

## LinuxCÑC

LinuxCNC en castellano

# Que es linuxCNC

 [linuxcncs](#)  [27 octubre, 2019](#)  [Linuxcnc basico](#)  [Editar](#)

El primer concepto que debe quedar bien sentado es que *linuxcnc* es, en esencia, un controlador software de posición. La misión de un controlador de este tipo es conocer en todo momento y bajo cualquier circunstancia la posición de un determinado punto en el espacio.

Para mecanizar, el punto de interés es aquel donde la herramienta ataca al material, que deberá ser mantenido bajo absoluto control en sus desplazamientos, modificaciones de situación por cambio de herramienta, cambios de sistema de referencia y muchas circunstancias mas. Esa es la misión de un controlador de posición.

El paquete *linuxcnc*, siguiendo la filosofía UNIX, es altamente modular. Esto significa que esta constituido por una gran cantidad de programas, módulos, archivos de datos, scripts (guiones), etc.

## Convenciones

– Cualquier palabra en cursiva, *como estas*, representan nombres con significado concreto; aplicaciones, paquetes, documentos, etc.

– Una linea como esta:

```
> git gui
```

representa una linea de comando de terminal de Linux. El símbolo utilizado, “>”, representa un prompt genérico, distinto de un usuario a otro. El resto de la linea es el contenido literal del comando a introducir en la linea de comandos

– Cualquier texto encerrado entre corchetes angulares “<>”, como `<este_texto>` significa que se debe sustituir “este\_texto” por lo que corresponda en su sistema, sin los caracteres “<” y “>”. Por ejemplo, si tu directorio de usuario, en el directorio de sistema /home, se llama “mi\_directorio”, la expresión:

```
> cd /home/<directorio_de_usuario>
```

significará en tu caso concreto:

```
> cd /home/mi_directorio
```

– Una palabra en negrita dentro de un texto normal representara generalmente el nombre de una orden, comando o programa, o bien se usa como resaltado de algo importante.

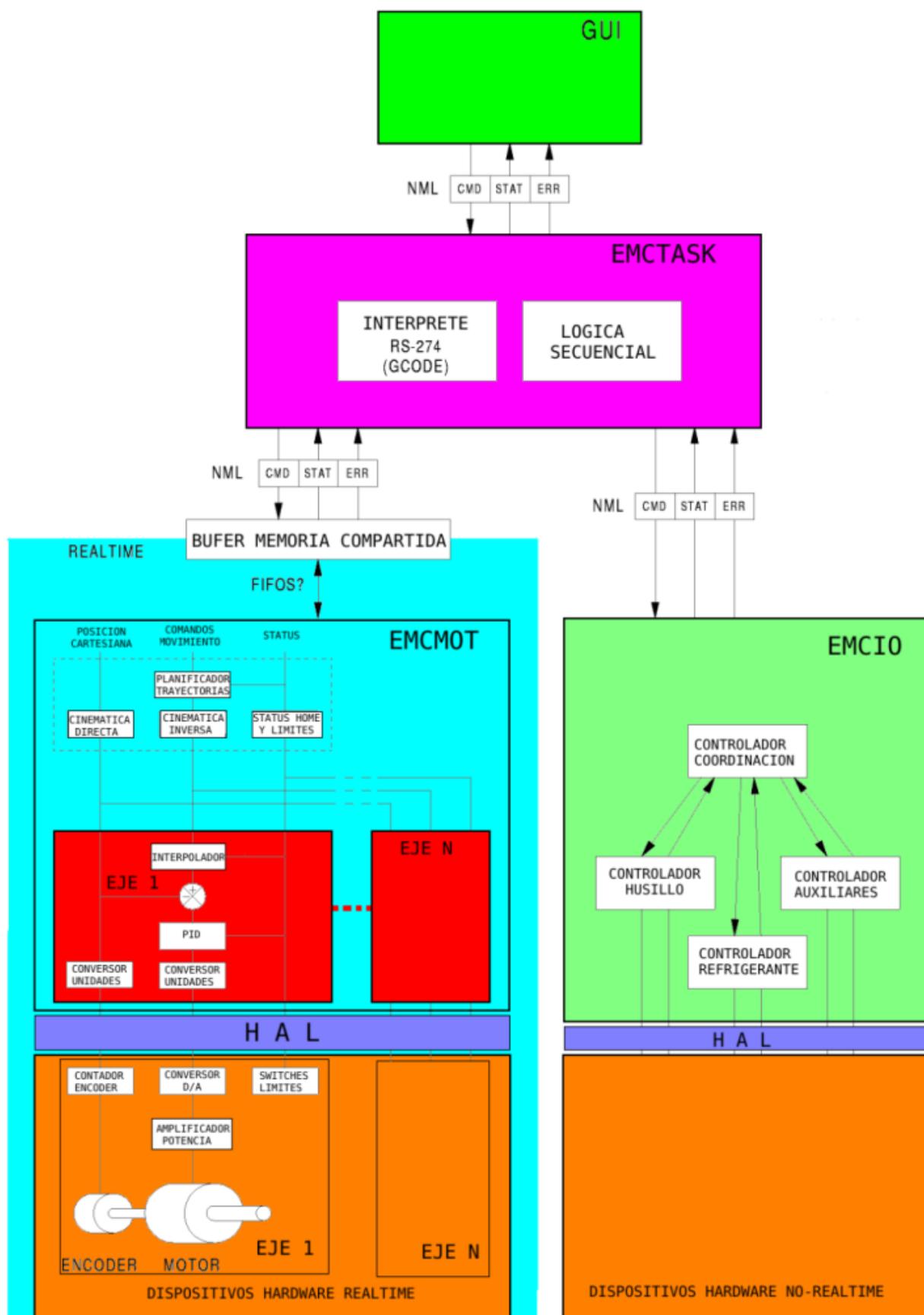
## Cómo funciona linuxCNC

Un controlador de movimiento de maquinas puede ser construido totalmente mediante hardware o bien parte de el puede ser software. *linuxcnc* entra en la segunda categoría.

Hay cinco componentes principales en el software *linuxcnc* que, en bloque, permiten su uso como controlador de una maquina herramienta... y mas cosas:

- un controlador de movimiento ([EMCMOT](#))
- un controlador de entradas/salidas discretas ([EMCIO](#))
- un ejecutor de tareas que los coordina ([EMCTASK](#))
- una interfaz gráfica de usuario ([GUI](#))
- una capa de abstracción de hardware ([HAL](#))

El esquema general de la organización del software que conforma *linuxcnc* es el de la siguiente figura:



Esto, visto por primera vez, asusta y mucho. Pero realmente es mas sencillo y lógico de lo que parece. Vamos a diseccionar este esquema, usando los mismos colores que las correspondientes partes.

- La comunicación entre las diferentes partes se realiza mediante el llamado NML o *lenguaje neutro de mensajes*. Los mensajes pueden estar constituidos por una orden o comando (cmd), por una información de estado (stat) o por información de errores (err).
- La interface gráfica de usuario, o GUI, interactúa con el operador del sistema. Es la parte de *linuxCNC* que vemos en el monitor de nuestro ordenador, a la que damos mas importancia.....y la que menos tiene, créeme.
- El planificador de tareas, EMCTASK, distribuye las tareas que se deben realizar para cumplir con los mensajes recibidos desde la GUI o desde el interprete del lenguaje RS274 (el programa que analiza y convierte el famoso g-code a ordenes internas que son las que realmente ejecuta la maquina). Este bloque está relacionado tanto con los bloques que necesitan tiempo real como con los que no lo necesitan. Reparte las tareas a uno u otro bloque en función de esa necesidad. Vamos a aclarar esto un poco mas. Supongamos una maquina con un cambiador de herramientas. Un cambio de herramientas es un proceso SECUENCIAL. Será mas rápido o mas lento, pero no necesita tiempo real (ya discutiremos ese concepto tan malentendido de tiempo real). Lo que si es cierto es que un cambio de herramienta necesita seguir una SECUENCIA perfectamente definida de acciones. EMCTASK se asegurara de que esa secuencia sea correcta y enviara la secuencia de ordenes que el modulo EMCIO usará para activar los accionadores del cambiador que sean necesarios. En el otro extremo tenemos los procesos que necesitan tiempo real. Fresar una circunferencia puede ser un caso. Se necesita una perfecta coordinación espacio-temporal entre dos ejes para que resulte una circunferencia perfecta. Ya no hay movimientos secuenciales, sino coordinados. EMCTASK enviara este tipo de ordenes a la parte de tiempo real de *linuxCNC*.
- Los mensajes de tareas que **no** precisan tiempo real pasan al modulo EMCIO que, mediante controladores software independientes y a través de la capa de abstracción de hardware HAL, provocan las acciones correspondientes en los dispositivos de hardware adecuados (lubricación, refrigeración, cambiadores, etc)
- Los mensajes de tareas que **si** precisan tiempo real pasan a una zona especial de memoria; el *buffer de memoria compartida*. El controlador software de movimiento, EMCOT, accede a esta memoria para procesar y mantener datos generales de la maquina (posición global, cinemática, posición home, limites, etc) y datos de cada uno de los ejes, EJE 1 a EJE N, mediante controladores software independientes. EMCOT puede devolver información a la memoria compartida, de donde EMCTASK y GUI pueden extraerla para actualizar sus datos.
- La capa de abstracción de hardware HAL actúa como conexión entre el hardware real y el resto del software. De esta forma, no es preciso “escribir” ordenes especiales para cada tipo de hardware para provocar las acciones correspondientes en los dispositivos de hardware que precisan tiempo real.

Es mas, las mismas ordenes funcionaran de igual manera en configuraciones de equipos diferentes.

- La parte restantes son los dispositivos físicos reales; motores, encoders, relés, etc, tanto los que necesitan control bajo tiempo real como los que no. **EMCTASK** ya hizo el trabajo con antelación para diferenciar a donde enviar según que tipo de orden.

A partir de aquí debería comenzar la disección de LinuxCNC, pero antes ¿que tal un poco de Linux puro?. Si no lo has hecho ya y tu experiencia con Linux es escasa (o nula), te recomiendo encarecidamente que veas la categoría **S.O. Linux**

Me gusta

Sé el primero en decir que te gusta.

---

#### Relacionado

El primer Post  
En "Miscelanea"

Compilación RIP (2.7.14)  
En "Linuxcnc avanzado"

Información para torno  
En "Sin categoría"

 [linuxcncs](#)  [27 octubre, 2019](#)  [Linuxcnc basico](#)  [Editar](#)

**LinuxCÑC.**