



FACULTAD DE INGENIERIA

Universidad de Buenos Aires

MAESTRÍA EN SISTEMAS EMBEBIDOS

MEMORIA DEL TRABAJO FINAL

Lectura de fiduciales para máquina CNC

Autor:
Esp. Ing. Pablo Slavkin

Director:
MEE. Ing. Norberto M. Lerendegui (IEEE)

Jurados:
Ing. Ariel Hernandez
Mg. Ing. Lucio Martinez (CNEA)
Dr. Daniel Minsky (CNEA/CONICET)

*Este trabajo fue realizado en la Ciudad de Bariloche,
entre marzo de 2020 y abril de 2021.*

Resumen

En el presente trabajo se presenta el desarrollo de un sistema electrónico capaz de dotar de visión artificial a una máquina de mecanizado por control numérico computarizado CNC de la fábrica española Wolfcut para permitir el alineamiento automático de piezas 2D mediante la lectura de marcas fiduciales. Utilizando Linux sobre la plataforma PocketBeagle, se desarrollaron los drivers del kernel en C para interactuar con un controlador de movimientos Weihong NK105, se implementó una interfaz de control web con HTML, JavaScript y Python, se habilitó la carga de archivos USB remota mediante la tecnología configFS, y se realizó el procesamiento de vídeo desde una cámara Wi-Fi mediante la biblioteca pyopencv.

Agradecimientos

A Juli, Valen, Maxi y León

Índice general

Resumen	I
1. Introducción general	1
1.1. Mecanizados CNC en 2D	1
1.2. Soluciones de mercado	4
1.3. Motivación y alcance	4
2. Introducción específica	7
2.1. Historia y principio de funcionamiento	7
2.1.1. Programa de movimientos	8
2.1.2. Controlador	10
2.1.3. Máquina	10
2.1.4. Tecnologías utilizadas	10
2.1.5. Plataforma PocketBeagle	13
2.1.6. Aplicación web	14
2.1.7. Cámara de vídeo	17
2.1.8. Trigonometría de alineación	17
2.1.9. Detection y tipos de marcas fiduciales	21
3. Diseño e implementación	23
3.1. Intervencion al NK105	23
3.2. Envio de arvhivos por USB	23
4. Ensayos y Resultados	25
4.1. Pruebas funcionales del hardware	25
4.1.1. Validación de tramas SPI	25
4.1.2. Validación de tramas UART	25
4.1.3. Precisión de centro y ángulo de las marcas	25
4.2. Pruebas funcionales del software	25
4.2.1. Ensayos con diferentes cámaras y resoluciones de video	25
4.2.2. Validación del error de rotación del GCode	25
4.2.3. Validación del error de alineación	25
4.2.4. Ensayo general de caja negra con diferentes archivos de mecanizado	25
5. Conclusiones	27
5.1. Conclusiones generales	27
5.2. Próximos pasos	27
Bibliografía	29

Índice de figuras

1.1. Ejemplos de piezas mecanizadas mediante máquinas CNC.	2
1.2. Esquema de corte de una letra A en una placa de material virgen.	3
1.3. Corte de la silueta de la letra A previamente impresa en el material.	3
1.4. Corte de la silueta de la letra A previamente impresa en el material con lectura de marcas.	4
1.5. Controlador NK105 de la firma Weihong elegido para extender sus funciones y dotarlo de lectura de marcas.	6
2.1. Jhon Parsons junto a una de sus máquinas, considerado el inventor de la máquina de control numérico NC.	7
2.2. Los tres bloques básicos de una máquina CNC.	8
2.3. Secuencia de pasos para operar una máquina CNC primigenia.	8
2.4. Secuencia de pasos para operar una máquina CNC moderna.	9
2.5. La etapa de control se suele separar en dos: controlador lógico y driver de potencia	10
2.6. Varias vistas de una máquina de 3 ejes perpendiculares como las que se estudia en este trabajo.	11
2.7. Maquinas CNC fabricadas por la empresa Wolfcut.	11
2.8. Diagrama de bloques del sistema implementado para con el objeto de identificar las tecnologías involucradas en cada bloque.	14
2.9. Plataforma de desarrollo	14
2.10. Capas de software relacionadas con la aplicación web utilizadas en este trabajo.	15
2.11. Aplicación IPWebcam que permite utilizar la cámara del móvil y enviar el vídeo por Wi-Fi.	18
2.12. Corrección de desplazamiento. A representa el sistema de coordenadas de la máquina y las dimensiones extraídas del archivo de corte, B representa el objeto real desplazado, escalado y rotado respecto del primero	19
2.13. Cálculo de rotación conociendo las coordenadas relativas de dos puntos.	20
2.14. Corrección de la rotación.	21
2.15. Corrección de la escala.	21
2.16. Ejemplo de diferentes tipos de marcas fiduciales.	22
3.1. Mando a distancia del controlador NK105 a) Comunicación por UART del estado de la botonera b) Envío de datos del controlador al LCD por SPI	23
3.2. Aplicación web desarrollada para controlar la máquina, transferir archivos, gestionar la detección de marcas, visualizar el trabajo a cortar en tiempo real, entre otras utilidades de configuración.	24

Índice de tablas

1.1.	Sistemas de reconocimiento de marcas	5
2.1.	Modelos de controladores	12
2.2.	Modelos de drivers	13
2.3.	Selección de la cámara	17

Dedicado a Fernando Sanchez

Capítulo 1

Introducción general

En el presente capítulo se introducen ejemplos de uso de las máquinas CNC y las dificultades que presentan para alinearse con las piezas a mecanizar. Luego se comparan soluciones de otros fabricantes y finalmente se comenta acerca de la motivación, el alcance y los objetivos de la propia.

1.1. Mecanizados CNC en 2D

En la actualidad, muchos de los procesos industriales que involucran el mecanizado de piezas como las que se muestran en la figura 1.1 se realizan utilizando máquinas de control numérico computarizado o CNC [1] (*computer numerical control*).

El proceso de mecanizar piezas utilizando esta tecnología se esquematiza en la figura 1.2 y consiste en una serie de pasos como los que se enumeran a continuación:

1. Posicionar la placa del material a cortar en la mesa de corte.
2. Posicionar la herramienta de corte en un punto de referencia de la placa.
3. Cargar el archivo que contiene la información de corte.
4. Cortar.

Hay casos en los cuales la placa a cortar está previamente impresa y el proceso de corte debe respetar su silueta con exactitud como se esquematiza en la figura 1.3a.

Al no haberse aplicado ninguna corrección ni alineamiento entre el sistema de movimientos de la máquina y la pieza, el software de corte no tiene la información de la posición, rotación y escala exacta de la pieza dispuesta en la mesa.

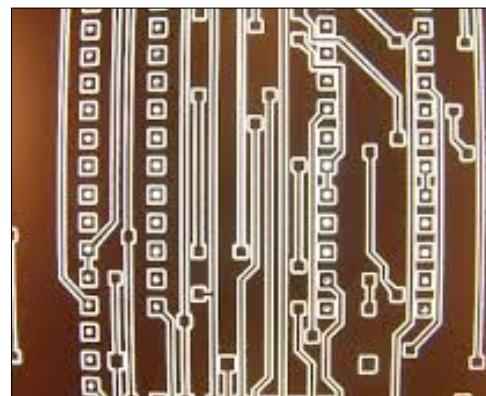
En el ejemplo mostrado en la figura ??, la máquina no puede seguir con exactitud el contorno de la letra A.

En la industria se encuentran muchos casos de uso en donde se presenta este problema, algunos de los cuales se enumeran a continuación:

- Alineación de placas de circuito impreso de 2 caras.
- Alineación de una pieza que requiere un nuevo proceso de mecanizado.
- Alineación luego de abortar un mecanizado debido a un corte de energía.
- Corrección por errores de escala entre diferentes máquinas.



(A) Máquina CNC ejecutando el corte de piezas en madera para mobiliario.



(B) Placa de circuito impreso realizado mediante el fresado del contorno de sus pistas.

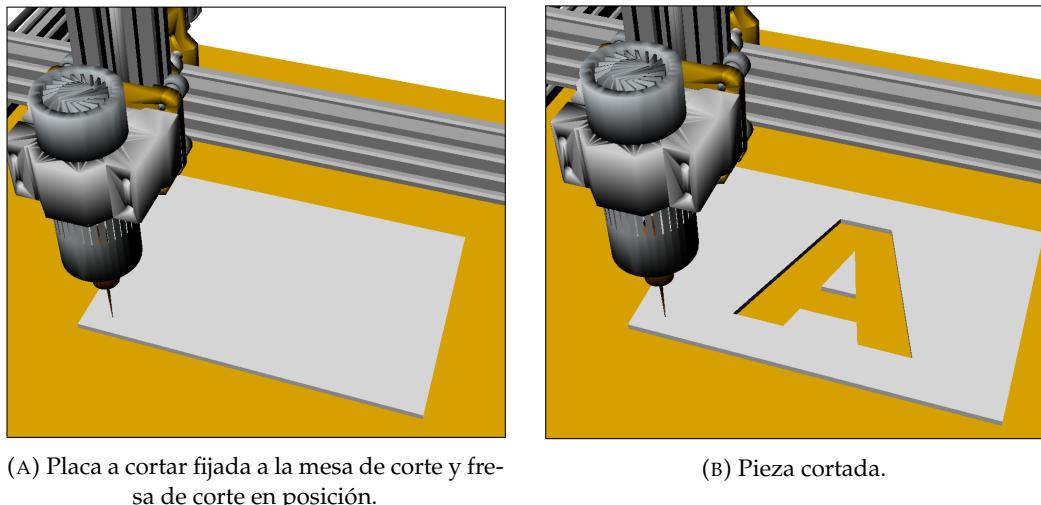


(C) Corte y fresado en placa de aluminio para obtener un repuesto de una máquina herramienta.



(D) Corte de logos y letras en madera.

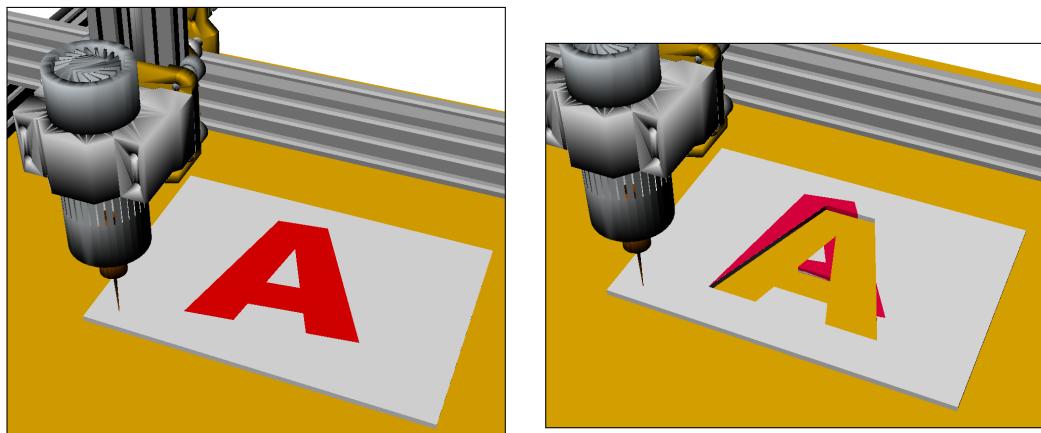
FIGURA 1.1. Ejemplos de piezas mecanizadas mediante máquinas CNC.



(A) Placa a cortar fijada a la mesa de corte y fresa de corte en posición.

(B) Pieza cortada.

FIGURA 1.2. Esquema de corte de una letra A en una placa de material virgen.



(A) Placa a cortar fijada a la mesa de corte y fresa de corte en posición.

(B) Pieza cortada con una notable desalineación entre la silueta previamente impresa y el corte.

FIGURA 1.3. Corte de la silueta de la letra A previamente impresa en el material.

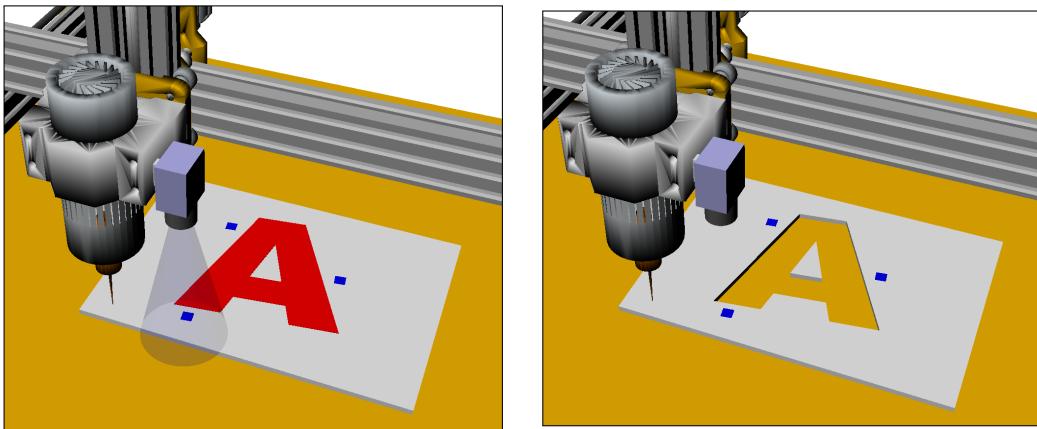
- Corrección por contracción y estiramiento del material debido a variaciones de temperatura.
- Corrección por deformación en piezas elásticas.

En el presente trabajo se aplican técnicas de visión artificial para reconocer los puntos de referencia que permiten corregir esta desalineación.

Estos puntos se incluyen en el proceso de diseño y se imprimen junto con el trabajo a mecanizar.

Una vez reconocidos con una cámara de vídeo montada en la máquina CNC se puede corregir el desplazamiento, el ángulo y la escala del objeto impreso relativos al sistema de movimiento de la máquina.

El resultado esperado se muestra en la figura 1.4.



(A) Placa a cortar impresa con la letra A y tres marcas fijada a la mesa con fresa de corte ya en posición.

(B) Alineación del trabajo de corte según las marcas azules de referencia. La pieza se muestra mecanizada siguiendo el contorno sin errores.

FIGURA 1.4. Corte de la silueta de la letra A previamente impresa en el material con lectura de marcas.

1.2. Soluciones de mercado

En la tabla 1.1 se destacan algunos desarrollos de software que permiten extenderse o adaptarse para soportar el reconocimiento de marcas.

Se puede ver que la mayoría de las soluciones del mercado están basadas en PC y eso aumenta el costo general del sistema, disminuye la fiabilidad y limita el acceso desde múltiples plataformas.

Se ha podido constatar que además del costo de hardware, licencias o extensiones de software, el costo de las cámaras utilizadas son privativos para el segmento de máquinas bajo a medio que es donde mejor aplican los resultados de este trabajo. No se ha encontrado ningún sistema que utilice una cámara con conexión inalámbrica, y tampoco aprovechando el uso de la cámara de un teléfono celular.

No se ha encontrado ninguna solución de código abierto y colaborativo que facilite el crecimiento del proyecto con la ayuda de la extensa comunidad de usuarios, desarrolladores y entusiastas.

1.3. Motivación y alcance

La principal motivación de este trabajo es lograr extender las capacidades de un controlador embebido de uso profesional y dotarlo de visión artificial para el reconocimiento de marcas fiduciales.

Con los argumentos y la experiencia de la empresa Wolfcut, se determinó que uno de los controladores de uso profesional más popular del mercado es el NK105 de la firma Weihong [9] que se muestra en la figura 1.5.

Este controlador solo cuenta con un comando remoto para todas las operaciones de manejo y configuración.

No provee una API definida por el fabricante ni una interfase física para poder conectarse y extender sus funciones.

Por otro lado, al estar basado internamente en una FPGA y no depender de una

TABLA 1.1. Se destacan algunos modelos y marcas de sistemas de reconocimiento de marcas y sus características principales.

Características	Imagen
EddingCNC [2]: Software basado en PC sobre Windows al cual varios fabricantes como GES [3] y Wolfcut [4] lo han extendido para soportar reconocimiento de marcas.	
myCNC [5]: Esta aplicación de la empresa pv-automation [6] ofrece un sistema de visión artificial y reconocimiento de marcas basado en una PC industrial y cámaras USB.	
Machinekit [7]: Es un software de control que opera sobre Linux al cual se le han hecho algunas intervenciones para el reconocimiento de marcas.	
Summa [8]: Es una línea de máquinas de corte de contornos principalmente en papel que cuenta con lectura de marcas integrado en sus sistemas.	

PC para funcionar, es reconocido por sus excelentes resultados de corte, su estabilidad en trabajos extensos y una excelente fiabilidad.

Es un controlador simple, pero de uso profesional con un mercado ya consolidado y muy extenso en todo el mundo. El alcance de este trabajo se limita a intervenir y dotar de lectura de marcas al controlador NK105 y obtener resultados comparables con otras soluciones de mercado.



FIGURA 1.5. Controlador NK105 de la firma Weihong elegido para extender sus funciones y dotarlo de lectura de marcas.

Capítulo 2

Introducción específica

En el presente capítulo se expone una breve reseña histórica de las máquinas CNC, su principio de funcionamiento y su uso en la industria. Luego se introducen las tecnologías mas relevantes involucradas en el desarrollo de este trabajo.

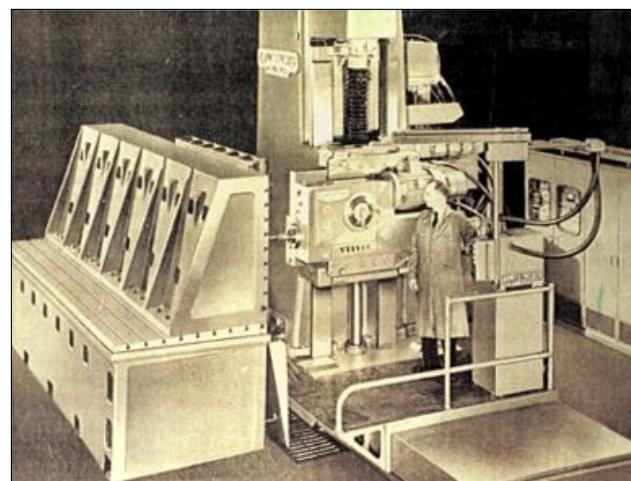
2.1. Historia y principio de funcionamiento

Hacia finales de la década del '40, el mecánico inventor Jhon Parson¹ retratado en la figura 2.1a, logró motorizar una agujereadora de banco de precisión y la automatizó mediante el uso de una cinta perforada, figura 2.1b.

A este invento se lo considera la primera máquina de control numérico o NC por sus siglas en inglés (*numerical control*).



(A) Fotografia de Jhon Parsons.



(B) Una de las primeras maquinas consideradas CNC.

FIGURA 2.1. Jhon Parsons junto a una de sus máquinas, considerado el inventor de la máquina de control numérico NC.

Luego de varias décadas, con el advenimiento de las computadoras, se reemplazaron las cintas perforadas por software, dando lugar a las máquinas de control numérico computarizado o CNC por sus siglas en inglés (*computer numerical control*).

A pesar del paso del tiempo y los avances tecnológicos, las partes principales de una máquina CNC siguen siendo las mismas que se muestran en la figura 2.2.

¹https://en.wikipedia.org/wiki/John_T._Parsons



FIGURA 2.2. Los tres bloques básicos de una máquina CNC.

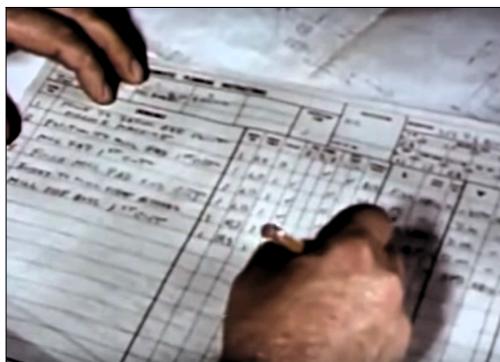
2.1.1. Programa de movimientos

El programa de movimientos consiste en una serie de instrucciones necesarias para obtener una determinada pieza y se escribe en un lenguaje conocido como GCode[10].

Este lenguaje fue creado por el Instituto tecnológico de Massachusetts en la década del '50 y especificado en el documento NIST-RS274-D [11].

Originalmente los ingenieros de mecanizado lo escribían manualmente en una planilla y luego, mediante una máquina de mecanografía, se transcribía a una cinta perforada que sería interpretada por el controlador de movimientos.

Se pueden ver algunas fotos de este primigenio proceso en la figura 2.3.



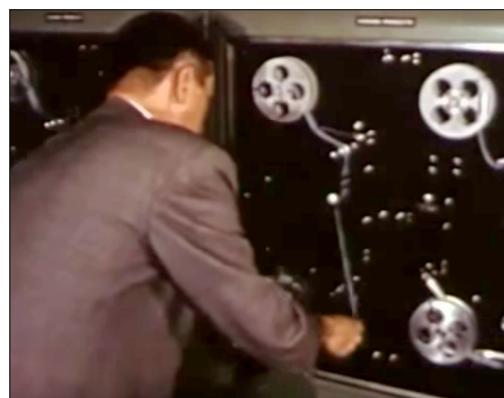
(A) Ingeniero escribiendo en papel la lista de operaciones para mecanizar una pieza en lenguaje GCode.



(B) Operadora transcribiendo la lista de operaciones a una cinta plástica perforada.



(C) Operadora preparando la máquina de procesamiento de cintas perforadas



(D) Lector de cinta multiperforada que controla los movimientos de la máquina.

FIGURA 2.3. Secuencia de pasos para operar una máquina CNC primigenia.

En el presente se diseña la pieza en tres dimensiones o 3D con la ayuda de programas de diseño asistido por computadora CAD (*computer aided design*).

Algunos programas de CAD reconocidos se listan a continuación:

- FreeCAD
- Blender
- Rhinoceros
- AutoCAD
- SolidWorks

La salida del software de CAD se procesa con uno de manufactura asistido por computadora CAM (*computer aided manufacturing*) en donde el diseñador puede configurar los métodos y estrategias de mecanizado.

Algunos programas de CAM reconocidos se listan a continuación:

- RhinoCAM
- Blender CAM
- FreeMILL
- Aspire
- Mastercam

El resultado final es un archivo de texto en lenguaje GCode que se almacena digitalmente y que será procesado por el controlador.

Esta secuencia es conocida como diseño CAD/CAM y se muestra en la figura 2.4

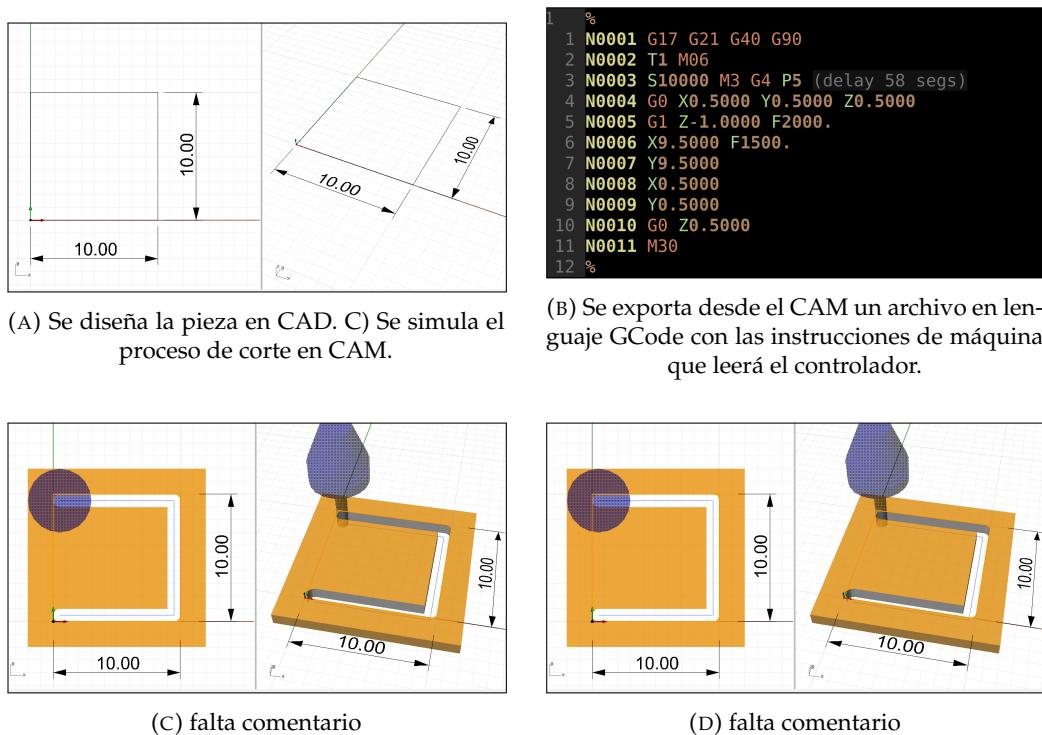


FIGURA 2.4. Secuencia de pasos para operar una máquina CNC moderna.

2.1.2. Controlador

El controlador de movimientos es un equipo electrónico capaz de leer un programa en lenguaje GCode y proveer las señales eléctricas adecuadas para mover la máquina.

Es usual que a la salida del controlador se conecten amplificadores de señal (*drivers*) que provean la potencia suficiente para mover los motores y mecanismos montados en la máquina.

De esta manera el controlador se compone de dos etapas, controlador lógico y drivers como se aprecia en la figura 2.5.



FIGURA 2.5. La etapa de control se suele separar en dos: controlador lógico y driver de potencia

En función de la complejidad requerida para la máquina y de los requisitos de potencia para los movimientos se dimensionan el controlador y los drivers.

En las tablas 2.1 y 2.2 se listan algunos modelos de controladores y drivers comerciales detallando las características principales.

2.1.3. Máquina

En términos generales la máquina es un conjunto de piezas electromecánicas que permiten mover el elemento de mecanizado en varias dimensiones.

En algunas máquinas el elemento de mecanizado permanece fijo y lo que se mueve es la pieza a mecanizar.

Suelen ser motorizadas, pero también las hay con actuadores lineales, sistemas hidráulicos o una combinación de todos estos.

Dependiendo el propósito de la máquina se definen los grados de libertad del movimiento. Es usual utilizar tres ejes perpendiculares para mesas de corte planos, seis ejes para centros de mecanizado de piezas complejas, seis para brazos roboticos y solo dos para corte y grabado de piezas planas con láser.

Para el desarrollo de este trabajo se estudian solamente máquinas de dos y tres ejes perpendiculares, dado que la empresa interesada comercializa principalmente este tipo de estructuras que se esquematiza en la figura 2.6.

Se muestran algunos modelos de máquinas fabricadas por Wolfcut en la figura 2.7.

2.1.4. Tecnologías utilizadas

En el diagrama de bloques de la figura 2.8 se muestran las partes que componen el sistema implementado.

Para poder luego explicar como se relacionan e interactúan entre si estos bloques,

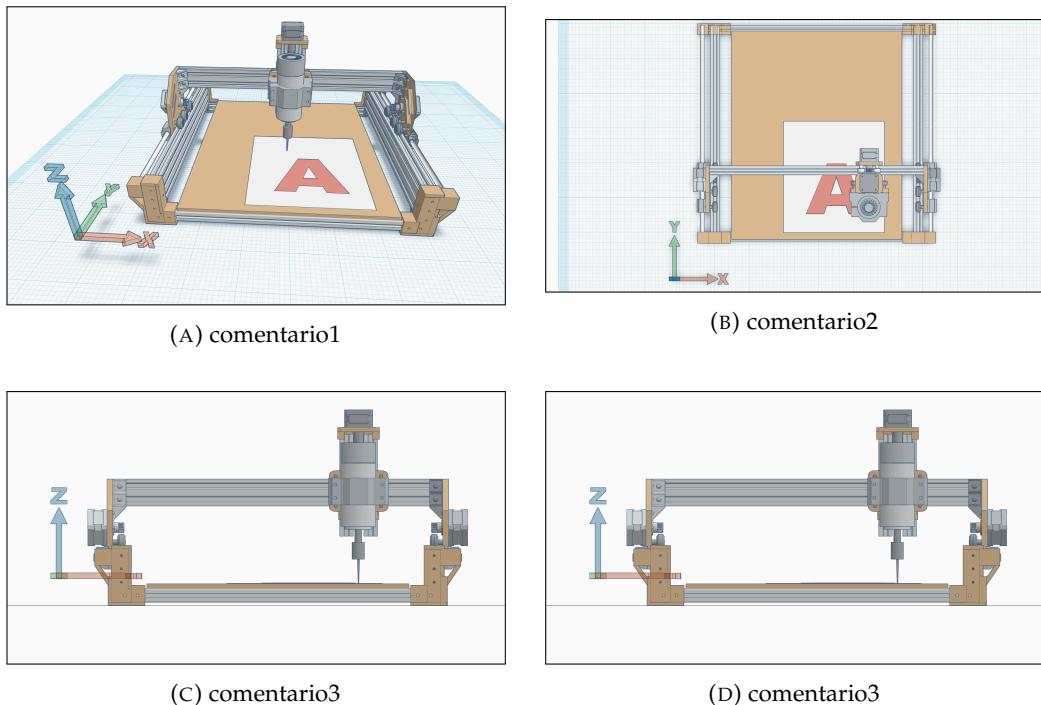
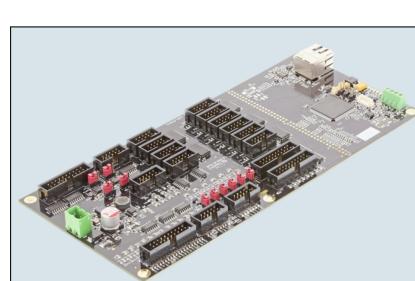


FIGURA 2.6. Varias vistas de una máquina de 3 ejes perpendiculares como las que se estudia en este trabajo.



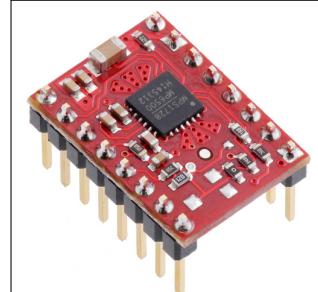
FIGURA 2.7. Maquinas CNC fabricadas por la empresa Wolfcut.

TABLA 2.1. Modelos de controladores CNC disponibles en el mercado

Características	Imagen
<p>Controlador dependiente de una PC y conexión por puerto paralelo. Solución económica para máquinas de baja performance.</p>	
<p>Controlador integrado de media performance, ideal para maquinas profesionales pero de baja complejidad. Este es el controlador que se usará en este trabajo para realizar los ensayos.</p>	
<p>Controlador basado en PC sobre Windows de media performance. Este es el controlador que usa actualmente la empresa Wolfcut en sus máquinas.</p>	
<p>Controlador autónomo profesional de gran performance y opciones de operación. Este tipo de controladores se utilizan en centros de mecanizado de altísima complejidad y prestaciones sumamente exigentes. Es inusual ver este tipo de controladores en las maquinas de gama media debido a su alto costo.</p>	

en las próximas secciones se describen las cualidades y funciones mas destacadas de cada uno.

TABLA 2.2. Modelos de drivers de motores

Características	Imagen
Driver para motores paso a paso pequeños, económicos, ideales para maquinas simples, impresoras 3D, y hobby.	
Driver para motores paso a paso medianos, ideales para máquinas de media precisión y mecánica ligera.	
Driver para motores BLDC (<i>brush less direct current motors</i>), de potencia media, adecuados para máquinas de extrema precisión y escalables en potencia.	

2.1.5. Plataforma PocketBeagle

PocketBeagle es un miembro de un ecosistema de plataformas de desarrollo BeagleBoard. [12]. Las características de esta plataforma, que se muestra en la figura 2.9.A, y que son relevantes para este trabajo son las siguientes:

- Controlador integrado SiP (system-in-package) Octavo Systems OSD3358-SM.
- Memoria de 512MB DDR3.
- Unidad de procesamiento de 32b Cortex-A8 @1-GHz.
- 72 pines de expansión, UART, SPI, I2C, entre otras.
- USB de alta velocidad.

Durante la carrera de maestría se obtuvo experiencia en el uso de otra plataforma de la misma familia, BeagleBoneBlack²sobre la cual se corrió un sistema operativo Linux y se desarrollaron drivers para manejar interfaces de comunicación. Dicha experiencia permitió argumentar que la PocketBeagle cuenta con las interfaces de comunicación necesarias y es capaz de correr el software requerido para este trabajo a una fracción del costo y tamaño.

La única falencia es que no cuenta con una interfaz Wi-Fi ni Ethernet pero se resolvió utilizando un adaptador USB como se destaca en la figura 2.9.B.

²<https://beagleboard.org/black>

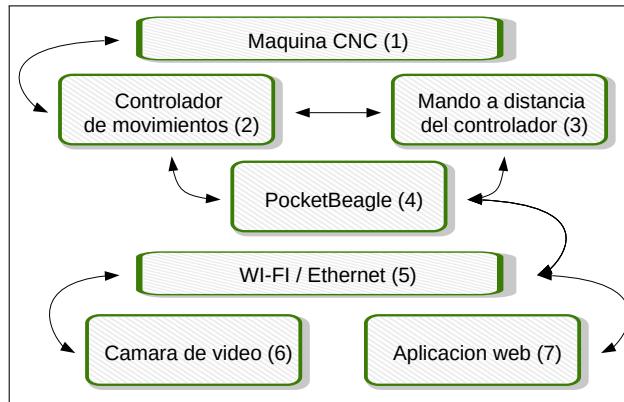


FIGURA 2.8. Diagrama de bloques del sistema implementado para con el objeto de identificar las tecnologías involucradas en cada bloque.

Se está utilizando una distribución oficial del sistema operativo Debian compilada para esta plataforma y se puede descargar desde este link oficial³.

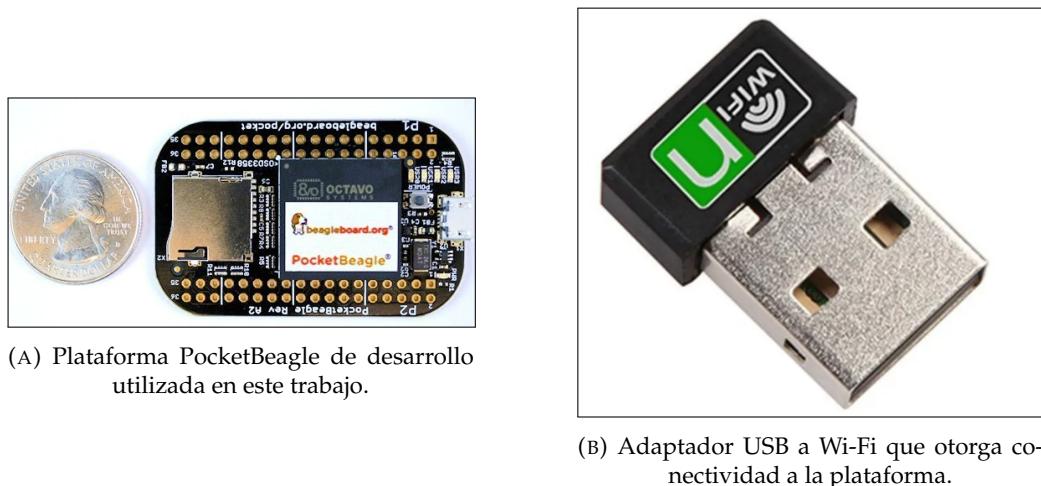


FIGURA 2.9. Plataforma de desarrollo

Se evaluaron plataformas mas potentes como la PYNQ-Z1 [13], pero dado que se trata de un accesorio para un controlador, se intentó mantener los costos y la complejidad justa para la aplicación.

2.1.6. Aplicación web

La interfaz de usuario se desarrolló utilizando tecnologías web para permitir acceder desde cualquier dispositivo a con un navegador web.

Los argumentos a favor de esta tecnología están basados en la falta de aplicaciones para el manejo de máquinas CNC que sean independientes del sistema operativo del ordenador del cliente.

Muchos usuarios utilizan herramientas de diseño sobre el sistema operativo macOS⁴, y deben contar con un segundo computador para poder interactuar con el CNC. Con la estrategia web, solo basta con abrir un navegador desde el mismo entorno

³<https://beagleboard.org/latest-images>

y computador de trabajo para operar el CNC.

Para cumplir con los requisitos planteados se utilizó un arreglo de tecnologías web muy variadas pero ligadas entre sí que se muestran en la figura 2.10.

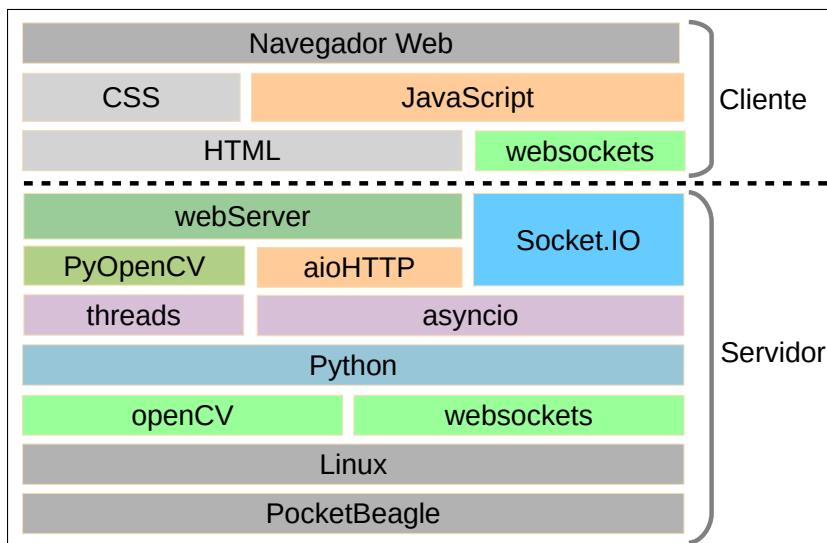


FIGURA 2.10. Capas de software relacionadas con la aplicación web utilizadas en este trabajo.

Para entender la función principal de cada capa se describen algunos detalles en la siguiente lista:

- Python [14]: Es un poderoso y popular lenguaje de programación en el cual se corre principalmente el servidor web, y las funciones de procesamiento de imágenes. La imagen del sistema operativo de la PocketBeagle ya cuenta con Python versión 3 instalado por defecto, lo que asegura compatibilidad.
- asyncio [15]: Es una biblioteca para Python que permite correr una única tarea que administra muchas de manera concurrente y cooperativa. Esto permite que, por ejemplo, una función de Python esté esperando datos de un archivo mientras otra procesa una imagen sin bloquear las funciones del servidor y trabajando todas de manera ordenada.
- aiohttp [16]: Es una biblioteca de Python que permite correr un servidor web utilizando la infraestructura de asyncio para realizar tareas de manera cooperativa y concurrente.
Es el motor del servidor web en este trabajo.
- HTML5.0 [17]: Es el lenguaje de marcas utilizado para visualizar contenidos en la web.
En este trabajo se utiliza para mostrar contenido estático y también para aprovechar un mecanismo nativo de la versión 5.0 para la reproducción de vídeo necesario para mostrar las capturas de la cámara.
- CSS [18]: Es un lenguaje que permite definir estilos, colores, formato y modo de presentación en pantalla de una página escrita en HTML.
Es indispensable para crear aplicaciones web atractivas y apropiadas para cada uso.

- JavaScript [19]: Es un lenguaje de programación intrínsecamente relacionado con HTML que permite la creación de páginas web dinámicas.
La mayoría de los navegadores modernos soportan este lenguaje y esto permite que la aplicación que se desarrolla pueda correr en cualquier plataforma que cuente con un navegador web.
Mas del 90 % de la aplicación web desarrollada está codificada en lenguaje JavaScript, los cálculos de alineación, la interacción con el usuario, etc.
También se utilizan bibliotecas de terceros para diferentes usos escritas en este lenguaje, lo que permite reutilizar código.

- WebGL [20]: Es una biblioteca gráfica (*Web graphics library*) escrita en JavaScript, que permite definir y visualizar objetos de tres dimensiones en una página web.
Esta íntimamente ligada con el desarrollo web y es por ello que puede aprovechar las tarjetas gráficas del ordenador del cliente para acelerar las tareas. De esta manera logra eficiencias similares a las aplicaciones nativas del sistema operativo.

- Three.js [21]: Es una biblioteca escrita en JavaScript, que utiliza la tecnología WebGL pero facilita la creación de objetos, cuenta con muchos ejemplos y casos de uso, abstracta al programador de los detalles de implementación y mejora el mantenimiento del código. Se utiliza en la aplicación para visualizar el movimiento de la máquina en 3D, los trazos de corte, las correcciones de rotación, etc.

- Websockets [22]: Es un protocolo de comunicaciones que opera sobre el protocolo TCP/IP, similar a HTTP, pero está diseñado con la premisa de lograr una comunicación bidireccional de baja latencia.
Es de gran importancia en la aplicación para lograr una rápida respuesta de operación.

- socket.IO [23]: Es una biblioteca de JavaScript que utilizando Websockets permite la comunicación bidireccional entre el servidor web y el o los clientes.
En este trabajo toda la comunicación entre los scripts de JavaScript que corren en el cliente y el servidor en Python se comunican utilizando esta biblioteca.

- OpenCV [24]: Es una biblioteca muy popular escrita en C++ para procesamiento de imágenes asistido por computadora.
Además de contar con potentes algoritmos de procesamiento muy útiles para este trabajo, es soportada por muchas plataformas asegurando la compatibilidad entre dispositivos. En particular la PocketBeagle utiliza la biblioteca libopencv-dev extraída de los repositorios oficiales de Debian.

- PyOpenCV [25]: Es una biblioteca de Python que permite utilizar las funciones de OpenCV.
Dado que este trabajo está escrito en Python, se utiliza esta biblioteca para el procesamiento de marcas, la cual a su vez utiliza libopencv-dev del sistema operativo.

2.1.7. Cámara de vídeo

Los criterios para la selección de la cámara de vídeo se basaron principalmente en la interfaz de comunicación, los costos, la calidad de imagen y la facilidad de adquisición en mercado local. Con dichos criterios se confeccionó la tabla 2.3 con las opciones mas destacadas.

TABLA 2.3. Tabla comparativa entre diferentes cámaras

Modelo	Interfaz	Calidad [0-5]	Accesibilidad [0-5]
Celular	Wi-Fi	4	5
Microscopio	USB	5	5
web-cam	USB	3	5
Industrial	Ethernet	5	1

Si bien la opción del microscopio USB es atractiva, la gran extensión que debe recorrer el cable por las bandejas portacables compartida con la alimentación de los motores, desalienta su uso.

Por otro lado las cámaras industriales con interfase Ethernet resuelven este problema, pero por sus altos costos y difícil accesibilidad en el mercado local, se decidió postergar su utilización.

Para este trabajo se optó por utilizar un celular la aplicación IPWebcam [26] para la transmisión de vídeo como se muestra en la figura 2.11.

Esta opción resuelve el problema del cable, pero también permite utilizar varios modelos de móvil simultáneamente sin cambiar el software y poder tomar imágenes desde diferentes ángulos simultáneamente.

También cuenta con una interfaz web desde la cual se pueden ajustar los parámetros de la cámara mas importantes: zoom, brillo, desplazamiento, resolución y calidad de imagen.

2.1.8. Trigonometría de alineación

El objetivo del método es poder conocer la dimensión, la posición y la rotación de un objeto relativo a la máquina.

Esto se debe a que tanto la pieza que se desea mecanizar como la máquina podrían estar distorsionadas y lo que importa es solo su relación.

Como se trata de una alineación en dos dimensiones, en geometría implica posicionar, escalar y rotar un plano respecto de otro.

Si se consideran dos planos A y B superpuestos como se muestra en la figura 2.12a en donde A en rojo representa el sistema de coordenadas de la máquina y las dimensiones establecidas en el archivo de corte, mientras que B en azul representa la el objeto real a mecanizar que se encuentra desplazado, rotado y escalado respecto al primero.

Conociendo las coordenadas de un solo punto en los dos sistemas de coordenadas, se puede establecer el desplazamiento y corregirlo como se realiza en la

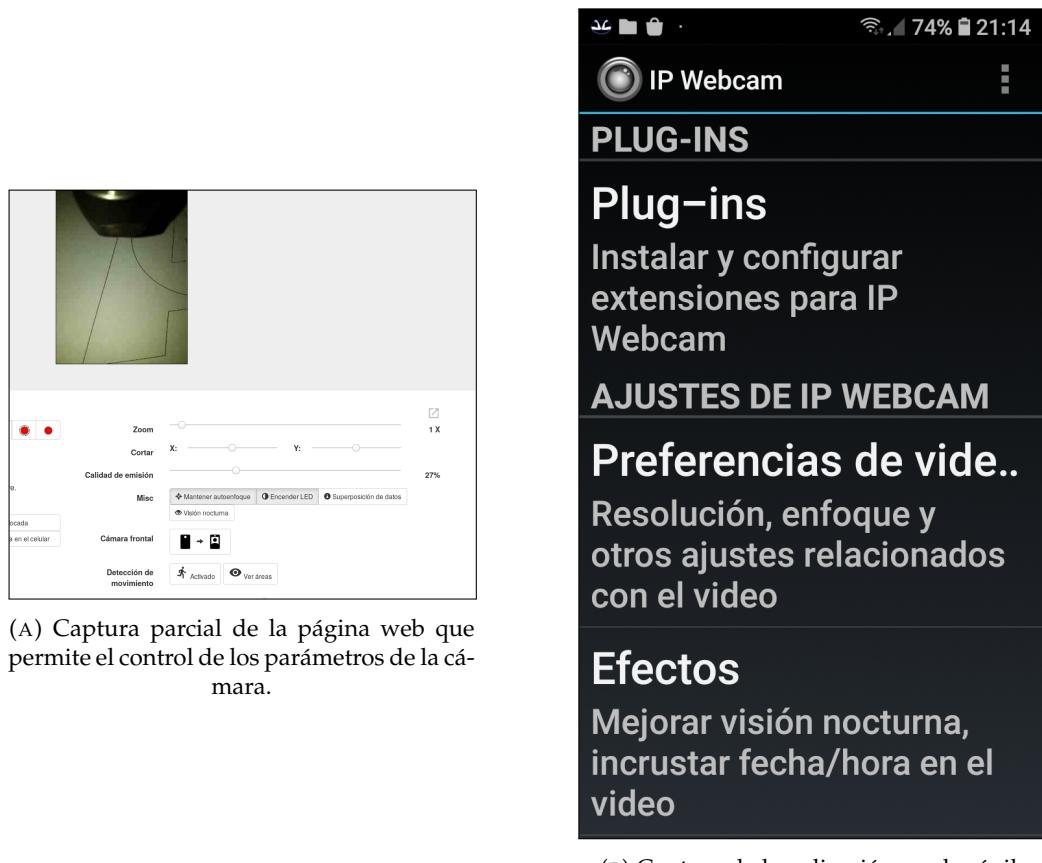


FIGURA 2.11. Aplicación IPWebcam que permite utilizar la cámara del móvil y enviar el vídeo por Wi-Fi.

figura 2.12b. El punto 1 en el sistema A es el $(2, 1)$ mientras que en el sistema B es el $(0, 0)$. La ecuación que corrige la posición de A es la 2.1

$$\begin{aligned} A_x(x) &= x + x_1 \\ A_y(y) &= y + y_1 \end{aligned} \quad (2.1)$$

Ahora, si se considera el punto 2, como se muestra en la figura ??, se puede calcular la rotación relativa entre B y A.

Como primer paso, aplicando trigonometría se calcula el ángulo que forma el punto 2 con el punto 1 en el plano A, luego el ángulo del punto 2 con el punto 1 pero en coordenadas del plano B.

Su diferencia es la rotación del plano A respecto al plano B.

Se puede ver gráficamente en las figuras 2.13 B y C y se expresa en las ecuaciones 2.2.

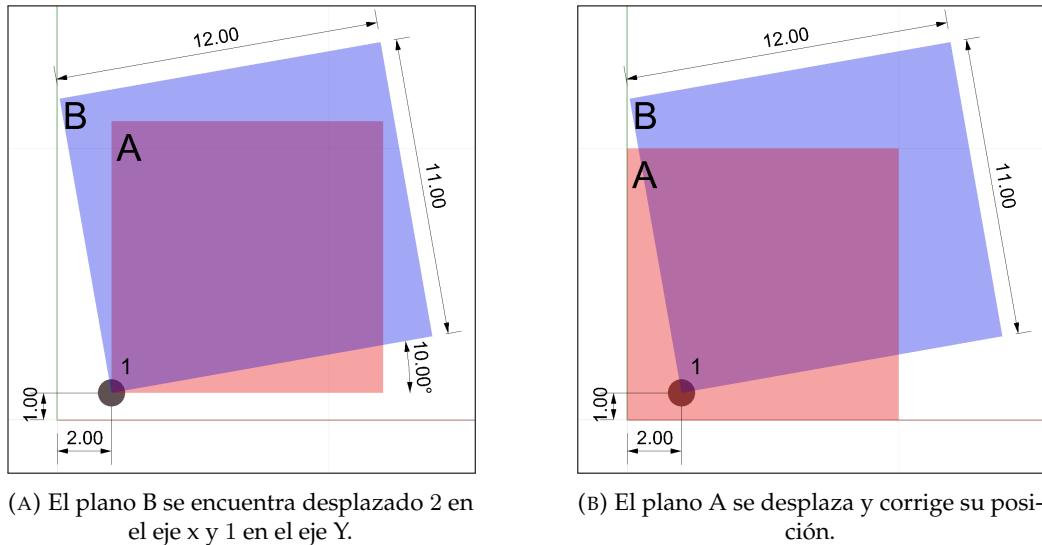


FIGURA 2.12. Corrección de desplazamiento. A representa el sistema de coordenadas de la máquina y las dimensiones extraídas del archivo de corte, B representa el objeto real desplazado, escalado y rotado respecto del primero

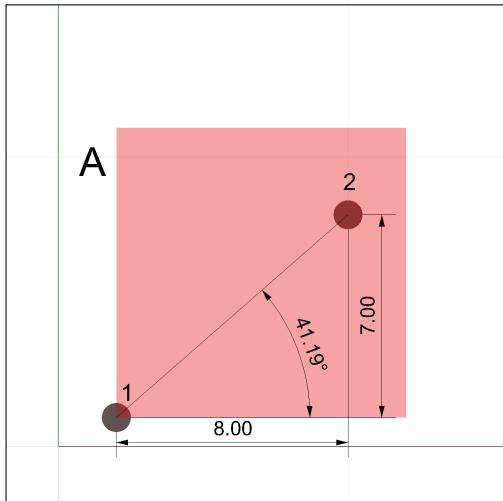
$$\begin{aligned}
 R_A &= \arctan\left(\frac{x_{2A}}{y_{2A}}\right) \\
 &= \arctan\left(\frac{7}{8}\right) \\
 &= 41.18^\circ \\
 R_B &= \arctan\left(\frac{x_{2B}}{y_{2B}}\right) \quad (2.2) \\
 &= \arctan\left(\frac{5,5}{9,09}\right) \\
 &= 31.18^\circ \\
 R_{AB} &= R_A - R_B \\
 &= 10^\circ
 \end{aligned}$$

Una vez obtenida la diferencia de ángulos se corrige rotando el plano A respecto del B como se muestra en la figura 2.14.

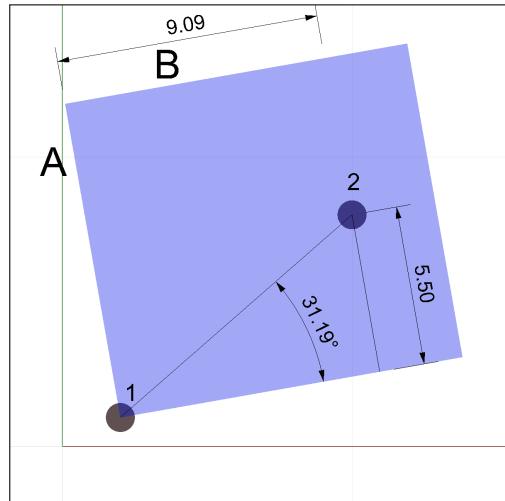
Para completar el proceso y conseguir la alineación final resta escalar el plano A relacionando las dimensiones con el plano B.

Se muestra gráficamente esta corrección en la figura 2.15 y se expresa en las ecuaciones 2.3.

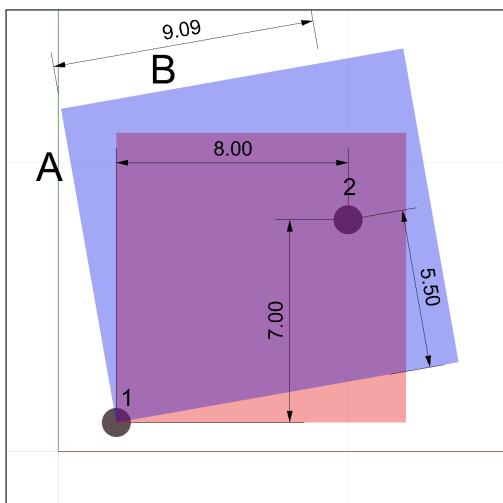
$$\begin{aligned}
 S_{Ax} &= \frac{x_B}{x_A} \\
 &= \frac{12}{10} \\
 &= 1,2 \quad (2.3) \\
 S_{Ay} &= \frac{y_B}{y_A} = \frac{11}{10} \\
 &= 1,1
 \end{aligned}$$



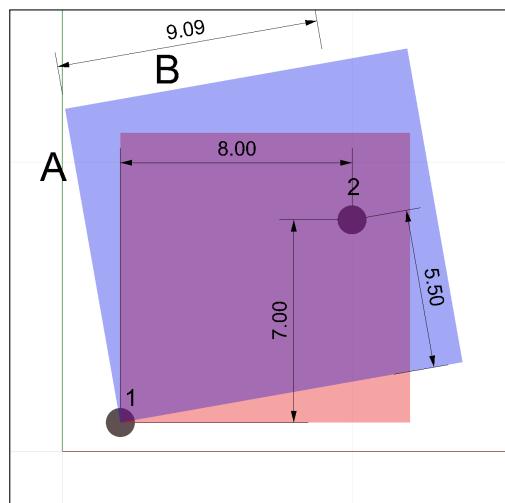
(A) Se calcula el ángulo en el plano B.



(B) Se calcula el ángulo en el plano A.



(C) Se calcula las coordenadas del punto 2 en cada sistema de coordenadas.



(D) Se calcula las coordenadas del punto 2 en cada sistema de coordenadas.

FIGURA 2.13. Cálculo de rotación conociendo las coordenadas relativas de dos puntos.

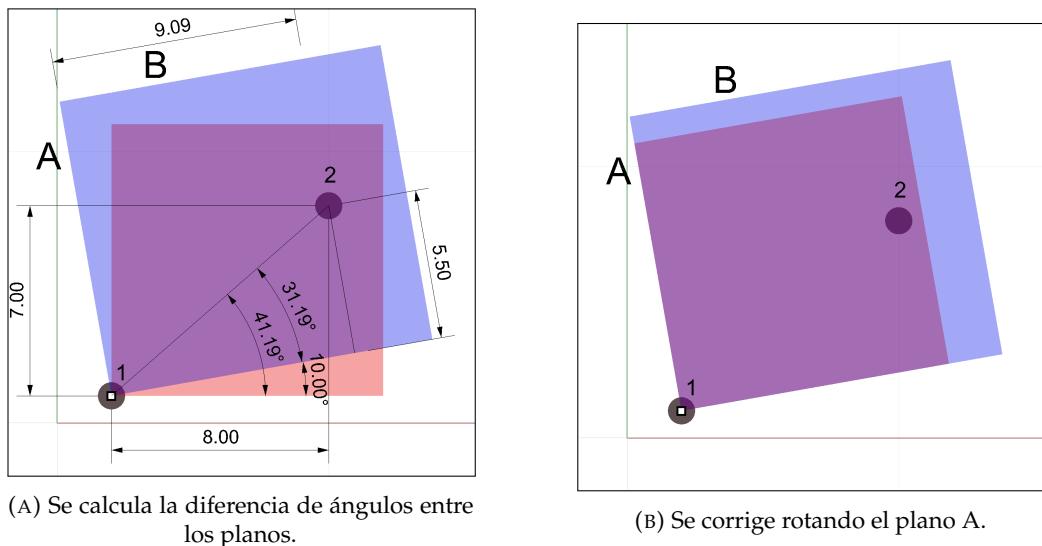


FIGURA 2.14. Corrección de la rotación.

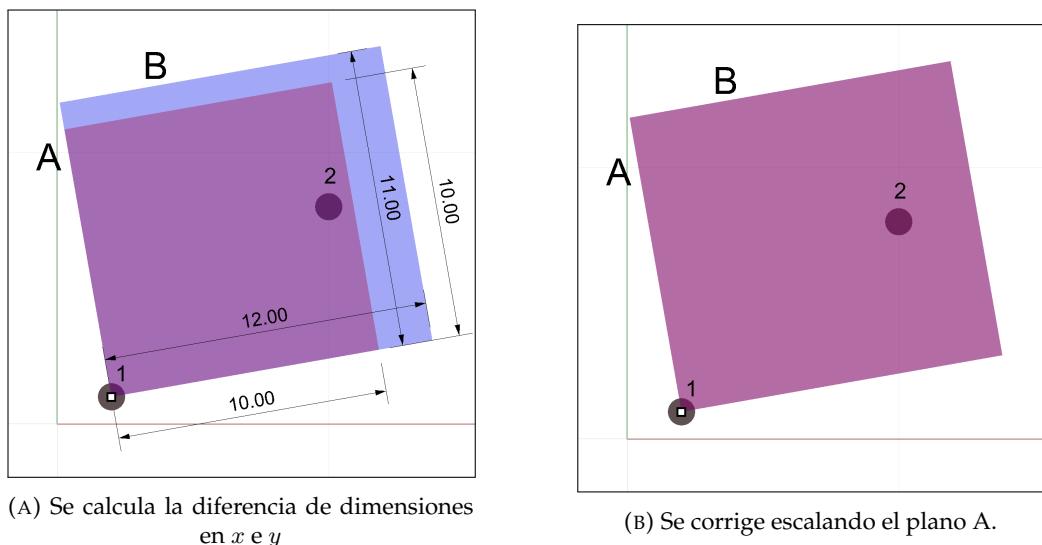


FIGURA 2.15. Corrección de la escala.

Estas ecuaciones se utilizaron para el desarrollo de la aplicación detallada en el capítulo 3.

2.1.9. Detection y tipos de marcas fiduciales

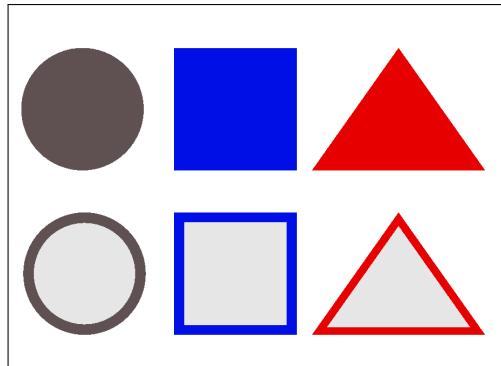
Es posible utilizar diversas figuras geométricas e incluso colores para identificar la posición de una marca, sin embargo en el mercado gráfico y de mecanizado es usual encontrar que las marcas son círculos o cuadrados de entre 1mm y 10mm de diámetro o de lado.

Para un efectivo reconocimiento de una marca utilizando una cámara de vídeo algunas consideraciones deben tenerse en cuenta como ser:

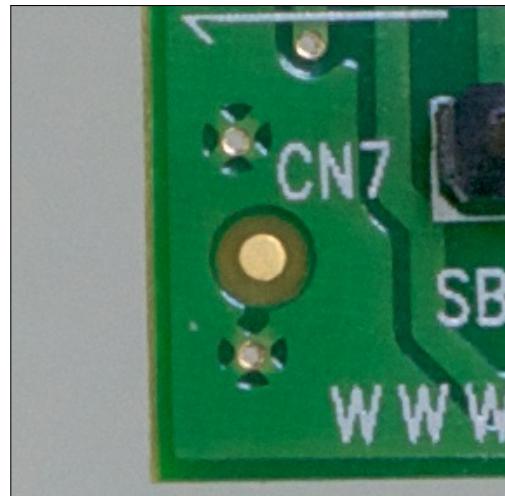
- Maximizar el contraste entre el fondo del objeto y la marca.
- Evitar irregularidades en el trazado del contorno.

- De ser posible que la marca esté pintada internamente y que no sea solo un contorno.
- Que esté alejada de bordes y otras figuras de la pieza.
- En el caso de marcas cuadradas es posible calcular el centro y también una aproximación del ángulo.
- En el caso de marcas circulares se logra mejor precisión en la detección del centro.

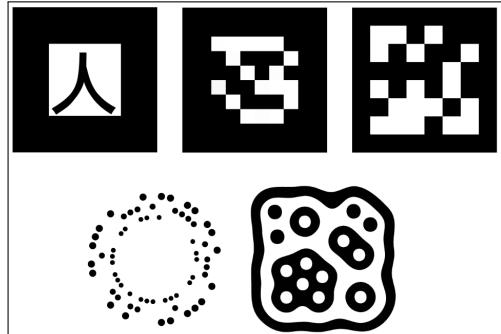
Algunos ejemplos de marcas fiduciales se muestran en la figura 2.16.



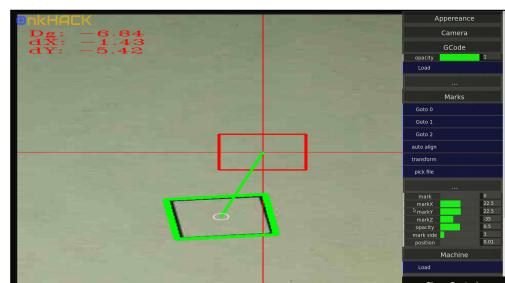
(A) Figuras geométricas llenas o contorno



(B) Ejemplo de marcas fiduciales en una placa de circuito impreso PCB.



(C) Marcas fiduciales codificadas. Se pueden utilizar no solo para reconocer la posición sino tambien leer un numero codificado.



(D) Reconocimiento de un contorno cuadrado.

FIGURA 2.16. Ejemplo de diferentes tipos de marcas fiduciales.

Para el alcance de este trabajo solo se utilizarán marcas geométricas llenas o contornos.

Capítulo 3

Diseño e implementación

En este capítulo se muestra en detalle la solución implementada destacando los bloques constitutivos y su interoperabilidad.

3.1. Intervención al NK105

Es importante mencionar que el controlador elegido para el trabajo no cuenta con una API de control y tampoco está dotado de algún canal de comunicaciones dedicado para tal fin, sin embargo su uso es tan popular y su desempeño es reconocido en la industria, por lo que aun con esta dificultad fue el controlador elegido. Para resolver la comunicación con el controlador se intervino el cable que conecta el mando a distancia con el controlador. Este cable cuenta con 2 interfaces, una UART que comunica el estado de los botones del mando a distancia y una interfase SPI para el envío de la información a la pantalla LCD. Estas interfaces se adaptaron en tensión y se conectaron a la SBC para poder procesar los datos como se muestra en la figura ??

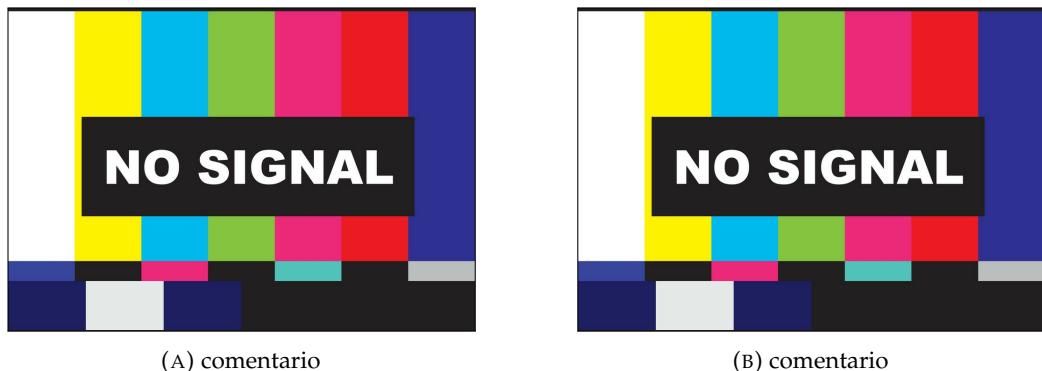


FIGURA 3.1. Mando a distancia del controlador NK105 a) Comunicación por UART del estado de la botonera b) Envío de datos del controlador al LCD por SPI

En la figura ?? se muestran los bloques de software utilizados para procesar la información proveniente del mando a distancia. Se aprovecharon los mecanismos de colas y names pipes que ofrece Linux para multiplexar los datos provenientes de la UART del mando a distancia con los datos generados por la propia SBC.

3.2. Envío de archivos por USB

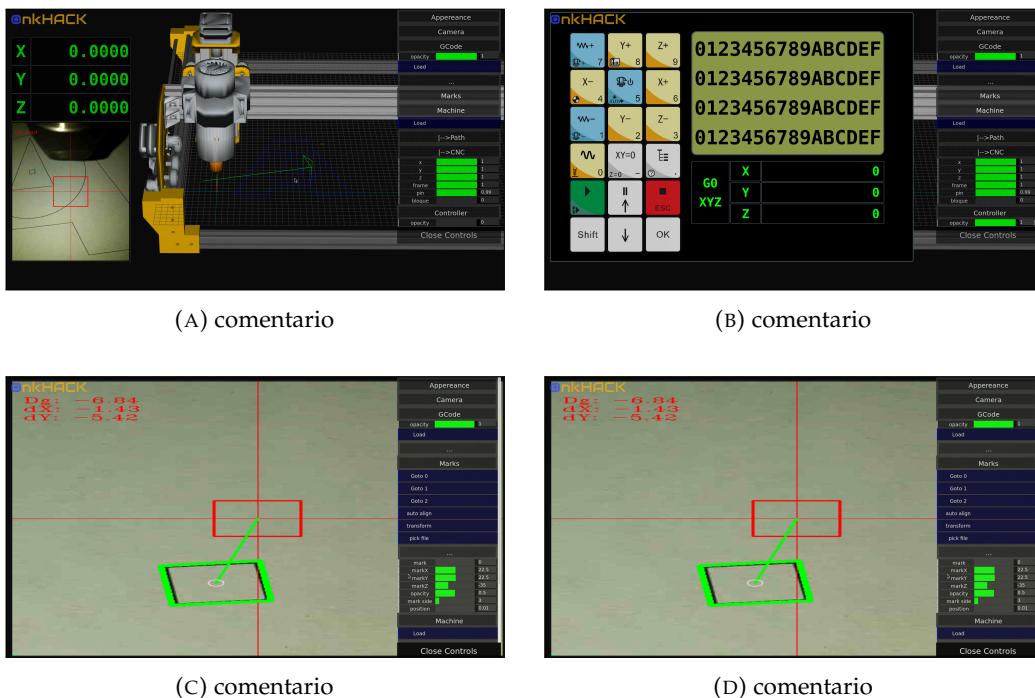


FIGURA 3.2. Aplicacion web desarrollada para controlar la maquina, transferir archivos, gestionar la detección de marcas, visualizar el trabajo a cortar en tiempo real, entre otras utilidades de configuracion.

Capítulo 4

Ensayos y Resultados

4.1. Pruebas funcionales del hardware

- 4.1.1. Validación de tramas SPI
- 4.1.2. Validación de tramas UART
- 4.1.3. Precisión de centro y ángulo de las marcas

4.2. Pruebas funcionales del software

- 4.2.1. Ensayos con diferentes cámaras y resoluciones de video
- 4.2.2. Validación del error de rotación del GCode
- 4.2.3. Validación del error de alineación
- 4.2.4. Ensayo general de caja negra con diferentes archivos de mecanizado

Capítulo 5

Conclusiones

5.1. Conclusiones generales

La idea de esta sección es resaltar cuáles son los principales aportes del trabajo realizado y cómo se podría continuar. Debe ser especialmente breve y concisa. Es buena idea usar un listado para enumerar los logros obtenidos.

Algunas preguntas que pueden servir para completar este capítulo:

- ¿Cuál es el grado de cumplimiento de los requerimientos?
- ¿Cuán fielmente se pudo seguir la planificación original (cronograma incluido)?
- ¿Se manifestó algunos de los riesgos identificados en la planificación? ¿Fue efectivo el plan de mitigación? ¿Se debió aplicar alguna otra acción no contemplada previamente?
- Si se debieron hacer modificaciones a lo planificado ¿Cuáles fueron las causas y los efectos?
- ¿Qué técnicas resultaron útiles para el desarrollo del proyecto y cuáles no tanto?

5.2. Próximos pasos

Acá se indica cómo se podría continuar el trabajo más adelante.

Bibliografía

- [1] https://en.wikipedia.org/wiki/Numerical_control.
- [2] <https://www.edingcnc.com/>.
- [3] <http://www.cnccamera.nl/src-gen/products.html>.
- [4] <https://www.wolfcut.es/>.
- [5] <https://www.pv-automation.com/en/>.
- [6] <https://www.pv-automation.com>.
- [7] <https://www.machinekit.io/>.
- [8] <https://www.summa.com/en/solutions/>.
- [9] <https://www.weihong.com.cn/en/products/controller/20160424/23.html>.
- [10] wikipedia.com. *La especificacion del lenguaje GCode.*
<https://en.wikipedia.org/wiki/G-code#Implementations>. Ene. de 2019.
(Visitado 10-11-2020).
- [11] Elena R. Messina Thomas R. Kramer Frederick M. Proctor. *The NIST RS274NGC Interpreter - Version 3.* 6556. Rev. 3. NIST National Institute of Standards y Technology. 2000.
- [12] *Ecosistema de placas de desarrollo de proposito general.*
<https://http://beagleboard.org/>. (Visitado 02-02-2021).
- [13] <https://www.xilinx.com/products/boards-and-kits/1-hydd4z.html>.
- [14] <https://www.python.org/>.
- [15] <https://docs.python.org/3/library/asyncio.html>.
- [16] <https://docs.aiohttp.org/en/stable/>.
- [17] <https://en.wikipedia.org/wiki/HTML5>.
- [18] <https://en.wikipedia.org/wiki/CSS>.
- [19] <https://en.wikipedia.org/wiki/JavaScript>.
- [20] <https://get.webgl.org/>.
- [21] <https://https://threejs.org/>.
- [22] <https://en.wikipedia.org/wiki/WebSocket>.
- [23] <https://socket.io/>.
- [24] <https://opencv.org/>.
- [25] <https://pypi.org/project/pyopencv/>.
- [26] <https://play.google.com/store/apps/details?id=com.pas.webcam&hl=en&gl=US>.