



# TRABAJO PRÁCTICO

Brazo Robot

Proyecto con Microcontroladores 1  
Prof. PhD. Vicente González

Elias Alvarez  
[eliasdavidalvarez@gmail.com](mailto:eliasdavidalvarez@gmail.com)

## Índice

Introducción.....	1
Contexto del proyecto .....	1
Objetivo General .....	1
Objetivos Específicos .....	1
Justificación .....	1
Análisis del Sistema .....	3
Partes en las que se divide el sistema y su funcionalidad .....	3
Brazo robótico electromecánico .....	3
Interfaz de Usuario .....	3
Sistema de ingreso de posiciones de los motores .....	3
Medio de almacenamiento .....	3
Relación entre las partes .....	3
Diseño del sistema .....	4
Diseño de hardware.....	4
Control de movimiento .....	4
Interfaz de usuario .....	4
Sensores y acondicionamiento de señal.....	4
Sincronización y control .....	4
Diseño 3D.....	4
Diseño de software.....	5
Comunicación entre PSoC y Arduino .....	5
Interfaz de usuario .....	5
Control de movimiento .....	6
Estados del sistema .....	6
Implementación .....	7
Construcción del prototipo .....	7
Programación del microprocesador.....	8
Pruebas de hardware y software .....	8

Resultados .....	8
Análisis de los resultados obtenidos .....	8
Comparativa con objetivos iniciales .....	9
Evaluación del desempeño .....	9
Logros alcanzados.....	9
Limitaciones del proyecto .....	9
Aportes a la investigación .....	9
Conclusiones.....	10
Resumen de los resultados .....	10
Logros alcanzados.....	10
Limitaciones del proyecto .....	10
Aportes a la investigación .....	10
Recomendaciones .....	10
Sugerencias para futuros trabajos .....	10
Mejoras al sistema .....	10
Anexo.....	12
Hipervínculos .....	12

# Introducción

## Contexto del proyecto

Este proyecto de brazo robótico se inspira en una versión inicial realizada en la escuela utilizando Arduino. Con los conocimientos adquiridos en la universidad, el objetivo es mejorar y expandir esa idea. El brazo está diseñado para ejecutar comandos en tiempo real y almacenar patrones de movimiento aprendidos para su reproducción en bucle. Este modelo de brazo robótico a pequeña escala se controla con un PSoC, aprovechando sus periféricos y capacidades de procesamiento para lograr los movimientos y funcionalidades requeridas.

## Objetivo General

Desarrollar un brazo robótico controlado mediante PSoC, integrando una interfaz de usuario y almacenamiento de movimientos para permitir tanto el seguimiento en tiempo real de comandos como la repetición de secuencias previamente aprendidas.

## Objetivos Específicos

1. Crear un brazo con movimientos en base, codo, muñeca y pinza.
2. Diseñar una interfaz que permita al usuario ingresar comandos.
3. Incorporar un Display y otros medios visuales que comuniquen al usuario el estado del sistema.
4. Configurar y utilizar los periféricos del PSoC para lograr una comunicación eficiente entre las partes del sistema.
5. Implementar una funcionalidad de almacenamiento de patrones de movimiento que permita al brazo aprender y reproducir secuencias en bucle.
6. Ajustar el control del brazo para que opere en la zona lineal de los potenciómetros, maximizando la precisión de los movimientos.
7. Almacenar los movimientos aprendidos en la EEPROM, evitando que se pierdan en caso de una interrupción de energía.

## Justificación

La robótica tiene aplicaciones cruciales en la automatización de tareas repetitivas y precisas, así como en la investigación y educación. Este proyecto de brazo robótico permite experimentar con el control de servomecanismos, los cuales simplifican la implementación en comparación con otras opciones. La capacidad de aprender y reproducir movimientos emula bien un proceso de automatización, mientras que el modo de seguimiento en tiempo

real simula entornos de robótica asistencial. Esto permite profundizar en los principios básicos de la robótica, a la vez que mejora las habilidades prácticas en el diseño y programación de sistemas embebidos.

# Análisis del Sistema

## Partes en las que se divide el sistema y su funcionalidad

### Brazo robótico electromecánico

Ejecuta los movimientos mediante la rotación controlada de los motores en cada articulación (base, codo, muñeca y pinza), permitiendo que el brazo alcance las posiciones deseadas.

### Interfaz de Usuario

Permite al usuario ingresar comandos y visualizar mensajes del sistema, facilitando una interacción intuitiva y en tiempo real para configurar y controlar el brazo robótico.

### Sistema de ingreso de posiciones de los motores

Recibe las posiciones específicas a las que deben moverse los motores, permitiendo que el usuario controle con precisión los movimientos del brazo.

### Medio de almacenamiento

Guarda de forma segura los patrones de movimiento aprendidos en memoria no volátil, para que no se pierdan en caso de apagado, permitiendo su reproducción en cualquier momento.

### Relación entre las partes

Todas las partes están conectadas y coordinadas a través del PSoC, que actúa como controlador central. La interfaz envía comandos al PSoC, que los interpreta y gestiona, asegurando que los motores se ajusten según las posiciones deseadas y que los movimientos se guarden para futuras ejecuciones.

# Diseño del sistema

## Diseño de hardware

El sistema del brazo robótico combina diversos componentes electrónicos y mecánicos para garantizar su funcionalidad. Los elementos principales son:

### Control de movimiento

El movimiento del brazo se logra mediante servomotores, los cuales se controlan con pulsos PWM generados por el PSoC. Este enfoque permite una integración eficiente y precisa gracias a las capacidades nativas de PWM del microcontrolador.

### Interfaz de usuario

1. **LCD RGB:** Se utiliza una pantalla LCD RGB para mostrar instrucciones y estados del sistema. La comunicación entre la LCD y el PSoC se realiza mediante el protocolo I2C.
2. **OLED:** Complementa al LCD con animaciones que representan visualmente el estado actual del sistema o indican errores mediante bitmaps almacenados en una tarjeta SD. Para su control, se utiliza un Arduino Mega conectado a la OLED mediante I2C y a la SD mediante SPI.
3. **Teclado matricial:** Permite ingresar comandos al sistema. El teclado incluye diodos para el barrido y se conecta al PSoC.

### Sensores y acondicionamiento de señal

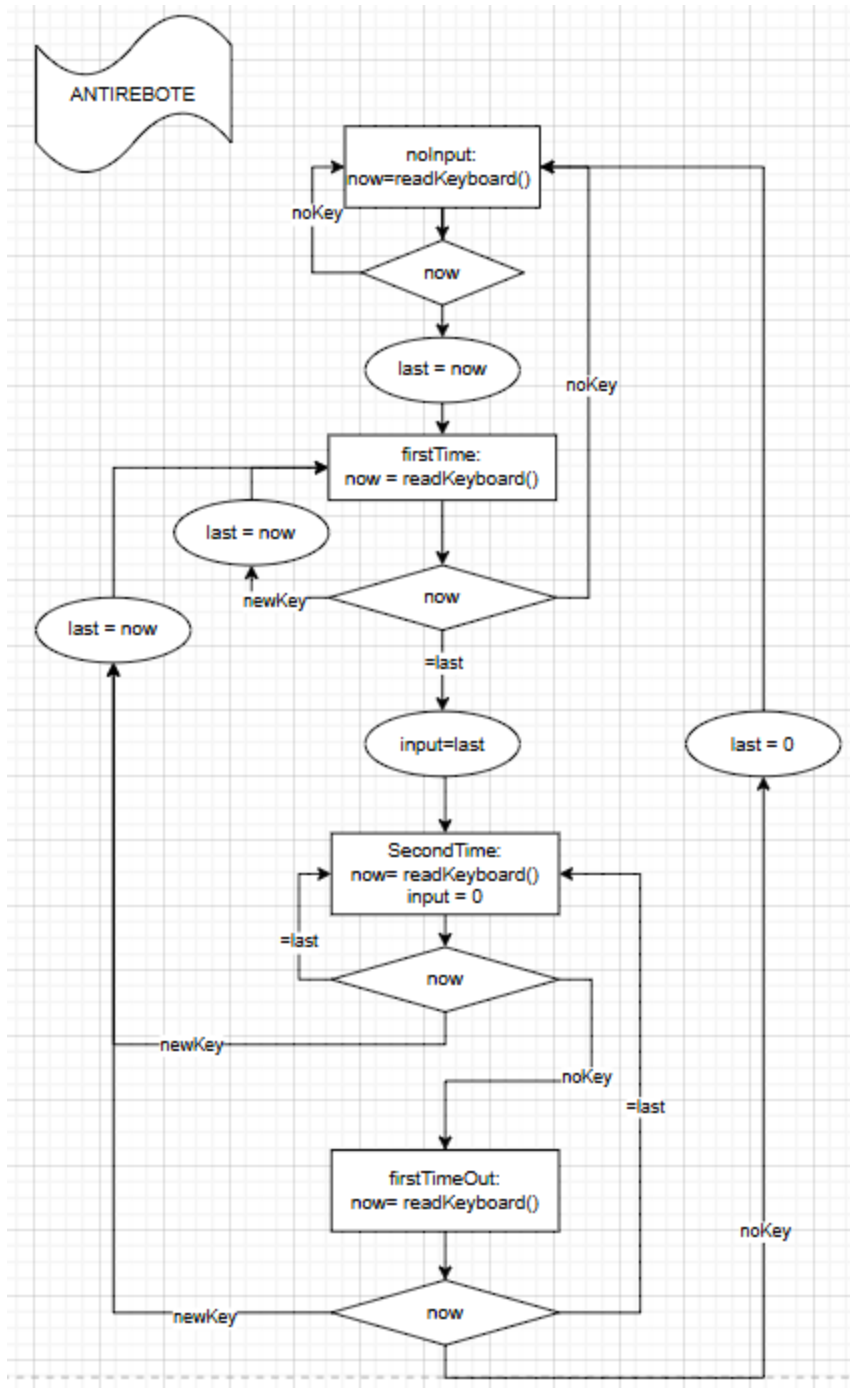
Los potenciómetros lineales, usados para medir los ángulos de las articulaciones, están conectados al ADC-SAR del PSoC a través de un MUX. Antes del MUX, se incluye un filtro pasivo pasa bajas para reducir el ruido y un amplificador operacional en configuración buffer para minimizar errores relacionados con la alta impedancia de entrada del ADC.

### Sincronización y control

El sistema utiliza timers y registros de estado para sincronizar las acciones. Las interrupciones se emplean para tareas prioritarias, mientras que los registros sticky permiten manejar eventos no prioritarios de manera eficiente.

### Diseño 3D

Se hizo uso del programa FUSION 360 para diseñar piezas en modelado 3D que permitan la síntesis del sistema en su totalidad.



## Diseño de software

El diseño de software se organiza en módulos para cada funcionalidad:

### Comunicación entre PSoC y Arduino

El PSoC envía instrucciones al Arduino mediante UART utilizando un protocolo de confirmación:

1. El PSoC envía un mensaje y activa un temporizador.
2. El Arduino responde con el mismo mensaje.
3. Si el PSoC no recibe o detecta errores en la respuesta, reenvía el mensaje hasta que la transmisión sea exitosa o alcance un límite de intentos.

### Interfaz de usuario

1. El teclado se maneja mediante interrupciones periódicas controladas por un temporizador que activa

la máquina de estados correspondiente al mismo para evitar rebotes.

2. La LCD RGB utiliza una versión adaptada de la librería Adafruit, mostrando información del estado y respondiendo a las entradas del usuario.



3. La OLED, controlada por el Arduino, muestra animaciones almacenadas en formato bitmap y gráficas en tiempo real, como barras de progreso basadas en las posiciones de los potenciómetros.

## Control de movimiento

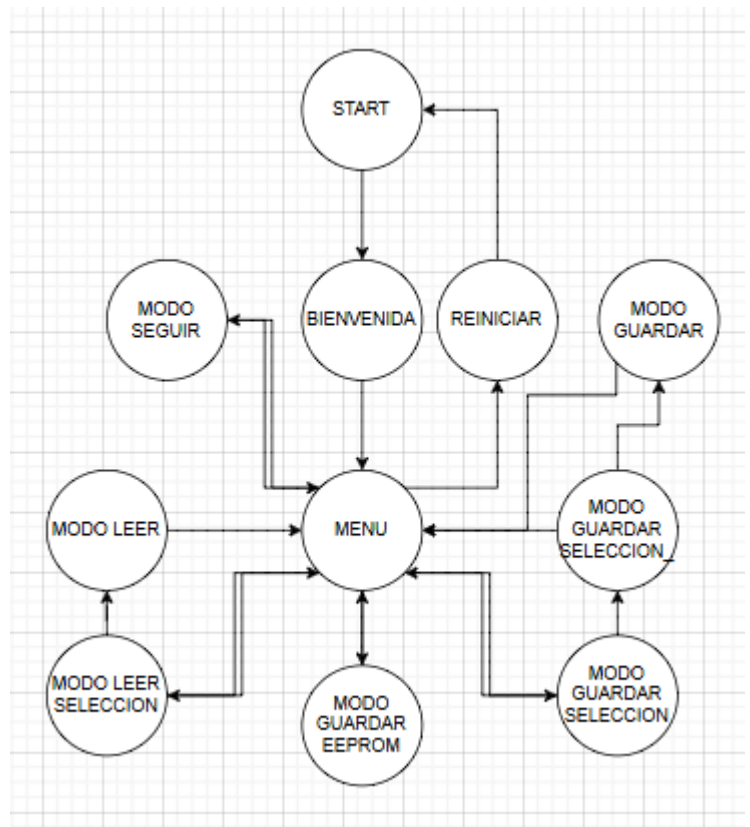
Los datos de los potenciómetros se ajustan para operar solo en su zona lineal. Los valores son procesados por el PSoC para generar los pulsos PWM necesarios para mover los servomotores.

- Se utiliza una curva de ajuste con umbrales para evitar la zona no lineal.
- Opcionalmente, se aplica un filtro de media móvil con tres niveles configurables.

## Estados del sistema

El sistema opera bajo una máquina de estados con transiciones controladas por las entradas del usuario:

- **START:** Inicialización y reinicio de componentes. Con ENT pasamos a BIENVENIDA
- **BIENVENIDA:** Espera hasta que el sistema esté listo. Pasa luego de un tiempo a MENU.
- **MENU:** Permite al usuario activar o desactivar funciones, cambiar configuraciones, o elegir modos de operación. En este modo, además de lo que denota la pantalla LCD, podemos con \* forzar el reinicio, con UP\_ARROW y DOWN\_ARROW aumentar o disminuir la media móvil y cambiar entre grados o porcentaje en la pantalla OLED con los botones LEFT\_ARROW y RIGHT\_ARROW respectivamente.
- **MODOS DE OPERACIÓN:** El sistema puede operar en varios modos de operación, cada uno con sus propias subfunciones:
  - **MODOS DE SEGUIR:** Seleccionados desde MENU, permiten controlar los motores en tiempo real. Seleccionados desde MENU, se activa el modo SEGUIR y se muestra la pantalla de configuración de los motores.
  - **MODOS DE LEER:** Seleccionados desde MENU, permiten leer los valores de los sensores. Seleccionados desde MENU, se activa el modo LEER y se muestra la pantalla de configuración de los sensores.
  - **MODOS DE GUARDAR:** Seleccionados desde MENU, permiten guardar los valores de los sensores en la memoria. Seleccionados desde MENU, se activa el modo GUARDAR y se muestra la pantalla de configuración de los sensores.



- **MODO\_GUARDAR:** Guarda posiciones en un vector con muestras periódicas o manuales.
- **MODO\_LEER:** Ejecuta en bucle uno de los vectores guardados.
- **MODO\_GUARDAR\_EEPROM:** Permite cargar o guardar datos en la EEPROM.
- **MODO\_LEER\_SELECCION** y **MODO\_GUARDAR\_SELECCION:** Selección del vector a leer o guardar.
- **MODO\_GUARDAR\_SELECCION\_:** Se elige si guardar los puntos con muestreo periódico o aperiódico mediante la tecla ENT.
- **REINICIAR:** Reinicia el sistema.

El flujo principal del software combina la máquina de estados con tareas periódicas y priorizadas mediante timers e interrupciones.

## Implementación

### Construcción del prototipo

La implementación comenzó con programas de prueba diseñados para verificar el funcionamiento individual de cada componente. Estos programas permitieron ingresar parámetros mediante el teclado u otros medios para asegurar que cada elemento del sistema funcionara correctamente. Una vez validados los componentes, se desarrollaron seis versiones del sistema, integrando progresivamente las funcionalidades necesarias para alcanzar el objetivo final. La versión 5.5 resultó ser la más eficiente y confiable.

El hardware fue montado en una Protoboard pequeña para conectar los componentes principales mediante jumpers y cables unifilares y una Protoboard grande externa para dejar libre el PSoC para otros proyectos de la universidad. El diseño mecánico del brazo se realizó inicialmente en la facultad utilizando Fusion 360. Sin embargo, debido a problemas con los formatos de archivo, se utilizó una impresora 3D de terceros con filamento PLA, obteniendo buenos resultados en términos de costo y calidad. Los potenciómetros se soldaron y ensamblaron con tuercas a la estructura impresa, mientras que las piezas fueron atornilladas a los motores. Se realizaron adaptaciones a las aspas de los servomotores para optimizar su acople. Aunque el diseño fue verificado previamente en Fusion 360, se presentaron problemas con el volumen adicional generado por los cables, lo que requirió expandir la caja utilizando tornillos largos. Además, este diseño se hizo modificando uno de los proyectos de FABRI CREATORS, ajustando las dimensiones y alterando los modelos a necesidad. Se dejó espacio como para poner un modelo pequeño del brazo 3D y así hacer

el ingreso de los datos más natural y similar a un juguete, junto al código que permite la multiplexión de la señal para lograr switchear entre ese y las perillas, pero se descartó la idea por falta de tiempo y más potenciómetros.

## Programación del microprocesador

El software se estructuró en librerías específicas para cada componente, exceptuando el servo y el potenciómetro, que se programaron como un módulo conjunto. Estas librerías incluyen funciones de comunicación que modifican variables globales internas, evitando el uso de externs. Uno de los mayores desafíos fue la implementación de interrupciones, que, aunque funcionaban bien en programas de prueba individuales, causaron problemas en la integración final. Esto se resolvió mediante el uso de timers para sincronizar las operaciones.

Para evitar la zona no lineal de los potenciómetros, se realizaron pruebas basadas en sus datasheets, ajustando por software los valores capturados. De manera similar, se corrigieron imprecisiones en los pulsos PWM generados por el PSoC. Se realizaron ajustes adicionales en los temporizadores para minimizar los tiempos de inactividad sin comprometer la estabilidad del sistema. Esto fue especialmente importante en el manejo del LCD, que presentó problemas de actualización rápida, y en la comunicación con el Arduino, que se saturaba con mensajes en las primeras versiones.

## Pruebas de hardware y software

Las pruebas incluyeron la calibración de potenciómetros, la validación del rango de movimiento de los servomotores, y la evaluación de la estabilidad de las señales PWM y de los temporizadores. Cada componente fue probado individualmente y en conjunto, identificando problemas específicos como ruido en las señales de entrada al ADC y tiempos de respuesta en la comunicación entre dispositivos. Estos problemas se solucionaron con filtros pasivos y ajustes en el software.

## Resultados

### Análisis de los resultados obtenidos

El brazo robótico cumplió con todas las expectativas planteadas al inicio del proyecto. Su funcionamiento es confiable y permite evitar los problemas identificados en las versiones preliminares. Las funciones de seguimiento en tiempo real, almacenamiento de movimientos y reproducción en bucle se lograron con éxito, mostrando estabilidad y precisión.

## Comparativa con objetivos iniciales

Todos los objetivos planteados se cumplieron. Sin embargo, quedaron aspectos que podrían mejorarse, como el diseño 3D de la caja, añadir más peso a la base para ser más robusto y permitir movimientos rápidos, la implementación de un PCB, y la incorporación de filtros analógicos de mayor orden para reducir el ruido en las entradas del ADC. También sería posible optimizar la memoria almacenando directamente los valores del PWM en lugar de los datos de los potenciómetros, lo que reduciría el espacio necesario en la EEPROM. A todo esto se le puede añadir la implementación del modelo 3D de potenciómetros para el seguimiento del brazo junto a un botón para switchear entre este y las perillas, que no se llegó a culminar.

## Evaluación del desempeño

El sistema es rápido y estable, con movimientos precisos del brazo. Sin embargo, el filtro de media móvil ralentiza ligeramente el sistema al priorizar la estabilidad. Se identificaron límites relacionados con la fuerza de los motores y la capacidad de la fuente de alimentación, los cuales podrían solucionarse fácilmente con componentes de mayor capacidad.

## Logros alcanzados

El principal logro fue implementar un sistema robótico funcional utilizando una amplia variedad de dispositivos estudiados. Destacaron la estabilidad y velocidad del sistema, que mejoraron significativamente en comparación con las primeras versiones.

## Limitaciones del proyecto

Las principales limitaciones fueron las restricciones mecánicas y eléctricas, así como la dependencia de componentes básicos como el PLA y los motores de baja potencia. Estas limitaciones no impidieron el cumplimiento de los objetivos, pero restringen el desempeño del sistema.

## Aportes a la investigación

Este proyecto demuestra cómo implementar un sistema robótico funcional utilizando dispositivos y técnicas accesibles, lo que podría servir como base para proyectos similares en educación y prototipos de bajo costo.

## Conclusiones

### Resumen de los resultados

El brazo robótico cumplió con todos los objetivos propuestos, destacándose por su estabilidad y velocidad. El sistema logró integrar múltiples dispositivos y resolver problemas técnicos a través de un diseño iterativo.

### Logros alcanzados

El proyecto logró implementar un sistema robótico funcional, mejorando significativamente la estabilidad y el rendimiento en comparación con versiones anteriores.

### Limitaciones del proyecto

El uso de motores de baja potencia y una fuente limitada restringieron el alcance del sistema. También se identificaron problemas con el ruido en las entradas del ADC y la dependencia de ajustes por software.

### Aportes a la investigación

Este proyecto sirve como modelo para la implementación de sistemas robóticos educativos y de bajo costo, demostrando cómo integrar múltiples tecnologías para resolver problemas complejos.

## Recomendaciones

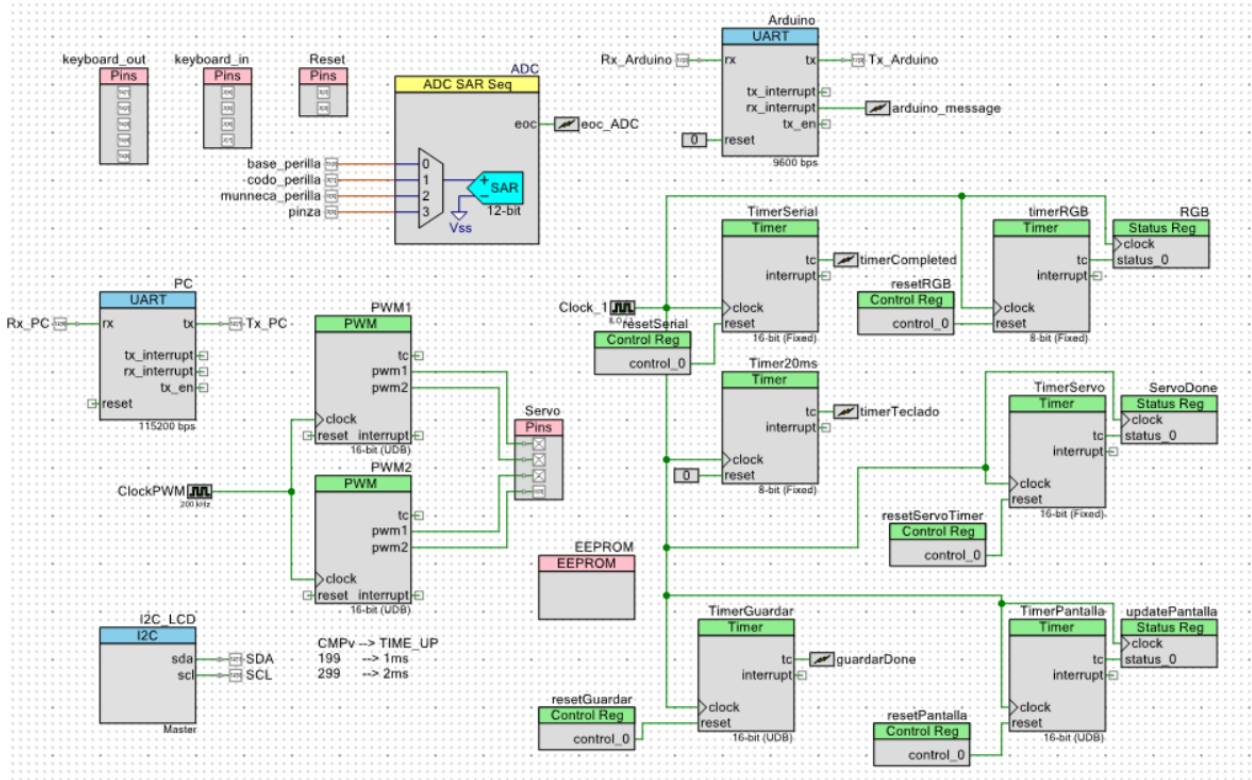
### Sugerencias para futuros trabajos

1. Probar las piezas y componentes a medida que se implementan su hardware y software en conjunto con las ya implementadas, para evitar problemas de integración tardía.
2. Implementar un PCB que reduzca las conexiones cableadas y mejore la confiabilidad del sistema.

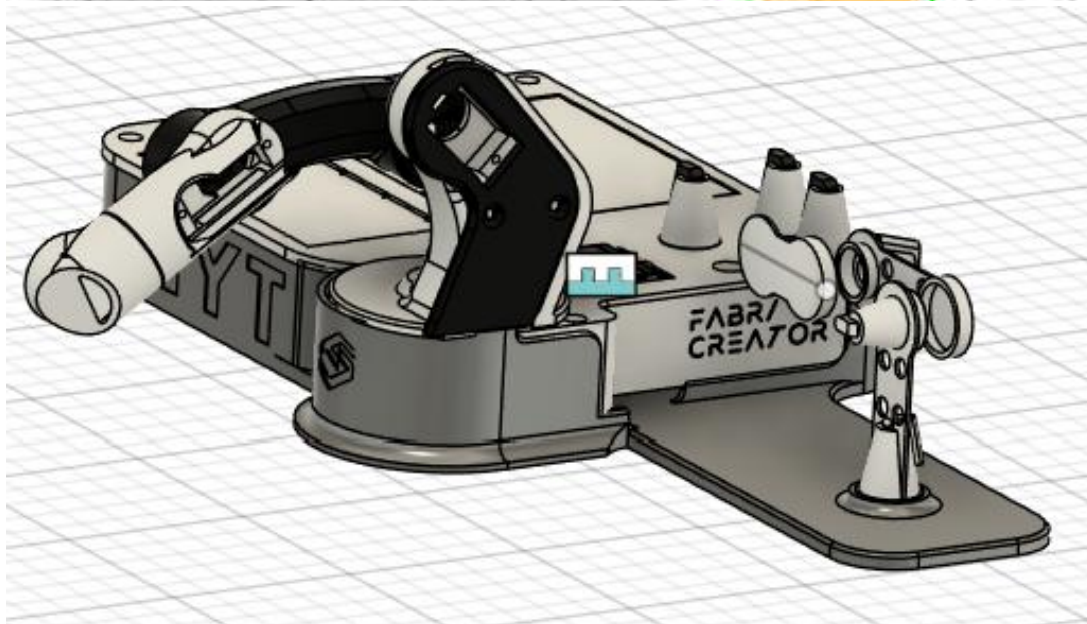
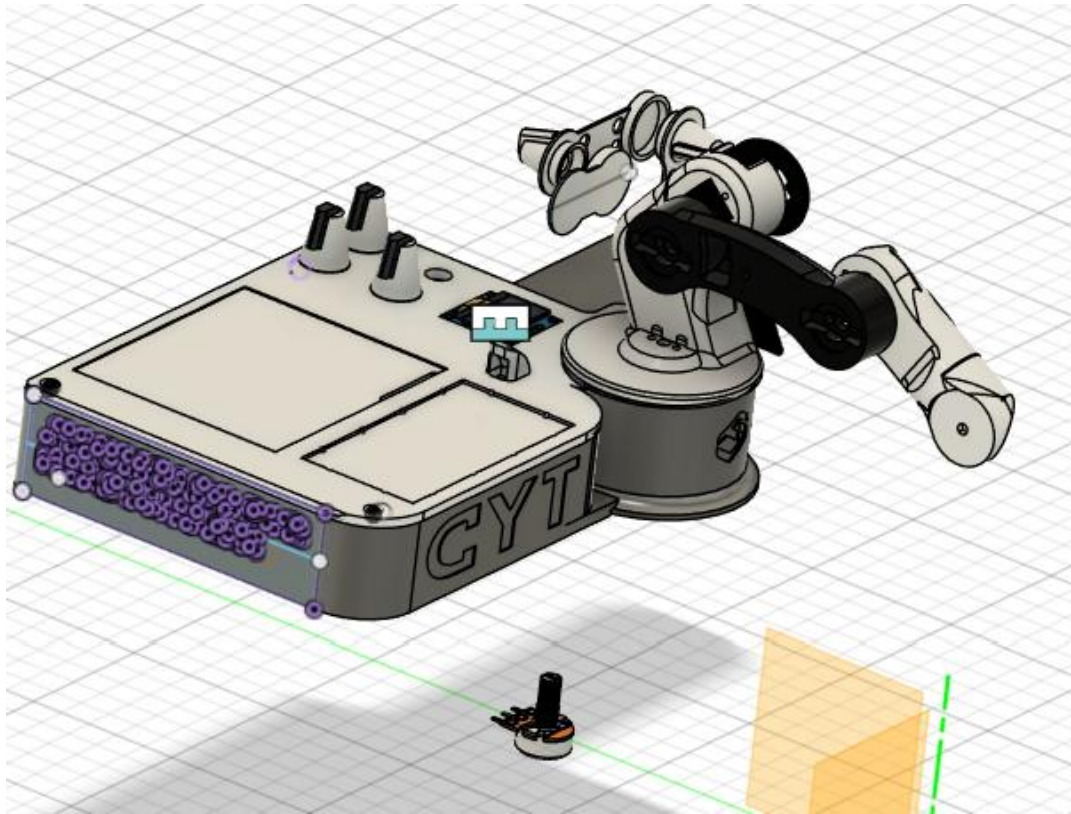
### Mejoras al sistema

1. Optimizar el almacenamiento, registrando valores de PWM en lugar de los de los potenciómetros.
2. Implementar interpolación para movimientos más suaves, permitiendo que al principio empiece acelerando suavemente y desaceleré cuando se acerca al punto destino.

3. Diseñar filtros analógicos de mayor orden para reducir el ruido en las señales del ADC.
4. Terminar de implementar el segundo brazo hecho de potenciómetros y el software para multiplexar su señal con la de las perillas en la caja.









# Hipervínculos

[Códigos utilizados para el Arduino.](#)

[Versiones de los programas del PSoC.](#)

[Diseño 3D.](#)

[Archivos varios utilizados en el proyecto.](#)