Logotipo

Descripción generada automáticamente

**Plataforma dibujante con reconocimiento de imágenes**

**Informe de avance 2**

Grupo 2

Caciani Toniolo, Melina 02866/1

Chanquía, Joaquín 02887/7

Ollier, Gabriel 02958/4

UNLP

Facultad de Ingeniería

Departamento de Electrotecnia

Taller de Proyecto I (E0306)

31 de octubre de 2024

# ÍNDICE

[INTRODUCCIÓN 3](#_Toc181103782)

[DISEÑO DE LA PCB 3](#_Toc181103783)

[COMPONENTES 3](#_Toc181103784)

[DESARROLLO MECÁNICO?? 5](#_Toc181103785)

[DESARROLLO DE SOFTWARE 6](#_Toc181103786)

[ARQUITECTURA DEL FIRMWARE 6](#_Toc181103787)

[MODULARIZACIÓN 6](#_Toc181103788)

[ENSAYOS 8](#_Toc181103789)

[MÓDULOS DE ELECTRÓNICA 8](#_Toc181103790)

[COMPONENTES MECÁNICOS 8](#_Toc181103791)

[PARTES IMPRESAS 9](#_Toc181103792)

[CRONOGRAMA PRELIMINAR 9](#_Toc181103793)

[BIBLIOGRAFÍA 10](#_Toc181103794)

# INTRODUCCIÓN

Incluya como mínimo en el informe una introducción resumiendo el contenido del mismo, distintas secciones explicando los avances en el desarrollo de hardware, software y la parte mecánica del proyecto, fundamentando las decisiones del diseño y una sección con ensayos o validaciones realizadas hasta el momento. Además, deberá incluir como anexos el cronograma ajustado según el estado de avance, las tareas individuales realizadas hasta la fecha con las horas invertidas por cada integrante hasta el momento y las tareas a futuro de cada uno.

El presente informe detalla los avances en el desarrollo de un robot capaz de dibujar figuras geométricas básicas y reconocer dibujos de dichas figuras. Además, como objetivo secundario, el robot será capaz de jugar al ta-te-ti, permitiendo una experiencia más entretenida.

En este informe se enumeran las modificaciones decididas relacionadas al diseño del robot, las pruebas realizadas y el diseño de la PCB

# DISEÑO DE LA PCB

Para el diseño de la PCB se comenzó utilizando uno de los ponchos predefinidos que se pueden encontrar en la página del proyecto CIAA:

<https://www.proyecto-ciaa.com.ar/devwiki/doku.php%3Fid=desarrollo:edu-ciaa:ponchos_modelos.html>

De los distintos modelos de ponchos que están subidos al repositorio ahí mencionado se utilizó inicialmente el poncho grande ya que este cubría la totalidad de la CIAA y utilizaba los cuatro tornillos de fijación ubicados en la misma.

Además de la plantilla del poncho se utilizaron las huellas de cada componente que será ubicado en la misma:

Conectores: Para el resto de los componentes, al no estar ubicados en el poncho se utilizaron conectores de diversa cantidad de pads dependientes de su uso. En el diagrama esquematico se utilizaron los símbolos “Screw\_Terminal\_01x0X” (en X va la cantidad de pines necesarios). Como este símbolo no contaba con una huella predefinida se utilizaron unas huellas encontradas en un repositorio de github que implementaba la huella para la inserción de borneras de 2, 3 y 4 pines:

<https://github.com/MrChunckuee-Electronics/KiCad_Libraries/tree/master> Interfaz de usuario gráfica

Descripción generada automáticamente

DRV8825: Para cada controlador se utilizó una huella ya predefinida en kicad con la ubicación de cada uno de los pines.

Forma

Descripción generada automáticamente

{

Capacitor: Para la conexión de la alimentación de los motores paso a paso se ubico un capacitor entre las pistas de tensión de motor y la malla de GND, la huella utilizada se encuentra en las librerías por defecto de kicad.

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja

Luego de agregar huellas para cada componente se conectaron los pad necesarios para cumplir la conexión dada por el diagrama esquemático del proyecto y siguiendo las pautas dadas por la catedra se llego al siguiente diseño final:

Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza media

El diseño esquematico tambien sufrió modificaciones a lo largo del diseño de la placa PCB como la adición de los conectores en lugar de la conexión directa entre la CIAA y el servo o la ESP32. Y el cambio del uso de varios pines por comodidad para la conexión.

Diagrama, Esquemático

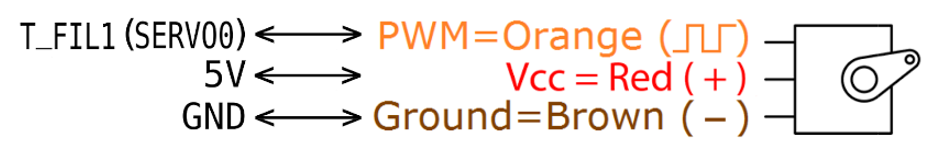
Descripción generada automáticamente

**DESARROLLO DE HARDWARE**

Esta etapa se utilizó para la prueba y familiarización de los componentes a utilizar donde

SERVO

Este dispositivo presenta una conexión simple con la placa, ya que solo necesita un canal de información y su respectiva alimentación, como se ve en la siguiente imagen.



ESP CAM

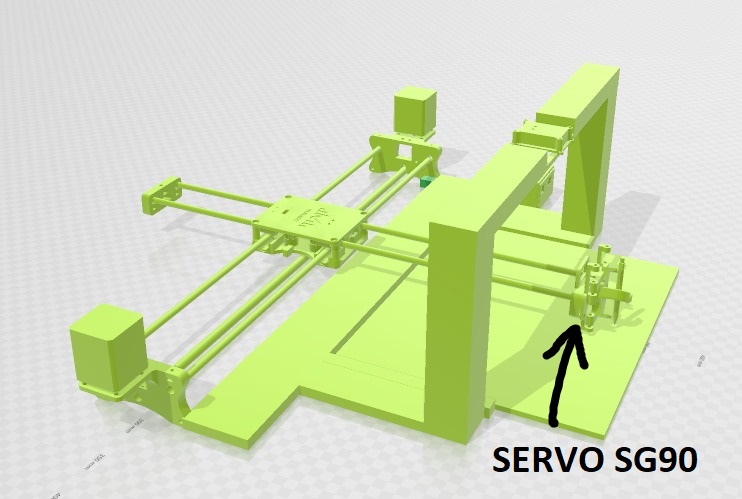
CONTROLADOR DRV8825 – MOTORES

En el siguiente diseño esquemático se muestran las conexiones necesarias para el funcionamiento de la plataforma dibujante. En la parte inferior izquierda se encuentran las dos tiras de pines correspondientes a la EDU-CIAA, por encima de esta se encuentran un servo y la ESP32-CAM. A la derecha se encuentran ambos controladores de motor DRV8825 y conexiones para poder conectar cada uno de los motores.



# DESARROLLO MECÁNICO??

A continuación, se presenta el modelo de la estructura del sistema. Puede llegar a sufrir modificaciones a lo largo de su desarrollo, pero este es el diseño preliminar ideado.Diagrama

Descripción generada automáticamente

# DESARROLLO DE SOFTWARE

## ARQUITECTURA DEL FIRMWARE

Se eligió una arquitectura **RTOS** (Sistema Operativo en Tiempo Real). Esta arquitectura permitirá gestionar mejor las tareas que ocurren simultáneamente, ya que su implementación trae ventajas como la priorización de tareas, escalabilidad, concurrencia de tareas, manejo eficiente de interrupciones y temporización.

## MODULARIZACIÓN

El software se organizará en módulos que aseguran que cada función del sistema pueda desarrollarse, probarse y mantenerse de manera independiente. Esto promueve una alta cohesión dentro de cada módulo y minimiza la dependencia entre ellos, favoreciendo la reutilización y el escalamiento del proyecto. Además de los módulos enumerados a continuación, obviamente existirá un módulo integrador (main) que coordine la interacción entre los módulos y dirija el flujo de trabajo.

* **RECONOCIMIENTO DE IMÁGENES**

Este módulo procesará las imágenes capturadas por la ESP32-CAM para identificar figuras geométricas básicas. Se utilizará la plataforma de Arduino, junto con sus librerías, para facilitar el procesamiento de las imágenes. Se obtendrán las imágenes mediante las librerías del ESP32-CAM, y se procesarán utilizando un algoritmo de inteligencia artificial de [Edge Impulse](https://edgeimpulse.com/), lo que permite reconocer la figura geométrica. El resultado obtenido se comunicará a la EDU-CIAA para su posterior procesamiento. El enfoque de este módulo será el área a la que se le dedicará más esfuerzo en este proyecto.

* **COMUNICACIÓN**

Este módulo manejará la comunicación entre los dos microcontroladores, enviará las imágenes capturadas al módulo de reconocimiento de imágenes para el procesamiento. Utilizará protocolo UART para enviar y recibir datos de manera eficiente.

* **CONTROL DE MOTORES**

Este módulo se encargará de manejar el movimiento de los motores paso a paso, los cuales controlan los ejes del robot que dibuja. Su función es convertir las instrucciones de dibujo, que provienen del sistema de reconocimiento de imágenes, en órdenes precisas para los motores, determinando tanto la cantidad de pasos como la dirección en la que deben moverse.

* **INTERFAZ DE USUARIO**

Este módulo se encargará de la interacción con el usuario para poder definir el modo de funcionamiento o para indicarle al jugador el siguiente paso que debe realizar. Aunque aún no se ha definido completamente cómo se implementará la interfaz, se está evaluando la posibilidad de utilizar la aplicación [MIT App Inventor](https://appinventor.mit.edu/) en conjunto con la conectividad Bluetooth del módulo ESP32. Esto permitiría que el usuario controle el robot desde un dispositivo móvil de manera inalámbrica, facilitando una experiencia de uso intuitiva y sin cables. De todas formas, esto no está completamente definido ya que no es una prioridad del proyecto, pero se contempla esta posibilidad.

* **JUEGO**

Este módulo tendrá la lógica del juego TA-TE-TI, que interactúa tanto con el sistema de control de motores como con el de reconocimiento de imágenes. El juego se controla a partir de las imágenes detectadas por la ESP32-CAM y ejecuta movimientos de acuerdo con las reglas del juego.

* **MODO DE PRUEBA**

Este módulo contendrá la lógica de la función de testeo de funcionamiento, en la cual se pide al jugador dibujar una figura en una sección de la hoja para posteriormente ser copiada por el robot dibujante.

# ENSAYOS

A continuación se enumeran los elementos necesarios para llevar a cabo este proyecto, obviando materiales como tornillos, tuercas, cables, resistencias, capacitores, etc.

### MÓDULOS DE ELECTRÓNICA

* 1 EDU-CIAA-NXP
* 1 ESP-32 CAM
* 2 DRV8825
* 2 Motor NEMA17 Double
* 1 Servo SG90

### COMPONENTES MECÁNICOS

* 2x varilla de rodamiento lineal M8 x 450 mm (eje X)
* 2x varilla de rodamiento lineal M8 x 350 mm (eje Y)
* 2x varilla lineal M3 x 75 mm (eje Z)
* 1x varilla roscada M8 x 480 mm (soporte)
* 8x rodamientos lineales LM8UU
* 2x polea GT2 16 dientes
* 1x correa GT2 de 2000 mm
* 5x rodamientos 624zz
* 4x idler\_18x13x6\_01.stl

### PARTES IMPRESAS

* Estructura eje Y delantero
* Estructura eje Y trasero
* Estructura eje X delantero
* Estructura eje X trasero
* Soporte lapicera
* Eje central superior
* Eje central inferior
* Soporte motor (x2)

# CRONOGRAMA PRELIMINAR



diagrama de Gantt

# BIBLIOGRAFÍA

* CIAA: <https://www.proyecto-ciaa.com.ar/index.html>
* Estructura robot dibujante: <https://www.thingiverse.com/thing:2349232>
* Edge impulse: <https://edgeimpulse.com/>
* MIT App Inventor: <https://appinventor.mit.edu/>

**Especificaciones Técnicas**

* [ESP32](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)
* [DRV8825](https://www.alldatasheet.com/html-pdf/432263/TI/DRV8825/24/1/DRV8825.html)
* [Motor Nema 17](https://pages.pbclinear.com/rs/909-BFY-775/images/Data-Sheet-Stepper-Motor-Support.pdf)
* [Servo SG90](https://www.alldatasheet.es/html-pdf/1572383/ETC/SG90/59/1/SG90.html)

**Herramientas utilizadas**

* [KiCad](https://www.kicad.org/)
* [3D Builder](https://apps.microsoft.com/detail/9wzdncrfj3t6?hl=en-us&gl=US)