

Implementación de un Sistema de Control para una Máquina de Control Numérico Computarizado (CNC) sobre un Sistema Embebido utilizando Herramientas de Software Libre

Carlos Ronquillo C. ⁽¹⁾, M.Sc. Jorge Magallanes B. ⁽²⁾
Facultad de Ingeniería en Electricidad y Computación (FIEC) ⁽¹⁾
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil-Ecuador
caronqui@espol.edu.ec ⁽¹⁾

Escuela Superior Politécnica del Litoral (ESPOL)
Máster en Investigación en Informática, Director del Proyecto de Graduación ⁽²⁾
jmagalla@espol.edu.ec ⁽²⁾

Resumen

El presente proyecto presenta la implementación de un Sistema de Control para una máquina de Control Numérico Computarizado utilizando únicamente software libre e implementándolo en un sistema embebido. Como prototipo experimental para las pruebas se utilizó una impresora 3D en reemplazo de maquinaria CNC como fresadoras y tornos, debido a las similitudes en su estructura de ejes y motores. El proyecto abordó dos de las etapas que conforman el proceso de manufactura CNC. Primero, utilizando una herramienta CAM (Computer-Aided Manufacturing) se generó del código que detalla la trayectoria que deberá seguir la herramienta de la máquina, todo en base a un diseño 3D CAD (Computer-Aided Design). La segunda etapa consiste en la ejecución del código utilizando un Software de Control CNC para generar las señales transmitidas desde el Sistema Embebido a los motores de la máquina vía puerto paralelo. También, se diseñó un controlador especializado para aprovechar el uso de pines de propósito general (GPIO) como alternativa al puerto paralelo. Se realizaron varias pruebas para asegurarse de la correcta configuración de la máquina, entre ellas el monitoreo de las señales de salida del puerto paralelo, la ejecución de código, además de pruebas individuales a los motores de la máquina.

Palabras Claves: control numérico computarizado, CNC, sistema embebido, software libre.

Abstract

This paper details the implementation of a Control System for a Computer Numerical Control machine using only free software and implementing it in an embedded system. As an experimental prototype for testing, a 3D printer has been used in replacement of CNC milling machines and lathes, due to similarities in structure and motors used. This project addressed two stages in the CNC manufacturing process. First, a CAM tool (Computer-Aided Manufacturing) is used to generate a code detailing the toolpath to be followed by the machine, all based on a 3D CAD model (Computer-Aided Design). The second stage consists in interpreting the code using a CNC Control Software to generate the signals to be transmitted from the Embedded System engines to the machine through parallel port. A custom driver was also designed to take advantage of general-purpose pins (GPIO) as an alternative to the parallel port driver. Several tests were performed to ensure the exact configuration of the machine, including the monitoring of the output signals through the parallel port, G-code execution, and individual test to each stepper motor.

Keywords: computer numerical control, CNC, embedded system, free software.

1. Introducción

En la actualidad países que basan su economía en recursos naturales como Ecuador pueden garantizar una buena calidad de vida a sus habitantes, sin embargo estos recursos no durarán para siempre, por eso es

necesario establecer una economía basada en la Industria, ofreciendo una gran cantidad de bienes y servicios que otros países requieran comprar.

Una forma de lograrlo es mejorando la eficiencia y calidad de nuestros métodos de manufactura, por

ejemplo la automatización de maquinaria usando Control Numérico Computarizado (CNC), este proyecto busca demostrar que se puede crear un Sistema de Control de bajo costo con el que cualquier empresario en el área de manufactura pueda mejorar la calidad de su producción y competir con otros mercados.

2. Antecedentes

En esta sección se detalla la justificación al realizar este proyecto y los objetivos generales y específicos.

2.1. Justificación

En Ecuador la producción nacional de maquinaria especializada para manufactura no es muy competitiva a nivel mundial, esto ha llevado a la Industria Manufacturera a buscar alternativas que le permitan mejorar la calidad de sus productos, como la importación de maquinaria especializada en automatización industrial o contratar terceros para que se encarguen de la comercialización y capacitación de la tecnología. Por otro lado, se encuentran las pequeñas y medianas empresas que no cuentan con los recursos económicos para invertir e importar esta maquinaria.

Esta propuesta busca apoyar a la producción nacional y a estas empresas, de manera que, se desarrolle localmente a bajo costo una maquinaria de control numérico computarizado (CNC), especializada para la manufactura en metal y madera que les permita la automatización de sus líneas de producción.

La inversión e implementación de esta tecnología por parte de empresas nacionales en la Industria Manufacturera para automatizar su maquinaria les permitirá alcanzar un nivel más alto de competitividad, mejorando la relación calidad-precio de su producción al poder obtener acabados más finos y detallados con gran precisión y disminuyendo tiempos y costos.

2.2. Objetivo General

Implementar un Sistema de Control de una Máquina CNC en un Sistema Embebido utilizando herramientas de Software Libre, que permita a la Industria Manufacturera Ecuatoriana mejorar su maquinaria y aumentar la calidad de la producción nacional.

2.3. Objetivos Específicos

- Implementar el Sistema en un hardware embebido con Sistema Operativo Linux y el Software de Control de Maquinaria LinuxCNC.
- Establecer la configuración adecuada del Sistema que permita adaptar los diferentes tipos de maquinaria CNC de acuerdo a sus características.

- Diseñar e implementar un controlador especializado que aproveche las características del sistema embebido, adaptándolo para que ofrezca más opciones de control y operación al Sistema.

3. Marco Teórico

3.1. Control Numérico Computarizado (CNC)

El Control Numérico Computarizado describe la automatización de Máquinas-Herramientas, es decir, una computadora controla los movimientos de los ejes de una máquina mediante un código el cual contiene instrucciones secuenciales de los movimientos de las herramientas, giros, las velocidades de corte, puntos de inicio y fin de las operaciones. Este código permite crear piezas y partes tallándolas a partir de un bloque de material cada vez que es ejecutado [1].

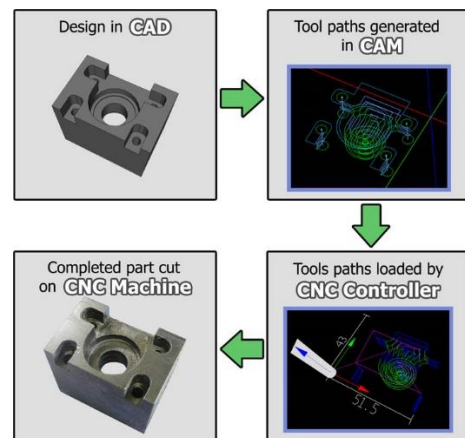


Figura 1. Proceso de Manufactura CNC. [2]

El proceso de diseño a proyecto finalizado en CNC puede resumirse como se aprecia en la Figura 1, todo empieza con la idea plasmada en un diseño mediante una Herramienta CAD (Computer-Aided Design), el cual contiene todos los detalles de su geometría y dimensiones, luego este diseño es convertido en datos para controlar a la máquina mediante una herramienta CAM (Computer-Aided Manufacturing), en donde se definirán las estrategias a seguir para cortar el material y los tipos de herramientas necesarias para obtener la pieza final, todo esto es almacenado en un archivo de código-G, la siguiente etapa interviene el Software de Control CNC, este interpretará los comandos del código-G y los transmitirá en forma de señales que la máquina-herramienta pueda entender para el movimiento de sus motores y herramientas, y así se finaliza con el producto o pieza terminada. [2] [3]

3.2. Máquinas CNC

Las máquinas CNC surgieron como medida para superar las limitaciones que se tenían con maquinaria manual, como como piezas con geometrías complejas,

utilizando un computador para calcular las trayectorias y manejar la máquina mediante control numérico se pueden crear piezas con gran velocidad y precisión. Es de notar que cualquier máquina-herramienta convencional como tornos y fresadoras pueden adaptarse para trabajar con CNC.

3.3. Herramientas CAD/CAM

CAD (Computer-Aided Design) es el proceso de diseño de una parte o pieza a elaborar, mediante vectores. El diseño CAD constituye la fase inicial del proceso de manufactura CNC donde la idea o el boceto de una persona son plasmados y representados de forma gráfica a escala real y con todas sus características geométricas [4].

CAM (Computer-Aided Manufacturing) constituye la segunda etapa en el proceso de manufactura CNC, las herramientas CAM básicamente permiten importar diseños vectoriales CAD, y convertirlos en una secuencia de códigos y funciones que puedan ser interpretados por el Sistema CNC, por ejemplo código-G, para que la máquina-herramienta realice las tareas necesarias para elaborar la parte o pieza diseñada [5].

3.4. Software de Control – LinuxCNC

El software de control es el encargado de recibir los diseños, representados en código-G y obtenidos mediante herramientas CAM, procesarlos y convertirlos en señales que serán enviadas a la tarjeta interfaz y luego a los motores que controlaran la máquina CNC.

LinuxCNC es un Software de Control CNC para máquinas-herramientas como fresadoras, tornos y enrutadores. LinuxCNC es un software gratuito de código abierto, soportado por una gran comunidad de desarrolladores. [6]

El manejo de una máquina CNC requiere, de un control preciso y en tiempo real del movimiento y otras tareas, por ello LinuxCNC requiere una plataforma con capacidad de cómputo en tiempo real. Para esto LinuxCNC utiliza un kernel de Linux con la extensión RTAI (Real Time Application Interface).

4. Implementación del Sistema

4.1. Arquitectura del Sistema

El Sistema estará conformado por varios componentes como se muestra en la Figura 2.

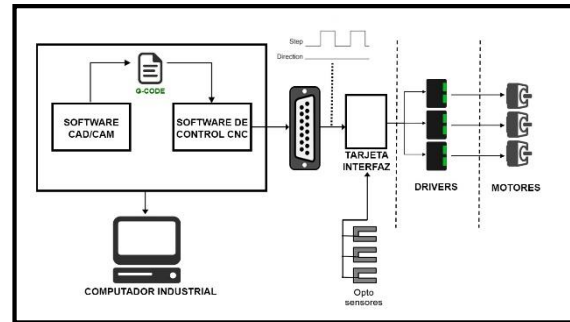


Figura 2. Arquitectura del Sistema – Componentes

Sistema de Control CNC

Implementado en el embebido Advantech PCM-9562 el cual contara con lo siguiente:

- SO Ubuntu con extensión RTAI
- Software CAM: PyCAM
- Software de Control: LinuxCNC
- Puerto Paralelo para manejo de señales

Hardware de Control CNC

- Tarjeta Interfaz
- Controladores de motor genéricos
- Actuadores: Motores de paso
- Máquina CNC: Impresora 3D Prusa Mendel

4.2. Instalación de LinuxCNC

La instalación se realizó utilizando el Live-CD obtenido en la página oficial de LinuxCNC. Esto brinda la ventaja de tener una distribución de Ubuntu automáticamente configurada y lista para usar LinuxCNC.

Para poder desarrollar el controlador de GPIO es necesario instalar LinuxCNC desde su código fuente en el repositorio Git:

```
sudo apt-get install libpthreads-dev
sudo apt-get build-dep linuxcnc
git clone git://git.linuxcnc.org/git/linuxcnc.git
linuxcnc-dev
```

De esta manera obtendremos la versión más reciente del código de LinuxCNC.

4.3. Creación de la configuración de la máquina

Para la correcta operación de la máquina se debe crear un archivo de configuración adecuado lo que se puede realizar fácilmente mediante el asistente de configuración de LinuxCNC.

Información básica de la máquina. Definiremos la configuración de ejes, las unidades de medida, los valores de timing de los controladores de motor de paso y la latencia.

Configuración del puerto paralelo. Aquí seleccionaremos las señales de salida (dirección y paso de los motores) y entrada (botones, sensores, switches) para los pines del puerto paralelo.

Configuración de ejes. Tendremos que definir las características de cada uno de los ejes de la máquina, entre ellas están las siguientes:

- Numero de pasos por revolución.
- Micropasos soportado por el controlador.
- Distancia de avance por revolución.
- Velocidad y aceleración máxima del motor.
- Límites del eje.

4.4. Generación del Toolpath

Se utilizó la herramienta PyCAM para la generación de las trayectorias que seguirá la máquina. PyCAM utiliza archivos CAD 3D y diseños 2D, para la creación de toolpaths y exportarlos como código-G.

El proceso de creación de toolpaths consiste en:

- Definición de las herramientas: Tendremos que definir las características de las herramientas que se usaran en el proceso: forma, diámetro, velocidad de rotación y avance.
- Definición de los procesos: Consiste en las estrategias que se usaran para el tallado, por ejemplo, desbaste, contorneado, superficial, grabado.
- Definición de áreas de límites: indicando el área en la que operara la herramienta sin exceder los límites.

Las trayectorias se generaran combinando todas nuestras especificaciones en una tarea de tallado como se muestra en la Tabla 1.

Tabla 1. Definición de Tareas de Tallado

Tarea	Herramienta	Proceso	Límites
Desbaste	Cilíndrica (5mm)	Remover Material	Margen 10%
Acabado	Esférica (3mm)	Acabado	1mm

Una vez generado el toolpath podremos previsualizarlo y realizar una simulación para ver los resultados (ver Figura 3) y luego exportarla a código-G.

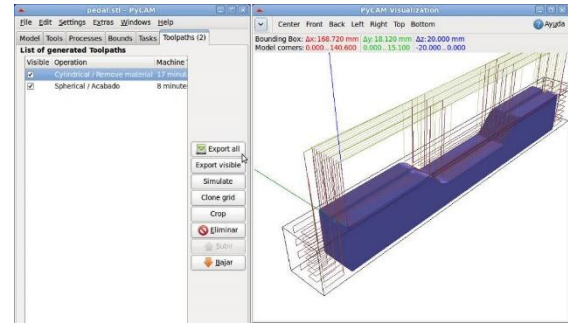


Figura 3. Toolpaths y su Simulación en PyCAM

4.5. Diseño del controlador GPIO

En la Figura 4 se muestra el diseño de la componente HAL, la capa de abstracción de hardware usada en LinuxCNC nos da la ventaja de poder comunicarnos con dispositivos de hardware más fácilmente mediante una representación abstracta del dispositivo, el controlador está compuesto de una estructura de datos llamada *viaggio_t* la cual va a estar asociada y almacenara los datos de entrada y salida de los pines de nuestra componente *hal_viaggio*, también se necesitan definir las funciones de lectura y escritura del puerto mediante funciones de bajo nivel para acceder a los registros de memoria de nuestros pines físicos, deberán ser implementadas de acuerdo a las indicaciones para acceder a los GPIO de nuestro embebido.

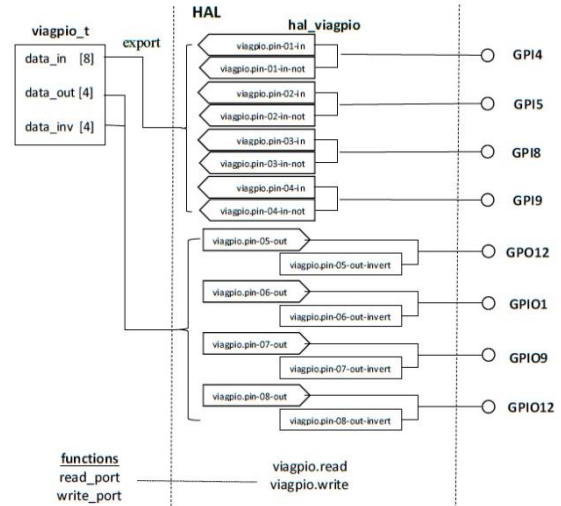


Figura 4. Diseño de la componente hal_viaggio

5. Resultados

5.1. Prueba de Señales de Salida Puerto Paralelo

Se realizó la prueba de las señales de salida del puerto paralelo con ayuda de un osciloscopio y utilizando LinuxCNC y los botones que permiten mover cada eje:

También se utilizó el control de velocidad en la interfaz de LinuxCNC para observar los cambios en la frecuencia de los pulsos al variar este parámetro.

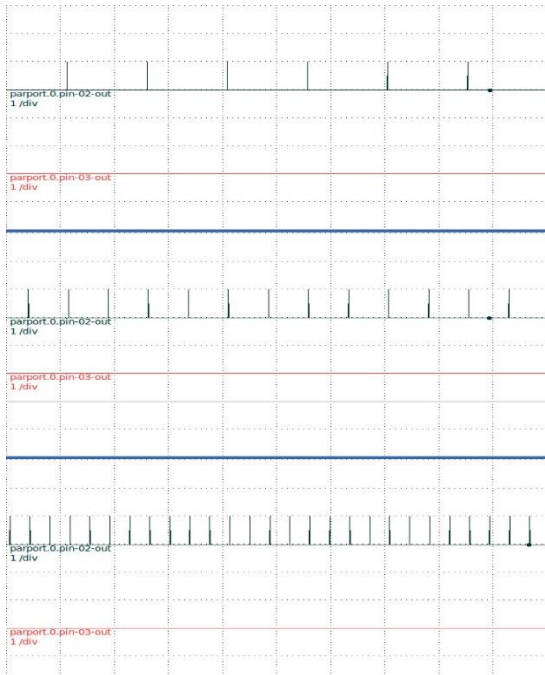


Figura 5. Señales de Salida a 50, 100, y 200 mm/s

En la Figura 5 se muestran las señales de salida de los pines 02 y 03 correspondientes a la señal de dirección y la de pasos del eje X. La primera imagen corresponde a una velocidad de 50mm/s moviéndose a la derecha, la segunda a una velocidad de 100mm/s moviéndose a la izquierda y la última a una velocidad de 200mm/s moviéndose a la derecha.

5.2. Pruebas de los Motores de paso – velocidad y aceleración máxima

Durante la creación del archivo de configuración con Stepconf se necesitaba especificar las velocidades y aceleración máximas de cada motor, ahora utilizaremos la herramienta de prueba (ver Figura 6) que se encontraba en esa sección usando el botón “Test this axis”.

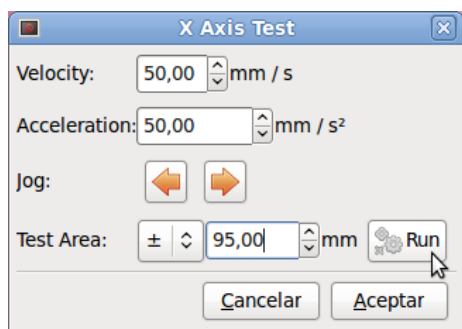


Figura 6. Herramienta de Prueba de Ejes

El procedimiento consiste en ubicar la herramienta (en el caso de la impresora el extrusor) en una posición central al eje de prueba y marcar la posición. Luego definiremos un área de prueba en la que se moverá la herramienta, el área de trabajo de la impresora es 200x200x100, pero usaremos un valor menor por seguridad, para X (± 95), Y (± 95), Z (± 45).

Iniciamos con valores bajos de velocidad y aceleración, al ejecutar el test, la herramienta comenzará a moverse a ambos extremos del eje a la velocidad utilizada, la detenemos y la herramienta debería volver a su posición inicial, si la posición es incorrecta significa que el motor se atascó o perdió pasos y reducimos la velocidad, en caso contrario podemos probar aumentándola. Una vez encontrada la velocidad máxima basados en el margen de error obtenido, la reducimos un 10% por seguridad y la asignamos a la máquina.

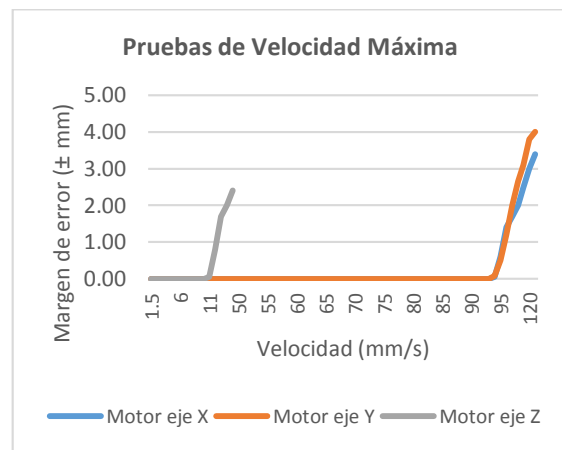


Figura 7. Experimento-Pruebas de velocidad

En la Figura 7a 7, se muestran los resultados del experimento realizado para medir la velocidad máxima, se puede observar comparando los resultados, que en los ejes X, Y, a medida que la velocidad supera alrededor de los 100mm/s los motores empezaban a atascarse y no podían regresar a su posición inicial. El eje Z es algo especial ya que consiste de 2 motores y utilizan un husillo para su movimiento, las velocidades utilizadas son menores considerando que se desplaza aproximadamente 1.4 mm/revolución, se puede observar que empezaba a fallar a medida que la velocidad se acercaba a 15mm/s.

Considerando estos resultados, las velocidades máximas para cada eje serían las siguientes:

Eje X = 90mm/s Eje Y = 90mm/s Eje Z = 12mm/s

Ajustándolas al valor de velocidad recomendado obtendríamos:

Eje X = 81mm/s Eje Y = 81mm/s Eje Z = 10mm/s

Una vez obtenida la velocidad máxima, es turno de la aceleración mediante el mismo procedimiento, pero esta vez ya conociendo la velocidad máxima esta vez empezamos a variar la aceleración.

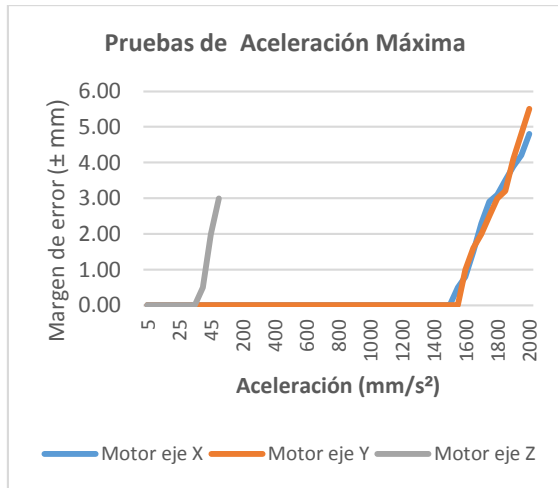


Figura 8. Experimento - Pruebas de aceleración

En la Figura 8 se muestran los resultados del experimento para encontrar la aceleración máxima, en el eje X, Y se observa que ambos ejes comenzaban a tener errores a medida que se acercaban a los 1600 mm/s², en el eje Z se manejaban aceleraciones menores, se observó que a medida que se acercaba a los 45 mm/s² empezaban a tener fallos.

En base a las pruebas se determinaron las aceleraciones máximas:

Eje X = 1500mm/s²
 Eje Y = 1500mm/s²
 Eje Z = 40mm/s²

Ajustando las aceleraciones a su valor recomendado resultaría en:

Eje X = 1350mm/s²
 Eje Y = 1350mm/s²
 Eje Z = 35mm/s²

5.3. Prueba del Sistema ejecutando Código G

Con la velocidad y aceleración de cada eje correctamente definida procedemos a realizar las pruebas del sistema CNC junto a la impresora ejecutando el código-G en LinuxCNC.

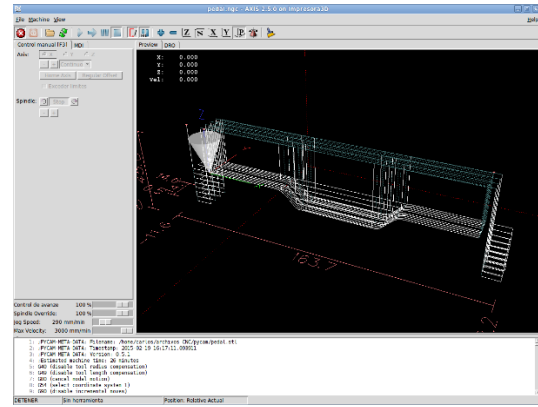


Figura 9. LinuxCNC con el código-G generado

La Figura 9 muestra la pre-visualización del código-G en LinuxCNC, la ejecución se realizó de forma correcta, no hubo ningún problema en el movimiento de los motores de la impresora.

5.4. Prueba del Controlador GPIO diseñado

Una vez compilado el controlador que creamos y definida la configuración de la máquina, realizamos la prueba de las señales de salida de los pines GPO utilizando un osciloscopio.

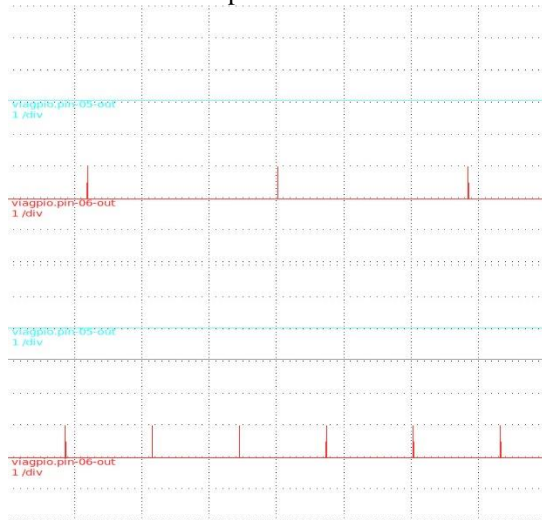


Figura 10 Señales de Salida de los pines GPIO

La Figura 10 muestra las señales de dirección en el pin viaggpio.pin-05-out (GPO12) y la señal de paso del eje X en el pin viaggpio.pin-06-out (GPIO01). En la primera imagen el movimiento es hacia la derecha (dir=Low) y en la segunda hacia la izquierda (dir=High), ambas con diferentes velocidades.

6. Conclusiones

Las pruebas realizadas y la baja latencia obtenida, demostraron que un Sistema embebido comercial como el que se utilizó, cumple perfectamente el rol de un

Sistema de Control CNC utilizando el software adecuado.

El Sistema Linux junto al software LinuxCNC demostraron ser un excelente medio como Sistema de control CNC, al utilizar un kernel de Linux modificado con la extensión RTAI garantiza que la generación de pulsos sea precisa y a tiempo, por lo que no es necesario de hardware adicional para este propósito como en otros sistemas.

El archivo de configuración de LinuxCNC desarrollado para la impresora 3D permitió que esta trabajara como cualquier maquinaria CNC de 3 ejes, su funcionamiento fue excelente durante las pruebas, esto nos demuestra que con los parámetros adecuados en la configuración se puede operar cualquier máquina-herramienta CNC, e incluso cualquier máquina que utilice motores de paso sin problemas.

El controlador GPIO desarrollado funciono adecuadamente, se obtuvo el mismo resultado comparándolo con el puerto paralelo, esto demuestra que se puede adaptar el Sistema para que aproveche estos pines GPIO como recurso para controlar otros dispositivos, sensores, botones, sin requerir otro puerto paralelo.

7. Recomendaciones

Aquí se detallaran algunas recomendaciones para futuras investigaciones de la misma temática.

- 1) En este proyecto no se cubrió el uso de servo motores por lo que podría ser de utilidad explorar las características y varias ventajas que estos poseen frente a los motores de paso. Otro caso de interés es el uso de máquina-herramientas de rotación como el torno.
- 2) Investigar y aprovechar los beneficios que ofrece el uso de software libre, LinuxCNC ofrece una gran cantidad de alternativas de

configuración, es posible crear una interfaz de usuario personalizada de acuerdo a nuestras necesidades, incluso utilizar joysticks para el movimiento manual de los ejes, además de que LinuxCNC cuenta con una gran comunidad de desarrolladores que aporta continuamente con nuevas funcionalidades.

- 3) Considerar la posibilidad de optimizar o mejorar el funcionamiento del LinuxCNC, se podría crear un método de control utilizando laser o cámaras para mejorar la precisión de la herramienta. Esta es otra de las libertades que ofrece usar software libre.

8. Agradecimientos

Agradezco al Centro de Visión y Robótica de la ESPOL, ya que gracias a ellos tuve la oportunidad de involucrarme en el desarrollo de este proyecto y poder realizar mi trabajo final de graduación.

9. Referencias

- [1] Miller, R. y Miller, M. R., Audel Automated Machines and Toolmaking, Wiley, 2004.
- [2] «The CNC Process» CNC 4 everyone, 15 de Febrero del 2012. [Disponible en línea], <http://www.cnc4everyone.com/beginning-with-cnc/the-cnc-process/>, fecha de consulta Diciembre 2014.
- [3] Bowman, M., CNC Milling in the Workshop, Crowood, 2013.
- [4] Overby, A., CNC Machining Handbook, McGraw-Hill/TAB Electronics, 2010.
- [5] «CAD and CAM» CNC 4 everyone, 25 de Febrero del 2012. [Disponible en línea]. <http://www.cnc4everyone.com/software/cad-and-cam/>, fecha de consulta Diciembre 2014.
- [6] «LinuxCNC» 6 Abril 2014. [Disponible en línea]. <http://www.linuxcnc.org/>, fecha de consulta Enero 2015.