

**Diseño e implementación de un controlador fuera de línea para maquinas CNC basadas en  
GRBL 1.1+**

Carlos Alfonso Guerrero Navarro

Universidad Nacional Abierta y a Distancia – UNAD  
Escuela de Ciencias Básicas Tecnología e Ingeniería – ECBTI  
Programa de Ingeniería Electrónica  
Corozal  
2021

**Diseño e implementación de un controlador fuera de línea para maquinas CNC basadas en  
GRBL 1.1+**

Carlos Alfonso Guerrero Navarro

Director

Ing. Wilson Jesus Arrubla Hoyos

Universidad Nacional Abierta y a Distancia – UNAD

Escuela de Ciencias Básicas Tecnología e Ingeniería – ECBTI

Programa de Ingeniería Electrónica

Corozal

2021

### **Dedicatoria**

A **Dios todo poderoso** que me dio la vida y que me brinda todo lo que necesito gracias a Él  
puedo seguir por el camino.

A **mis padres y hermanos** quienes me apoyaron en todo momento y me dieron mucho ánimo  
para seguir adelante.

*Carlos Alfonso Guerrero Navarro*

## **Agradecimientos**

Agradezco primeramente a **Dios**, quien me llenó de sabiduría y entendimiento para idear, desarrollar y culminar este proyecto. Sin Él, nada soy.

A **mis padres, hermanos y amigos** que de una u otra forma me apoyaron para avanzar hacia mi camino como profesional.

A **Lizeth Leguia** y **Jesus Pion**, quienes siempre me dieron de su apoyo y conocimiento, motivándome a dar lo mejor de mí, logrando así éste y muchos otros logros más.

Por último, y no menos importantes, a **los profesores y tutores** que he tenido a lo largo de mi preparación académica, especialmente a los Ing. **Wilson Arrubla** y **Fernando Diaz**, de la UNAD. Quienes siempre confiaron en mí y me tuvieron en cuenta para las diferentes oportunidades que se presentaban en la institución, como participar en el semillero de investigación ENGINE, pudiendo así participar en eventos de investigación local y nacional, adquiriendo experiencia y momentos que nunca olvidaré.

A todos ellos: Dios los bendiga, guarde y multiplique todo lo que han hecho por mí.

*Carlos Alfonso Guerrero Navarro*

## Resumen

Existen maquinas con diferentes diseños que emplean el control numérico computarizado para un sinfín de soluciones, por ejemplo, las maquinas CNC DIY, que son fabricadas por aficionados de la electrónica para resolver problemas específicos, y que presentan una serie de desafíos por resolver.

Este proyecto se enfocó en el diseño e implementación de un sistema embebido capaz de controlar una maquina CNC basada en el firmware GRBL en la versión 1.1 o superior, con el propósito de suplir las funciones que normalmente un computador ejerce sobre estas máquinas.

Para el desarrollo de este proyecto se identificaron y caracterizaron los procesos y/o comandos necesarios para controlar el estado de una maquina basada en GRBL 1.1, con el fin de establecer los comandos para ejecutar acciones directamente desde el controlador; se diseñó y desarrolló un sistema embebido para el control de una maquina CNC, seleccionando los componentes y software ideales para su óptimo funcionamiento; y, por último, se validó el funcionamiento y operó la solución propuesta a través de simulaciones y, posteriormente, en una maquina CNC, para probar y/o corregir las funciones que este ejecuta.

Al finalizar, se desarrolló el sistema embebido o controlador que sea capaz de reemplazar las funciones que realiza un computador en una maquina CNC, pudiendo controlar sus principales movimientos y/o acciones. Además, se verificó que los componentes electrónicos y código de programación escrito, se unen de tal manera que cumplen las funciones para las cuales fueron escogidos.

## Tabla de contenido

<b>Lista de tablas.....</b>	<b>8</b>
<b>Lista de Figuras.....</b>	<b>9</b>
<b>Introducción .....</b>	<b>11</b>
<b>Descripción del Proyecto .....</b>	<b>12</b>
Definición y Formulación del Problema.....	12
Justificación .....	16
Objetivos .....	17
Objetivo General.....	17
Objetivos Específicos.....	17
<b>Marco Referencial.....</b>	<b>18</b>
Marco Teórico.....	18
Antecedentes .....	18
Fundamentos .....	22
<b>Metodología .....</b>	<b>24</b>
Fases Para el Desarrollo del Proyecto.....	24
Fase 1. Identificación y Caracterización.....	24
Fase 2. Diseño y Desarrollo .....	29
Fase 3. Validación.....	37
<b>Resultados.....</b>	<b>39</b>

Parámetros.....	40
Menú Control .....	41
Menú Configuración. ....	43
Menú de la tarjeta microSD. ....	43
Implementación del controlador fuera de línea para maquinas CNC basadas en GRBL 1.1+. ....	47
Plan de Mejoras a Corto y Mediano Plazo.....	49
Corto Plazo.....	49
Mediano Plazo .....	49
<b>Conclusiones .....</b>	<b>50</b>
<b>Referencias .....</b>	<b>51</b>

## Lista de tablas

<b>Tabla 1.</b> Listado de comandos del código G.....	27
<b>Tabla 2.</b> Comandos en tiempo real de GRBL.....	29
<b>Tabla 3.</b> <i>Comparación de tarjetas Arduino.</i> .....	31
<b>Tabla 4.</b> <i>Costos</i> .....	34
<b>Tabla 5.</b> Parámetros de configuración de GRBL .....	53



## Lista de Figuras

<b>Figura 1.</b> <i>Maquina CNC DIY</i> .....	12
<b>Figura 2.</b> <i>Atmega328P</i> .....	13
<b>Figura 3.</b> <i>Árbol de causas - efectos del problema</i> .....	14
<b>Figura 4.</b> <i>Arduino UNO</i> .....	15
<b>Figura 5.</b> <i>Esquema Arduino-GCode-Sender</i> .....	19
<b>Figura 6.</b> <i>Aplicativo Android Grbl Controller</i> .....	20
<b>Figura 7.</b> <i>Tarjeta G-Sender</i> .....	21
<b>Figura 8.</b> <i>Tarjeta MKS TFT24</i> .....	21
<b>Figura 9.</b> <i>Pines de GRBL</i> .....	24
<b>Figura 10.</b> <i>Arduino CNC Shield V3.10</i> .....	25
<b>Figura 11.</b> <i>Arduino MEGA</i> .....	30
<b>Figura 12.</b> <i>LCD 2004 I2C</i> .....	32
<b>Figura 13.</b> <i>Encoder rotativo KY-040</i> .....	33
<b>Figura 14.</b> <i>Módulo microSD SPI</i> .....	34
<b>Figura 15.</b> <i>Plano de conexiones</i> .....	35
<b>Figura 16.</b> <i>Caja</i> .....	36
<b>Figura 17.</b> <i>Arduino IDE</i> .....	37
<b>Figura 18.</b> <i>Árbol de funciones</i> .....	39
<b>Figura 19.</b> <i>Establecer conexión por software</i> .....	40
<b>Figura 20.</b> <i>Pantalla "Estado de la maquina"</i> .....	40
<b>Figura 21.</b> <i>Pantalla "Menú principal"</i> .....	41
<b>Figura 22.</b> <i>Pantalla "Menú Control"</i> .....	41

<b>Figura 23.</b> Pantalla " <i>Desbloquear GRBL</i> " .....	41
<b>Figura 24.</b> Pantallas " <i>Mover Eje</i> " .....	42
<b>Figura 25.</b> Pantalla " <i>Configuración</i> " .....	43
<b>Figura 26.</b> Pantalla " <i>Menú Tarjeta – Seleccionar Archivo</i> " .....	43
<b>Figura 27.</b> Pantalla " <i>Menú Tarjeta – Confirmación de Archivo</i> " .....	44
<b>Figura 28.</b> Pantalla " <i>Menú Tarjeta – Mecanizando</i> " .....	44
<b>Figura 29.</b> Pantalla " <i>Modificar en Mecanizado</i> " .....	45
<b>Figura 30.</b> Pantalla " <i>Mecanizado Cancelado</i> " .....	45
<b>Figura 31.</b> Pantalla " <i>Modificar Velocidad de Movimientos</i> " .....	46
<b>Figura 32.</b> Pantalla " <i>Modificar Velocidad del Motor de Corte</i> " .....	46
<b>Figura 33.</b> Pantalla " <i>Mecanizado completado</i> " .....	46
<b>Figura 34.</b> Sistema ensamblado. ....	47

## Introducción

En la comunidad de personas que se involucran en el mundo de las maquinas basadas en control numérico computarizado fabricadas por uno mismo (CNC DIY), enfrentan el problema de tener que manejarlas o enviar las ordenes desde un computador, debido a que el firmware GRBL no trae integrada la opción para hacerlo directamente, generando sobrecostos operativos, al incrementar el consumo energético y adquisición de equipos de cómputo.

El propósito de este proyecto será el de diseñar e implementar un sistema embebido, el cual se llamará “controlador fuera de línea para maquinas CNC basadas en GRBL 1.1+”. Este sistema permitirá suplir las funciones que cumple un computador en una maquina CNC: enviar órdenes y procesar las respuestas que ésta envía.

Para desarrollar dicho sistema, se identificarán y caracterizarán los procesos y/o comandos necesarios para controlar el estado y movimientos de una maquina CNC basada en GRBL 1.1. Posteriormente se diseñará y desarrollará el sistema embebido, seleccionando los compontes y software ideales para su óptimo funcionamiento y, como último paso, se validará el funcionamiento de la solución propuesta en una maquina CNC.

Como resultado, en este documento se mostrará el desarrollo de un controlador fuera de línea, basado en un sistema embebido, donde el usuario tenga la opción de visualizar los diferentes estados de la máquina, contralor el movimiento de sus ejes, realizar trabajos de mecanizado y entre otras.

## Descripción del Proyecto

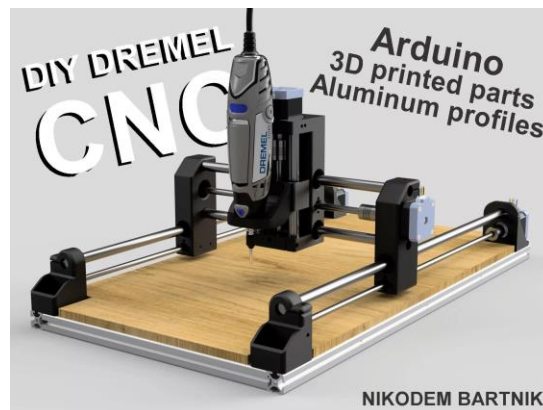
### Definición y Formulación del Problema

Las maquinas CNC DIY son ensambladas para un amplio campo de aplicaciones, ya sea para la fabricación de Circuitos Impresos (PCB), corte, grabado y/o tallado de madera, metal, acrílico y otros materiales, todo esto a través de diferentes herramientas, ya sean cuchillas, laser, brocas de corte o tallado, etc.

Este tipo de máquinas son ampliamente conocidas, debido a que sus planos, esquemas y firmware, se comparten de forma gratuita en muchas páginas de internet, como en [www.thingiverse.com](http://www.thingiverse.com), donde se pueden encontrar diseños como el de la figura 1, ésta es una maquina con partes impresas en 3D, varillas de acero, 1 motortool y otros componentes, las cuales conforman lo que se conoce como maquinas CNC hechas por uno mismo (CNC DIY).

### Figura 1

*Maquina CNC DIY*



*Nota.* Adaptado de *DIY Dremel CNC design and parts* [Ilustración]. Nikodem Bartnik.

<https://www.thingiverse.com/thing:3004773>

Estas máquinas, en su mayoría, están basadas en el firmware GRBL, el cual es un código abierto de programación, este se carga y ejecuta en el microcontrolador AT328p (Figura 2), de la familia AVR de ATmega, con el cual se reciben las órdenes del computador, se procesan y, por último, ejecutadas a través de movimientos u otras acciones en la CNC.

## Figura 2

*Atmega328P*



*Nota.* Adaptado de *ATmega328P* [Ilustración]. Microchip.

<https://www.microchip.com/wwwproducts/en/atmega328p>

De la figura 3 se pueden observar las raíces y consecuencias del problema central que pretende solucionar este proyecto. Desde la implementación del firmware GRBL, los desarrolladores que ha tenido este firmware no agregaron las opciones de visualización y/o control directamente en él, debido a ciertas limitantes de las que se hablarán más adelante. La falta de la incorporación de estas funciones, genera la necesidad de tener la CNC conectada en todo momento a un computador, para que a través de él se envíen las ordenes que la maquina debe realizar, ya sea solo para hacer pruebas o realizar un trabajo CAM. Esto causa mayor consumo energético por parte del computador que, dependiendo del tipo de trabajo a realizar en la CNC, puede tomar horas o incluso días. Además, los trabajos en CNC, por lo general, generan mucho polvo, este puede adentrarse en los componentes del computador, haciendo que su refrigeración y/o vida útil se vea afectada, llegando a ocasionar que tenga afectaciones mínimas o grandes, como quemarse por el recalentamiento.

Por último, otra consideración a tener en cuenta, es el volumen que ocupa el computador, ya que se debe poner en un lugar donde no obstruya al personal y a la misma máquina, además que sea de fácil acceso para su manipulación.

**Figura 3**

*Árbol de causas - efectos del problema*



GRBL, es un código optimizado solo para funcionar con el microcontrolador ya mencionado, el cual es el componente central de la placa de desarrollo Arduino UNO (Figura 4), desde donde parte las limitantes para la implementación de periféricos para el control de la CNC desde su propia electrónica. Lo anterior es debido a 2 razones principales:

- Baja disponibilidad de memoria libre para agregar las líneas de código necesarias para los periféricos, como LCD, enconder y lector de tarjetas microSD.

- Casi nula disponibilidad de pines físicos, para la conexión de los periféricos.

Lo anterior tomado de la sección de “preguntas frecuentes de GRBL”, donde los mismos desarrolladores del firmware recomiendan el uso de otros sistemas para la implementación de otras funciones diferentes a las que ya vienen en GRBL por defecto. (Jeon, 2009/2021).

#### **Figura 4**

*Arduino UNO*



*Nota.* Adaptado de *Arduino UNO* [Ilustración]. Arduino. <https://store.arduino.cc/usa/arduino-uno-rev3>

Por tal razón, se plantea la pregunta **¿Cómo desarrollar un sistema embebido que permita realizar las funciones de un computador en una CNC?**

## Justificación

En base a la definición del problema planteada anteriormente, se propone el Diseño de un Controlador Fuera de Línea Para Maquinas CNC Basadas en GRBL 1.1+. Lo anterior se hace necesario al querer disminuir el consumo energético, abaratar costos al no tener que comprar un computador solo para que funcione con la CNC, prevenir daños en el mismo y hacer un sistema que sea más compacto. Basado en (Gana Energía, 2018), un computador consume en promedio 2.2 kWh y, según (Empresas Públicas de Medellín, 2020) el costo del kWh para la zona no residencial (industrial y comercial) en el mes de septiembre fue de \$625.49 COP. Si esto se calcula con un mecanizado que tarde 3 días, estaría consumiendo 158.4 kWh, costando \$99.078 COP, lo que se puede apreciar que es un costo excesivo. Ayudando de igual forma al medio ambiente, a través del ahorro energético y de los residuos electrónicos que se puedan generar del computador.

Para lo anterior, se propone desarrollar un sistema embebido el cual pueda contar con los componentes electrónicos necesarios, para que a través de él se puedan enviar las órdenes a la maquina CNC, así como realizar trabajos CAM de larga duración, guardando los archivos que contengan el código G en una tarjeta microSD, leer dicho archivo y enviar los comandos a la máquina.

Las personas que se verán beneficiadas, serán principalmente aquellas que desarrollaron por sí mismos su máquina CNC, ya sea por un diseño propio o tomándolo de otro autor, los llamados proyectos DIY.



## Objetivos

### *Objetivo General*

Desarrollar un sistema embebido mediante el uso de un controlador fuera de línea para manipular una maquina CNC basada en GRBL 1.1.

### *Objetivos Específicos*

Identificar y caracterizar los procesos y/o comandos necesarios para controlar el estado de una maquina basada en GRBL 1.1.

Diseñar y desarrollar un sistema embebido para el control de una maquia CNC, seleccionando los compontes y software ideales para su óptimo funcionamiento.

Validar el Funcionamiento y operar la solución propuesta en una maquina CNC.

## Marco Referencial

### Marco Teórico

#### *Antecedentes*

En los últimos años, las CNC han tenido un mayor auge, ya que pasaron de ser usadas (en su mayoría) en la industria, a ser implementadas por personas para el desarrollo de su hobby o en pequeñas empresas. Los 2 últimos actores, suelen implementar CNC del tipo fabricadas por uno mismo, más conocidas como DIY, ya que estas son más económicas, fácil de adquirir sus partes y posterior ensamblaje. Así mismo hay gran variedad de modelos y/o diseños que pueden ser descargados de internet de forma gratuita. Donde también existen desarrollares de software para estas máquinas, desde el firmware que ejecuta la CNC a través de su electrónica, hasta programas para enviar las órdenes a ellas. Como GRBL y Universal G Code Sender, respectivamente.

Estas máquinas de tipo DIY, suelen estar conectadas a un computador, ya que su propia electrónica y/o firmware, no trae las opciones para interconectar periféricos para el control de la misma sin un computador. Por lo anterior, empresas y desarrolladores de sistemas embebidos, han diseñados sistemas llamados *Controladores Fuera de Línea*, el cual su función es reemplazar las funciones que ejerce el computador sobre la CNC, cabe notar que estos sistemas no reemplazan a la electrónica de la CNC, solo es un periférico.

La siguiente información no fue tomada de base de datos científicas, debido a que no se encontró información relacionada en ellas. Por lo anterior, se mencionarán trabajos que la misma comunidad de CNC ha realizado con el tiempo. Entre esas se tiene:

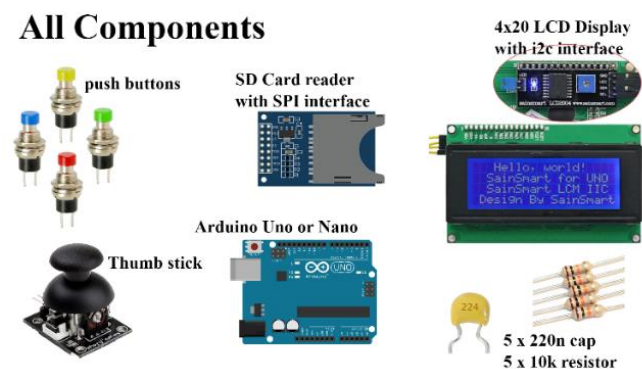
Existen algunos proyectos realizados por la comunidad CNC, como el *Arduino-GCode-Sender* en el cual, como se observa en la figura 5, a través de una tarjeta Arduino UNO, 4

pulsadores, un joystick y otros componentes, realiza el control de una maquina CNC. Este se conecta como un periférico a la tarjeta electrónica principal de la CNC, a través de los pines de UART de comunicación serial (TX-RX) (Venneker, 2016/2021).

Su código de programación está abierto al público, por lo que se puede descargar y modificar como se desee. Por lo anterior, este código se tomó como base de partida para la generación del proyecto diseñado y explicado en este informe. Tomando la sección de lectura de la tarjeta microSD, la cual es una de las funciones principales para el sistema que se desarrollará en este informe.

## Figura 5

### *Esquema Arduino-GCode-Sender*



*Nota.* Adaptado de *Arduino-GCode-Sender* [Figura]. Github.

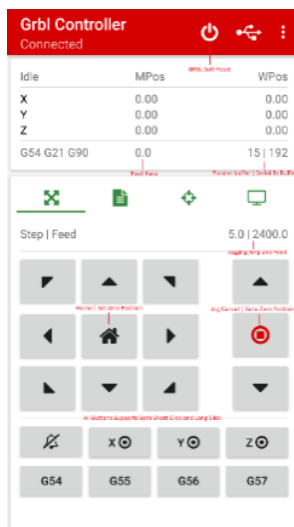
<https://github.com/bvenneker/Arduino-GCode-Sender/blob/master/BOM.svg>

Otro proyecto desarrollado fue un aplicativo móvil para Android, llamado **Grbl Controller**, el cual a través de conexión bluetooth entre el celular y la CNC, se envían órdenes y se controla la máquina. Este se encuentra disponible en la Play Store de Android para su descarga, tiene una versión gratuita con algunas funciones limitadas y publicidad flotante entre ventanas; y otra paga, la cual está abierta en todas sus funciones y sin publicidad (Silicon Circuits, 2017/2021).

La figura 6 muestra la pantalla de visualización de la aplicación instalada en un celular.

## Figura 6

*Aplicativo Android Grbl Controller*



*Nota.* Adaptado de *Grbl Controller* [Figura]. Github.

<https://raw.githubusercontent.com/zeevy/grblcontroller/master/doc/screenshots/JoggingTab.png>

Una empresa china desarrollo una **tarjeta electrónica**, como se aprecia en la figura 7, **con una LCD 16x2**, un lector de tarjeta microSD, un microcontrolador AT328P y otros componentes, los cuales permiten un control de la maquina CNC. Esta está limitada en cuando la posibilidad de modificaciones en tanto el Hardware, como el software, ya que al ser una tarjeta fabrica, ya trae todos los componentes soldados en ella, específicos para el funcionamiento de la misma. En cuanto al código de programación, la empresa fabricante no lo comparte, por lo que tampoco se le pueden realizar ajustes o agregar opciones de acuerdo a usuario final lo desee. (Lekn Tech, 2020/2021).

## Figura 7

### Tarjeta G-Sender



Nota. Adaptado de G-Sender [Figura]. Github. <https://github.com/LEKN-TECH/G-Sender/blob/master/doc/G-Sender%20Manual.pdf>

Otra empresa que desarrolló una tarjeta electrónica, llamada **MKS TFT24**, como se observa en la figura 8, con pantalla TFT táctil, con este puede leer la información almacenada en una tarjeta de memoria SD y ejecutar los comandos para controlar la CNC. Esta, al igual que la tarjeta anterior, no está disponible para su modificación en ningún sentido. (Makerbase, 2016/2021).

## Figura 8

### Tarjeta MKS TFT24



Nota. Adaptado de MKS TFT24 [Figura]. Github. <https://github.com/makerbase-mks/MKS-TFT/issues/189>

## ***Fundamentos***

**Arduino.** Es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar. Las placas Arduino pueden leer entradas (sensores de luz, botones o un mensaje de Twitter) y convertirlo en una salida, activando un motor o encendiendo un LED. Se le puede decir a la tarjeta qué hacer enviando un conjunto de instrucciones al microcontrolador en ella. Para hacerlo, se utiliza el lenguaje de programación C y el Software Arduino IDE. (Wikipedia, 2019).

**CAD/CAM.** Significa diseño asistido por computadora y manufactura asistida por computadora. El software CAD/CAM se usa para diseñar y manufacturar prototipos, productos terminados y tiradas de producción. Un sistema con CAD/CAM integrado ofrece una solución completa desde el diseño a la manufactura. (Autodesk, 2018).

**CNC.** Definen el control numérico computarizado (CNC) como una tecnología que controla los movimientos de una máquina, por medio de un programa preparado, que contiene datos codificados alfanuméricos. CNC puede controlar los movimientos de la pieza de trabajo o herramienta, los parámetros de entrada, tales como profundidad de corte, la velocidad, encendido y apagado del husillo, etc. (Schmid & Kalpakjian, 2000).

**Código G.** Es un lenguaje mediante el cual las personas pueden decir a máquinas-herramienta controladas por computadora qué hacer y cómo hacerlo. Esos "qué" y "cómo" están definidos mayormente por instrucciones sobre adonde moverse, que tan rápido moverse y que trayectoria seguir. Las máquinas típicas que son controladas con código-G son fresadoras, cortadoras, tornos e impresoras 3D. (Diaz del Castillo Rodriguez, 2008)

**DIY.** Son las siglas Do It Yourself (Hágalo Usted Mismo), es la práctica de la fabricación o reparación de objetos por uno mismo, generalmente se utiliza para ahorrar dinero, pues así se

obtienen múltiples beneficios: se desarrolla una actividad manual interesante y motivadora, y al mismo tiempo se aprende con la práctica, se logran manualidad y destrezas. (Wikipedia, 2021)

**GRBL.** Lo define como un firmware de código abierto y alto rendimiento, para el control de movimiento en máquinas CNC. (Jeon, 2009/2021).

## Metodología

### Fases Para el Desarrollo del Proyecto

Este proyecto se enmarcó en un periodo de 3 meses y 3 fases, donde las actividades se distribuyeron de la siguiente forma:

#### *Fase 1. Identificación y Caracterización*

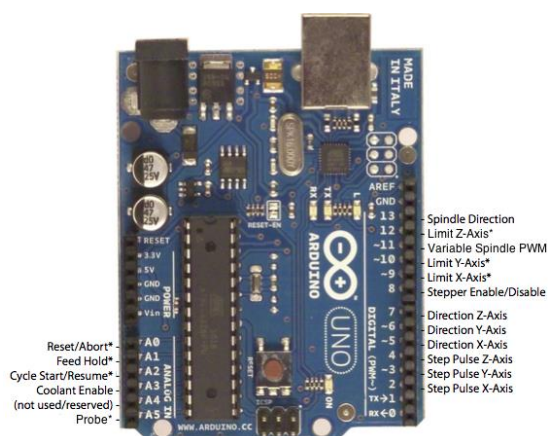
En esta fase se identificaron y caracterizaron los procesos y/o comandos necesarios para controlar el estado de una maquina basada en GRBL 1.1. Dividiéndose en 2 actividades:

**Actividad 1.** Se estudiaron los diferentes parámetros de configuración de GRBL involucrados en el funcionamiento y comunicación de GRBL y la CNC.

La configuración inicial parte de la interconexión de los diferentes componentes electrónicos que integran y hacen funcionar la fusión entre la CNC y GRBL. Donde los pines configurados por defecto (y que no se pueden cambiar) en el Arduino que ejecuta GRBL, están descritos en la figura 9.

#### **Figura 9.**

*Pines de GRBL.*



\* - Indicates input pins. Held high with internal pull-up resistors.

*Nota.* Adaptado de *Conexión de Grbl* [Figura]. Github. [Connecting Grbl · gnea/grbl Wiki](https://github.com/gnea/grbl/wiki) ·

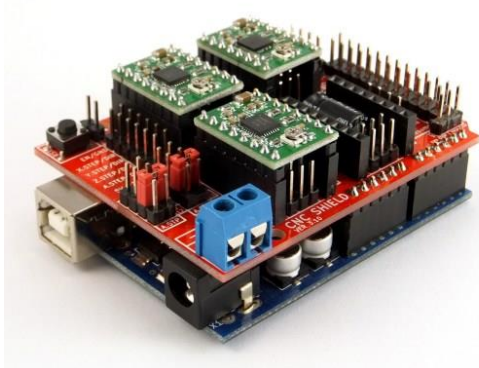
[GitHub](https://github.com)



De igual forma, en la figura 9 se pueden observar donde se conecta cada pin de los Drivers de los motores de cada eje (X-Y-Z), así como de los sensores de final de carrera (para detectar los topes o límites de cada eje) y otros periféricos, como un refrigerante para el motor de corte (spindle), PWM para controlar la potencia de un láser o la velocidad de giro del spindle, etc. Por lo general, se suelen usar módulos o PCB prediseñadas para insertar sobre el Arduino, también conocidos como “Shields”, en este caso, sería la CNC SHIELD CNC V3.10, la cual, como se mencionó, es insertada en los pines de conexión del Arduino y, en este shield, se insertan y conectan los demás componentes, como se observa en la figura 10.

**Figura 10.**

*Arduino CNC Shield V3.10.*



*Nota.* Adaptado de *Arduino CNC Shield* [Figura]. Protoneer. [Arduino CNC Shield – 100% GRBL Compatible | Protoneer.co.nz](https://www.protoneer.co.nz/arduino-cnc-shield-100-grbl-compatible/)

**Actividad 2.** Se investigaron los principales comandos de código G necesarios para el funcionamiento del controlador.

En general, GRBL asume que todos los caracteres y datos de transmisión que se le envían son código G y analizará e intentará ejecutarlos tan pronto como sea posible. Sin embargo, GRBL también tiene dos tipos de comandos de sistema separados que están fuera de la transmisión normal de código G. Un tipo de comando del sistema se transmite a GRBL como

código G, pero comienza con un carácter para decirle a GRBL que no es código G. El otro está compuesto por un conjunto especial de caracteres que inmediatamente ordenarán a GRBL que realice una tarea en tiempo real. No es parte del flujo de código G. Los comandos del sistema de GRBL hacen cosas como controlar el estado de la máquina, informar los parámetros guardados o lo que GRBL está haciendo, guardar o imprimir la configuración de la máquina, ejecutar un ciclo de homing (buscar posición 0,0,0) o hacer que la máquina se mueva más rápido o más lento de lo programado. Estos comandos son enviados desde un computador o cualquier otro dispositivo que se conecte por puerto serial al Arduino que ejecuta GRBL.

**Comandos del Sistema \$.** Los comandos del sistema proporcionan controles adicionales para el usuario, como imprimir comentarios sobre el estado actual del analizador de código G o ejecutar el ciclo de homing. Estos comandos pueden ser:

- **\$\$ - Ver y escribir la configuración de GRBL.** Establece parámetros como los pasos por mm que debe dar el motor para desplazarse una distancia determinada, la velocidad, distancia y aceleración máxima de avance, etc. Las diferentes opciones de observan en el anexo tabla 5.
- **\$# - Ver parámetros de código G.** Los parámetros del código G almacenan los valores de desplazamiento de coordenadas para las coordenadas de trabajo G54-G59, las posiciones predefinidas G28/G30, el desplazamiento de coordenadas G92, los desplazamientos de longitud de herramienta y el sondeo.
- **\$G - Ver el estado del analizador de código G.** Este comando imprime todos los modos de código G activos en el analizador de código G de Grbl. En la tabla se observan los diferentes comandos de código G compatibles con GRBL:

**Tabla 1.***Listado de comandos del código G.*

<b>Grupo Modal</b>	<b>Palabras de los miembros</b>
Modo de movimiento	<b>G0</b> , G1, G2, G3, G38.2, G38.3, G38.4, G38.5, G80
Selección del sistema de coordenadas	<b>G54</b> , G55, G56, G57, G58, G59
Selección de plano	<b>G17</b> , G18, G19
Modo de distancia	<b>G90</b> , G91
Modo de distancia Arc IJK	<b>G91.1</b>
Modo de velocidad de alimentación	G93, <b>G94</b>
Modo Unidades	G20, <b>G21</b>
Compensación de radio de corte	<b>G40</b>
Desplazamiento de longitud de herramienta	G43.1, <b>G49</b>
Modo de programa	<b>M0</b> , M1, M2, M30
Estado del husillo	M3, M4, <b>M5</b>
Estado del refrigerante	M7, M8, <b>M9</b>

*Nota.* Esta tabla resalta en negrita aquellos comandos que vienen activados por defecto en GRBL.

- **\$X - Bloqueo de alarma de apagado.** El modo de alarma de Grbl es un estado en el que algo ha salido críticamente mal, como llegar al límite de un eje o abortar el movimiento de uno o varios ejes, o si Grbl no conoce su posición. De forma predeterminada, si se tiene habilitado el homing y se enciende el Arduino, Grbl entra en el estado de alarma, porque no conoce su posición. El modo de alarma bloqueará todos los comandos de código G hasta que se haya realizado el ciclo de localización *\$H*.
- **\$H - Ejecutar ciclo de homing.** Este comando es la única forma de realizar el ciclo de homing en Grbl. Algunos otros controladores de movimiento designan un comando especial de código G para ejecutar un ciclo de homing, pero esto es incorrecto de acuerdo con los estándares de código G.
- **\$J=line – Ejecutar movimientos en ejes.** Este comando realizará movimientos en los ejes de la máquina CNC. Los comandos Jog se comportan de manera casi idéntica a la transmisión normal de código G. Cada comando Jog devolverá un 'OK' cuando el movimiento de jogging se haya analizado y esté configurado para su ejecución. Si un comando no es válido o supera un límite de software, Grbl devolverá un “*error:*”.

***Comandos en tiempo real de Grbl v1.1.*** Son caracteres de control únicos que se pueden enviar a Grbl para ordenar y realizar una acción en tiempo real. Esto significa que se pueden enviar en cualquier momento y en cualquier lugar, y Grbl responderá de inmediato, independientemente de lo que esté haciendo en ese momento. Entre estos comandos, se tiene:

**Tabla 2.***Comandos en tiempo real de GRBL.*

<b>Comando</b>	<b>Descripción</b>
0x18 (ctrl-x)	Soft-Reset
~	Pausar/Reanudar movimientos
0x85	Cancelar movimiento Jog
0x90	Establecer velocidad de movimientos al 100% de lo programado en el mecanizado
0x91	Aumentar velocidad de movimiento un 10%
0x92	Disminuir velocidad de movimiento un 10%
0x93	Aumentar velocidad de movimiento un 1%
0x94	Disminuir velocidad de movimiento un 1%
0x99	Establecer velocidad de giro del spindle al 100% de lo programado en el mecanizado
0x9A	Aumentar velocidad de movimiento un 10%
0x9B	Disminuir velocidad de giro del spindle un 10%
0x9C	Aumentar velocidad de giro del spindle un 1%
0x9D	Disminuir velocidad de giro del spindle un 1%

***Fase 2. Diseño y Desarrollo***

En la segunda fase se diseñó y desarrolló un sistema embebido para el control de una maquina CNC, seleccionando los componentes y software ideales para su óptimo funcionamiento.

Dividiéndose en 3 actividades:

**Actividad 1.** Caracterizar los componentes electrónicos.

Para la caracterización de los componentes electrónicos, se tuvo en cuenta su precio, disponibilidad en el mercado, funcionalidad y facilidad de uso para la programación/interconexión entre ellos. Permitiendo que las personas tengan fácil acceso a ellos y no se les complique al momento de conectar un componente con el otro.

Como cerebro del sistema embebido, se escogió un microcontrolador con disponibilidad de memoria para el almacenamiento del algoritmo. Así como fácil acceso a sus pines, sin tener que soldar regletas o conectores en ellas. Por lo anterior, lo ideal para este proyecto, fue seleccionar uno de los diferentes tipos de Arduino que existen en el mercado.

**Arduino MEGA.** Es una placa de microcontrolador basada en el ATmega2560. Tiene 54 pines digitales de entrada/salida (de los cuales 15 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UART (puertos serie de hardware), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un encabezado ICSP y un botón de reinicio. (Arduino, 2020). En la figura 11 se observa la placa de desarrollo Arduino Mega, con sus diferentes componentes y pines.

**Figura 11**

*Arduino MEGA.*



*Nota.* Adaptado de *Arduino Mega 2560 R3* [Figura]. Naylamp Mechatronics.

<https://naylampmechatronics.com/arduino-tarjetas/9-arduino-mega-2560.html>

Aunque sus pines no son usados en su totalidad, la selección principal de esta tarjeta se debió a la capacidad de memoria del microcontrolador para almacenar el programa de control para el sistema, donde se puede ver la comparación de este y otras tarjetas contra el espacio que necesita el programa para ser almacenado en la tabla 3.

**Tabla 3.**

*Comparación de tarjetas Arduino.*

<b>Arduino</b>	<b>Microcontrolador</b>	<b>Memoria disponible</b>	<b>Memoria requerida por el programa</b>	<b>Precio (\$COP)</b>
UNO				\$28560
NANO	ATmega328	32 kB		\$23800
PRO MINI			36.98 kB	\$15470
LEONARDO	ATmega32u4	32 kB		\$24990
MEGA	ATmega2560	256 kB		\$40460

*Nota.* Información y precios tomados de <https://www.didacticaselectronicas.com/>

Para la interfaz con el usuario, se necesitó una pantalla con el tamaño suficiente para mostrar toda la información en 1 ventana, como: posición actual de los ejes, menús, y/u otros datos. En el mercado hay gran variedad de pantallas, ya sean LCD, táctiles, shield con botones incorporados, etc. Donde cada una ofrece gran variedad de funciones y, hablando de versatilidad, la mejor sería la táctil, ya que minimiza las conexiones al no tener que agregar botones físicos, sino que se añaden a la propia pantalla, pero estas son costosas y necesitan otra programación aparte, por lo que complicaría la interconexión entre ella y el Arduino.

**LCD 2004 con módulo de comunicación I2C.** Es una pantalla alfanumérica de 20 columnas y 4 filas, la cual dispone de 2 pines de alimentación, 4 de control, 8 de comunicación y 2 para encender el led (en este caso de color azul) de retroiluminación. Para reducir la cantidad de pines antes mencionada, y así minimizar los cables de conexión que irían en el interior de la caja del sistema, se implementó un módulo de comunicación I2C para LCD, con la cual se reduce la cantidad de pines a solo 4: 2 para alimentación y 2 en comunicación.

La figura 12 muestra el módulo de comunicación i2c (circuito superior) y la pantalla LCD 2004 (circuito inferior).

**Figura 12.**

*LCD 2004 I2C*



*Nota.* Adaptado de *LCD 2004 I2C* [Figura]. Amazon. <https://images-na.ssl-images-amazon.com/images/I/61cGkRPqtfL.AC.SX679.jpg>

De igual forma, ya que la pantalla no fue táctil, se necesitaron físicos botones para acceder a las diferentes funciones del sistema, para ahorrar líneas de conexión y no tener que instalar varios botones, se escogió el uso de un encoder rotativo con pulsador.

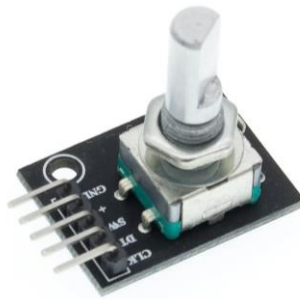


**Encoder rotativo con pulsador.** Este es un dispositivo que es capaz de convertir el movimiento angular de su perilla, en pulsos electrónicos, los cuales el microcontrolador procesa y determina la dirección del giro y la cantidad de pulsos generados. De igual forma, este encoder cuenta con un botón interno, el cual se activa al presionar la perilla hacia abajo. Este dispositivo se seleccionó debido a su gran versatilidad, ya que permite desplazarse a través de los diferentes menús que contiene el sistema, así como poder variar los valores de estado en la máquina. Lo anterior gracias a que, gracias al giro que se genera en la perilla, el microcontrolador determinar si debe desplazarse hacia un sentido u otro en el menú, y al presionar su botón interno, se hace la selección de alguna opción, así como entrar o salir de un menú.

Como se observa en la figura 13, cuenta con 5 pines: 2 de alimentación, 2 donde se generan los pulsos y 1 para el botón interno.

### Figura 13

*Encoder rotativo KY-040*



*Nota.* Adaptado de *Sensor Encoder Rotatorio KY-040* [Figura]. Electronilab.

<https://electronilab.co/tienda/sensor-encoder-rotatorio-ky-040/>

Debido a que una de las funciones que debió cumplir el sistema, es el de poder enviar los comandos contenidos en un archivo de código G a la CNC, fue necesario agregar un lector de dispositivos de almacenamiento, ya sea para memorias SD, microSD o USB.

**Módulo para tarjeta microSD SPI.** En este aspecto no fue tan difícil la elección del módulo, debido a que no hay mucha variedad en el mercado, y la diferencia entre uno y el otro no son tan grandes. Principalmente se escogió este en concreto debido a que en la PCB ya cuenta con un regulador de voltaje a 3.3v, los cuales son necesarios para alimentar la tarjeta microSD. Cuenta con 2 pines de alimentación y 4 de comunicación.

En la figura 14 se observa el módulo microSD seleccionado.

**Figura 14.**

*Módulo microSD SPI*



*Nota.* Adaptado de *Modulo para MicroSD y MicroSDHC* [Figura]. Electronicos Caldas.

<https://www.electronicoscaldas.com/es/conectores-para-tarjetas-de-memoria-y-sim-card/930-modulo-para-memoria-micro-sd-y-micro-sdhc.html>

**Costos de los materiales.** En la tabla 4 se puede apreciar el costo final que tuvo el sistema.

**Tabla 4.**

*Costos*

Componente	Cantidad	Precio
Arduino IDE	1	\$0
Arduino MEGA	1	\$40460
LCD 2004 I2C	1	\$23800

Módulo microSD SPI	1	\$5355
Encoder rotativo con pulsador	1	\$2856
Pulsador	1	\$630
Cables, estaño y otros materiales	NA	\$7000
<b>Total</b>		<b>\$80101</b>

*Nota.* Precios tomados de <https://www.didacticaselectronicas.com/>

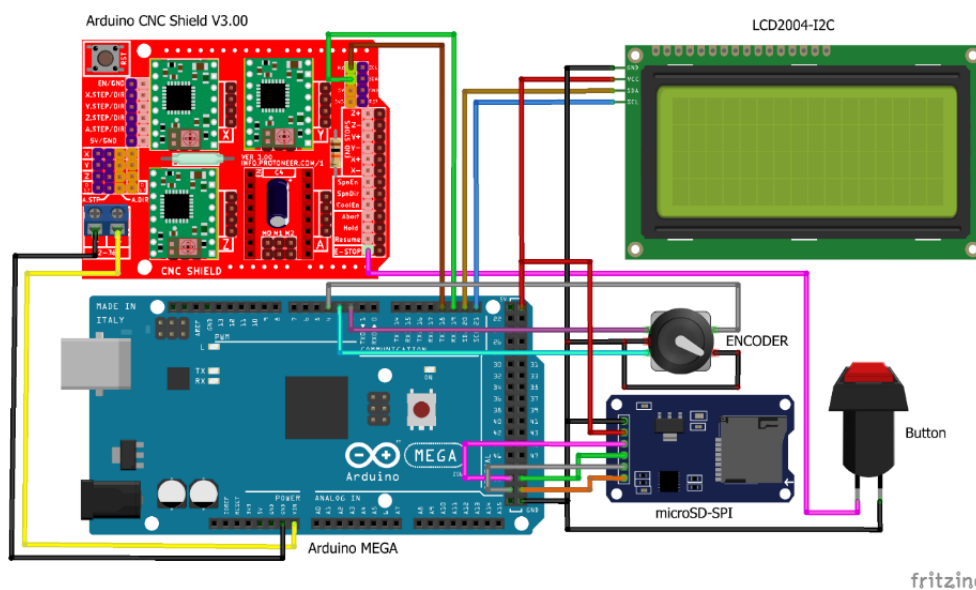
### Actividad 2. Diseño del plano esquemático de conexiones.

Una vez seleccionados los componentes electrónicos, se establecieron las líneas de conexión que permitirán la comunicación entre ellos. Este se diseñó de manera que fuera bastante fácil de interpretar por el usuario final que desee replicar este proyecto.

La figura 15 representa el plano de conexiones de los diferentes componentes.

### Figura 15.

#### Plano de conexiones

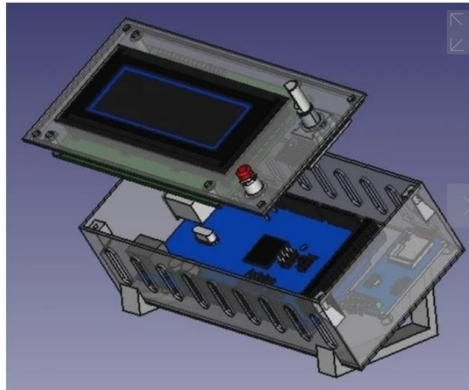


Aparte del plano esquemático, también se diseñó una caja para ensamblar el sistema, para así tenerlo más organizado y visualmente agradable. Este se realizó a través del software de

diseño 3D de Freecad, la cual puede ser impresa 3D. Quedando la caja como se observa en la figura 16.

**Figura 16.**

*Caja*



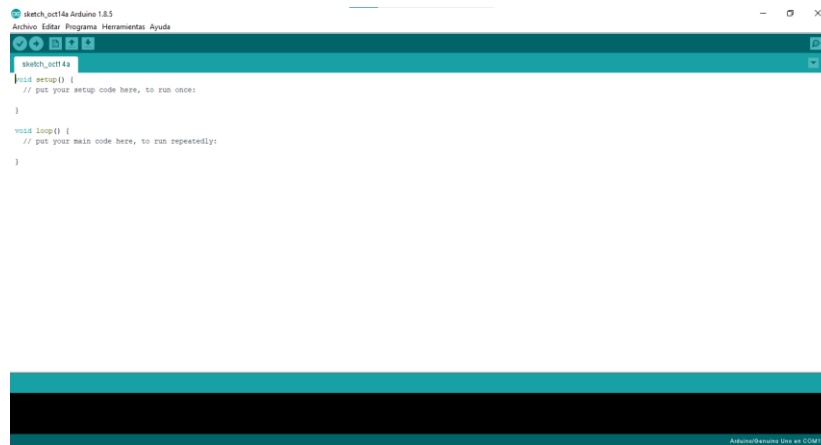
**Actividad 3.** Escribir el algoritmo que controlará al sistema embebido.

Este algoritmo, debido a que se usó una tarjeta Arduino, se desarrolló o escribió a través del entorno del mismo fabricante: Arduino IDE.

**Arduino IDE.** Es un entorno de desarrollo integrado (IDE) de Arduino es una aplicación multiplataforma (para Windows, macOS, Linux) que está escrita en el lenguaje de programación Java. Se utiliza para escribir y cargar programas en placas compatibles con Arduino. (Wikipedia, 2019). La figura 17 muestra la pantalla principal del software en mención.

## Figura 17.

### Arduino IDE



Los archivos pueden ser descargados y visualizados a través del siguiente enlace, donde se encuentra el código, diagramas de flujo de cada función y archivos de simulación.

[https://1drv.ms/f/s!AqYxPMAevUnJvjt1giANPd\\_dnCyT](https://1drv.ms/f/s!AqYxPMAevUnJvjt1giANPd_dnCyT)

### Fase 3. Validación

En la última fase se validó el funcionamiento de la solución propuesta en una maquina CNC. Dividiéndose en 2 actividades:

#### Actividad 1. Implementar la solución.

Una vez seleccionados los componentes y hecha la programación, se procedió a realizar la respectiva simulación del sistema. Con el fin de apreciar posibles fallas y/o errores en los parámetros, ya sea de hardware o software, previniendo así que, al conectar propiamente a una maquina CNC, genere un mal funcionamiento y/o daño en ella.

La visualización de la simulación, se puede observar a través del siguiente enlace, la cual se realizó con ayuda de Proteus.

<https://youtu.be/AmNcOBmoC54>

**Actividad 2.** Operar la implementación.

Después de simulada y verificada cada una a unas las funciones del sistema embebido, se procedió a conectarlo y operarlo con una maquina CNC, ejecutando cada una de las funciones programadas y realizando un mecanizado completo, verificando que no salte ninguna línea de código o realice alguna acción que entorpezca el proceso que se esté realizando en la CNC.

## Resultados

Como se ha mencionado a lo largo de este informe, el sistema se basa en un Arduino MEGA, una LCD2004, un módulo ENCODER rotatorio con pulsador y un módulo para lectura de tarjetas microSD.

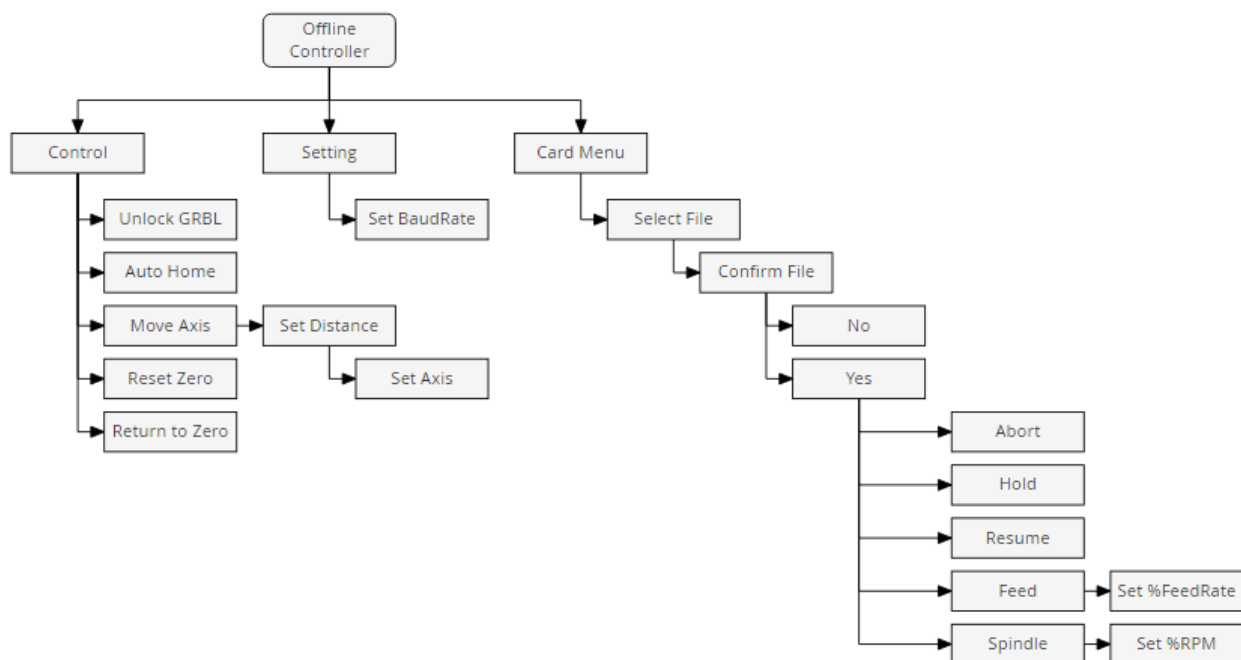
Al interconectar todos estos componentes, solo resta hacerlo con la CNC, lo cual se hace a través de 4 cables: 2 de alimentación (12V o 5V y GND) y 2 de comunicación (TX - RX).

Para entrar a menús y/o seleccionar opciones, se hace presionando el pulsador del encoder y para moverse a través de ellos, se gira el encoder hacia un lado u otro, dependiendo si se quiere subir o bajar.

Las opciones programadas y a través de que menús se pueden acceder a ellas, se pueden visualizar a continuación en la figura 18.

**Figura 18.**

*Árbol de funciones.*

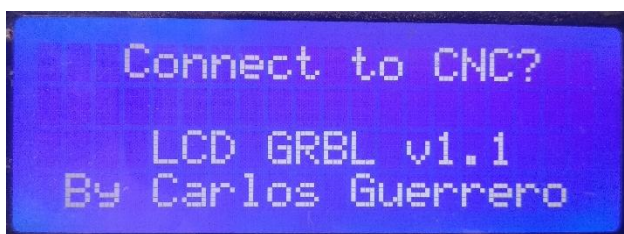


## Parámetros

Al encender el sistema, la primera opción que saldrá en la LCD será si se quiere o no conectarlo con la CNC (Figura 19), donde se debe entender que, aunque ya está conectado físicamente con los cables, no lo está por software, razón por la cual se debe realizar esta acción. Lo anterior con el fin de, si el usuario por un motivo u otro no quiere manipular la CNC con este sistema, no realice la conexión por software, así podrá conectar la CNC por cable USB al computador sin problemas.

### Figura 19.

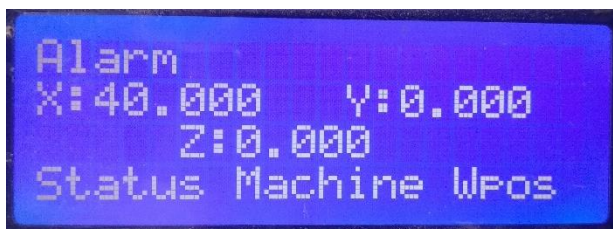
*Establecer conexión por software*



Posterior a esto, se podrá acceder a la vista principal en la LCD, en la cual se observa el estado general de la maquina: si esta está habilitada para recibir órdenes o en alarma, también se observa las coordenadas de la posición de la maquina en X, Y, Z (Figura 20).

### Figura 20.

*Pantalla "Estado de la maquina"*

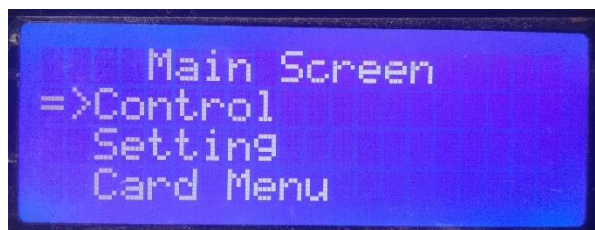


Al presionar el pulsador del encoder, se accede al menú principal (Figura 21), donde hay 3 opciones: control, configuración y menú de la tarjeta



**Figura 21.**

*Pantalla "Menú principal".*

**Menú Control.**

En esta sección están las opciones para manipular el comportamiento de la máquina, como se observa en la figura 22.

**Figura 22.**

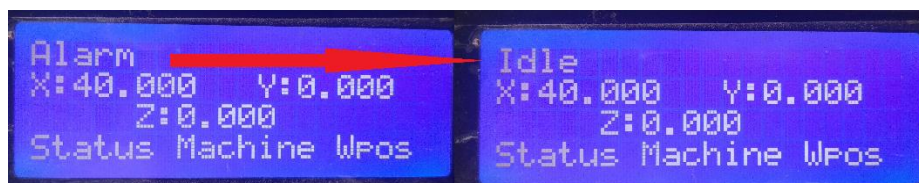
*Pantalla "Menú Control"*



**Desbloquear GRBL.** sirve para desactivar el estado de la máquina y así esta pueda recibir y cumplir las órdenes que posteriormente se le envíen. Pasando del estado "Alarm" a "Idle", como se observa en la figura 23.

**Figura 23.**

*Pantalla "Desbloquear GRBL"*



**Auto Home.** Si la maquina tiene sensores en los extremos inferiores de los ejes, hará que los ejes se muevan hasta que los sensores sean activados, es decir, la maquina buscará la posición de referencia (0,0,0).

**Mover ejes.** En esta opción, se puede establecer el intervalo de distancia que se desea mover el eje, ya sea cada 10mm, 1mm o 0.1mm a la vez. Después, se selecciona el eje que se debe mover. Y, por último, se llega a la pantalla en la cual se podrá observar que eje y cuanta distancia se moverá, así como la posición actual de la máquina, que se irá actualizando a medida que se mueva el eje. Las diferentes opciones se observan en la figura 24.

### Figura 24.

*Pantallas "Mover Eje"*



**Reestablecer cero.** Después de mover los ejes de la máquina y, si se quiere que esa posición actual sea el origen (0, 0, 0), entonces esta opción lo hace.

**Regresar a cero.** Si la posición actual es diferente a la del origen, entonces esta opción hace que los ejes se muevan hasta la posición (0, 0, 0) que esté almacenada. Esta opción es diferente a Auto Home, ya que esta busca la posición a través de los sensores, en cambio “regresar a cero” va hasta la posición (0, 0, 0) que ya esté configurada.

### ***Menú Configuración.***

En esta opción se configura los baudios para la comunicación con la maquina CNC, como se visualiza en la figura 25.

#### **Figura 25.**

*Pantalla "Configuración"*

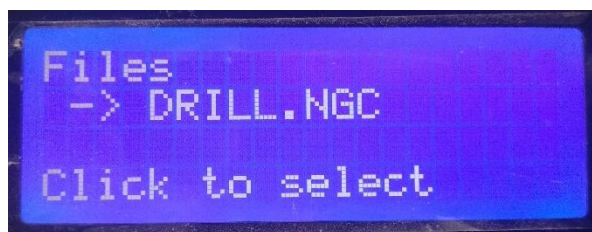


### ***Menú de la tarjeta microSD.***

En esta opción se muestran los archivos que están en la tarjeta micro SD de 1 a la vez (Figura 26).

#### **Figura 26.**

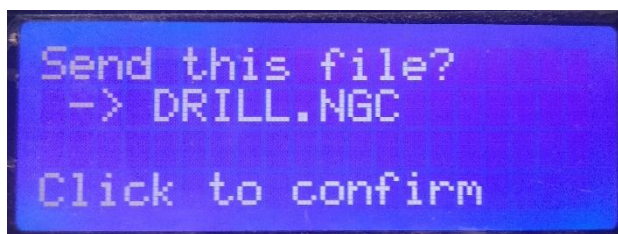
*Pantalla "Menú Tarjeta – Seleccionar Archivo"*



Después de seleccionar el archivo, el sistema le pedirá la confirmación para enviar los comandos a la CNC (Figura 27).

**Figura 27.**

*Pantalla "Menú Tarjeta – Confirmación de Archivo"*



Al confirmar y empezar a enviar los comandos a la CNC, saldrá una pantalla en la cual se puede observar el tiempo transcurrido desde que empezó el mecanizado, el porcentaje de comandos enviados/total de comandos en el archivo, posición actual de los ejes y el nombre del archivo que se está mecanizando (Figura 28).

**Figura 28.**

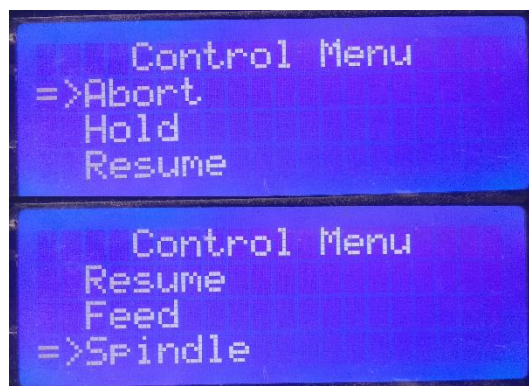
*Pantalla "Menú Tarjeta – Mecanizando"*



En esta pantalla, mientras se está realizando un mecanizado, se puede acceder a otro menú, uno con el que se puede modificar parámetros de la máquina que solo se tiene acceso cuando se está en movimiento (Figura 29).

**Figura 29.**

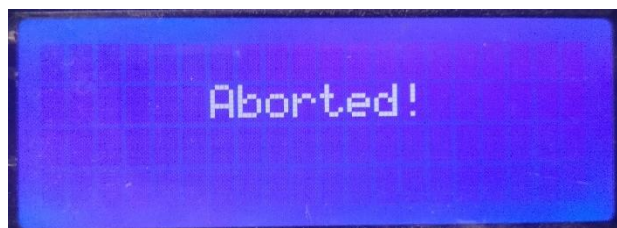
*Pantalla "Modificar en Mecanizado"*



**Abort.** Como su nombre lo indica, aborta/cancela el mecanizado que se esté realizando, al hacerlo saltará la pantalla de la figura 30.

**Figura 30.**

*Pantalla "Mecanizado Cancelado"*



**Hold.** Pausa el mecanizado.

**Resume.** Si el mecanizado está pausado, lo reanuda.

**Feed.** Si se desea modificar la velocidad con los que la maquina mueve los ejes, para hacer un mecanizado más rápido o más lento, se hace en esta opción. Donde el valor por defecto es al 100% de la velocidad, mínimo 10% y máximo 200%, como se observa en la figura 31.

**Figura 31.**

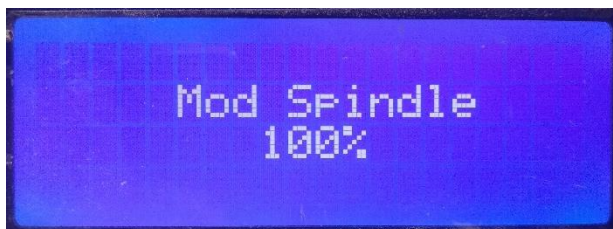
*Pantalla "Modificar Velocidad de Movimientos"*



**Spindle.** Al igual que la opción anterior, esta modifica la velocidad de giro del motor de corte. Donde el valor por defecto es al 100% de la velocidad, mínimo 10% y máximo 200%, como se visualiza en la figura 32.

**Figura 32.**

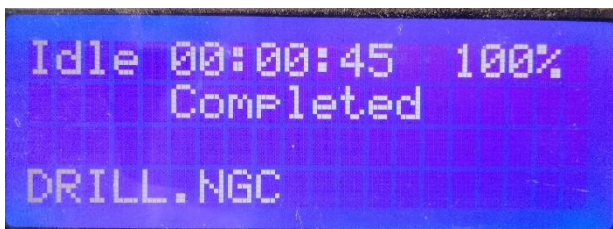
*Pantalla "Modificar Velocidad del Motor de Corte"*



Al finalizar el mecanizado, saldrá en pantalla el mensaje "Completado" (Figura 33), para salir de ella se debe presionar el pulsador. Y finalmente se regresa a la pantalla de Estado de la Maquina.

**Figura 33.**

*Pantalla "Mecanizado completado"*



## Implementación del controlador fuera de línea para maquinas CNC basadas en GRBL 1.1+

Para la implantación del sistema, se hizo uso de una maquina CNC con partes impresas en 3D y, necesariamente, ejecutando GRBL en su firmware. Dicha maquina es del tipo DIY, donde sus partes fueron diseñadas a través del software de Freecad, con ejes que se deslizan sobre varillas aceradas de 8mm, con motores nema 17 y tracción a través de tornillos sin fin de 8mm.

Al momento de ensamblar todo el sistema dentro de la caja diseñada, quedó como se observa en la figura 34.

### Figura 34.

*Sistema ensamblado.*



Las diferentes funciones, menús y puesta en marcha del Controlador Fuera de Línea, se puede observar a través del siguiente enlace:

<https://youtu.be/sHZ4-rndJ4I>

El proyecto se compartió en su totalidad como de código abierto, donde cualquier persona que lo desee, puede replicarlo y/o modificarlo a su gusto. Se puede acceder a él a través del siguiente enlace:

[https://github.com/cguerrero1205/LCD\\_GRBL](https://github.com/cguerrero1205/LCD_GRBL)

Aunque es difícil saber cuántas personas han llegado a aplicarlo, ya que ellas no están en la obligación de mencionar si lo hicieron o no, se puede saber cuántas han descargado los archivos de impresión 3D de la caja antes mencionada. Lo anterior se puede observar accediendo al siguiente enlace, página donde se compartió todos los archivos necesarios para la impresión 3D de dichas partes:

<https://www.thingiverse.com/thing:4354456>

A fecha de hoy en que escribo esta sección del informe (18/10/21), se han realizado 231 descargas de dichos archivos.

De igual forma, indagando en la web, se encontró un usuario de YouTube el cual replicó el proyecto. El anterior modificó el nombre inicial que se muestra al encender el sistema, pero el resto de funciones son exactamente las mismas. El usuario subió 2 videos: 1 de demostración y otro con la ejecución de un ejemplo de funcionamiento.

Se recomienda, si se va a observar los videos, se baje el volumen del computador.

<https://youtu.be/Qk2Sx1z2Zq4>

<https://youtu.be/1BGa6r9NG4o>



## **Plan de Mejoras a Corto y Mediano Plazo**

### ***Corto Plazo***

Añadir archivos de corte para realizar la caja en MDF, acrílico o el material que tenga a la mano el interesado en implementar el proyecto.

Añadir opción para el control de máquinas CNC que cuentan con LASER.

Añadir la opción de cambiar el lenguaje entre español e inglés.

### ***Mediano Plazo***

Diseñar una PCB en la cual se puedan montar y soldar todos los componentes, minimizando los cables de conexión y problemas por malas conexiones de los mismos. Además de minimizar el tamaño final de la caja.

Añadir la programación necesaria para la implementación de autonivelación desde el mismo sistema embebido.

Añadir un menú para modificar las configuraciones de GRBL desde el sistema embebido.

## Conclusiones

Los comandos identificados y caracterizados que fueron investigados pudieron ser usados de tal forma que, desde el sistema embebido, se puede controlar acciones de la maquina CNC, tales como búsqueda del punto de origen de los ejes, reestablecer posición de origen, etc.

El diseño y desarrollo ejecutado para el sistema, permite que el usuario final tenga fácil acceso a los componentes electrónicos, tanto por el costo como la facilidad en la interconexión de cada uno de ellos. Además de ejecutar las acciones con los componentes necesarios, de manera óptima y eficiente.

Al validar el funcionamiento a través de las simulaciones, se pudo evidenciar y corregir cualquier fallo que pusiera en riesgo la integridad física de la maquina CNC. Procediendo posteriormente a la operación física del sistema, conectándolo a una maquina CNC y verificando que cada una de las funciones programadas se ejecutara de manera adecuada.

Como mejoras o trabajos a futuro, se puede la añadir opción para el control de máquinas CNC que cuentan con LASER, así como añadir la programación necesaria para la implementación de autonivelación desde el mismo sistema embebido.

## Referencias

- Arduino. (2020). *Arduino Mega 2560 Rev3*. <https://store.arduino.cc/usa/mega-2560-r3>
- Autodesk. (2018). *CAD/CAM*. <https://latinoamerica.autodesk.com/solutions/cad-cam>
- Diaz del Castillo Rodriguez, F. (2008). *Programación automática de máquinas CNC*.  
[http://olimpia.cuautitlan2.unam.mx/pagina\\_ingenieria/mecanica/mat/mat\\_mec/m4/master\\_cam.pdf](http://olimpia.cuautitlan2.unam.mx/pagina_ingenieria/mecanica/mat/mat_mec/m4/master_cam.pdf)
- Empresas Públicas de Medellín. (2020). *Tarifas y Costo de Energía Eléctrica—Mercado Regulado—Septiembre*.  
[https://www.epm.com.co/site/Portals/2/Documentos/tarifas/energia2020/Publicacion\\_Tarifas\\_Energia\\_16\\_Septiembre\\_2020.pdf?ver=2020-09-16-132848-990](https://www.epm.com.co/site/Portals/2/Documentos/tarifas/energia2020/Publicacion_Tarifas_Energia_16_Septiembre_2020.pdf?ver=2020-09-16-132848-990)
- Gana Energía. (2018, noviembre 11). ¿Cuánto Consumo el Ordenador? *Gana Energía*.  
<https://ganaenergia.com/blog/que-consumo-nos-supone-utilizar-el-ordenador/>
- Jeon, S. (2021). *Grbl/grbl* [C]. Grbl CNC controller. <https://github.com/grbl/grbl> (Original work published 2009)
- Lekn Tech. (2021). *G-Sender*. <https://github.com/LEKN-TECH/G-Sender> (Original work published 2020)
- Makerbase. (2021). *MKS-TFT*. <https://github.com/makerbase-mks/MKS-TFT> (Original work published 2016)
- Schmid, S., & Kalpakjian, S. (2000). *Manufacturing Engineering and Technology* (4ta ed.). Prentice Hall.
- Silicon Circuits. (2021). *Grbl Controller* [Java]. <https://github.com/zeevy/grblcontroller> (Original work published 2017)

Venneker, B. (2021). *Bvenneker/Arduino-GCode-Sender* [Arduino].

<https://github.com/bvenneker/Arduino-GCode-Sender> (Original work published 2016)

Wikipedia. (2019). Arduino IDE. En *Wikipedia, la enciclopedia libre*.

[https://es.wikipedia.org/w/index.php?title=Arduino\\_IDE&oldid=119240592](https://es.wikipedia.org/w/index.php?title=Arduino_IDE&oldid=119240592)

Wikipedia. (2021). Hágalo usted mismo. En *Wikipedia, la enciclopedia libre*.

[https://es.wikipedia.org/w/index.php?title=H%C3%A1galo\\_usted\\_mismo&oldid=137143](https://es.wikipedia.org/w/index.php?title=H%C3%A1galo_usted_mismo&oldid=137143817)

817

## Anexos

Tabla 5.

*Parámetros de configuración de GRBL*

<b>Configuración y valores de ejemplo</b>	<b>Descripción</b>
\$0=10	Pulso de paso, microsegundos
\$1=25	Paso de retardo inactivo, milisegundos
\$2=0	Paso puerto invertido, máscara
\$3=0	Puerto de dirección invertido, máscara
\$4=0	Paso habilitar invertir, booleano
\$5=0	Limitar pines invertidos, booleanos
\$6=0	Pin de sonda invertido, booleano
\$10=1	Informe de estado, máscara
\$11=0.010	Desviación de la unión, mm
\$12=0.002	Tolerancia de arco, mm
\$13=0	Reporte pulgadas, booleano
\$20=0	Límites suaves, booleanos
\$21=0	Límites duros, booleanos
\$22=1	Ciclo de Homing, booleano
\$23=0	Homing dir invertir, máscara
\$24=25.000	Alimentación homing, mm/min
\$25=500.000	Búsqueda de homing, mm/min
\$26=250	Homing debounce, milisegundos
\$27=1.000	Homing pull-off, mm
\$30=1000.	Velocidad máxima del husillo, RPM
\$31=0.	Velocidad mínima del husillo, RPM
\$32=0	Modo láser, booleano

\$100=250.000	X pasos/mm
\$101=250.000	Pasos Y/mm
\$102=250.000	Pasos Z/mm
\$110=500.000	X Velocidad máxima, mm/min
\$111=500.000	Y Velocidad máxima, mm/min
\$112=500.000	Velocidad máxima Z, mm/min
\$120=10.000	X Aceleración, mm/seg <sup>2</sup>
\$121=10.000	Aceleración Y, mm/seg <sup>2</sup>
\$122=10.000	Aceleración Z, mm/seg <sup>2</sup>
\$130=200.000	X Recorrido máximo, mm
\$131=200.000	Y Max recorrido, mm
\$132=200.000	Z Max recorrido, mm

---

*Nota.* Adaptado de *Configuración de Grbl* [Tabla]. Github. [Configuración grbl v1.1 · gnea/grbl](#)

[Wiki · GitHub](#)