plasma cutter, selecting **Mesh Mode** will automatically override the **Cut Mode** for the currently selected material and set it to cut mode 2 (CPA). When **Mesh Mode** is disabled, the **Cut Mode** will be return to the default cut mode for the currently selected material.

It is also possible to start a **Mesh Mode** cut without receiving an Arc OK signal by checking the **Ignore Arc OK** check button in the CONTROL section of the MAIN Tab.

Both **Mesh Mode** and **Ignore Arc OK** can be enabled/disabled at any time during a job.

9.30. Ignore Arc OK

Ignore Arc OK mode disables the THC, will begin a cut without requiring an Arc OK signal, and will ignore a lost Arc OK signal during a cut.

This mode can be selected by:

1. Checking the **Ignore Arc OK** check button in the CONTROL section of the MAIN Tab.
2. Setting HAL pin **motion.digital-out-01** to 1 via G-code.
    - `M62 P1` will enable **Ignore Arc OK** (Synchronized with Motion)
    - `M63 P1` will disable **Ignore Arc OK** (Synchronized with Motion)
    - `M64 P1` will enable **Ignore Arc OK** (Immediately)
    - `M65 P1` will disable **Ignore Arc OK** (Immediately)

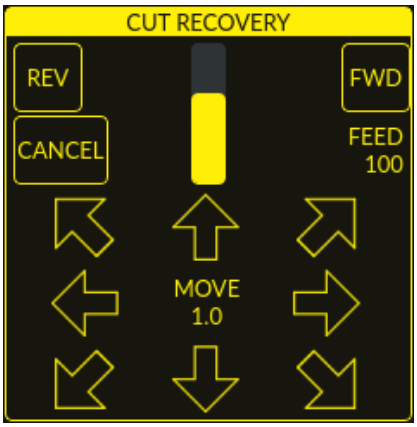
It is important to thoroughly understand the difference between **Synchronized with motion** and **Immediate**:

- `M62` and `M63` (Synchronized with Motion) - The actual change of the specified output (P2 (THC) for example) will happen at the beginning of the next motion command. If there is no subsequent motion command, the output changes will not occur. It is best practice to program a motion code (G0 or G1 for example) right after a M62 or M63.
- `M64` and `M65` (Immediate) - These commands happen immediately as they are received by the motion controller. Since these are not synchronized with motion, they will break blending. This means if these codes are used in the middle of active motion codes, the motion will pause to activate these commands.

This mode may also be used in conjunction with **Mesh Mode** if the user doesn't require an Arc OK signal to begin the cut.

Both **Mesh Mode** and **Ignore Arc OK** can be enabled/disabled at any time during a job.

9.31. Cut Recovery



This feature will produce a CUT RECOVERY panel that will allow the torch to be moved away from the cut path during a paused motion event in order to position the torch over a scrap portion of the material being cut so that the cut restarts with a minimized arc-divot. The CUT RECOVERY panel will display automatically over top of the JOGGING panel when motion is paused.

It is preferable to make torch position adjustments from the point at which paused motion occurred, however if moving along the cut path is necessary prior to setting the new start point, the user may use the paused motion controls (**REV**, **FWD**, and a **JOG-SPEED** slider) at the top of the CUT RECOVERY panel. Once the user is satisfied with the positioning of the torch along the cut path, moving off the cut path is achieved by pressing the **DIRECTION** buttons. Each press of the **DIRECTION** button will move the torch a distance equivalent to the **Kerf Width** parameter of the currently selected material.

The moment the torch has been moved off the cut path, the paused motion controls (**REV**, **FWD**, and a **JOG-SPEED** slider) at the top of the CUT RECOVERY panel will become disabled.

Once the torch position is satisfactory, press **CYCLE RESUME** and the cut will resume from the new position and travel the shortest distance to the original paused motion location. The CUT RECOVERY panel will close and the JOGGING panel will display when the torch returns to the original paused motion location.

Pressing **CANCEL MOVE** will cause the torch to move back to where it was positioned before the direction keys were used to offset the torch. It will not reset any **REV** or **FWD** motion.

Pressing **CYCLE STOP** will cause the torch to move back to where it was positioned before the direction keys were used to offset the torch and the CUT RECOVERY panel overlay will return to the JOGGING panel. It will not reset any **REV** or **FWD** motion.

If an alignment laser has been set up then it is possible to use the laser during cut recovery for very accurate positioning of the new start coordinates. If either the X axis offset or Y axis offset for the laser would cause the machine to move out of bounds then an error message will be displayed.

To use a laser for cut recovery when paused during a cut:

1. Click the **LASER** button.
2. **LASER** button will change to disabled, the HAL pin named qtplasmac.laser_on will be turned on and the X and Y axis will offset so that the laser cross hairs will indicate the starting coordinates of the cut when it is resumed.
3. Continue the cut recovery as described above.

If a laser offset is in effect when **CANCEL MOVE** is pressed then this offset will also be cleared.

Note Cut recovery movements will be limited to a radius of 10â☐☐mm (0.4") from either the point the program was paused, or from the last point on the cut path if paused motion was used.

Important PUDDLE JUMP IS DISABLED DURING CUT RECOVERY

9.32. Run From Line

If the user has the Run From Line option enabled in the GUI SETTINGS section of the SETTINGS Tab then they will have the ability to start from any line in a G-code program via the following methods:

1. Clicking any line in the Preview Window

2. Clicking any line in the G-code Window

Note that the "Run From Line"-function will run from the beginning of the selected line.

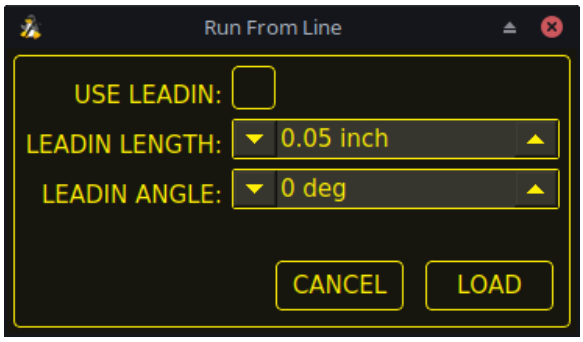
It is important to note that G-code programs can be run from any selected line using this method, however a leadin may not be possible depending on the line selected. In this case, an error message will be displayed to let the user know the leadin calculation was not possible.

Once the user has selected the starting place, the **CYCLE START** button will blink **"SELECTED *nn*"** where *nn* is the corresponding line number selected. Clicking this button will bring up the following Run From Line dialog box:

It is not possible to use Run From Line from within a subroutine. If the user selects a line within a subroutine and clicks **"SELECTED *nn*"** then an error message will be displayed that includes the O-code name of the subroutine.

It is not possible to use Run From Line if previous G-code has set cutter compensation active. If the user selects a line while cutter compensation is active and clicks **"SELECTED *nn*"** then an error message will be displayed.

It is possible to select a new line while Run From Line is active.



Name	Description
USE LEADIN	This radio button will allow the user to start the selected line with a leadin.
LEADIN LENGTH	If USE LEADIN is selected, this will set the length of the lead in the machine units.
LEADIN ANGLE	If USE LEADIN is selected, this will set the angle of approach for the leadin. The angle is measured such that positive increases in value move the leadin counter-clockwise: 0 Degrees = 3 o'clock position 90 Degrees = 12 o'clock position 180 Degrees = 9 o'clock position 270 Degrees = 6 o'clock position
CANCEL	This button will cancel the Run From Line dialog box and any selections.
LOAD	This button will load a temporary "rfl.ngc" program with any selected leadin parameters applied. If the leadin cannot be calculated for the selected line, the following error message will be displayed: "Unable to calculate a leadin for this cut Program will run from selected line with no leadin applied"

After pressing **LOAD**, the blinking "SELECTED *nn*" button will change to **RUN FROM LINE CYCLE START** button. Click this button to start the program from the beginning of the selected line.

Run From Line selections may be canceled in the following ways:

1. Click the background of the preview window - this method will cancel a selection of either a cut line in the preview window, or a G-code line in the G-code window.
2. Click the text of the first line of the G-code program in the G-code display - this method will cancel a selection of either a cut line in the preview window, or a G-code line in the G-code window.
3. Clicking **RELOAD** in the G-code window header - this method will cancel the Run From Line process if LOAD was clicked on the Run From Line dialog box and "rfl.ngc" is displayed as the loaded file name in the G-code window header. This will return the user to the originally loaded file.

9.33. Scribe

A scribe may be operated by QtPlasmaC in addition to the plasma torch.

Using a scribe requires the use of the LinuxCNC tool table. Tool 0 is assigned to the plasma torch and Tool 1 is assigned to the scribe. The scribe X and Y axes offsets from the plasma torch need to be entered into the LinuxCNC tool table. This is done by editing the tool table via the main GUI, or by editing the **tool.tbl** file in the ***<machine_name>*** configuration directory. This will be done after the scribe can move to the work piece to help determine the appropriate offset.

The plasma torch offsets for X and Y will always be zero. The tools are selected by the **T*n* M6** command followed by a **G43 H0** command which is required to apply the offsets. The tool is then started with a **M3 $*n* S1** command. For *n*, use 0 for torch cutting or 1 for scribing.

To stop the scribe, use the G-code command **M5 $1**.

If the user has not yet assigned the HAL pins for the scribe in the configuration wizard then they may do so by using the appropriate configuration wizard or by manually editing the HAL file, see modifying QtPlasmaC.

There are two HAL output pins used to operate the scribe, the first pin is used to arm the scribe which moves the scribe to the surface of the material. After the Arm Delay has elapsed, the second pin is used to start the scribe. After the On Delay has elapsed, motion will begin.

Using QtPlasmaC after enabling the scribe requires the selection of either the torch or the scribe in each G-code file as a LinuxCNC tool.

The first step is to set the offsets for the scribe by following the procedure described in Peripheral Offsets.

The final step is to set the scribe delays required:

1. **Arm Delay** - allows time for the scribe to descend to the surface of the material.
2. **On Delay** - allows time for the scribe to start before motion begins.

Save the parameters in the Config tab.

After the above directions are completed, the scribe may be tested manually by issuing a **M3 $1 S1** command in the MDI input. The user may find it helpful to use this method to scribe a small divot and then try to pulse the torch in the same location to align the offsets between the scribe and the torch.

To use the scribe from G-code:

```
...
M52 P1 (paused motion on)
F#<_hal[plasmac.cut-feed-rate]>
T1 M6 (select scribe)
G43 H0 (apply offsets for current tool)
M3 $1 S1 (start the scribe)
.
M5 $1 (stop the scribe)
.
T0 M6 (select torch)
G43 H0 (apply offsets for current tool)
G0 X0 Y0 (parking position)
M5 $-1 (end all)
```

It is a good idea to switch back to the torch at the end of the program before the final rapid parking move so the machine is always in the same state at idle.

The user can switch between the torch and the scribe any number of times during a program by using the appropriate G-codes.

Issuing **M3 S1** (without $*n*) will cause the machine to behave as if an **M3 $0 S1** had been issued and issuing **M5** (without $*n*) will cause the machine to behave as if an **M5 $0** had been issued. This will control the torch firing by default in order to provide backward compatibility for previous G-code files.

Warning If there is an existing manual tool change parameter set in the *<machine_name>*.hal file then QtPlasmaC will convert it to an automatic tool change.

9.34. Spotting

To achieve spotting to mark the material prior to drilling etc., QtPlasmaC can pulse the torch for a short duration to mark the spot to drill.

Spotting can be configured by following these steps:

1. Set the arc voltage **Threshold** in the Spotting section of the PARAMETERS Tab. Setting the voltage threshold to zero will cause the delay timer to begin immediately upon starting the torch. Setting the voltage threshold above zero will cause the delay timer to begin when the arc voltage reaches the threshold voltage.
2. Set the **Time On** in the Spotting section of the PARAMETERS Tab. When the **Time On** timer has elapsed, the torch will turn off. Times are adjustable from 0 to 9999 milliseconds.

The torch is then turned on in G-code with the **M3 $2 S1** command which selects the plasma torch as a spotting tool.

To turn the torch off, use the G-code command **M5 $2**.

For more information on multiple tools, see multiple tools.

LinuxCNC (QtPlasmaC) requires some motion between any **M3** and **M5** commands. For this reason, a minimal movement at a high speed is required to be programmed.

An example G-code is:

```
G21 (metric)
F99999 (high feed rate)
.
.
G0 X10 Y10
M3 $2 S1 (spotting on)
G91 (relative distance mode)
G1 X0.000001
G90 (absolute distance mode)
M5 $2 (spotting off)
.
.
G0 X0 Y0
G90
M2
```

Note The **high feed rate** of 99999 is to ensure that the motion is at the machine's highest feed rate.

Important SOME PLASMA CUTTERS WILL NOT BE SUITABLE FOR THIS FEATURE.
IT IS RECOMMENDED THAT THE USER CARRY OUT SOME TEST SPOTTING TO ENSURE THAT THE PLASMA CUTTER IS CAPABLE OF UTILIZING THIS FEATURE.

9.35. Virtual Keyboard Custom Layouts

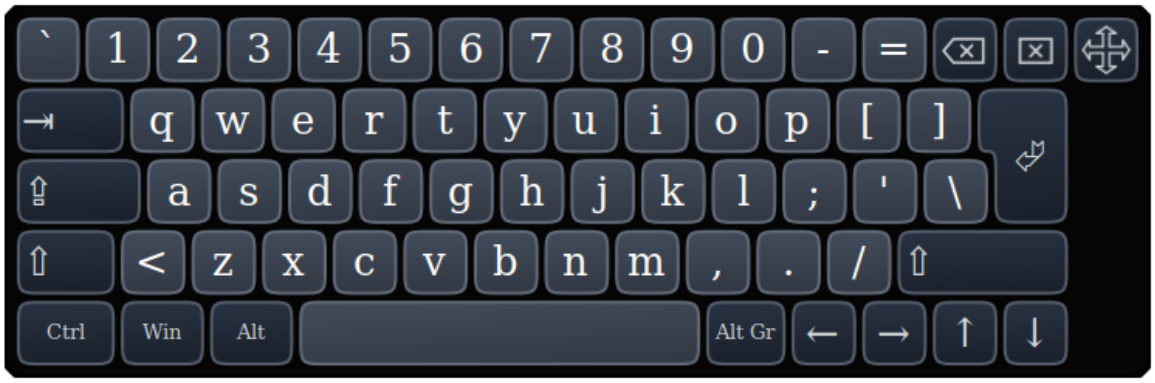
Virtual keyboard support is available for only the "onboard" onscreen keyboard. If it is not already on the system it may be installed by typing the following in a terminal:

```
sudo apt install onboard
```

The following two custom layouts are used for soft key support:



Number keypad - used for the CONVERSATIONAL Tab and the PARAMETERS Tab



Alpha-numeric keypad - used for G-code editing and file management.

If the virtual keyboard has been repositioned and on the next opening of a virtual keyboard it is not visible then clicking twice on the onboard icon in the system tray will reposition the virtual keyboard so the move handle is visible.

9.36. Keyboard Shortcuts

Below is a list of all available keyboard shortcuts in QtPlasmaC.

Note All keyboard shortcuts are disabled by default.

In order to utilize them, **KB Shortcuts** must be enabled in the **GUI SETTINGS** section of the SETTINGS Tab.

Keyboard Shortcut	Action
ESC	Aborts current automated motion (example: a running program, a probe test, etc.) as well as an active torch pulse (behaves the same as clicking CYCLE STOP).
F1	Toggles the GUI E-STOP button (if the GUI E-STOP button is enabled).
F2	Toggles the GUI power button.
F9	Toggles the "Cutting" command, used to begin or end a manual cut.
F12	Show stylesheet editor.
ALT+RETURN	Places QtPlasmaC into Manual Data Input (MDI) mode. Note that ALT + ENTER will achieve the same result. In addition, pressing RETURN (or ENTER) with no entry in the MDI will close the MDI window.
`, 1-9, 0	Changes jog speed to 0%, 10%-90%, 100% of the value present in the DEFAULT_LINEAR_VELOCITY variable in the **[DISPLAY]** section of the *<machine_name>*.ini file.

Keyboard Shortcut	Action
SHIFT+`, 1-9, 0	Changes rapid speed to 0%, 10%-90%, 100%.
CTRL+1-9, 0	Changes feed rate to 10%-90%, 100%.
CTRL+HOME	Homes all axes if they are not yet homed and have a homing sequence set in the *<machine_name>*.ini file. If they are already homed, they will no longer be homed.
CTRL+R	Cycle Start if the program is not already running. Cycle Resume if the program is paused.
END	Touches off X and Y to 0.
DEL	Allows the user to use a laser to set an origin with or without rotation. See the LASER section for detailed instructions.
SPACE BAR	Pauses motion.
CTRL+SPACE BAR	Clears notifications.
O	Opens a new program.
L	Loads the previously opened program if no program is loaded. Reloads the current program if there is a program loaded.
â☐☐	Jogs the X axis positive.
â☐☐	Jogs the X axis negative.
â☐☐	Jogs the Y axis positive.
â☐☐	Jogs the Y axis negative.
PAGE UP	Jogs the Z axis positive.
PAGE DOWN	Jogs the Z axis negative.
[	Jogs the A axis positive.
]	Jogs the A axis negative.
.	Jogs the B axis positive.
,	Jogs the B axis negative.
SHIFT (+ Jog Key)	The shift key is used with any jog key to invoke a rapid jog.
+ (+Jog Key)	The plus key can be used with any jog key to invoke a rapid jog (behaves the same as SHIFT).
- (+Jog Key)	The minus key can be used with any jog key to invoke a slow jog (10% of the displayed jog speed) If SLOW jogging is already active, the axis will jog at the displayed jog speed.

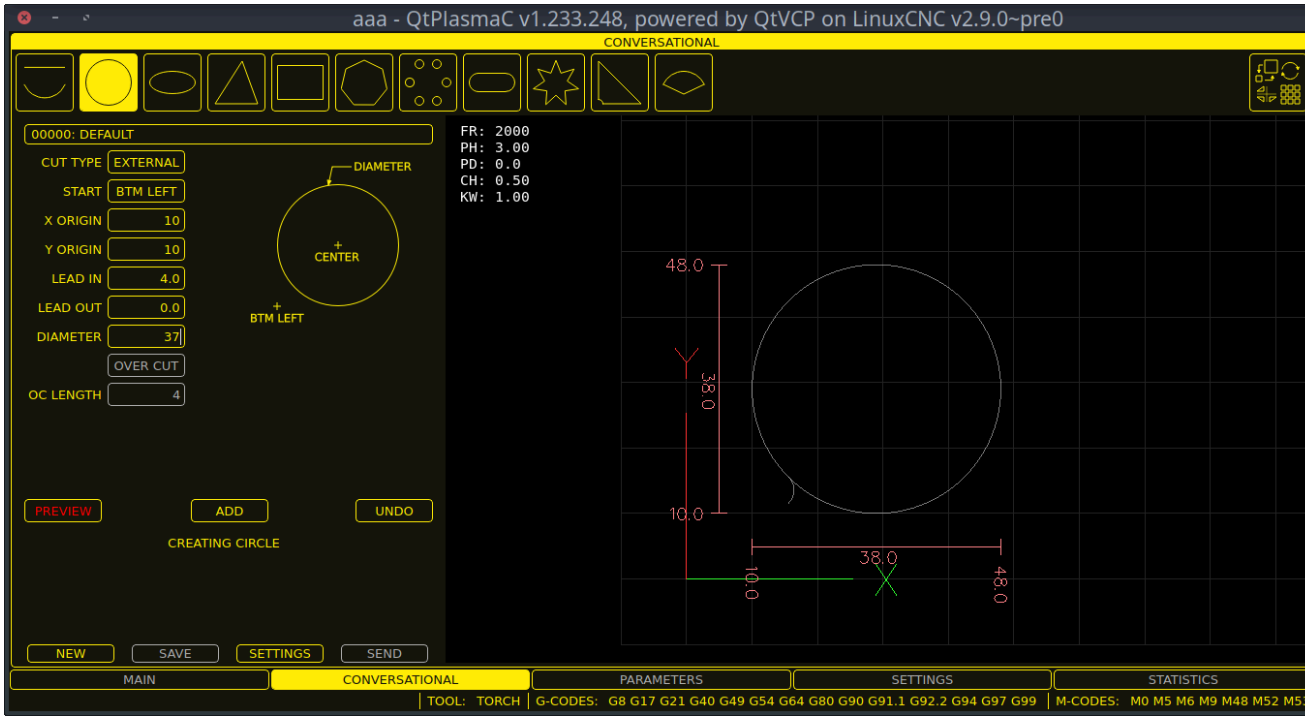
9.37. MDI

In addition to the typical G and M codes that are allowed by LinuxCNC in MDI mode, the MDI in QtPlasmaC can be used to access several other handy features. The following link outlines the features and their use: [sub:qtvcp:widgets:mdiline][MDILine Widget]

Note M3, M4, and M5 are not allowed in the QtPlasmaC MDI.

In addition, pressing RETURN (or ENTER) with no entry in the MDI will close the MDI window.

10. Conversational Shape Library



The **Conversational Shape Library** consists of several basic shapes and functions to assist the user with generating quick G-code at the machine to cut simple shapes quickly. This feature is found on the CONVERSATIONAL Tab.

> **Note** The Conversational Library is not meant to be a CAD/CAM replacement as there are limitations to what can be achieved.

Blank entries in the shape input boxes will use the current setting at the time the G-code was generated. For example, if **X start** was left blank then the current X axis position would be used.

All leadins and leadouts are arcs except for **Circles** and **Stars**:

Circles:

- If the circle is external then any leadin or leadout will be an arc.
- If the circle is internal and a **small hole** then any leadin will be perpendicular and there will be no lead out.
- If the circle is internal and not a **small hole** then any leadin and leadout will be an arc. If the leadin has a length greater than half the radius then the leadin will revert to perpendicular and there will be no leadout. If the leadout has a length greater than half the radius then there will be no leadout.

Stars:

- The leadin is at the same angle as the first cut and the leadout is at the same angle as the last cut.

> **Note** A **small hole** is a circle that is smaller than the SMALL HOLE DIAMETER specified in the CONVERSATIONAL SETTINGS page.

> **Note** The holes in a BOLT CIRCLE shape will also abide by the above rules.

The cut order will occur in the same order as the shape was built.

Pressing **Return** on the keyboard while editing parameters will automatically show the preview of the shape if there are enough parameters entered to create the shape. Clicking any of the available check boxes will do the same.

The general functions are as follows:

Name	Description
Material Drop Down	Allows the user to select the desired material for cutting. If "VIEW MATERIAL" is selected on the SETTINGS Tab, a visual reference showing key material cut settings will be displayed on the Conversational Preview Window. Examples are: Feed Rate, Pierce Height, Pierce Delay, Cut Height, and Kerf Width (for Conversational only). Cut Amps will be shown if PowerMax communications are enabled.
NEW	Removes the current G-code file and load a blank G-code file.
SAVE	Opens a dialog box allowing the current shape to be saved as a G-code file.
SETTINGS	Allows the changing of the global settings.
SEND	Loads the current shape into LinuxCNC (QtPlasmaC). If the last edit was not added then it will be discarded.
PREVIEW	Displays a preview of the current shape provided the required information is present.
CONTINUE	This button is used for lines and arcs only. Allows another segment to be added to the current segment/segments.
ADD	Stores the current shape into the current job.
UNDO	Reverts to the previously stored state.
RELOAD	Reloads the original G-code file or a blank file if none was loaded.

If there is a G-code file loaded in LinuxCNC (QtPlasmaC) when the CONVERSATIONAL Tab is selected, that code will be imported into the conversational as the first shape of the job. If this code is not required then it can be removed by pressing the **NEW** button.

If there is an added shape that is unsaved or unsent then it is not possible to switch tabs in the GUI. To re-enable switching tabs it is necessary to either **SAVE** the shape, **SEND** the shape, or press **NEW** to remove the shape.

If **NEW** is pressed to remove an added shape that is unsaved or unsent then a warning dialog will be displayed.

Note All distances are in machine units relative to the current User Coordinate System and all angles are in degrees.

10.1. Conversational Settings

Global settings for the shape library can be set by pressing the **SETTINGS** button in the CONVERSATIONAL Tab. This will display all of the available settings parameters that are used for G-code program creation. These include:

- **Preamble**
- **Postamble**
- **Origin (Center or Bottom Left)**
- **Leadin length**
- **Leadout length**
- **Small hole diameter**
- **Small hole speed**
- **Preview Window Grid Size**

Any internal circle that has a diameter less than **Small hole diameter** is classified as a small hole and will have a straight leadin with a length that is the lesser of either the radius of the hole or the specified leadin length. It will also have its feed rate set to **Small hole speed**.

Preamble and Postamble may be entered as a string of G-Codes and M-Codes separate by spaces. If the user wishes for the generated G-code to have each code on an individual line then this is made possible by separating the codes with **\n**.

This will place all codes on the same line:

```
G21 G40 G49 G64p0.1 G80 G90 G92.1 G94 G97
```

This will place each code on its own line:

```
G21\nG40\nG49\nG64p0.1\nG80\nG90\nG92.1\nG94\nG97
```

Pressing the **RELOAD** button will discard any changed but unsaved settings.
Pressing the **SAVE** button will save all the settings as displayed.
Pressing the **EXIT** button will close the setting panel and return to the previous shape.

10.2. Conversational Lines And Arcs



Lines and arcs have an additional option in that they may be strung together to create a complex shape.
There are two line types and three arc types available:

1. **Line** given a start point and an end point.
2. **Line** given a start point, length, and angle.
3. **Arc** given a start point, way point, and end point.
4. **Arc** given a start point, end point, and radius.
5. **Arc** given a start point, length, angle, and radius.

To use lines and arcs:

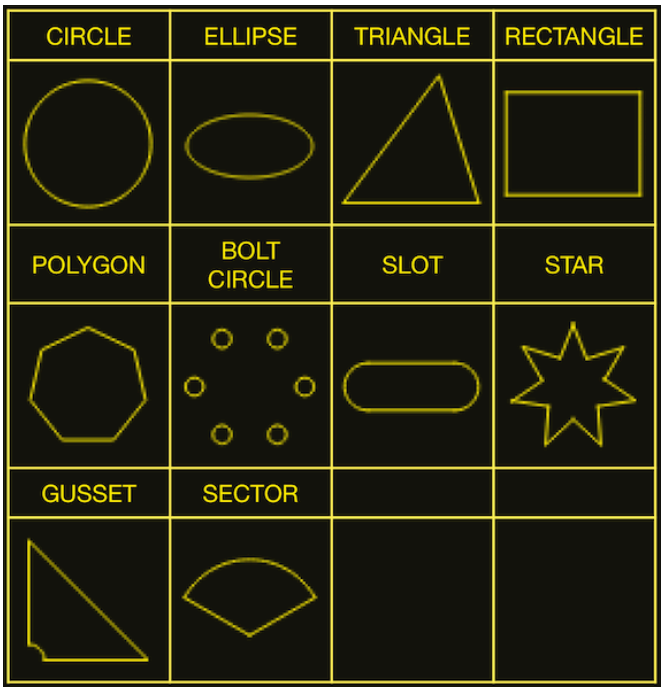
1. Select the **Lines and Arcs** icon.
2. Select the type of line or arc to create.
3. Choose the material from the MATERIAL drop down. If no material is chosen, the default material (00000) will be used.
4. Enter the desired parameters.
5. Press **PREVIEW** to see the shape.
6. If satisfied with the shape press **CONTINUE**.
7. Change the line or arc type if needed and continue this procedure until the shape is complete.
8. Press **SEND** to send the G-code file to LinuxCNC (QtPlasmaC) for cutting.

If the user wishes to create a closed shape, they will need to create any required leadin as the first segment of the shape. If a leadout is required it will need to be the last segment of the shape.

> **Note** At this stage there is no automatic option for a leadin/leadout creation if the shape is closed.

10.3. Conversational Single Shape

The following shapes are available for creation:



To create a shape:

1. Select the corresponding icon for the shape to create. The available parameters will be displayed.
2. Choose the material from the MATERIAL drop down. If no material is chosen, the default material (00000) will be used.
3. Enter the appropriate values and press **PREVIEW** to display the shape.
4. If the shape is not correct, edit the values and press **PREVIEW** and the new shape will be displayed. Repeat until satisfied with the shape.
5. Press **ADD** to add the shape to the G-code file.
6. Press **SEND** to send the G-code file to LinuxCNC (QtPlasmaC) for cutting.

For **CIRCLE**, the **OVER CUT** button will become valid when a CUT TYPE of INTERNAL is selected and the value entered in the DIAMETER field is less than the Small Hole Diameter parameter in the Conversational SETTINGS section.

For **BOLT CIRCLE** the **OVER CUT** button will become valid if the value entered in the HOLE DIA field is less than the SMALL HOLES DIAMETER parameter in the Conversational SETTINGS section.

For the following shapes, KERF OFFSET will become active once a LEAD IN is specified:

1. TRIANGLE
2. RECTANGLE
3. POLYGON
4. SLOT
5. STAR
6. GUSSET

10.4. Conversational Group Of Shapes

Multiple shapes can be added together to create a complex group.

The cut order of the group is determined by the order in which the individual shapes are added to the group.

Once a shape is added to the group it cannot be edited or removed.

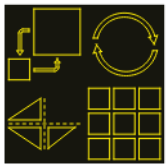
Groups cannot have shapes removed, only added to.

To create a group of shapes:

1. Create the first shape as in **Single Shape**.
2. Press **ADD** and the shape will be added to the group.
3. If the user wishes to add another version of the same shape then edit the required parameters and press **ADD** when satisfied with the shape.

4. If the user wishes to add a different shape, select that shape and create it as in **Single Shape**.
5. Repeat until all the required shapes to complete the group have been added.
6. Press **SEND** to send the G-code file to LinuxCNC (QtPlasmaC) for cutting.

10.5. Conversational Block



The Conversational Block feature allows block operations to be performed on the current shape or group of shapes displayed in the CONVERSATIONAL Tab. This can include a G-code file not created using the Conversational Shape Library that has been previously loaded from the MAIN Tab.

A previously saved Block G-code file may also be loaded from the MAIN Tab and then have any of its operations edited using the Conversational Block feature.

Block operations:

- Rotate
- Scale
- Array
- Mirror
- Flip

To create a block:

1. Create a shape, a group, or use a previously loaded G-code file.
2. Click the **Block** icon to open the Block tab.
3. Enter the appropriate values in the Block tab and press **PREVIEW** to display the resulting changes.
4. If the result is not correct, edit the values and press **PREVIEW** and the new result will be shown. Repeat until satisfied with the result.
5. Press **ADD** to complete the procedure.
6. Press **SEND** to send the G-code file to LinuxCNC (QtPlasmaC) for cutting, or **SAVE** to save the G-code file.

   COLUMNS & ROWS
   : specifies the number of duplicates of the original shape arranged in columns and rows as well as the offset distance from the original shape.

   ORIGIN
   : offset the result from the origin coordinates.

   ANGLE
   : rotate the result.

   SCALE
   : scale the result.

   ROTATION
   : rotate the shape within the result.

   MIRROR
   : mirror the shape about its X coordinates within the result.

   FLIP
   : flip the shape about its Y coordinates within the result.

If the result is an array of shapes then the cut order of the result is from the left column to the right column, starting at the bottom row and ending at the top row.

10.6. Conversational Saving A Job

The current job displayed in the Preview Panel may be saved at any time by using the bottom **SAVE** button. If the G-code has been sent to LinuxCNC (QtPlasmaC) and the user has left the CONVERSATIONAL Tab, the user may still save the G-code file from the GUI. Alternatively, the user could click the CONVERSATIONAL Tab which will reload the job, at which time they can press the **SAVE** button.

11. Error Messages

11.1. Error Logging

All errors are logged into the machine log which is able to be viewed in the STATISTICS Tab. The log file is saved into the configuration directory when QtPlasmaC is shutdown. The five last logfiles are kept, after which the oldest logfile is deleted each time a new log file is created. These saved log files may be viewed with any text editor.

11.2. Error Message Display

By default, QtPlasmaC will display error messages via a Operator Error popup window. In addition, QtPlasmaC will alert the user that an error has been sent to the machine log by displaying the message **"ERROR SENT TO MACHINE LOG"** in the lower left portion of the status bar.

The user may opt to disable the Operator Error popup window, and view the error messages by going to the STATISTICS Tab by changing the following option to **False** in the **[SCREEN_OPTIONS]** of the *<machine_name>*.prefs file in the *<machine_name>* directory:

```
desktop_notify
```

Note *<machine_name>*.prefs must be edited with QtPlasmaC closed or any changes will be overwritten on exit.

Additionally, it is possible for **ERROR SENT TO MACHINE LOG** to flash to get the user's attention by adding or editing the following option in the **[GUI_OPTIONS]** section of the *<machine_name>*.prefs file:

```
Flash error = True
```

11.3. Critical Errors

There are a number of error messages printed by QtPlasmaC to inform the user of faults as they occur. The messages can be split into two groups, **Critical** and **Warning**.

Critical Errors will cause the running program to pause, and the operator will need to clear the cause of the error before proceeding.

If the error was received during cutting then forward or reverse motion is allowed while the machine is paused to enable the user to reposition the machine prior to resuming the cut.

When the error is cleared the program may be resumed.

These errors indicate the corresponding sensor was activated during cutting:

- **breakaway switch activated, program is paused**
- **float switch activated, program is paused**
- **ohmic probe activated, program is paused**

These errors indicate the corresponding sensor was activated before probing commenced:

- **ohmic probe detected before probing program is paused**
- **float switch detected before probing program is paused**
- **breakaway switch detected before probing program is paused**

The Arc OK signal was lost during cutting motion, before the **M5** command was reached:

- **valid arc lost program is paused**

The Z axis reached the bottom limit before the work piece was detected:

- **bottom limit reached while probing down program is paused**

The work piece is too high for any safe rapid removes:

- **material too high for safe traverse, program is paused**

One of these values in MATERIAL section of the PARAMETERS Tab is invalid (For example: if they are set to zero):

- **invalid pierce height or invalid cut height or invalid cut volts, program is paused**

No arc has been detected after attempting to start the number of times indicated by **Max Starts** in the ARC frame of the CONFIGURATION section of the PARAMETERS Tab:

- **no arc detected after <*n*>d start attempts program is paused**
- **no arc detected after <*n*>d start attempts manual cut is stopped**

THC has caused the bottom limit to be reached while cutting:

- **bottom limit reached while THC moving down program is paused**

THC has caused the top limit to be reached while cutting:

- **top limit reached while THC moving up program is paused**

These errors indicate move to pierce height would exceed the Z Axis MAX_LIMIT for the corresponding probe method:

- **pierce height would exceed Z axis maximum limit condition found while moving to probe height during float switch probing**
- **pierce height would exceed Z axis maximum limit condition found while moving to probe height during ohmic probing**

These errors indicate the move to pierce height would exceed the Z axis maximum safe height for the corresponding probe method:

- **pierce height would exceed Z axis maximum safe height condition found while float switch probing**
- **pierce height would exceed Z axis maximum safe height condition found while ohmic probing**

11.4. Warning Messages

Warning messages will not pause a running program and are informational only.

These messages indicate the corresponding sensor was activated before a probe test commenced:

- **ohmic probe detected before probing probe test aborted**
- **float switch detected before probing probe test aborted**
- **breakaway switch detected before probing probe test aborted**

This indicates that the corresponding sensor was activated during a consumable change:

- **breakaway, float, or ohmic activated during consumable change, motion is paused WARNING: MOTION WILL RESUME IMMEDIATELY UPON RESOLVING THIS CONDITION!**

> **Warning** CONSUMABLE CHANGE MOTION WILL RESUME IMMEDIATELY UPON RESOLVING THE CORRESPONDING SENSOR ACTIVATION.

This indicates that the corresponding sensor was activated during probe testing:

- **breakaway switch detected during probe test**

This indicates that probe contact was lost before probing up to find the zero point:

- **probe trip error while probing**

This indicates that the bottom limit was reached during a probe test:

- **bottom limit reached while probe testing**

This indicates that the move to pierce height would exceed the Z Axis MAX_LIMIT during the corresponding probe method:

- **pierce height would exceed Z axis maximum limit condition found while moving to probe height during float switch probe testing**
- **pierce height would exceed Z axis maximum limit condition found while moving to probe height during ohmic probe testing**

This indicates that the safe height has been reduced due to THC raising the Z axis during cutting:

- **safe traverse height has been reduced**

This indicates that the value for the Arc Voltage was invalid (NAN or INF) when QtPlasmaC launched.

- **invalid arc-voltage-in**

12. Updating QtPlasmaC

12.1. Standard Update

QtPlasmaC update notices are posted at https://forum.linuxcnc.org/plasmac/37233-plasmac-updates .

Users are strongly encouraged to create a Username and subscribe to the above thread to receive update notices.

For a standard ISO installation, LinuxCNC will only be updated when a new minor release has been released. QtPlasmaC will then automatically update its configuration the first time it is run after a LinuxCNC update.

LinuxCNC is normally updated by entering the following commands into a terminal window (one at a time):

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

12.2. Continuous Update

Enhancements and bug fixes will not be available on a standard installation until a new minor release of LinuxCNC has been released. If the user wishes to update whenever a new QtPlasmaC version has been pushed, they could use the LinuxCNC Buildbot repository rather than the standard LinuxCNC repository by following the instructions at http://buildbot.linuxcnc.org/ .

13. Modify An Existing QtPlasmaC Configuration

There are two ways to modify an existing QtPlasmaC configuration:

1. Running the appropriate configuration wizard and loading the .conf file saved by the wizard.
2. Manually edit the INI and/or the HAL file of the configuration.

Important Any manual modification to the *<machine_name>*.ini and *<machine_name>*.hal files will not be registered in PnCconf or StepConf.

Note If unsure of the HAL pin's full name, the user may start LinuxCNC and run **HalShow** for a full listing of all HAL pins.

14. Customizing QtPlasmaC GUI

Styling of the QtPlasmaC GUI is done with Qt stylesheets and some customization may be achieved by the use of a custom stylesheet. This allows the user to change some GUI items such as color, border, size, etc. It cannot change the layout of the GUI.

Information on Qt stylesheets is available here.

There are two methods available to apply custom styles:

1. Add A Custom Style: use this for minor style changes.
2. Create A New Style use this for a complete style change.

14.1. Add A Custom Style

Adding style changes to the default stylesheet is achieved by creating a file in the *<machine_name>* configuration directory. This file MUST be named qtplasmac_custom.qss. Any required style changes are then added to this file.

For example the user may want the arc voltage display in red, a green Torch On LED of a larger size and a larger Torch Enable button. This would be done with the following code in qtplasmac_custom.qss:

```
#arc_voltage {
    color: #ff0000 }

#led_torch_on {
    qproperty-diameter: 30;
    qproperty-color: green }

#torch_enable::indicator {
    width: 30;
    height: 30}
```

14.2. Create A New Style

Custom stylesheets are enabled by setting the following option in the **[GUI_OPTIONS]** section of the *<machine_name>*.prefs file. This option must be set to the filename of the stylesheet as shown below.

```
Custom style = the_cool_style.qss
```

The filename may be any valid filename. The standard extension name is .qss but this is not mandatory.

There are some constraints on the custom stylesheet for QtPlasmaC, e.g., the jog buttons, cut-recovery buttons, and the conversational shape buttons are image files and are not able to be custom styled.

The custom style file requires a header in the following format:

```
/*****************************
Custom Stylesheet Header

color1 = #000000
#QtPlasmaC default = #ffee06

color2 = #e0e0e0
#QtPlasmaC default = #16160e

color3 = #c0c0c0
#QtPlasmaC default = #ffee06

color4 = #e0e0e0
#QtPlasmaC default = #26261e

color5 = #808080
#QtPlasmaC default = #b0b0b0

*****************************/
```

The colors may be expressed in any valid stylesheet format.

The above colors are used for the following widgets. So any custom styling will need to take these into account. The colors shown below are the defaults used in QtPlasmaC along with the color name from the SETTINGS Tab.

Color	Parameter	Affects
color1 (#ffee06)	Foreground	foreground of jog buttons foreground of latching user buttons foreground of camera/laser buttons foreground of conversational shape buttons background of active conversational shape buttons
color2 (#16160e)	Background	background of latching user buttons background of camera/laser buttons background of G-code editor active line background of conversational shape buttons
color3 (#ffee06)	Highlight	background of active latching user buttons background of active camera/laser buttons foreground of G-code editor cursor
color4 (#36362e)	Alt Background	background of G-code display active line

14.3. Returning To The Default Styling

The user may return to the default styling at any time by following the following steps:

1. Close QtPlasmaC if open.
2. Delete qtplasmac.qss from the machine config directory.
3. Delete qtplasmac_custom.qss from the machine config directory (if it exists).
4. Open *<machine_name>*.prefs file.
5. Delete the **[COLOR_OPTIONS]** section.
6. Delete the Custom style option from the **[GUI_OPTIONS]** section.
7. Save the file.

The next time QtPlasmaC is loaded all custom styling will be removed and the default styling will return.

Below is an example of the section and options to be deleted from *<machine_name>*.prefs:

```
[COLOR_OPTIONS]
Foreground = #ffee06
Highlight = #ffee06
LED = #ffee06
Background = #16160e
Background Alt = #36362e
Frames = #ffee06
Estop = #ff0000
Disabled = #b0b0b0
Preview = #000000
```

14.4. Custom Python Code

It is possible to add custom Python code to change some existing functions or to add new ones. Custom code can be added in two different ways: a user command file or a user periodic file.

A user command file is specified in the DISPLAY section of the *<machine_name>*.ini file and contains Python code that is processed only once during startup.

```
USER_COMMAND_FILE = my_custom_code.py
```

A user periodic file must be named user_periodic.py and must be loaded in the machines config directory. This file is processed every cycle (usually 100 ms) and is used for functions that require regular updating.

14.5. Custom G-code Filter

All incoming G-code is parsed by a G-code filter to ensure it is suitable for QtPlasmaC. It is possible to extend this filter with custom python code executed from a file in the configuration directory to aid in converting different flavours of G-code to a format suitable for QtPlasmaC.

The name of this file is custom_filter.py and it will be automatically used if it exists.

There are three preset methods available for use:

Name	Function
custom_pre_process	This does basic processing of each line before any processing is done in the filter.
custom_pre_parse	This parses any G-code from a line **before** any parsing done in the filter.
custom_post_parse	This parses any G-code from a line **after** any parsing done in the filter.

These methods are applied by the following procedure:

- Define the method with an argument for the incoming data.
- Add any required code to manipulate the data.
- Return the resultant data.
- Attach the new method.

For example to remove any code beginning with *G71* and change *M2* to *M5 $0* and *M2*:

```
def custom_pre_parse(data):
    if data[:3] == 'G71':
        return(None)
    if data == 'M2':
        return(f'M5 $0\n\n{data}')
    return(data)
self.custom_pre_parse = custom_pre_parse
```

In addition to these it is also possible to override any existing method in the filter the same way. This requires defining the same number of arguments as the existing method, noting that *self* in the original does not constitute an argument.

```
def new_method_name(data):
    if data[:3] == 'G71':
        return(None)
    return(data)
self.old_method_name = new_method_name
```

Note The existing filter code may be observed in the file /bin/qtplasmac_gcode.
The file sim/qtplasmac/custom_filter.py has example skeleton code for custom filtering.

15. QtPlasmaC Advanced Topics

15.1. Custom User Buttons

The QtPlasmaC GUI offers user buttons that can be customized by adding commands in the USER BUTTON ENTRIES section of the SETTINGS Tab in the *<machine_name>*.prefs file.

The number of user buttons varies by display type and resolution as follows:

- 16:9 and 4:3 - Minimum 8, Maximum 20

- 9:16 - Minimum 15, Maximum 20

The user will need to run QtPlasmaC at the desired screen size to determine how many user buttons are available for use.

All *<machine_name>*.prefs file settings for the buttons are found in the **[BUTTONS]** section.

Button Names

The text that appears on the button is set the following way:

```
n Name = HAL Show
```

Where *n* is the button number and **HAL Show** is the text.

For text on multiple lines, split the text with a \ (backslash):

```
n Name = HAL\Show
```

If an ampersand is required to be displayed as text then two consecutive ampersands are required:

```
n Name = PIERCE&&CUT
```

Button Code

Buttons can run the following:

1. External commands
2. External python scripts
3. G-code commands
4. Toggle a HAL pin
5. Toggle the alignment laser HAL pin
6. Pulse a HAL pin
7. Probe test
8. Ohmic Test
9. Cut Type
10. Change consumables
11. Load a G-code program
12. Pulse the torch on
13. Single unidirectional cut
14. Framing a job
15. Begin/End a manual cut
16. Display/Hide an offsets viewer
17. Load the latest modified NGC file found in a directory
18. Display/Hide the online HTML user manual

External Commands

To run an external command, the command is preceded by a % character.

```
n Code = %halshow
```

External Python Scripts

To run an external Python script, the script name is preceded by a % character and it also requires a .py extension. It is valid to use the ~ character as a shortcut for the users home directory.

```
n Code = %halshow
```

G-code

To run G-code, just enter the code to be run.

```
n Code = G0 X100
```

To run an existing subroutine.

```
n Code = o<the_subroutine> call
```

<machine_name>.ini file variables can be entered by using the standard LinuxCNC G-code format. If expressions are included then they need to be surrounded by brackets.

```
n Code = G0 X#<_ini[joint_0]home> Y1
n Code = G53 G0 Z[#<_ini[axis_z]max_limit> - 1.001]
```

<machine_name>.prefs file variables and also *<machine_name>*.ini variables can be entered by enclosing each option in **{}** . You must put a space after each **}** if there are any following characters. If expressions are included then they need to be surrounded by brackets.

```
BUTTON_n_CODE = G0 X{LASER_OFFSET X axis} Y{LASER_OFFSET Y axis}
BUTTON_n_CODE = G0 X{JOINT_0 HOME} Y1
BUTTON_n_CODE = G53 G0 Z[{AXIS_Z MAX_LIMIT} - 1.001]
```

Multiple codes can be run by separating the codes with a "\" (backslash) character. The exception is the special commands which are required to be a single command per button.

```
n Code = G0 X0 Y0 \ G1 X5 \ G1 Y5
```

External commands and G-code may be mixed on the same button.

```
n Code = %halshow \ g0x.5y.5 \ %halmeter
```

Toggle HAL Pin

The following code will allow the user to use a button to invert the current state of a HAL bit pin:

```
n Code = toggle-halpin the-hal-pin-name
```

This code is required to be used as a single command and may only control one HAL bit pin per button.

The button colors will follow the state of the HAL pin.

After setting the code, upon clicking, the button will invert colors and the HAL pin will invert pin state. The button will stay "latched" until the button is clicked again, which will return the button to the original colors and the HAL pin to the original pin state.

There are three External HAL Pins that are available to toggle as an output, the pin names are `qtplasmac.ext_out_0`, `qtplasmac.ext_out_1`, and `qtplasmac.ext_out_2`. HAL connections to these HAL pins need to be specified in a postgui HAL file as the HAL pins are not available until the QtPlasmac GUI has loaded.

For toggle-halpin buttons, it is possible for the user to mark the associated HAL pin as being required to be turned "ON" before starting a cut cycle by adding "runcritical" after the HAL pin in the button code. If **TORCH**

ENABLE is checked and **CYCLE START** is pressed while the "runcritical" button is not "ON" then the user will receive a dialog warning them as such and asking to CONTINUE or CANCEL.

```
n Code = toggle-halpin the-hal-pin-name runcritical
```

Toggle Alignment Laser HAL Pin

The following code will allow the user to use a button to invert the current state of the alignment laser HAL bit pin:

```
n Code = toggle-laser
```

This code is also able to be used as a multiple command with G-code or external commands but may control only the alignment laser HAL bit pin.

The button colors will follow the state of the alignment laser HAL pin.

After setting the code, upon clicking, the button will invert colors and the alignment laser HAL pin will invert pin state. The button will stay "latched" until the button is clicked again, which will return the button to the original colors and the alignment laser HAL pin to the original pin state.

The following code would allow the user to use a button to invert the current state of the alignment laser HAL bit pin and then move the X and Y axes to the offset for the alignment laser as specified in the *<machine_name>*.prefs file:

```
n Code = G0 X{LASER_OFFSET X axis} Y{LASER_OFFSET Y axis} \ toggle-laser
```

The position of the "toggle-laser" command is not important as it is always the first command actioned regardless of position.

Pulse HAL Pin

The following code will allow the user to use a button to pulse a HAL bit pin for a duration of 0.5 seconds:

```
n Code = pulse-halpin the-hal-pin-name 0.5
```

This code is required to be used as a single command and may only control one HAL bit pin per button.

The pulse duration is specified in seconds, if the pulse duration is not specified then it will default to one second.

The button colors will follow the state of the HAL pin.

After setting the code, upon clicking the button, the button will invert colors, the HAL pin will invert pin state, and the time remaining will be displayed on the button. The button color and the pin state will stay inverted until the pulse duration timer has completed, which will return the button to the original colors, the HAL pin to the original pin state, and the original button name.

An active pulse can be canceled by clicking the button again.

There are three External HAL Pins that are available to pulse as an output, the pin names are qtplasmac.ext_out_0, qtplasmac.ext_out_1, and qtplasmac.ext_out_2. HAL connections to these HAL pins need to be specified in a postgui HAL file as the HAL pins are not available until the QtPlasmac GUI has loaded.

Probe Test

QtPlasmaC will begin a probe and when the material is detected, the Z axis will rise to the Pierce Height currently displayed in the MATERIAL section of the PARAMETERS Tab. If the user has "View Material" selected in the GUI SETTINGS section of the SETTINGS Tab, this value will be displayed in the top left corner of the PREVIEW Window next to **PH:**.

QtPlasmaC will then wait in this state for the time specified (rounded to no decimal places) before returning the Z axis to the starting position. An example of a 6 second delay is below. If there is no time specified then the probe time will default to 10 seconds.

```
n Code = probe-test 6
```

Note Enabling a user button as a Probe Test button will add an external HAL pin that may be connected from a pendant etc. HAL connections to this HAL pin needs to be specified in a postgui HAL file as the HAL pin is not available until the QtPlasmac GUI has loaded.

Ohmic Test

QtPlasmaC will enable the Ohmic Probe Enable output signal and if the Ohmic Probe input is sensed, the LED indicator in the SENSOR Panel will light. The main purpose of this is to allow a quick test for a shorted torch tip.

```
n Code = ohmic-test
```

Note Enabling a user button as an Ohmic Test button will add an external HAL pin that may be connected from a pendant etc. HAL connections to this HAL pin needs to be specified in a postgui HAL file as the HAL pin is not available until the QtPlasmac GUI has loaded.

Cut Type

This button if selected will toggle between the two cut types, Pierce and Cut (default cutting mode) or Pierce Only.

```
n Code = cut-type
```

Change Consumables

Pressing this button moves the torch to the specified coordinates when the machine is paused to allow the user easy access to change the torch consumables.

Valid entries are Xnnn Ynnn Fnnn. At least one of the X or Y coordinates are required, Feed Rate (F) is optional.

The X and Y coordinates are in absolute machine coordinates. If X or Y are missing then the current coordinate for that axis will be used.

Feed Rate (F) is optional, if it is missing or invalid then the feed rate of the current material will be used.

There are three methods to return to the previous coordinates:

1. Press the **Change Consumables** button again - the torch will return to the original coordinates and the machine will wait in this position for the user to resume the program.
2. Press **CYCLE RESUME** - the torch will return to the original coordinates and the program will resume.
3. Press **CYCLE STOP** - the torch will return to the original coordinates and the program will abort.

```
n Code = change-consumables X10 Y10 F1000
```

Note Enabling a user button as a Change Consumables button will add an external HAL pin that may be connected from a pendant etc. HAL connections to this HAL pin needs to be specified in a postgui HAL file as the HAL pin is not available until the QtPlasmac GUI has loaded.

Load

Loading a G-code program from the directory specified by the **PROGRAM_PREFIX** variable in the *<machine_name>*.ini file (usually ~/linuxcnc/nc_files) is possible by using the following format:

```
n Code = load G-code.ngc
```

If the user's G-code file is located in a sub-directory of the **PROGRAM_PREFIX** directory, it would be accessed by adding the sub-directory name to the beginning of the G-code file name. Example for a sub-directory named **plasma**:

```
n Code = load plasma/G-code.ngc
```

Note that the first "/" is not necessary as it will be added automatically.

Torch Pulse

Pulse the torch on for a predetermined time. The time must be specified in seconds using up to one decimal place. The maximum allowable time is 3 seconds, anything specified above that value will be limited to 3 seconds. An example of a 0.5 second pulse is below. It there is no time specified then it will default to 1 second. Pulse times with more than one decimal place will be rounded to one decimal place.

Pressing the button again during the countdown will cause the torch to be turned off, as will pressing *Esc* if keyboard shortcuts are enabled in the SETTINGS Tab.

If the button is released before the countdown is complete then the torch will turn off at countdown completion, holding the button on until after the countdown has completed will cause the torch to remain on until the button has been released.

```
n Code = torch-pulse 0.5
```

Note Enabling a user button as a Torch Pulse button will add an external HAL pin that may be connected from a pendant etc. HAL connections to this HAL pin needs to be specified in a postgui HAL file as the HAL pin is not available until the QtPlasmac GUI has loaded.

Single Cut

Run a single unidirectional cut. This utilises the automatic Single Cut feature.

```
n Code = single-cut
```

Framing

Framing is the ability to move the torch around the perimeter of a rectangle that encompasses the bounds of the current job.

The laser enable HAL pin (qtplasmac.laser_on) will be turned on during the framing moves and any X/Y offsets for the laser pointer in the *<machine_name>*.prefs file will also be applied to the X/Y motion. After the framing motion is completed, the torch will move to the X0 Y0 position to clear any applied laser offsets and qtplasmac.laser_on will be turned off.

Upon starting a Framing cycle, it is important to note that by default the Z axis will be moved to a height of [AXIS_Z]MAX_LIMIT - 5â□□mm (0.2") before X/Y motion begins.

The velocity for the XY movements of the Framing motion can be specified so that Framing motion always occurs at a set velocity. This can be achieved by adding the feed rate (F) as the as the last portion of the button code. If the feed rate is omitted from the button code, framing motion velocity will default to the feed rate for the currently selected material.

The following GUI buttons and Keyboard Shortcuts (if enabled in the SETTINGS Tab) are valid during Framing motion:

1. Pressing **CYCLE STOP** or the ESC keyboard shortcut - Stops Framing motion.
2. Pressing **CYCLE PAUSE** or the SPACE BAR keyboard shortcut- Pauses Framing motion.
3. Pressing **CYCLE RESUME** or the CTRL+r keyboard shortcut- Resumes paused Framing motion.
4. Changing the **FEED SLIDER** or any of the CTRL+0-9 keyboard shortcuts - Slows the feed rate.

Note IF THE FEED RATE IS CHANGED FOR THE FRAMING MOTION, IT WILL BE NECESSARY TO RETURN THE FEED SLIDER TO 100% BEFORE PRESSING CYCLE START AND CUTTING THE LOADED JOB.

```
n Code = framing
```

It is possible for the user to omit the initial default Z movement and run the framing sequence at the current Z height by adding "usecurrentzheight" after "framing".

```
n Code = framing usecurrentzheight
```

To specify a feed rate:

```
n Code = framing F100
```

or:

```
n Code = framing usecurrentzheight F100
```

Enabling a user button as a framing button will add an external HAL pin that may be connected from a pendant etc. HAL connections to this HAL pin needs to be specified in a postgui HAL file as the HAL pin is not available until the QtPlasmac GUI has loaded.

Manual Cut

Manual Cut functions identically to the **F9** button to begin or end a manual cut.

```
n Code = manual-cut
```

Offset Viewer

This allows the showing/hiding of an offset viewing screen that displays all machine offsets. All relative offsets can be edited and the G54 ~ G59.3 work system coordinates are able to be given custom names.

```
n Code = offsets-view
```

Load Latest File

This allows the loading of the last modified file in a directory. The directory name is optional and if omitted will default to the last directory a file was loaded from.

```
n Code = latest-file /home/me/linuxcnc/nc_files/qtplasmac-test
```

User Manual

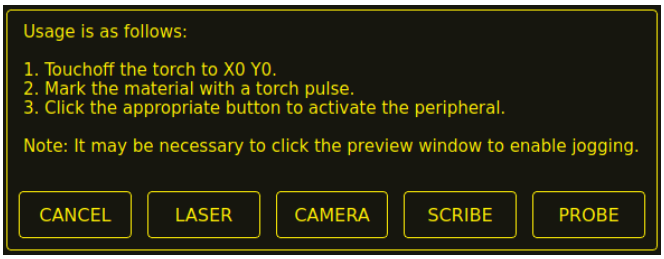
This allows the showing/hiding of the online HTML user manual specific to the version of LinuxCNC currently running. Note that internet access is required for this functionality.

```
n Code = user-manual
```

15.2. Peripheral Offsets (Laser, Camera, Scribe, Offset Probe)

Use the following sequence to set the offsets for a laser, camera, scribe, or offset probe:

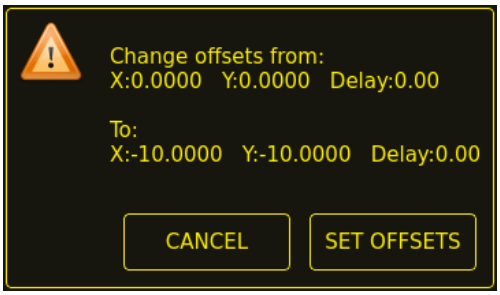
1. Place a piece of scrap material under the torch.
2. The machine must be homed and idle before proceeding.
3. Open the SETTINGS tab.
4. Click the SET OFFSETS button which opens the Set Peripheral Offsets dialog.



5. Click the XoYo button to set the torch position to zero.
6. Make a mark on the material by one of:
    a. Jog the torch down to pierce height then pulse the torch on to make a dimple in the material.
    b. Place marking dye on the torch shield then jog the torch down to mark the material.
7. Click the appropriate button to activate the peripheral.
8. The Get Peripheral Offsets dialog will now be showing.



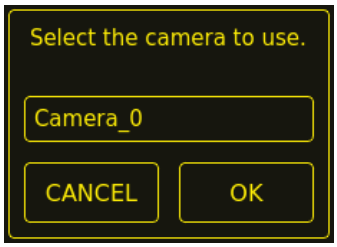
9. Raise the Z axis so the torch and peripheral are clear of the material.
10. Jog the X/Y axes so that the peripheral is centered in the mark from the torch.
11. Click the GET OFFSETS button to get the offsets and a confirmation dialog will open.



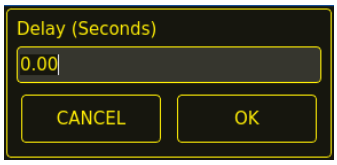
12. Click SET OFFSETS and the offsets will now be saved.

Canceling may be done at any stage by pressing the CANCEL button which will close the dialog and no changes will be saved.

If CAMERA was selected at item 7 above and more than one camera exists then a camera selection dialog will show. The appropriate camera needs to be selected before the Get Peripheral Offsets dialog will appear.



If PROBE was selected at item 7 above then a delay dialog will show prior to the confirmation dialog at item 11. This is for the delay required for the probe to deploy to its working position.



Note It may be necessary to click the preview window to enable jogging. By following the above procedure the offsets are available for use immediately and no restart of LinuxCNC is required.

15.3. Keep Z Motion

By default, QtPlasmaC will remove all Z motion from a loaded G-code file and add an initial Z movement to bring the torch near the top of travel at the beginning of the file. If the user wishes to use their table with a marker, a drag knife, diamond scribe, etc. mounted in the torch holder, QtPlasmaC has the ability to retain the Z movements when executing a program by adding the following command in a G-code file:

```
#<keep-z-motion> = 1
```

Omitting this command, or setting this value to anything but 1 will cause QtPlasmaC to revert to the default behavior of stripping all Z motion from a loaded G-code file and making an initial Z movement to bring the torch near the top of travel at the beginning of the file.

15.4. External HAL Pins

QtPlasmaC creates some HAL pins that may be used to connect a momentary external button or pendant etc.

HAL connections to these HAL pins need to be specified in a postgui HAL file as the HAL pins are not available until the QtPlasmac GUI has loaded.

The following HAL bit pins are always created. The HAL pin has the identical behaviour of the related QtPlasmaC GUI button.

User Button Function	HAL Pin	GUI Function
Toggle machine power	qtplasmac.ext_power	POWER
Run the loaded G-code program	qtplasmac.ext_run	CYCLE START
Pause/Resume the loaded G-code program	qtplasmac.ext_pause	CYCLE PAUSE
Abort the loaded G-code program	qtplasmac.ext_abort	CYCLE STOP
Touchoff X & Y axes to zero	qtplasmac.ext_touchoff	XoYo
Use a laser to set an origin with or without rotation	qtplasmac.ext_laser_touchoff	LASER
Toggle qtplasmac.laser_on pin	qtplasmac.ext_laser_toggle	N/A
Run/Pause/Resume the loaded G-code program	qtplasmac.ext_run_pause	CYCLE START, CYCLE PAUSE, CYCLE RESUME in sequence
Torch height override plus	qtplasmac.ext_height_ovr_plus	OVERRIDE
Torch height override minus	qtplasmac.ext_height_ovr_minus	OVERRIDE -
Torch height override reset	qtplasmac.ext_height_ovr_reset	OVERRIDE RESET TO 0.00
Torch height override scale	qtplasmac.ext_height_ovr_scale	N/A

User Button Function	HAL Pin	GUI Function
Toggle jogging speed between fast and slow	qtplasmac.ext_jog_slow	JOGGING FAST/SLOW
Toggle THC enable	qtplasmac.ext_thc_enable	THC ENABLE
Toggle torch enable	qtplasmac.ext_torch_enable	TORCH ENABLE
Toggle corner Lock enable	qtplasmac.ext_cornerlock_enable	VELOCITY ANTI DIVE ENABLE
Toggle voidlock enable	qtplasmac.ext_voidlock_enable	VOID ANTI DIVE ENABLE
Toggle use auto volts	qtplasmac.ext_auto_volts_enable	AUTO VOLTS
Toggle ohmic probe enable	qtplasmac.ext_ohmic_probe_enable	OHMIC ENABLE
Toggle mesh mode	qtplasmac.ext_mesh_mode	MESH MODE
Toggle arc ignore OK	qtplasmac.ext_ignore_arc_ok	IGNORE OK
Forward along the programmed path	qtplasmac.ext_cutrec_fwd	CUT RECOVERY FWD
Reverse along the programmed path	qtplasmac.ext_cutrec_rev	CUT RECOVERY REV
Cancel any Cut Recovery movement	qtplasmac.ext_cutrec_cancel	CUT RECOVERY CANCEL MOVE
Move up	qtplasmac.ext_cutrec_n	CUT RECOVERY arrow up
Move down	qtplasmac.ext_cutrec_s	CUT RECOVERY arrow down
Move right	qtplasmac.ext_cutrec_e	CUT RECOVERY arrow right
Move left	qtplasmac.ext_cutrec_w	CUT RECOVERY arrow left
Move up-right	qtplasmac.ext_cutrec_ne	CUT RECOVERY arrow up-right
Move up-left	qtplasmac.ext_cutrec_nw	CUT RECOVERY arrow up-left
Move down-right	qtplasmac.ext_cutrec_se	CUT RECOVERY arrow down-right
Move down-left	qtplasmac.ext_cutrec_sw	CUT RECOVERY arrow down-left

The following HAL pins which allow the use of an MPG to control height override are always created.

Function	HAL Pin
Enable MPG height control	qtplasmac.ext_height_ovr_count_enable
MPG height change	qtplasmac.ext_height_ovr_counts

The following HAL bit pins are only created if the function is specified in a custom user button. The HAL pin has the identical behaviour of the related custom user button.

User Button Function	HAL Pin
Probe Test	qtplasmac.ext_probe

User Button Function	HAL Pin
Torch Pulse	qtplasmac.ext_pulse
Ohmic Test	qtplasmac.ext_ohmic
Change Consumables	qtplasmac.ext_consumables
Framing	qtplasmac.ext_frame_job

The following HAL bit output pins are always created and can be used by either the Toggle HAL Pin or Pulse HAL Pin custom user buttons to change the state of an output.

HAL Pin
qtplasmac.ext_out_0
qtplasmac.ext_out_1
qtplasmac.ext_out_2

15.5. Hide Program Buttons

If the user has external buttons and/or a pendant that emulates any of the program buttons, CYCLE START, CYCLE PAUSE, or CYCLE STOP then it is possible to hide any or all of these GUI program buttons by adding the following options to the **[GUI_OPTIONS]** section of the *<machine_name>*.prefs file:

```
Hide run = True
Hide pause = True
Hide abort = True
```

For the 16:9 or 4:3 GUIs, the hiding of each of these GUI buttons will expose two more custom user buttons in the GUI.

15.6. Tuning Mode 0 Arc OK

Mode 0 Arc OK relies on the arc voltage to set the Arc OK signal. This is accomplished by sampling the arc voltage every servo thread cycle. There needs to be a specified number of consecutive samples, all within a specified threshold for the Arc OK signal to be set. These voltages are also required to be within a specified range.

There are two settings in the PARAMETERS Tab for setting the range, these are:

- **OK High Volts** which is the upper value of the voltage range. The default is 250â□□V.
- **OK Low Volts** which is the lower value of the voltage range. The default is 60â□□V.

Both of these values may be changed by direct entry or by the use of the increment/decrement buttons.

There are also two HAL pins that have been provided to allow the user to tune the set point. These HAL pins are:

- plasmac.arc-ok-counts which is the number of consecutive readings within the threshold that are required to set the Arc OK signal. The default is 10.
- plasmac.arc-ok-threshold which is the maximum voltage deviation that is allowed for a valid voltage to set the Arc OK signal. The default is 10.

The following example would set the number of valid consecutive readings required to 6:

```
setp plasmac.arc-ok-counts 6
```

These settings if used should be in the custom.hal file of the configuration.

15.7. Lost Arc Delay

Some plasma power sources/machine configurations may lose the Arc OK signal either momentarily during a cut, or permanently near the end of a cut causing QtPlasmaC to pause the program and report a "valid arc lost" error.

There is a HAL pin named `plasmac.arc-lost-delay` that may be used to set a delay (in seconds) that will prevent a paused program/error if the lost Arc OK signal is regained, or the **M5** command is reached before the set delay period expires.

It is important to note that the THC will be disabled and locked at the cutting height at the time the Arc OK signal was lost.

The following code would set a delay of 0.1 seconds:

```
setp plasmac.arc-lost-delay 0.1
```

It is recommended that the user set this pin in the custom.hal file.

This setting should only be used if the user experiences the above symptoms. It should also be noted that the user could use the appropriate Ignore Arc OK G-code commands to achieve a similar result.

15.8. Zero Window

Small fluctuations in the arc voltage displayed while the machine is at idle are possible depending on many different variables (electrical noise, incorrect THCAD tuning, etc.).

After all contributing factors have been mitigated, if a small fluctuation still exists it is possible to eliminate it by widening the voltage window for which QtPlasmaC will display 0â□□V.

The pin for adjusting this value is named plasmac.zero-window and the default value is set to 0.1. To change this value, add the pin and the required value to the custom.hal file.

The following example would set the voltage window to be displayed as 0â□□V from -5â□□V to +5â□□V:

```
setp plasmac.zero-window 5
```

15.9. Tuning Void Sensing

In addition to the **Void Slope** setting in the PARAMETERS Tab there are two HAL pins to aid in the fine tuning of void anti-dive. These HAL pins are:

- **plasmac.void-on-cycles** which is the number of times the slope rate needs to be exceeded to activate void anti-dive. The default is 2.
- **plasmac.void-off-cycles** which is the number of cycles without the slope rate being exceeded to deactivate void anti-dive. The default is 10.

The following example would set the number of on cycles required to 3:

```
setp plasmac.void-on-cycles 3
```

The objective is to have as low a value of Void Slope as possible without any false triggering then adjust on and off cycles to ensure clean activation and deactivation of void anti-dive. In most cases it should not be necessary to change on and off cycles from the default value.

These settings if used should be in the custom.hal file of the configuration.

15.10. Max Offset

Max Offset is the distance (in millimeters) away from the Z MAX_LIMIT that QtPlasmaC will allow the Z axis to travel while under machine control.

The pin for adjusting this value is named plasmac.max-offset and the default value (in millimeters) is set to 5. To change this value, add the pin and the required value to the custom.hal file. It is not recommended to use

values less than 5â□□mm as offset overrun may cause unforeseen issues.

The following example would set the distance from Z MAX_LIMIT to 10â□□mm:

```
setp plasmac.max-offset 10
```

15.11. Enable Tabs During Automated Motion

By default, all tabs except the MAIN Tab are disabled during automated motion. It is possible for every tab but the CONVERSATIONAL Tab to be enabled during automated motion by setting the following HAL pin True:

```
setp qtplasmac.tabs_always_enabled 1
```

Warning

It is the responsibility of the operator to ensure that the machine is equipped with a suitable, working hardware E-stop. If using only a touchscreen to navigate the QtPlasmaC GUI, there is no way to stop automated machine motion on any tab but the MAIN tab.

15.12. Override Jog Inhibit Via Z+ Jog

It is possible to override the jog inhibit by using the GUI or keyboard to jog in the Z+ direction rather than checking the Override Jog box on the SETTINGS Tab.

This is done by changing the following option to **True** in the **[GUI_OPTIONS]** of the <*machine_name*>.prefs file in the <*machine_name*> folder:

```
Override jog inhibit via Z+
```

15.13. QtPlasmaC State Outputs

The plasmac HAL component has a HAL pin named **plasmac.state-out** which can be used to interface with user-coded components to provide the current state of the component.

Table 18. Different states QtPlasmaC could encounter

State	Name	Description
0	IDLE	idle and waiting for a start command
1	PROBE_HEIGHT	move down to probe height
2	PROBE_DOWN	probe down until material sensed
3	PROBE_UP	probe up until material not sensed, this sets the zero height
4	ZERO_HEIGHT	not used at present
5	PIERCE_HEIGHT	move up to pierce height
6	TORCH_ON	turn the torch on
7	ARC_OK	wait until arc ok detected
8	PIERCE_DELAY	wait for pierce delay time
9	PUDDLE_JUMP	xy motion begins, move to puddle jump height
10	CUT_HEIGHT	move to cut height
11	CUT_MODE_01	cutting in either mode 0 or mode 1

State	Name	Description
12	CUT_MODE_2	cutting in mode 2
13	PAUSE_AT_END	pause motion at end of cut
14	SAFE_HEIGHT	move to safe height
15	MAX_HEIGHT	move to maximum height
16	END_CUT	end the current cut
17	END_JOB	end the current job
18	TORCHPULSE	a torch pulse is active
19	PAUSED_MOTION	cut recovery motion is active while paused
20	OHMIC_TEST	an ohmic test is active
21	PROBE_TEST	a probe test is active
22	SCRIBING	a scribing job is active
23	CONSUMABLE_CHANGE_ON	move to consumable change coordinates
24	CONSUMABLE_CHANGE_OFF	return from consumable change coordinates
25	CUT_RECOVERY_ON	cut recovery is active
26	CUT_RECOVERY_OFF	cut recovery is deactivated

The DEBUG state is for testing purposes only and will not normally be encountered.

15.14. QtPlasmaC Debug Print

The plasmac HAL component has a HAL pin named **plasmac.debug-print** which if set to 1 (true) will print to terminal every state change as a debug aid.

15.15. Hypertherm PowerMax Communications

Communications can be established with a Hypertherm PowerMax plasma cutter that has a RS485 port. This feature enables the setting of **Cut Mode**, **Cutting Amperage** and **Gas Pressure** automatically from the **Cut Parameters** of the material file. In addition, the user will be able to view the PowerMax's **Arc On Time** in hh:mm:ss format on the STATISTICS Tab.

If **Gas Pressure** is set to Zero then the PowerMax will automatically calculate the required pressure from the **Cut Mode**, **Cut Current**, torch type, and torch length.

Changing the cutting mode will set the gas pressure to zero causing the machine to use its automatic gas pressure mode.

The maximum and minimum values of these parameters are read from the plasma cutter and the related spin-buttons in the Cut Parameters are then limited by these values. Gas pressure cannot be changed from zero until communications have been established.

This feature is enabled by setting the correct port name for the PM_PORT option in the **[POWERMAX]** section of the *<machine_name>*.prefs file. If the PM_PORT option is not set in the *<machine_name>*.prefs file then the widgets associated with this feature will not be visible.

Example showing enabling the Hypertherm PowerMax Communications on USB0:

```
[POWERMAX]
Port = /dev/ttyusb0
```

If the user is unsure of the name of the port, there is a Python script in the configuration directory that will show all available ports and can also be used to test communications with the plasma unit prior to enabling this

feature in the QtPlasmaC GUI.

To use the test script follow these instructions:

For a package installation (Buildbot) enter the following command in a terminal window:

```
pmx485-test
```

For a run in place installation enter the following two commands in a terminal window:

```
source ~/linuxcnc-dev/scripts/rip-environment
pmx485-test
```

The gas pressure units display (psi or bar) is determined by the data received during initial setup of the communication link and is then shown next to the Gas Pressure setting in the MATERIAL section of the PARAMETERS Tab.

The PowerMax machine will go into remote mode after communications have been established and may only be controlled remotely (via the QtPlasmaC GUI) at this point. The connection can be validated by observing the PowerMax display.

To switch the PowerMax back to local mode the user can either:

1. Disable PowerMax Comms from the MAIN Tab
2. Close LinuxCNC which will put the PowerMax into local mode during shutdown.
3. Turn the PowerMax off for 30 seconds and then power it back on.

> **Tip** If PowerMax communications is active then selecting Mesh Mode will automatically select CPA mode on the PowerMax unit.

> **Note** To use the PowerMax communications feature it is necessary to have the Python pyserial module installed.
> If pyserial is not installed an error message will be displayed.

To install pyserial, enter the following command into a terminal window:

```
sudo apt install python3-serial
```

A typical connection diagram is shown in the appendix of this document as well as confirmed working interfaces.

16. Internationalisation

It is possible to create translation files for QtPlasmaC to display in the language of the current locale.

To create and or edit a translation file requires that LinuxCNC has been installed as run in place.

The following assumes that the linuxcnc git directory is ~/linuxcnc-dev.

All language files are kept in ~/linuxcnc-dev/share/screens/qtplasmac/languages.

The qtplasmac.py file is a Python version of the GUI file used for translation purposes.

The .ts files are the translation source files for the translations. These are the files that require creating/editing for each language.

The .qm files are the compiled translation files used by pyqt.

The language is determined by an underscore plus the first two letters of the locale, for example if an Italian translation was being done then it would be `_it`. It will be referred to as `_xx` in this document, so `qtplasmac_xx.ts` in this document would actually be `qtplasmac_it.ts` for an Italian translation.

The default locale for QtPlasmaC is `_en` which means that any translation files created as `qtplasmac_en.*` will not be used for translations.

If any of the required utilities (pyuic5, pylupdate5, linguist) are not installed then the user will need to install the required development tools:

```
sudo apt install qttools5-dev-tools pyqt5-dev-tools
```

Change to the languages directory:

```
cd ~/linuxcnc-dev/share/qtvcp/screens/qtplasmac/languages
```

If any text changes have been made to the GUI then run the following to update the GUI Python file:

```
pyuic5 ../qtplasmac.ui > qtplasmac.py
```

The user can either create a new translation source file for a non-existing language translation or modify an existing translation source file due to changes being made to some text in a QtPlasmaC source file. If modifying an existing translation that has had no source file changes then this step is not required.

Create or edit a .ts file:

```
./langfile xx
```

Note this command is a script which runs the following: $ pylupdate5 **.py ../**.py ../../../../../lib/python/qtvcp/lib/qtplasmac/*.py -ts qtplasmac_xx.ts

The editing of the translation is done with the linguist application:

```
linguist
```

1. Open the TS file and translate the strings

It is not necessary to provide a translation for every text string, if no translation is specified for a string then the original string will be used in the application. The user needs to be careful with the length of strings that appear on widgets as space is limited. If possible try to make the translation no longer than the original.

When editing is complete save the file:
`File -> Save`

Then create the .qm file:
`File -> Release`

Then create links to the compiled .qm file for the other QtPlasmaC GUIs.

```
$ ln -s qtplasmac_en.qm ../../qtplasmac_4x3/languages/
$ ln -s qtplasmac_en.qm ../../qtplasmac_9x16/languages/
```

QtPlasmaC will be translated to the language of the current locale on the next start so long as a .qm file exists in that language.

Users are welcome to submit translation files for inclusion into QtPlasmac. The preferred method is to submit a pull request from the users GitHub account as described in the contributing to LinuxCNC documentation. The only files required to be committed are qtplasmac_xx.ts and qtplasmac_xx.qm.

17. Appendix

17.1. Example Configurations

There are example configuration files which use the QtPlasmaC GUI to simulate plasma cutting machines.
They can be found in the LinuxCNC chooser under: Sample Configurations -> sim -> qtplasmac
Three versions are available in both metric and imperial units:

1. qtplasmac_l - 16:9 format, minimum resolution 1366x768
2. qtplasmac_p - 9:16 format, minimum resolution 786x1366
3. qtplasmac_s - 4:3 format, minimum resolution 1024x768

Each sample configuration includes a popup control panel to simulate various inputs to the GUI such as:

1. ARC VOLTAGE
2. OHMIC SENSE
3. FLOAT SWITCH
4. BREAKAWAY SWITCH
5. ESTOP

17.2. NGC Samples

There are some sample G-code files in the ~/linuxcnc/nc_files/examples/plasmac directory.

17.3. QtPlasmaC Specific G-codes

Description	Code
Begin cut	`M3 $0 S1`
End cut	`M5 $0`
Begin scribe	`M3 $1 S1`
End scribe	`M5 $1`
Begin center spot	`M3 $2 S1`
End center spot	`M5 $2`
End all the above.	`M5 $-1`
Select a material.	`M190 Pn` *n* denotes the material number.
Wait for material change confirmation.	`M66 P3 L3 Qn` + *n* is delay time (in seconds). This value may need to be increased for very large material files.
Set feed rate from material.	`F#<_hal[plasmac.cut-feed-rate]>`
Enable Ignore Arc OK	`M62 P1` (synchronized with motion) `M64 P1` (immediate)
Disable Ignore Arc OK	`M63 P1` (synchronized with motion) `M65 P1` (immediate)
Disable THC	`M62 P2` (synchronized with motion) `M64 P2` (immediate)
Enable THC	`M63 P2` (synchronized with motion) `M65 P2` (immediate)
Disable Torch	`M62 P3` (synchronized with motion) `M64 P3` (immediate)
Enable Torch	`M63 P3` (synchronized with motion) `M65 P3` (immediate)

Description	Code
Set velocity to a percentage of feed rate.	`M67 E3 Q`*n* (synchronized with motion) `M68 E3 Q`*n* (immediate) *n* is the percentage to set 10 is the minimum, below this will be set to 100% 100 is the maximum, above this will be set to 100% **It is recommended to have M68 E3 Q0 in both the preamble and postamble.**
Cutter compensation - left of path	`G41.1 D#<_hal[plasmac.kerf-width]>`
Cutter compensation - right of path	`G42.1 D#<_hal[plasmac.kerf-width]>`
Cutter compensation off	`G40` **Note that M62 through M68 are invalid while cutter compensation is on.**
Cut holes at 60% feed rate	`#<holes> = 1` for holes less than 32â☐☐mm (1.26") diameter
Cut holes at 60% feed rate, turn torch off at hole end, continue hole path for over cut.	`#<holes> = 2` for holes less than 32â☐☐mm (1.26") diameter over cut length = 4â☐☐mm (0.157")
Cut holes and arcs at 60% feed rate.	`#<holes> = 3` for holes less than 32â☐☐mm (1.26") diameter for arcs less than 16â☐☐mm (0.63") radius
Cut holes and arcs at 60% feed rate, turn torch off at hole end, continue hole path for over cut.	`#<holes> = 4` for holes less than 32â☐☐mm (1.26") diameter for arcs less than 16â☐☐mm (0.63") radius over cut length = 4â☐☐mm (0.157")
Specify hole diameter for #<holes> = 1-4.	`#<h_diameter> =` *n* (*n* is the diameter, use the same units system as the rest of the G-code file)
Specify hole velocity for #<holes> = 1-4.	`#<h_velocity> =` *n* (*n* is the percentage, set the percentage of the current feed rate)
Specify over cut length.	`#<oclength> =` *n* (*n* is the length, use the same units system as the rest of the G-code file)
Specify pierce-only mode.	`#<pierce-only> =` *n* (*n* is the mode, 0=normal cut mode, 1=pierce only mode)
Create or edit materials. Options: 0 - Create temporary default 1 - Add if not existing 2 - Overwrite if existing else add new	mandatory parameters: (o=<*option*>, nu=<*nn*>, na=<*ll*>, ph=<*nn*>, pd=<*nn*>, ch=<*nn*>, fr=<*nn*>) optional parameters: (kw=<*nn*>, th=<*nn*>, ca=<*nn*>, cv=<*nn*>, pe=<*nn*>, gp=<*nn*>, cm=<*nn*>, jh=<*nn*>, jd=<*nn*>)
Keep Z Motion	`#<keep-z-motion> = 1`

17.4. QtPlasmaC G-code Examples

Description	Example

Select material and do a normal cut	`M190 P3`
	`M66 P3 L3 Q1`
	`F#<_hal[plasmac.cut-feed-rate]>`
	`M3 $0 S1`
	.

Description	Example
	. M5 $0
Set velocity to 100% of CutFeedRate	M67 E3 Q0 or M67 E3 Q100
Set velocity to 60% of CutFeedRate	M67 E3 Q60
Set velocity to 40% of CutFeedRate	M67 E3 Q40
Cut a hole with 60% reduced speed using velocity setting	G21 (metric) G64 P0.05 M52 P1 (allow paused motion) F#<_hal[plasmac.cut-feed-rate]> G0 X10 Y10 M3 $0 S1 (start cut) G1 X0 M67 E3 Q60 (reduce feed rate to 60%) G3 I10 (the hole) M67 E3 Q100 (restore feed rate to 100%) M5 $0 (end cut) G0 X0 Y0 M2 (end job)
Cut a hole with 60% reduced speed using the #<holes> command	G21 (metric) G64 P0.05 M52 P1 (allow paused motion) #<holes> = 1 (velocity reduction for holes) F#<_hal[plasmac.cut-feed-rate]> G0 X10 Y10 M3 $0 S1 (start cut) G1 X0 G3 I10 (the hole) M5 $0 (end cut) G0 X0 Y0 M2 (end job)
Cut a hole with over cut using torch disable	G21 (metric) G64 P0.05 M52 P1 (allow paused motion) F#<_hal[plasmac.cut-feed-rate]> G0 X10 Y10 M3 $0 S1 (start cut) G1 X0 M67 E3 Q60 (reduce feed rate to 60%) G3 I10 (the hole) M62 P3 (turn torch off) G3 X0.8 Y6.081 I10 (continue motion for 4â□□mm) M63 P3 (allow torch to be turned on) M67 E3 Q0 (restore feed rate to 100%) M5 $0 (end cut) G0 X0 Y0 M2 (end job)
Cut a hole with over cut using the #<holes> command	G21 (metric) G64 P0.05 M52 P1 (allow paused motion) #<holes> = 2 (over cut for holes) F#<_hal[plasmac.cut-feed-rate]> G0 X10 Y10 M3 $0 S1 (start cut) G1 X0 G3 I10 (the hole)

Description	Example
	M5 $0 (end cut) G0 X0 Y0 M2 (end job)
Cut a hole with 6.5â□□mm over cut using the #<holes> command	G21 (metric) G64 P0.05 M52 P1 (allow paused motion) #<holes> = 2 (over cut for holes) <oclength> = 6.5 (6.5â□□mm over cut length) F<_hal[plasmac.cut-feed-rate]> G0 X10 Y10 M3 $0 S1 (start cut) G1 X0 G3 I10 (the hole) M5 $0 (end cut) G0 X0 Y0 M2 (end job)
Select scribe and select torch at end of scribing	. . M52 P1 (paused motion on) F#<_hal[plasmac.cut-feed-rate]> T1 M6 (select scribe) G43 H0 (apply offsets) M3 $1 S1 (start plasmac with scribe) . . T0 M6 (select torch) G43 H0 (apply offsets) G0 X0 Y0 (parking position) M5 $1 (end)
Hole center spotting.	(Requires a small motion command or nothing happens) G21 (metric) F99999 (high feed rate) G0 X10 Y10 M3 $2 S1 (spotting on) G91 (relative distance mode) G1 X0.000001 G90 (absolute distance mode) M5 $2 (spotting off) G0 X0 Y0 G90 M2
Create temporary default material	(o=0, nu=2, na=5mm Mild Steel 40A, ph=3.1, pd=0.1, ch=0.75, fr=3000)
Edit material, if not existing create a new one	(o=2, nu=2, na=5mm Mild Steel 40A, ph=3.1, pd=0.1, ch=0.75, fr=3000, kw=1.0)

17.5. Mesa THCAD

The Mesa THCAD is a common way of obtaining the arc voltage from a plasma cutter and is also useful for ohmic sensing of the material during probing. The THCAD may be used for parallel port configurations as well as configurations using Mesa Electronics hardware. The THCAD is available in three different models, THCAD-5, THCAD-10, and THCAD-300.

There is a mode jumper on each THCAD card which should be set to **UNIPOLAR**

There is a frequency divider jumper on each THCAD card which should be set according to the hardware type:

Input Device	Recommended Setting
Parallel Port with very low latency	F/32
Parallel Port recommended starting point	F/64
Parallel Port with higher latency, or when cutting thick material	F/128
Mesa Card	F/32

This value is required to be entered into PnCconf during installation.

Note If using a parallel port it may be necessary for the user to adjust the jumper setting and the subsequent scaling values on the Parameters Tab to achieve optimal results. Symptoms may include random torch raises or dives during otherwise stable cutting. Halscope plots may be useful in diagnosing these issues.

Located on the rear of the THCAD is a calibration sticker showing:

```
    THCAD-nnn

0V  121.1 kHz
5V  925.3 kHz
```

or similar values, these values are required to be entered into PnCconf during installation.

PnCconf has entries for all required THCAD parameters and will calculate and configure any required settings. The calculations used are as follows:

Voltage Scale

```
vs = r / ((f - z) / d / v)
```

Voltage Offset

```
vo = z / d
```

r = divider ratio (see below).

f = full scale value from calibration sticker.

z = 0â□□V value from calibration sticker.

d = value from jumper above.

v = full scale voltage of THCAD

Divider Ratio

THCAD-5 or THCAD-10

If connecting to a plasma CNC port then the divider ratio is selected from the plasma machine. A common ratio used is 20:1.

If connecting to the plasma machines full arc voltage then a common setup for a THCAD-10 is to use a 1â□□MÎ© resistor from arc negative to THCAD negative and a 1â□□MÎ© resistor from arc positive to THCAD positive. The divider ratio is obtained by:

```
r = (total_resistance + 100000) / 100000
```

THCAD-300

```
r = 1
```

Important IF THE USER IS USING A HF START PLASMA POWER SUPPLY THEN EACH OF THESE RESISTANCES SHOULD BE MADE UP OF SEVERAL HIGH VOLTAGE RESISTORS.

Caution IF THE USER IS USING A HF START PLASMA POWER SUPPLY THEN OHMIC SENSING IS NOT RECOMMENDED.

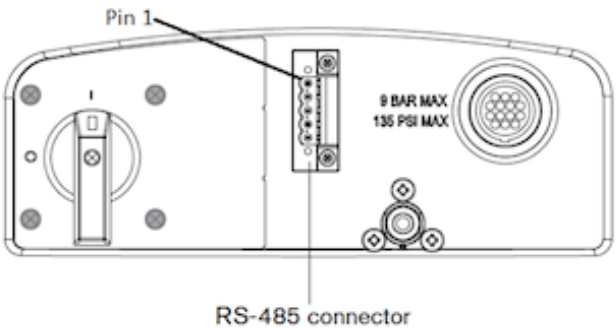
Note These values can be calculated by using this online calculator.

Note There is a lowpass filter available which may be useful if using a THCAD and there is a lot of noise on the returned arc voltage.

17.6. RS485 Connections

Hypertherm RS485 Wiring Diagram (wire colors inside the Hypertherm in parentheses):

Connection at Machine Pin #	Connection at Breakout Board
1 - Tx+ (Red)	->RXD+
2 - Tx- (Black)	->RXD-
3 - Rx+ (Brown)	->T/R+
4 - Rx- (White)	->T/R-
5 - GND (Green)	->GND



RS485 interfaces that are known to work:

DTECH DT-5019 USB to RS-485 converter adapter:

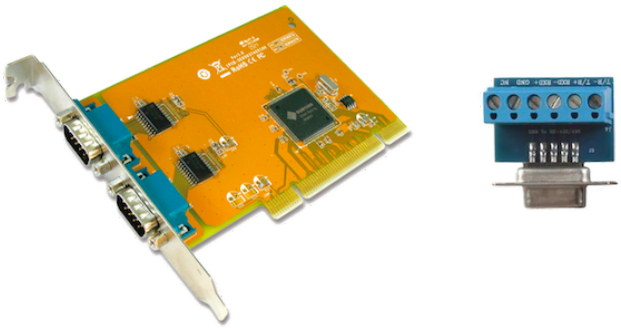


The following is necessary to convert a motherboard Serial connection or Serial card (RS232) to RS485:

DTECH RS-232 to RS-485 converter:



Serial card example (Sunnix SER5037A PCI Card shown with Breakout Board):

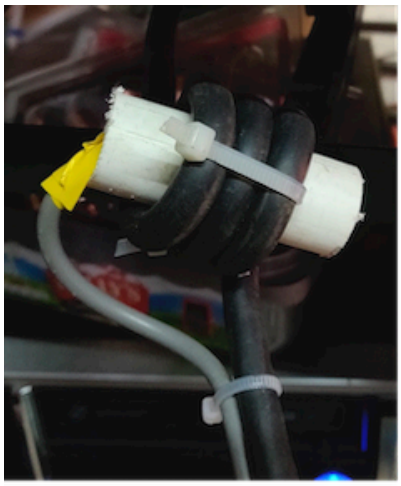


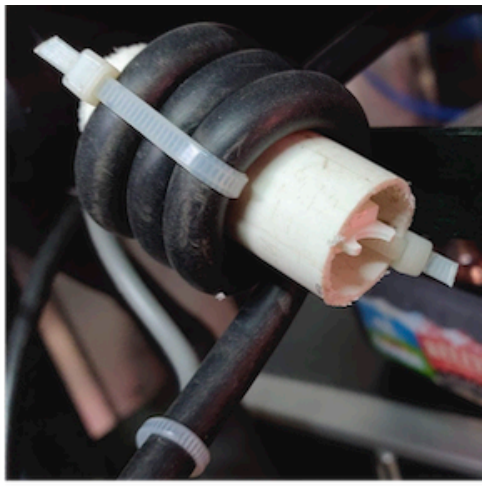
17.7. Arc OK With A Reed Relay

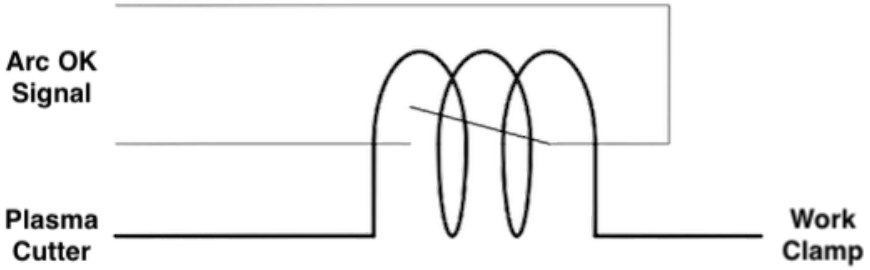
An effective and very reliable method of obtaining an Arc OK signal from a plasma power supply without a CNC port is to mount a reed relay inside a non-conductive tube and wrap and secure three turns of the work lead around the tube.

This assembly will now act as a relay that will switch on when current is flowing through the work lead which only occurs when a cutting arc has been established.

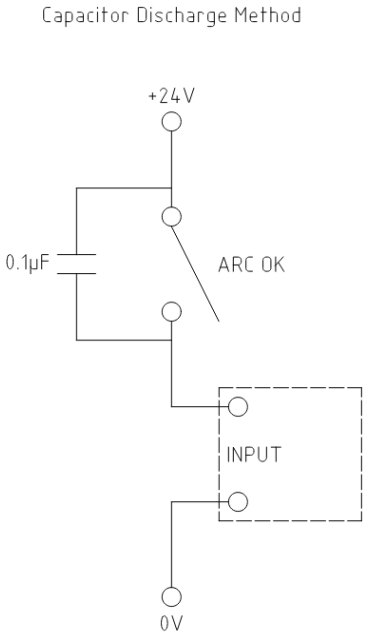
This will require that QtPlasmaC be operated in Mode 1 rather than Mode 0. See the QtPlasmaC Modes sections for more information.

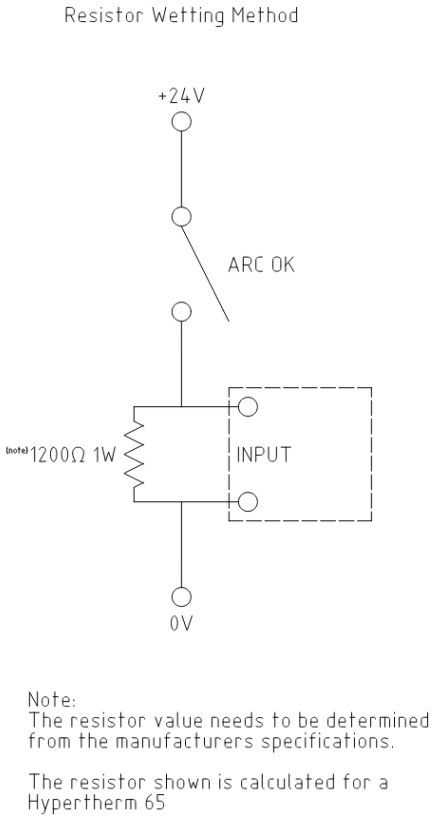






17.8. Contact Load Schematics





A full description is at Contact Load.

18. Known Issues

18.1. Keyboard Jogging

There is a known issue with some combinations of hardware and keyboards that may affect the autorepeat feature of the keyboard and will then affect keyboard jogging by intermittent stopping and starting during jogging. This issue can be prevented by disabling the Operating System's autorepeat feature for all keys. QtPlasmaC uses this disabling feature by default for all keys only when the MAIN Tab is visible, with the following exceptions when autorepeat is allowed with the MAIN Tab visible: G-code editor is active, MDI is active. When QtPlasmaC is shut down, the Operating System's autorepeat feature will be enabled for all keys.

If the user wishes to prevent QtPlasmaC from changing the Operating System's autorepeat settings, enter the following option in the **[GUI_OPTIONS]** section of the *<machine_name>*.prefs file:

```
    Autorepeat all == True
```

This issue does not affect any jogging using the GUI jog buttons.

> **Note** Disconnecting and reconnecting a keyboard during an active QtPlasmaC session will cause the autorepeat feature to re-enable itself automatically which may cause intermittent stopping and starting during jogging. The user must restart QtPlasmaC to disable the autorepeat feature again.

19. Support

Online help and support is available from the PlasmaC section of the LinuxCNC Forum.

The user can create a compressed file containing the complete machine configuration to aid in fault diagnosis by pressing following the directions in the backup section. The resulting file is suitable for attaching to a post on the LinuxCNC Forum to help the community diagnose specific issues.

---

1. In the US, the letter V is commonly used as a symbol (Voltage) and as a unit (Volt).