CRAFTING.BE

When engineering is in blood

ОТОБРАЗИТЬ МЕНЮ

Погружение в LinuxCNC

15.08.2013 / VALBER / 10 КОММЕНТАРИЕВ

Статья для тех кто плохо знает английский и не любит читать километры форумов

Недавно я участвовал в проекте одной перспективной молодой команды Jamlab , как видно я там есть)) Это статья скорее как фиксация некоего этапа работы , показывающая что я **сделал**, это важно иногда фиксировать навыки в виде балов в Skills, или новых выученных трюков в Feats(Передаю привет всем ролевикам).

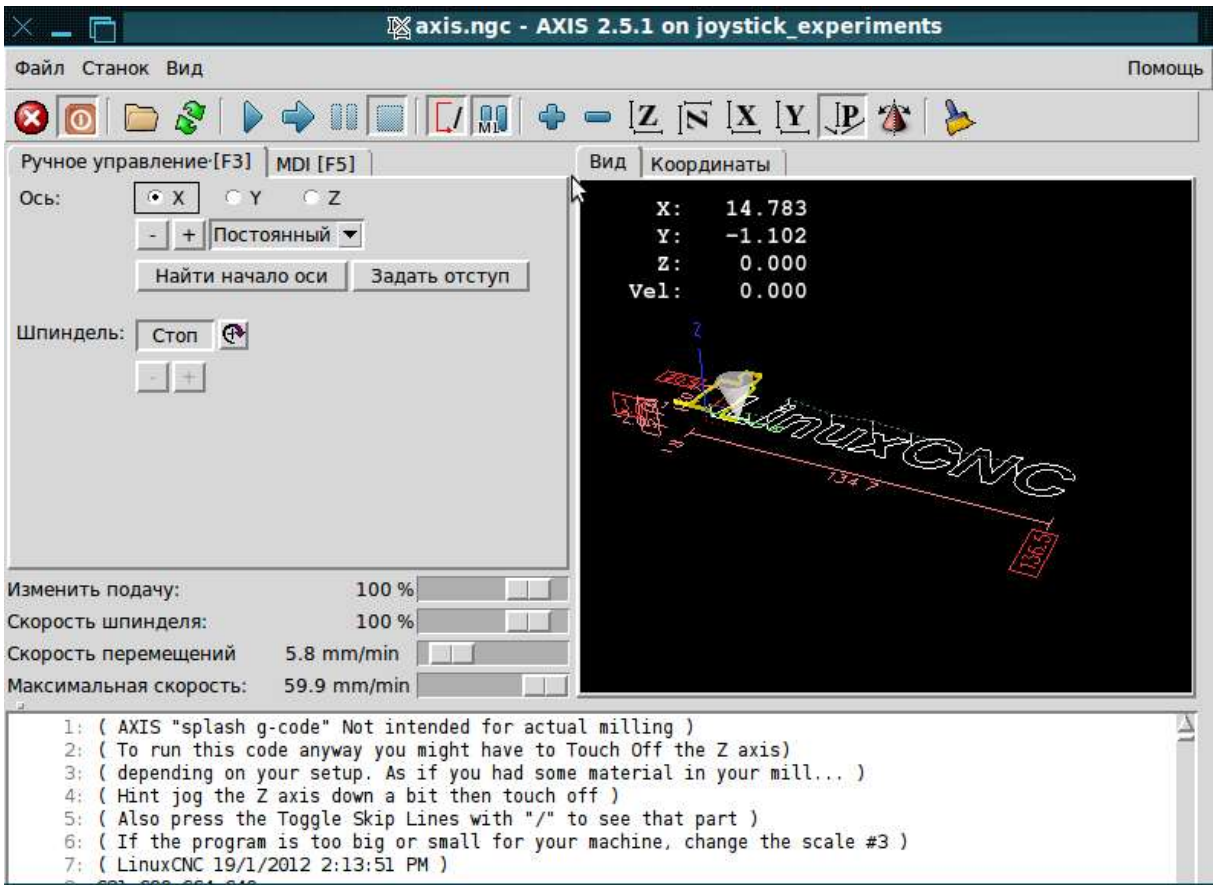


Table of Contents

Что такое LinuxCNC?

Linux обладает замечательными свойствами, его можно поставить куда хочешь даже на ATmega микроконтроллер. Или с помощью него можно сделать из обычного компьютера что то специфичное. Например на заре Linux его любили за то что он позволяет на дешевом оборудование создавать небольшие сервера, программно маршрутизировать пакеты и.т.д.

LinuxCNC — набор утилит который позволяет сделать из вашего компьютера стойку управления ЧПУ. Он позволяет программно генерировать шаги в случае управление по типу STEP-DIR-ENABLE, обрабатывать информацию с датчиков, позволяет вам собрать собственную заточенную под ваш станок — панель управления.

Ну а также LinuxCNC поддерживает работу с промышленными платами с аппаратной обработкой G-code — так называемые Mesa платы.

Состав LinuxCNC

RealTime Linux ядро

Простое ядро(ванильное) в тех.процессах связанных с реальным временем. Реализаций real time в linux несколько, конкретно linuxcnc использует RTAI и хотя

RTAI ушел далеко вперед(3.9 версия реализована) , на данный момент поддерживаются ядра 2.4 и 2.6 , вы скажите фи, а я отвечу что на производстве железо меняется крайне медленно, что большинство софта использует DOS а также что современный многоядерный монстр может показать… результаты на jitter time хуже чем одноядерный комп.

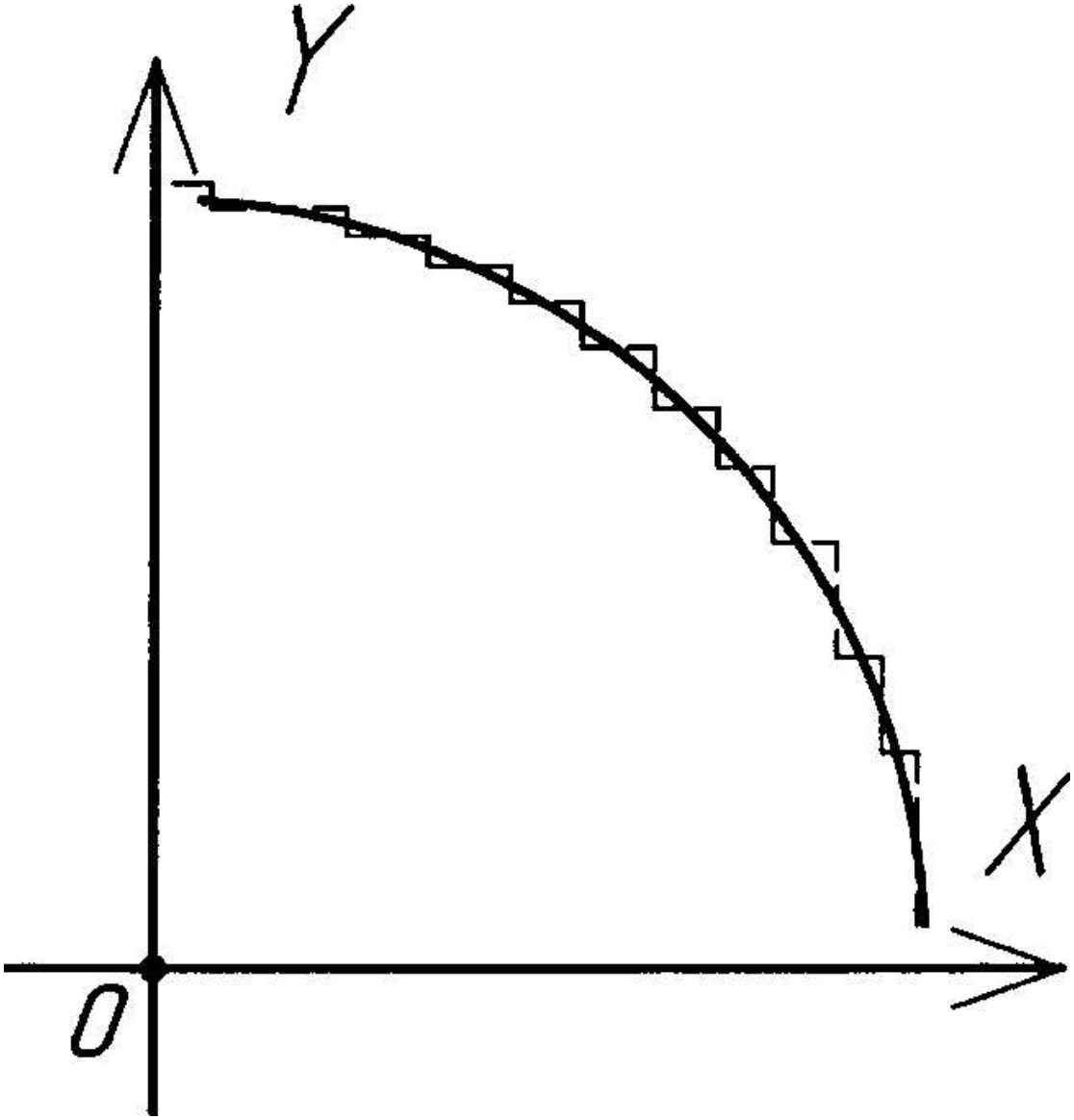
Драйвера

На данный момент хорошо реализованы и опробованы на практике, с работой в реальном времени , это параллельный порт(до 3-х штук) а также RS232 или COM-PORT.

Также ведутся работы(не могу оценить степень их активности), по реализации real time управления через Ethernet.

USB — использует буфферицацию и и говорить о настоящем real time не получается, в общем, с этим все сложно.

Итак представьте что вы делает сложную дугу, дугу можно представить как набор ступенчатых шажков.



Чем меньше размер шагов тем ближе мы к реальному изображению кривой, конечно это зависит и от минимального шага станка.. ну да ладно. В общем если наша задача увеличить частоту с которой компьютер передает сигналы управляющей плате, то нужно уменьшать параметр jitter time, один из способов это увеличение частоты процессора, также можно избавляться от «лишних» процессов в системе, ну там браузер выключить, музыку на этом компьютере не слушать, compiz отрубить, network demon-а развеять.... ИЗБАВИТСЯ ОТ ГРАФИЧЕСКОГО СЕРВЕРА, но это совсем хардкор и теоретически linuxcnc предполагает клиент-серверную модель, так что интерфейс будет работать на компе с браузерами и прочими кофеварками, а преобразователь G-code и вся управляющая логика на другом.

Другой вариант использовать другой вариант и на железе с помощью ПЛИС преобразовывать G-code в управляющие сигналы, количество читаемых/передаваемых компьютером команд заметно уменьшиться, для этого и существуют MESA платы стоят они дороже чем PCI-parrport переходник.

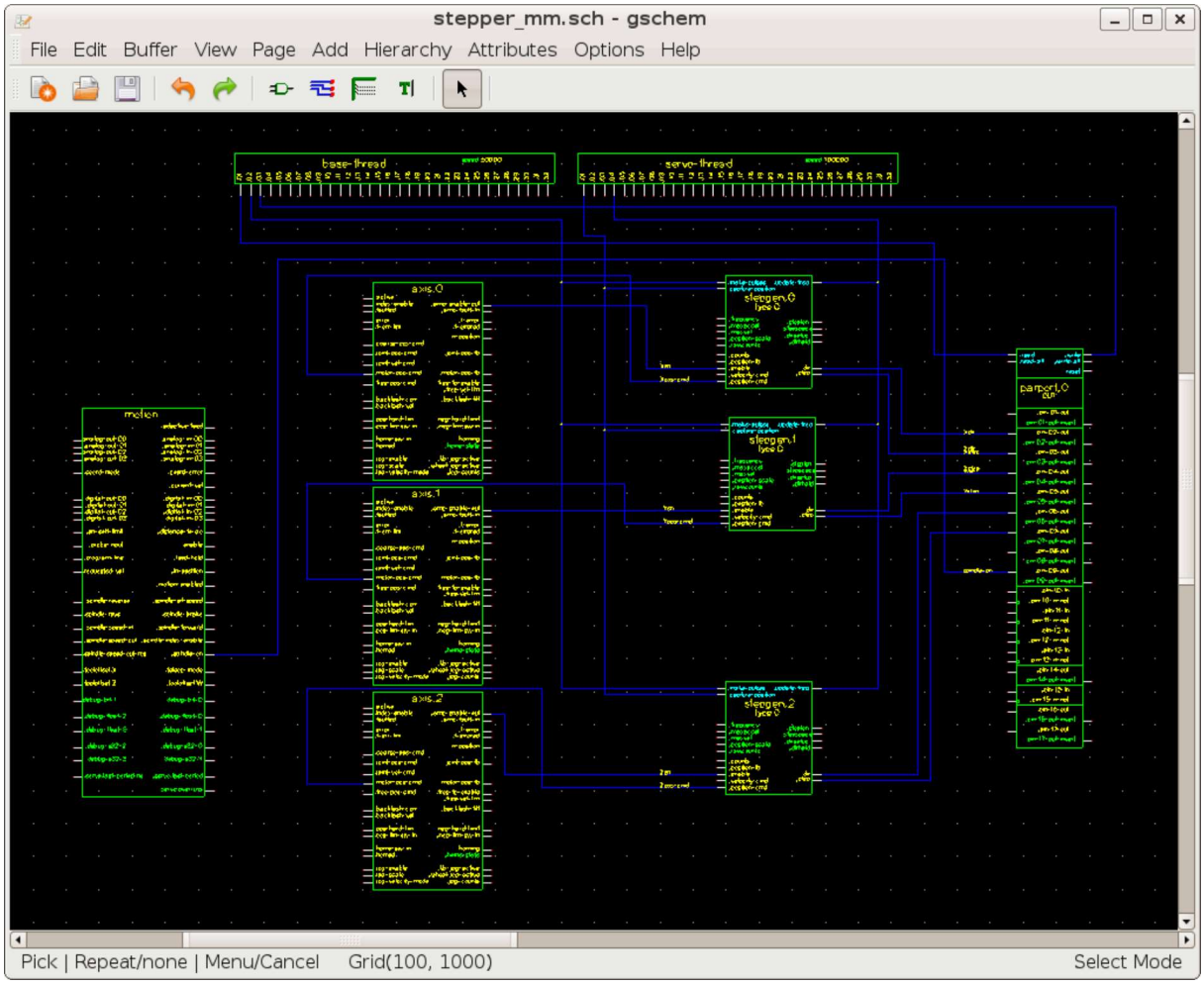
Чтобы знать больше читайте: мой перевод, официальной wiki , списки поддерживаемого оборудования wiki

HAL

Главная и самая интересная часть linuxcnc — Hardware Abstraction Layer. Это такая специальная прослойка которая позволяет обычному пользователю получать доступ к оборудованию(ядру) Linux. Раньше был ещё один HAL , который потом заменил проект udev , так вот это разные вещи, HAL Linuxcnc нужен только для linuxcnc.

Итак каждый компонент в HAL представлен черным ящиком с некоторым количеством ножек, каждая ножка обладает такими свойствами, как вход/выход а также тип передаваемого по ней сигнала. Ещё есть *псевдо* ножки, это константы, параметров компонента.

Все это похоже на блоксхемы и в итоге можно представить как вот такую вот схему(чем-то напоминает LabView, но увы удобного редактора нет).



Пример конвертера GEDA2HAL

Ещё стоит заметить что в HAL всегда работают два типа компонентов ,работают они в разных потоках loadrt загружает компоненты работающие в real time потоке, loadusr загружает компоненты работающие не real time т.е. с периодом >=200 мс ... это например интерфейс или джойстик подключенный по USB.

Также с помощью специального синтаксиса макросов и языка C, можно создавать свои компоненты, с помощью утилиты comp.

Также можно вручную вводить команды с помощью halcmd , с поддержкой автодополнения, ведь файлы с расширением .hal всего лишь сценарии написанные на этом языке.

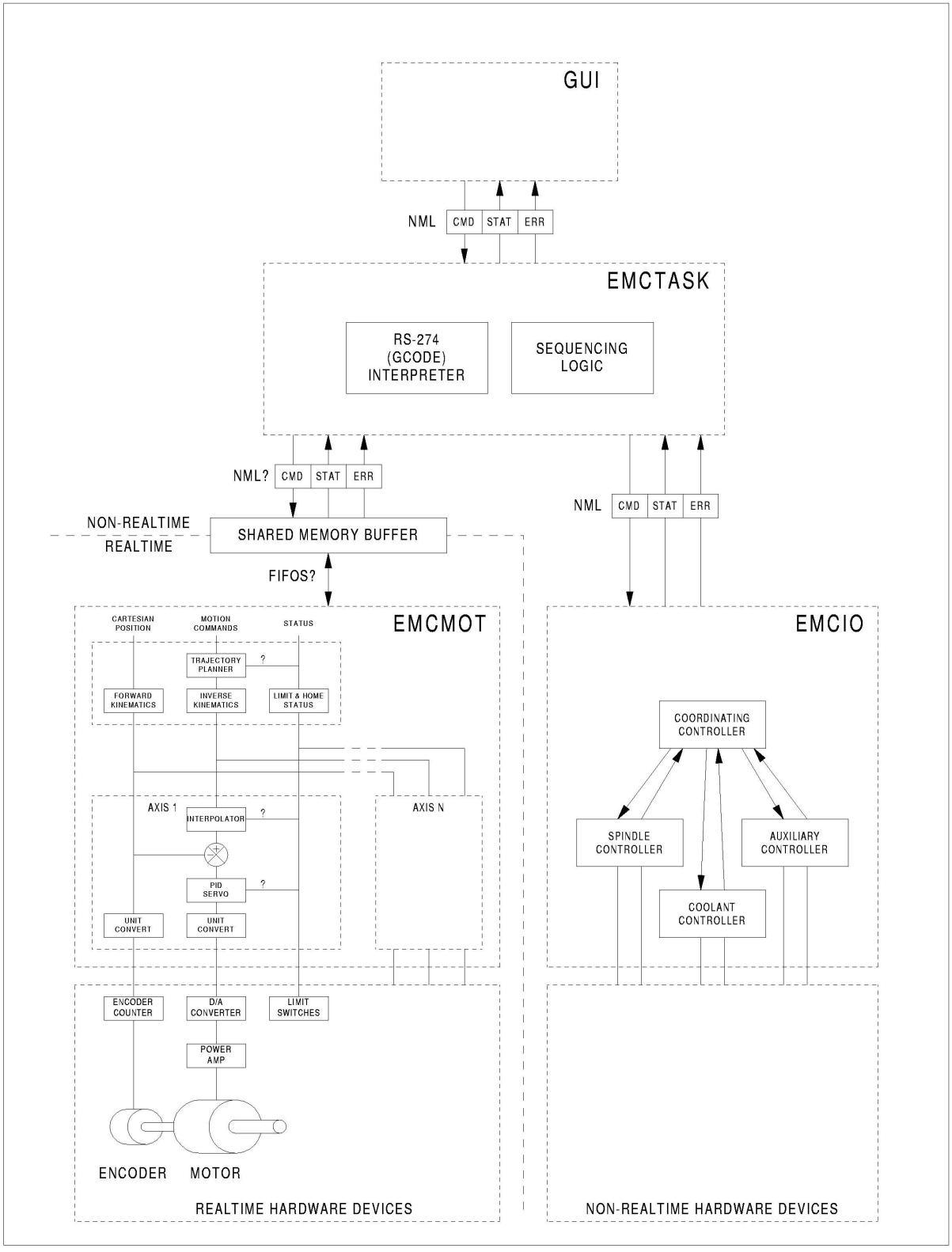
Подсветка синтаксиса

Для того чтобы .hal файлы настроек не выглядели однообразно, сделал подсветку для Emacs

```
;;Подсветка синтаксиса для HAL в LinuxCNC
;;GPLv2
(define-generic-mode hal-linuxcnc-generic-mode
  '(("#".?\n) ("/*"."*/"))
  '("or" "and")
  '(;;("\\(#<_?[A-Za-z0-9_]+>\\)" (1 font-lock-type-face))
    ;;("\\([NnGgMmFfSsTtOo]\\)" (1 font-lock-function-name-face))
    ("\\(net\\|loadrt\\|loadusr\\|addf\\|setp\\|linksp\\|newsig\\)" (1
font-lock-function-name-face))
    ("\\
(axis\\|stepgen\\|parport\\|pwmgen\\|iocontrol\\|motion\\|classicladder\
\)" (1 font-lock-builtin-face))
    ("\\([ \t]+-+[A-Za-z0-9_]+\\)" (1 font-lock-string-face))
    ("\\(AXIS_\\|EMCMOT\\|TRAJ\\)\\(servo\\|base\\)-thread" (1 font-
lock-constant-face))
    ("\\(step\\|dir\\|enable\\)" (1 font-lock-type-face))
    ("\\([0-9]+\\)" (1 font-lock-constant-face))
    )
  '("\\.hal\\'")
  nil
  "Generic mode for .hal file. LinuxCNC HAL")
```

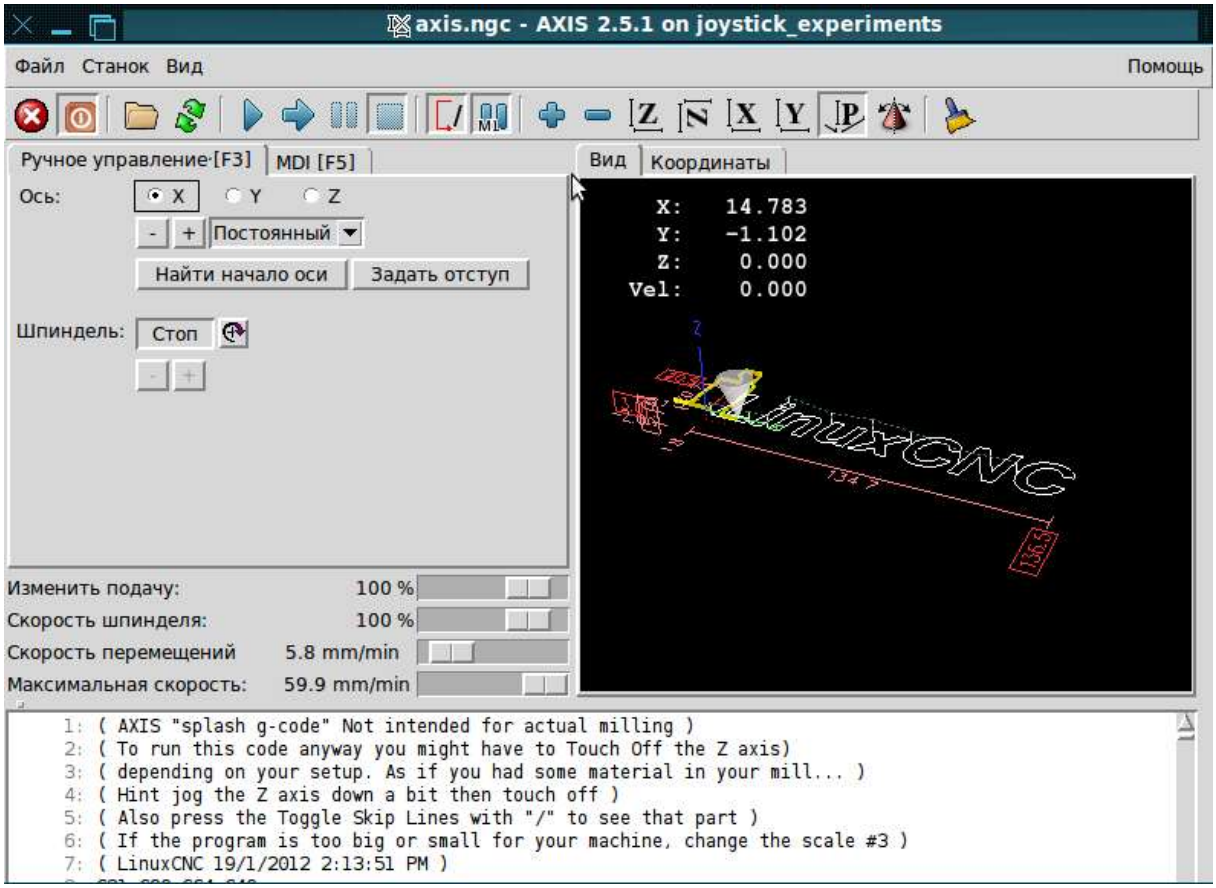
Внутренняя архитектура

Здесь представлены компоненты HAL а также их деления на real time и non real time



в linuxcnc входит свой собственный интерпретатор G-code RS274NGC , который можно дополнять пользовательскими M-кодами, написанными на bash и .hal.

Набор графических интерфейсов

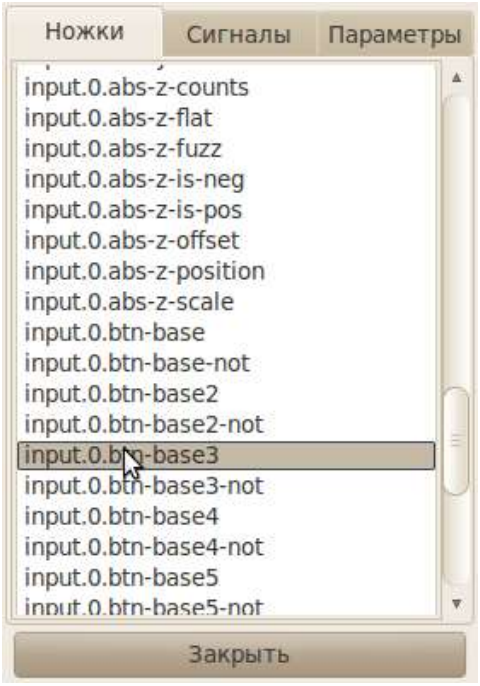


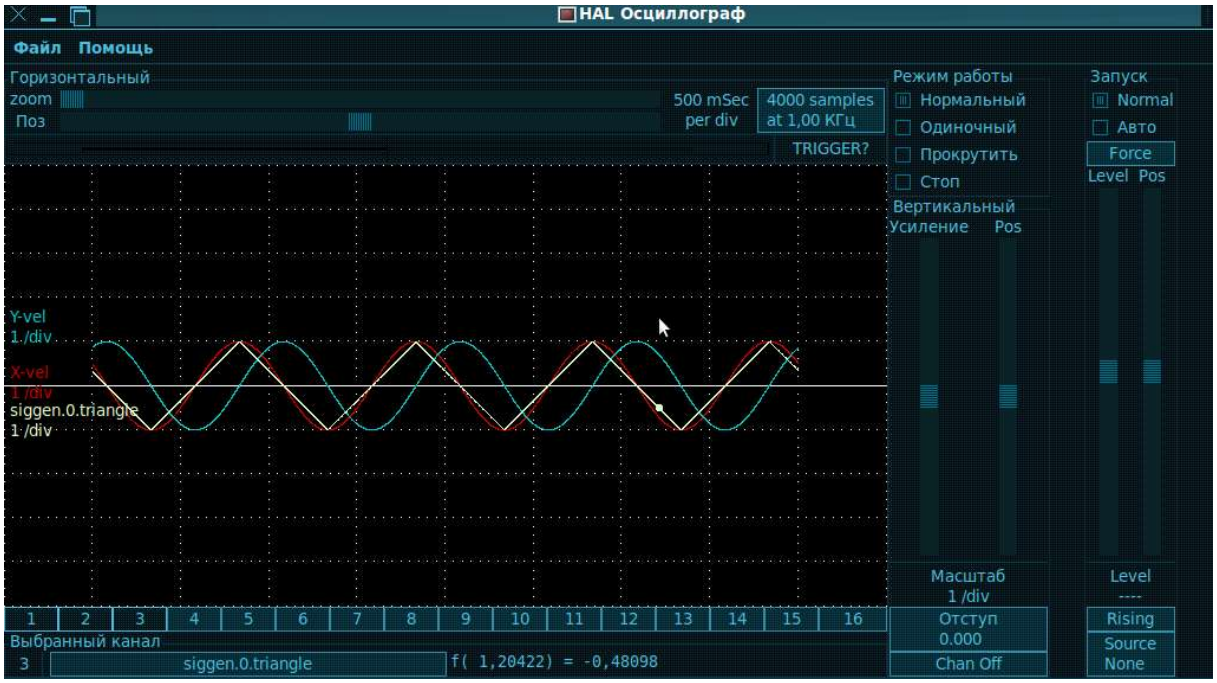
- Главный интерфейс который часто изображен называется — AXIS , можно дополнять этот интерфейс с помощью python , а также графических элементов из Tk/Gtk библиотек.
- TkLinux — по проще и менее жрущий
- Touchy интерфейс для планшетов
- Keystick самое минимальное из минимально возможного.

В общем, AXIS это основной интерфейс и его можно расширять PyVCP и GladeVCP , второе краше и Gtk виджеты лучше приспособлены для touch интерфейсов.

Измерительные и вспомогательные утилиты.

halmeter и halscope позволяют отслеживать сигналы проходящие между компонентов, разница такая же как между вольтметром и осциллографом.





Есть ещё Latency Test , которые измеряет минимальное время в которое может откликаться система — jitter time

С чего начать?

Если вы решили освоиться с linuxcnc

- cnc-club.ru — здесь переводы о том что есть что.
- JamLAB переводы в том числе и мои
- LinuxCNC wiki
- LOR

Рубрики: САПР Теги: cam, cnc, gcode, hal, linuxcnc

« AVR — Assembler! Использование макросов в KiCAD »