

# Control de Movimientos mediante Bus EtherCAT y LinuxCNC

I. Rosales González, V. Torres López y J.M. Gómez de Gabriel

Dto. Ingeniería de Sistemas y Automática  
Universidad de Málaga  
{narogon, vetorres, jmgomez}@uma.es

## Resumen

*En este artículo se describe la aplicación del bus de campo EtherCAT al control de movimientos mediante software de código abierto. EtherCAT es un protocolo de comunicación en tiempo real, de alta velocidad y muy baja latencia, basado en Ethernet que en este caso se utiliza para la coordinación del control de movimientos de un conjunto de motores sin escobillas. En concreto se ha desarrollado el control de una máquina de ensamblado de circuitos con componentes SMD, con una cinemática no convencional, basada en un robot delta de cuatro grados de libertad. Se detallan los pasos seguidos y los resultados del uso de éste bus en lugar de las tradicionales líneas analógicas 0-10V. Este artículo pretende servir de guía en la construcción de máquinas-herramienta modernas con código abierto, y su uso para la educación de los futuros ingenieros en automática.*

**Palabras clave:** Control de movimientos, CNC, EtherCAT, LinuxCNC, Comunicaciones en tiempo real

## 1. Introducción

En los últimos años la robótica y la automatización se han expandido desde el ámbito industrial para llegar a un gran número de usuarios, los cuales pueden participar de ellas debido principalmente a la bajada de costes así como al aumento de las capacidades de los actuales ordenadores.

La tendencia, por tanto, es transformar los sistemas de hardware y software cerrados por otros basados en licencias de código abierto que permitan la participación y la colaboración de un gran número de usuarios. Un ejemplo es el software de control de máquinas-herramienta, LinuxCNC que cuenta con una comunidad activa de más de 5000 usuarios en su foro[4] o un chat activo las 24 horas del día lo que lo convierten en una plataforma en continua expansión con un soporte muy rápido y eficaz.

La bajada de precios en el hardware, particularmente por la entrada en escena del mercado on-

line que posibilita la compra en casi cualquier lugar del mundo, acerca en gran medida estas tecnologías a usuarios finales. Esto ha dado lugar a diseños novedosos en muchos campos, entre los que se puede destacar la creación de impresoras 3D.

Por su lado los puertos de comunicación han tendido también a normalizarse, instalándose buses muy generalizados, como puede ser: USB, Ethernet, o CANbus; en la mayoría de los elementos.

En cuanto a la conectividad del hardware, en tiempo real, la normalización de protocolos de comunicación por parte de CiA (*CAN in Automation*) ha dado lugar a varias alternativas sobre Ethernet, véase Ethernet Powerlink[2] o EtherCAT, de código abierto y con respaldo de grandes empresas del sector.

LinuxCNC es un paquete de software libre para el control de máquinas CNC. La principal ventaja es que resuelve completamente el problema de la gestión de trayectorias y control automático de forma gratuita y abierta, lo que lo hace muy interesante para su uso docente. Como contrapartida, a pesar de su popularidad, como todo software libre su uso para casos distintos a los más convencionales y bien documentados requiere de ciertos conocimientos muy específicos y difíciles de alcanzar para usuarios noveles.

EtherCAT es un protocolo de comunicación de código abierto y alto rendimiento que pretende utilizar el bus Ethernet en un entorno industrial. El uso de este protocolo con LinuxCNC está en sus inicios por lo que no resulta trivial su puesta en marcha. Sin embargo, tal y como se demuestra en el caso práctico que se ha llevado a cabo en el contexto del presente artículo no es un objetivo inalcanzable.

Tanto LinuxCNC como EtherCAT master son software libre. El primero bajo licencia GPL (*General Public License*) y el segundo bajo la segunda versión de esta (GPLv2). Resumiendo, estas licencias garantizan la libertad de copia, distribución y modificación de los programas. Por otro lado la obligación del usuario será respetar las mismas y no coartarlas de cara a cualquier usuario, sin em-

bargo esto no restringe los usos comerciales.

Además LinuxCNC cuenta con licencia LGPL (*Lesser General Public License*) que permite la coexistencia de código libre y no libre simultáneamente. Esto quiere decir que no existirá la obligación de distribuir abiertamente ni permitir la modificación de los programas o códigos creados por nosotros que funcionen sobre el software bajo licencia LGPL.

Por tanto el modelo de negocio que tiene más sentido en este caso es la oferta de servicios adicionales al propio software: instalación y puesta en marcha, adaptación a las necesidades del cliente, soporte técnico y resolución de problemas, etc; en conjunción con el diseño e instalación del hardware necesario.

Éste es un artículo que tiene por objeto el servir de guía para la configuración y puesta en marcha del controlador de una máquina CNC con EtherCAT usando LinuxCNC.

El documento está dividido en 4 secciones. En las secciones 2 y 3 se presentan los conceptos básicos de EtherCAT y LinuxCNC. A continuación, en la sección 4, se describe el caso práctico que se ha llevado a cabo. Por último la sección 5 es una guía de inicio para la instalación y uso de LinuxCNC con EtherCAT.

## 2. Protocolo EtherCAT

EtherCAT es un protocolo industrial abierto. Desarrollado específicamente para aplicaciones en las que el tiempo real es un elemento crucial, usa el ancho de banda disponible en una red Ethernet estándar con una gran eficiencia. Esto se consigue mediante un circuito integrado de aplicación específica (ASIC), en todos los nodos de la red. El circuito ASIC se encarga del intercambio de datos del proceso «sobre la marcha», de manera que se reduce al mínimo el retardo del telegrama en cada nodo. Así, se dispone de un sistema de bus de campo de gran velocidad.

Las características principales del protocolo son: tiempo real (determinismo y sincronismo mediante relojes distribuidos), redundancia ante cortes en el bus de comunicación y una topología de conexión flexible.

El protocolo EtherCAT está impulsado por *The EtherCAT Technology Group* (ETG) del cual forman parte más de 2500 empresas del sector de la automatización, entre las que Beckhoff es la desarrolladora principal.

### 2.1. Principio de funcionamiento

El protocolo transmite la información mediante el estándar de tramas de Ethernet IEEE 802.3, siendo su *ethertype* el 0x88a4, lo que evita modificar el marco Ethernet estándar.

La principal ventaja de este sistema radica en que los datos ya no tienen que ser enviados, procesados, y devueltos por cada nodo uno a uno, si no que en cambio, gracias al FMMU (*Fieldbus Memory Management Unit*), los datos son procesados al vuelo sufriendo solo un retraso de nanosegundos al paso por cada nodo.

Una característica fundamental del protocolo es que se comparte entre todos los esclavos un único reloj que funcionara como referencia temporal del sistema. Cada esclavo relacionará su reloj interno con el reloj de referencia, midiendo los retrasos entre los paquetes del par de ida y del par de regreso, de manera que en todo momento el tiempo en todos los elementos de la red queda completamente determinado y sincronizado.

La topología de la red EtherCAT es muy robusta, ya que forma un «anillo virtual» con ambos pares de un cable Ethernet. Las configuraciones que permite el protocolo son muy numerosas y variadas. Es posible realizar el conexionado en serie, con ramas, en árbol, etc. Incluso si se cuenta con más de una tarjeta Ethernet en el *master* se puede conectar como primer y último elemento de la red, lo que permitirá mantener la conectividad incluso si se corta la comunicación en algún punto de la red.

Una máquina EtherCAT, en cuanto a su conectividad, se comporta como una máquina de estados finitos. Cuatro, más uno opcional para actualizar el *firmware*, son los posibles estados en los que se puede encontrar: INIT, PREOP, SAFE-OP, OP y BOOTSTRAP (figura 1). En función del estado en el que se encuentre se podrán realizar intercambios de datos de forma síncrona, asignar valores de configuración o reprogramar la *EEPROM* interna entre otras actividades tal y como se resume en la tabla 1.

	Comunicaciones
Init	No se permite
Pre-OP	SDO
Safe-OP	SDO y consulta PDO
OP	Completa
Bootstrap	FoE

Tabla 1: Comunicaciones en función del estado

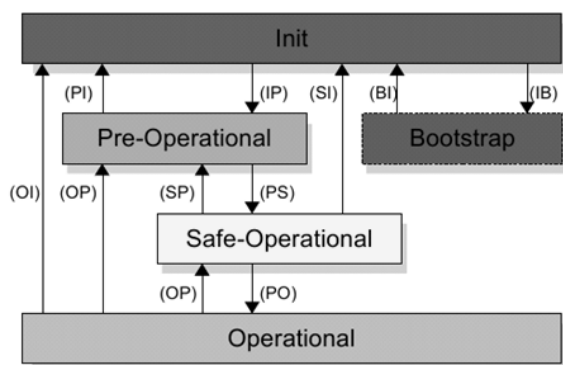


Figura 1: Máquina de estados EtherCAT

## 2.2. Tipo de datos

El protocolo EtherCAT permite transmitir diversos tipos de datos.

- CoE - CAN over EtherCAT
- SoE - SERCOS over EtherCAT
- EoE - Ethernet over EtherCAT
- FoE - File over EtherCAT
- VoE - Vendor over EtherCAT

Los que más nos pueden interesar son EoE que permite a un esclavo comunicarse a través de internet y CoE que será el más usado por la mayoría de los esclavos. Este estándar es una herencia del Bus CAN, y se ha mantenido en la actualidad tanto para redes que usan EtherCAT como otras como pueden ser Ethernet POWERLINK[2] o Profibus.

Dentro de los estándares que conforman el CAN y más concretamente el CANopen cabe destacar el CiA 402[5] que es el que describe el comportamiento de servodrivs, variadores de frecuencia y motores paso a paso.

El protocolo CAN ordena los datos según su propósito en índices numerados en hexadecimal que van desde el 0 hasta el FFFF. En la tabla 2 se ve la clasificación.

## 3. LinuxCNC

LinuxCNC es un software libre de control numérico de máquinas, principalmente enfocado para máquinas herramienta (fresadoras, tornos...) aunque puede ser utilizado para muchas otras aplicaciones que requieran de un control.

El software debe ejecutarse sobre un *kernel* en tiempo real, imprescindible para el control realimentado de movimientos. Actualmente es habitual

0000	No se usa
0001 - 001F	Tipos de datos estáticos
0020 - 003F	Tipos de datos complejos
0040 - 005F	Tipos de datos específicos
0060 - 0FFF	Reservado
1000 - 1FFF	Perfil de comunicación
2000 - 5FFF	Datos definidos por fabricante
6000 - 9FFF	Datos normalizados
A000 - FFFF	Reservado

Tabla 2: Clasificación de índices CANopen

usar Ubuntu 10.04 (*Lucid Lynx*) con el módulo RTAI (*Real Time Application Interface*) instalado en su núcleo.

Para la configuración e interconexión del numeroso hardware que soporta LinuxCNC, este incorpora la HAL (*Hardware Abstraction Layer*). En esta pseudocapa de hardware se cargarán todos los bloques que se necesiten. Bloques que en algunos casos representarán hardware real y en otros funciones de software. Como por ejemplo cálculos de posición para cinemáticas no triviales o funciones lógicas que puedan requerirse. Será importante diferenciar si las funciones que se cargarán realizan cálculos en punto flotante o no, pues el tiempo de ejecución varía en gran medida.

Cada bloque que se cargue en la HAL cuenta con unos pines (variables) de entrada y salida mediante los cuales reciben o envían información. Los enlaces entre pines son denominados señales en la nomenclatura de LinuxCNC.

Algunos componentes de tiempo real básicos para el funcionamiento son: Axis, Motion, Iocontrol, Halui, bloques lógicos y conversores de formato

Para definir que bloques se cargarán en nuestro entorno en tiempo real se crean archivos *.hal* que contendrán también todas las señales que se desee establecer y los valores iniciales de aquellos pines que se deseen inicializar.

Para la configuración de una máquina se crea un archivo *.ini* que incluye todos los parámetros que definen el sistema: número de ejes, grados de libertad, velocidades máximas, aceleraciones, errores de seguimiento permitidos, etc. Será también en este archivo donde se indique que archivos *.hal* han de cargarse. El esquema de configuración se muestra en la figura 2.

Además, existe la posibilidad de crear paneles de control virtuales que mostrarán por pantalla valores en tiempo real, mediante cuadros de texto, barras de colores, leds, contadores..., y permiten el control de variables por parte del usuario mediante botones, diales... todos ellos totalmente configu-

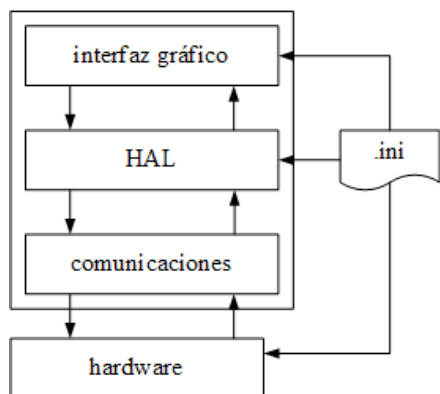


Figura 2: Esquema de configuración

rables mediante un sencillo lenguaje de etiquetas o programando en Python si se precisa una mayor personalización. Estos paneles de control serán bloques en el entorno HAL que habrán de conectarse mediante las señales anteriormente descritas.

Existen también diversas disposiciones gráficas que se pueden elegir para la pantalla principal del software. Unas enfocadas a entornos más visuales y otras más sencillas con solo texto, pasando por algunas pensadas para entornos táctiles. La más común de ellas se muestra en la figura 3.

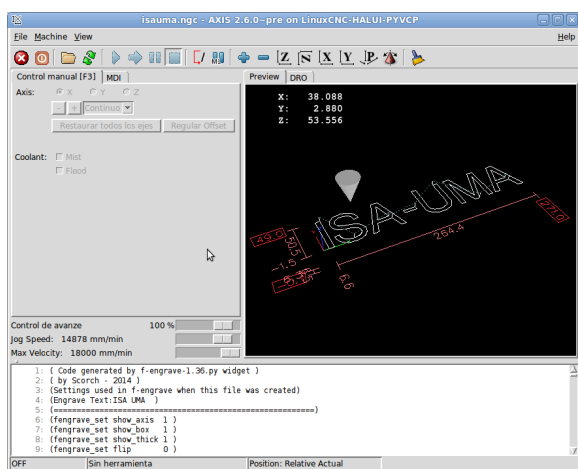


Figura 3: Pantalla principal LinuxCNC

LinuxCNC utiliza el estándar rs274[6], más conocido como código G, habitual de las máquinas herramienta, añadiendo algunos comandos especiales que aumentan las posibilidades de control.

La posibilidad de crear funciones que trabajan en tiempo real como bloques HAL es infinita puesto que el usuario solo tiene que programar el código en C y añadir una pequeña cabecera que define los pines de entrada y salida. Estos archivos se guardaran con extensión *.comp* y se pasan por un precompilador con el mismo nombre incluido en LinuxCNC que se encarga de añadir el resto de

rutinas necesarias para el funcionamiento del bloque en tiempo real.

Otra de las ventajas de LinuxCNC es que permite controlar salidas mediante el mismo código G. Utiliza los comandos M64 y M65 para la activación o desactivación de los bits *motion.digital-out-xx* que al enlazarse con los pines del bloque EtherCAT quedan conectados a las salidas físicas deseadas.

#### 4. Caso práctico

El departamento de Ingeniería de Sistemas y Automática de la Escuela Técnica Superior de Ingeniería Industrial de Málaga dispone de un robot experimental paralelo en configuración delta (figura 4), que fue premiado en el concurso OMRON 2011/2012 de la CEA. Su movimiento originalmente era gestionado por un controlador de automatización de máquinas de la serie NJ5 de OMRON, programado mediante (*Sysmac Studio*). La función principal del robot en la actualidad es la colocación de componentes electrónicos de montaje superficial (SMD, *Surface Mount Device*) en un circuito impreso. Sin embargo, se podría adaptar con facilidad para otros fines simplemente cambiando el efector final.

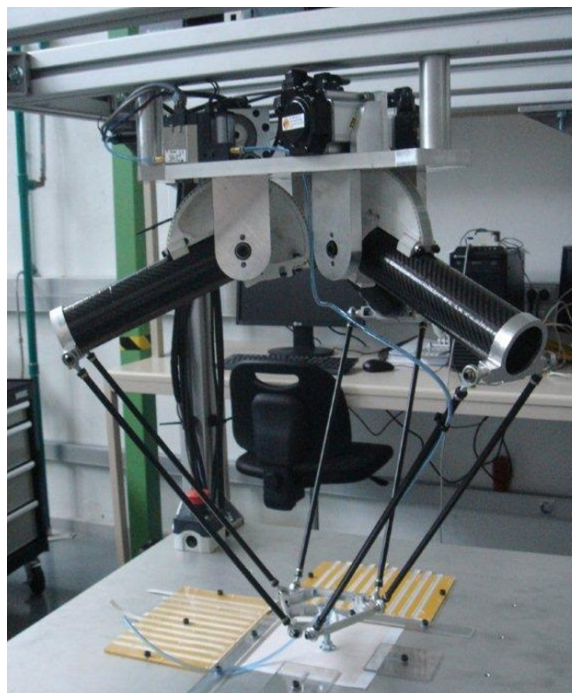


Figura 4: Robot delta de ensamblado SMD

La máquina cuenta con 4 servomotores de 200 W con *encoders* de resolución 1.000.000 (un millón) pasos/vuelta. Los *drives* son cuatro R88-KN02H-ECT de la marca OMRON.

El efector final es una boquilla de aspiración co-

nectada a un bloque de inyección de vacío de SMC (ZX1101-K15L0Z-E55CN-Q), que se encarga de obtener los componentes SMD y retenerlos hasta que se realice el movimiento a la posición final, donde se irán colocando según su disposición en la PCB.

Lo primero que se realizó fue la retirada del controlador NJ5 y las tarjetas de entradas/salidas adjuntas al mismo así como la conexión directa del cable Ethernet al PC que actúa como maestro del Bus.

Para comprobar la conectividad de los *drives* con el pc y su correcto funcionamiento mediante EtherCAT se usó el software de Beckhoff *TwinCAT 2*. Para ello se tuvo que incluir el archivo *xml* de descripción de los *drives*, obtenido en la página web del fabricante, en la base de datos de *TwinCAT*. Este paso permite la experimentación con conceptos propios del protocolo de manera muy sencilla y visual antes de pasar a su configuración en LinuxCNC.

La empresa Ingenieurgesellschaft IgH es la desarrolladora de Etherlab, una serie de programas enfocados a la creación y gestión de aplicaciones que usen la comunicación a través de EtherCAT en tiempo real. Entre ellos se encuentra «EtherCAT master for linux[3]» que incluye el driver para el S.O. y las funciones que el gestor de red necesita, por lo que se ha de instalar para poder dotar a la tarjeta de red Ethernet de la capacidad de transmisión y recepción de paquetes EtherCAT. Este driver no es compatible con todas las tarjetas Ethernet. Se deberá comprobar previamente si la que se va a utilizar lo es. En concreto estos son los modelos compatibles: 8139too , e100 , e1000 , e1000e , r8169.

Por último, en cuanto a conectividad, se precisa de un componente HAL que interactúe con este driver y permita la disponibilidad de los datos en tiempo real en LinuxCNC. Hay varias versiones que se pueden descargar, muchas de las cuales incluyen configuraciones por defecto para terminales Beckhoff como [13]. Sin embargo, son componentes creados por usuarios y no están totalmente revisados por lo que es posible que surjan pequeños problemas, especialmente con su uso en configuraciones distintas a las del programador original, que se deben subsanar. Uno de los problemas más comunes que se han encontrado son las distintas versiones de firmware de los terminales Beckhoff que hacen que algunos de los ajustes por defecto no funcionen.

El driver de IgH precisa de una descripción de la red que defina los elementos EtherCAT conectados (figura 5). Se tendrá que crear un archivo

*xml* tomando como referencia los archivos de descripción tal y como los distribuyen los fabricantes. Este archivo debe incorporar entre otra la siguiente información: tiempo de ciclo, fabricante de cada esclavo, modelo de cada esclavo, datos que se transmitirán en tiempo real (PDO), etc.

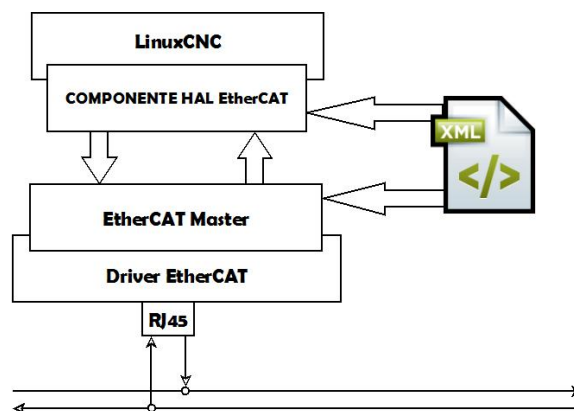


Figura 5: Configuración red EtherCAT

Los datos que se transmiten en tiempo real han de estar definidos antes de poder pasar al estado operativo para que maestro y esclavo sepan que datos han de transmitir y recibir en todo momento.

En el caso de los servos de OMRON utilizados[9] se cuenta con una serie de agrupamientos predefinidos de datos para transmitirse en tiempo real (*Fixed PDO*). Para nuestro caso se ha seleccionado el segundo grupo que permite entre otras cosas el control en posición y el cambio de modo para poder realizar la calibración inicial de la posición.

En cuanto a la situación del motor el driver se comporta como una máquina de estados finitos (figura 6). Estos estados se relacionan con el tipo de comunicación que tiene lugar. Por ejemplo para encender los motores es necesario que las comunicaciones EtherCAT estén en estado operativo (OP) y no haya errores de sincronismo.

Los índices que corresponden al control de estado y al estado actual del servodrive según el estándar CiA 402 son 0x6040 y 0x6041.

En el componente HAL de EtherCAT que se ha de cargar estos índices serán indexados como 16 bits (tal y como se especificó en el archivo *xml* de descripción de red), los cuales al enlazarse mediante señales podrán controlarse o mostrarse por pantalla de manera sencilla. Por facilidad para el usuario se ha creado una pequeña función que codifica el nombre del estado actual de la máquina según los valores de los bits siguiendo la descripción del estándar. En la figura 7 se muestra una posible salida gráfica.

Es posible realizar la configuración de los servos

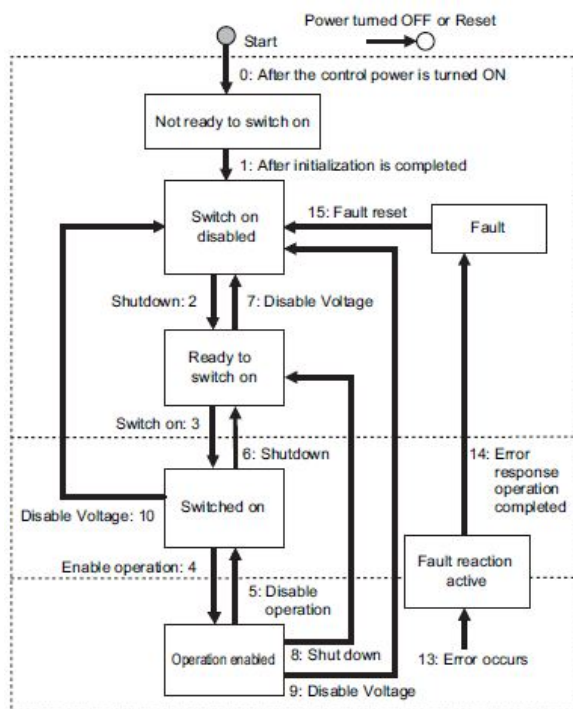


Figura 6: Diagrama de estados CiA 402



Figura 7: Panel de visualización de estados

usando directamente EtherCAT parametrizando los esclavos uno a uno mediante ordenes textuales. Sin embargo es mucho más sencillo hacerlo utilizando la aplicación de OMRON *CX-Drive* que cuenta con utilidades diseñadas específicamente para ello. La conexión se realizó mediante el puerto mini-usb de los servos y el puerto usb del ordenador. Esto no quiere decir que no sea posible configurar todos estos parámetros mediante EtherCAT, lo que habría que hacer dando valores uno a uno vía texto, si no que la manera gráfica y automatizada del software del fabricante es más sencilla.

Dado que el robot a controlar es de tipo paralelo es imprescindible disponer del modelo cinemático. Este modelo debe ser programado y cargado en la HAL tal y como describe el manual de LinuxCNC[7]

Además de los servos se han incorporado otros esclavos de otro fabricante, como muestra de la versatilidad del protocolo, que se han usado para incorporar un *joystick* que permite al usuario final el movimiento manual en los ejes X e Y, así como un par de botones que gobiernan el eje Z.

El *joystick* es un serie I de la marca PML y se compone, en esencia, de dos potenciómetros accionados con la palanca. Se alimenta con una tensión de 10 V en continua y se miden las tensiones de salida de cada potenciómetro mediante una tarjeta Beckhoff de entradas analógicas  $\pm 10$  V diferencial (EL3102).

Los dos botones que hacen incrementar o disminuir la posición Z del efector final se alimentan con 24 V en continua y se conectan con una tarjeta de entradas digitales Beckhoff (EL1014).

Con todo esto se obtiene un controlador en tiempo real que gobierna el robot delta en modo automático, siguiendo la trayectoria definida mediante un archivo descrito en código G, y un modo manual mediante el *joystick*, la botonera y una interfaz gráfica software.

La precisión, repetibilidad y usabilidad conseguidas son en cualquier caso, como mínimo, equivalentes a los obtenidos antes de los cambios realizados. Además, tras sustituir el controlador propietario por uno libre las posibilidades de mejora, ampliación y modificación de la máquina son infinitas. La facilidad de uso ha mejorado también enormemente dado que no hay que restringirse al código de programación de OMRON.

## 5. Guía de instalación e inicio

Con la siguiente guía se pretende orientar, al menos en las etapas iniciales, a docentes y estudiantes para que sean capaces de hallar e instalar todos los elementos necesarios, para hacer funcionar su máquina mediante EtherCAT y el programa LinuxCNC.

Se necesitaran:

1. Linux 10.04 con kernel RTAI
2. IgH EtherCAT Master
3. LinuxCNC
4. Componente HAL EtherCAT
5. Archivos de descripción de cada elemento conectado en la red.

### 5.1. Linux

Aunque es posible la instalación de LinuxCNC sobre un sistema operativo Linux que cuente con

un módulo de tiempo real es preferible instalar la versión de Linux con LinuxCNC preinstalado que facilitan en su página oficial[8].

Para su instalación se recomienda hacer uso de una memoria USB para lo cual se debe utilizar algún software que la convierta en una unidad desde la que se pueda arrancar el ordenador. Por su sencillez se sugiere usar Universal-USB-Installer[12].

## 5.2. IgH EtherCAT master

Se descargará tanto EtherCAT-master v1.5.2 desde la página de IgH[3] como el directorio «debian»[1] que permitirá empaquetar el driver en un paquete *.deb* e instalarlo como tal.

Se descomprime el primer archivo descargado:

```
# tar xvfj ethercat-1.5.2.tar.bz2
```

Se copia el directorio «debian» dentro del directorio recién creado y se empaqueta el driver:

```
# cd ethercat-1.5.2
# cd debian
# ./configure -a
# cd ..
# dpkg-checkbuilddeps
# dpkg-buildpackage
```

Si tras ejecutar la comprobación de dependencias falta algún paquete necesario habrá de instalarse de la manera habitual en Ubuntu:

```
# sudo apt-get install paquete-que-falta
```

Y volver a ejecutar la comprobación hasta que todo este correcto.

Si el empaquetado ha acabado correctamente se han obtenido dos archivos: ethercat.1.5.2-1.i386.deb, ethercat-dev.1.5.2-1.i386.deb

Se instala primero ethercat.1.5.2-1.i386.deb:

```
# sudo dpkg -i ethercat.1.5.2-1.i386.deb
```

Cuando acabe la instalación se debe modificar un archivo de configuración para incluir la MAC y el driver de la tarjeta Ethernet que se va a usar:

```
# sudo edit /etc/default/ethercat
```

Para conseguir ambos datos se puede ejecutar:

```
# lshw -c networks
```

Se guardan los cambios en el archivo de configuración y se actualiza el driver.

```
# sudo update-ethercat-config
```

Por último instalamos el paquete para desarrolladores

```
# sudo dpkg -i ethercat-dev.1.5.2-1.i386.deb
```

Y reiniciamos el ordenador para que arranque usando los drivers correctos.

En este momento si se ejecuta en una terminal

```
# ethercat master
```

se podrá ver información del maestro EtherCAT. Si se tiene conectado algún esclavo, se pueden enumerar ejecutando:

```
# ethercat slaves
```

## 5.3. LinuxCNC y componentes HAL

El siguiente paso es instalar una versión de LinuxCNC que incluya el componente HAL para EtherCAT. Se podría bajar cualquier versión e incluir el componente nosotros mismos modificando cuantos archivos fueran necesarios para satisfacer todas las referencias. Sin embargo, de nuevo por sencillez, se utilizará una versión ya modificada por un usuario.

Se descarga desde el repositorio[14] donde esta alojado. Se descomprime el *.zip* descargado y se instala siguiendo los siguientes comandos:

```
# cd src
# ./autogen.sh (Crea el programa de configuración)
# ./configure
```

Si tras ejecutar el configurador falta alguna librería debe instalarse como en el caso anterior (subsección 5.2) y volver a ejecutarlo hasta que todas las comprobaciones sean correctas.

```
# make clean
# make
# sudo make setuid
```

Dado que se ha instalado una versión portable (*Run In Place*) de LinuxCNC habrá que ejecutar el *script rip-environment* que configura las rutas de la versión en la terminal desde la cuál se vaya a iniciarlo.

```
# . scripts/rip-environment
```

## 5.4. Descripción de la red

Se tiene que configurar el master indicándole que esclavos se va a encontrar conectados en la red y que información se va a compartir con cada uno de ellos en tiempo real. Para ello hay que crear un archivo de etiquetas (formato *.xml*) que interpretará el configurador (figura 5).

En el siguiente enlace[11] se pueden encontrar varias configuraciones que servirán de guía para la creación de cualquier otra.

Los datos necesarios para la creación del archivo se obtienen principalmente del archivo de descripción del esclavo que debe facilitar cada fabricante. Se puede utilizar el software TwinCAT de Beckhoff para que interprete el archivo, que en la mayoría de los casos es bastante confuso, y tomar los datos de su visualización gráfica.

Además será en este archivo donde se describan las actuaciones que hay que realizar en la transición al estado operativo (OP) o de tiempo real. En el caso de un servo, por ejemplo, habrá de definirse el modo de operación para que desde el inicio sepa cual es la consigna que debe tener en cuenta (posición, velocidad...).

### 5.5. Componente HAL

Los comandos para configurar el driver y cargar el componente HAL son

```
# loadusr -W lcec.conf ruta-archivo-xml
# loadrt lcec
```

Se puede comprobar el funcionamiento de los mismos desde el *halrun* (terminal de la HAL). Para ello se abre *halrun* y se introducen ambos comandos. Si se ejecuta *show*, deberán aparecer todos los pines de las variables que se declararon en el archivo *.xml*

## 6. Conclusiones

Se espera que la lectura de estas líneas realmente contribuya a superar las dificultades que presenta la instalación, configuración y puesta en marcha del driver EtherCAT con LinuxCNC para el control de una máquina de control numérico no convencional.

Con el caso práctico presentado se ha demostrado que los beneficios de una modernización en los buses y protocolos tradicionales a los modernos protocolos sobre Ethernet, en especial EtherCAT justifican considerablemente la inversión tanto económica como en investigación. Es por ello que día a día crece la expansión de estas tecnologías.

Por otro lado con la guía de instalación se consigue eliminar cualquier barrera que pueda suponer para docentes y estudiantes el primer acercamiento a estas técnicas.

### Agradecimientos

Antonio Pérez, Andrea Maidana y Fernando Sabarit diseñaron y construyeron el robot delta sobre el que se ha trabajado en el caso práctico de este artículo.

OMRON facilitó gran parte de los componentes utilizados en la construcción del manipulador.

Los colaboradores del proyecto LinuxCNC, especialmente Sasha Ittner y Andreas Schiffler han hecho posible el uso de EtherCAT con este sistema operativo.

## Referencias

- [1] Archivos necesarios para empaquetar driver EtherCAT. [En línea] <https://github.com/narogon/linuxcnc/tree/add-hal-ethercat/src/hal/drivers/ethercat/master/debian>. Consultado: 01/06/2014
- [2] “::: EPSG - Homepage :::”, Sitio oficial Ethernet POWERLINK Standardization Group. [En línea]. <http://www.ethernet-powerlink.org/>. Consultado: 01/06/2014
- [3] EtherCAT Master v1.5.2. Ingenieurgesellschaft IgH. [En línea]. <http://etherlab.org/en/ethercat/index.php>. Consultado: 01/06/2014
- [4] Foro oficial LinuxCNC. [En línea]. <http://linuxcnc.org/index.php/english/forum/index>. Consultado: 01/06/2014
- [5] IEC 61800-7-201, IEC 61800-7-301. CiA 402: CANopen device profile for drives and motion control. 2007.
- [6] ISO/TR 6983/2009. Automation systems and integration — Numerical control of machines — Program format and definition of address words. Genova, Suiza. 2009
- [7] LinuxCNC team. LinuxCNC user’s manual v2.5. [En línea] 2014. <http://www.linuxcnc.org>.
- [8] linuxcnc.org, Autores varios. [En línea]. <http://www.linuxcnc.org>. Consultado 01/06/2014.
- [9] OMRON. G5-series WITH BUILT-IN EtherCAT COMMUNICATIONS User’s Manual. 2011. I576-E1-02.
- [10] Pose, Florian. IgH EtherCAT master 1.5.2 Documentation. Essen : s.n., 2013.
- [11] Repositorio de descripciones de redes EtherCAT. [En línea]. <https://github.com/narogon/linuxcnc/tree/add-hal-ethercat/configs/EtherCAT>. Consultado: 01/06/2014



- [12] Universal usb installer v1.9.4.5 [En línea] 2014. <http://www.pendrivelinux.com/universal-usb-installer-easy-as-1-2-3/>. Consultado: 01/06/2014
- [13] Versión LinuxCNC con componente HAL EtherCAT incluido. Sascha Ittner. [En línea]. <https://github.com/sittner/linuxcnc/tree/add-hal-ethercat>. Consultado: 01/06/2014
- [14] Versión LinuxCNC con componente HAL EtherCAT incluido. [En línea]. <https://github.com/narogon/linuxcnc/tree/add-hal-ethercat>