

Sistema Seguidor de Rostros

Proyecto final Sistemas de Control II

El Senado Galáctico



FACULTAD DE CIENCIAS EXACTAS, FÍSICAS y NATURALES

Universidad Nacional de Córdoba
Facultad de Ciencias Exactas, Físicas y Naturales
Escuela de Ingeniería Electrónica

21 de junio de 2023

Contenidos

- 1 Introducción
- 2 Planificación de las tareas
- 3 Análisis del hardware a utilizar
- 4 Reconocimiento facial
- 5 Herramientas utilizadas
- 6 Programación
- 7 Modelado del sistema
- 8 Implementación
- 9 Cierre

Introducción

La siguiente presentación resume el desarrollo del prototipo de seguidor de rostros implementado por el grupo **El Senado Galáctico**, conformado por los alumnos:

- Cruz, Enrique Luis Fernando
- Perez Perera, Eduardo André
- Recalde, Santiago
- Villar, Federico Ignacio

Diagrama de Gantt

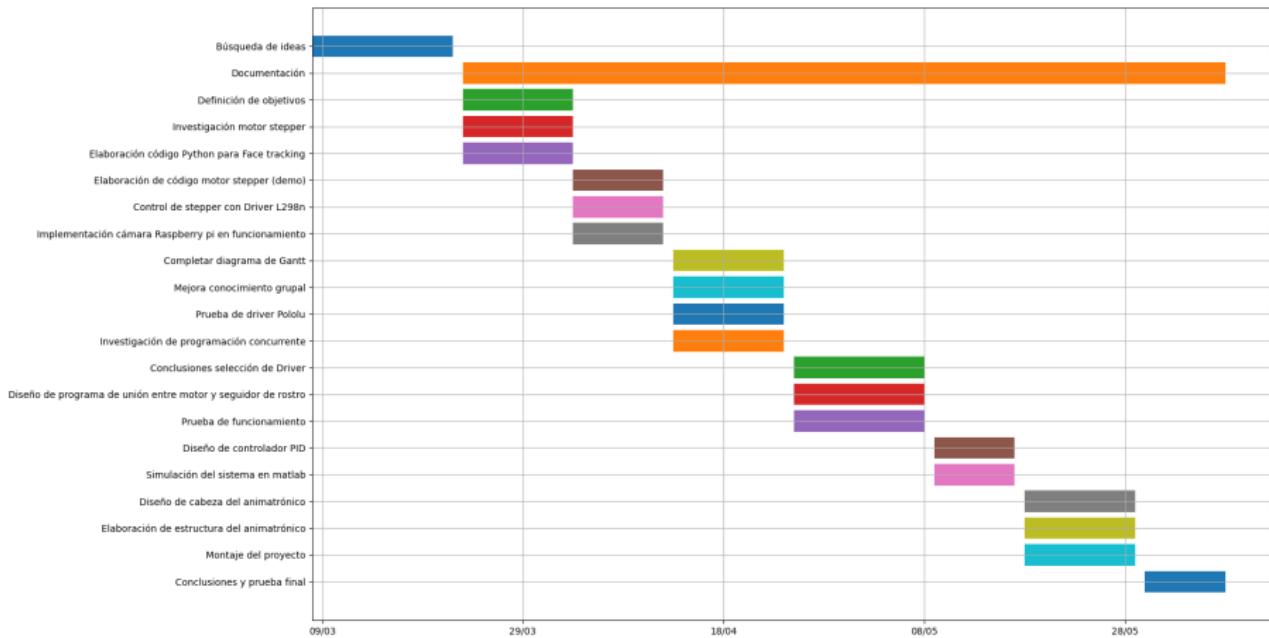


Figura 1: Diagrama de Gantt.

Motor paso a paso

El motor paso a paso es un tipo de motor de corriente continua que convierte pulsos eléctricos en desplazamientos angulares de un eje. Se puede clasificar de acuerdo a diversos criterios.



Figura 2: Motor bipolar Nema 17.

Controladores de motor paso a paso

Existen varias alternativas para controlar un motor paso a paso, entre ellos: L298N (puente H), ULN2003, **Pololu A4988**, Pololu DRV8825, y algunas alternativas más costosas con ventajas en cuanto a insonorización principalmente, o microstepping.



Figura 3: Driver Pololu A4988.

Software del reconocimiento facial

Desarrollado en Python, haciendo uso de dos librerías principales:

- Dlib
- OpenCV



Figura 4: Reconocimiento facial logrado.

Hardware para reconocimiento facial

Para poder realizar el reconocimiento facial, en un principio se tomó como computador de referencia un Macbook Air M1, pero luego, para fines de ahorrar espacio, se optó por realizar el reconocimiento con una Raspberry Pi 4B de 2GB de RAM junto con una Raspberry Pi Camera rev1.3.



Figura 5: Raspberry Pi 4B con una cámara conectada.

Herramientas de software utilizadas

Se utilizaron las siguientes herramientas de software:

- Visual Studio Code
- Thonny IDE
- Micropython
- dlib
- OpenCV
- Raspbian (Raspberry OS)

Evaluación alternativas de programación

Se evaluaron diferentes formas para programar el prototipo, estas fueron:

- Software y hardware: mezclar diferentes microcontroladores en donde cada uno realice una acción diferente.
- Programación concurrente o paralela: es la solución óptima. Se basa en un principio de ejecución de procesos en paralelo, logrando así una gran velocidad de reacción.
- Secuencialidad: esta es la forma implementada. Se trata de comprobar posiciones, error y corregir de forma secuencial durante un tiempo determinado.

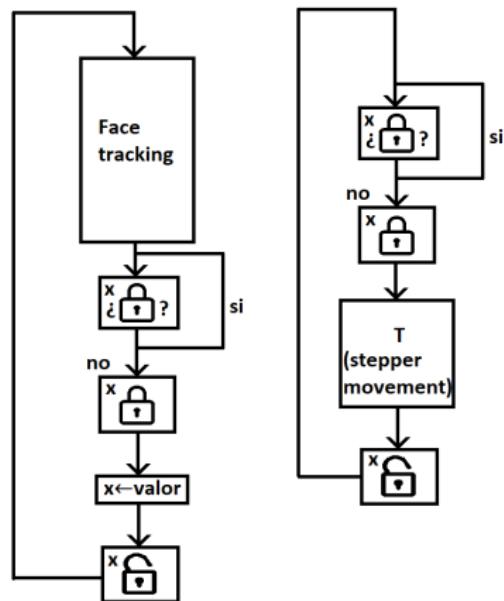


Figura 6: Concurrencia posible de implementar.

Modelado del sistema (análisis del error)

Se modeló el sistema a partir del error obtenido a partir de un análisis trigonométrico.

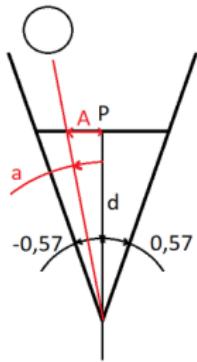


Figura 7: Análisis del error en el sistema.

Se llega a una expresión:

$$a = \operatorname{tg}^{-1}(0.00265A)$$

Que sirve para luego modelar el sistema y realizar la programación.

Modelado del sistema (función de transferencia)

Del análisis del funcionamiento de un motor paso a paso, y, con los valores obtenidos del modelado del error en el sistema, es posible obtener una función de transferencia a lazo cerrado para el sistema controlado. Se tiene:

$$G(s) = \frac{2.094}{0.5672s + 2.094}$$

Cuya respuesta al escalón de altura 0.28 radianes es la siguiente:

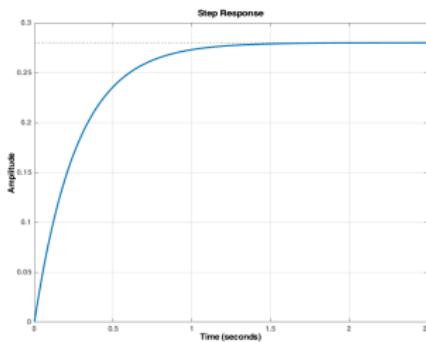


Figura 8: Respuesta del sistema controlado.

Conexiones en el sistema (esquema topológico)

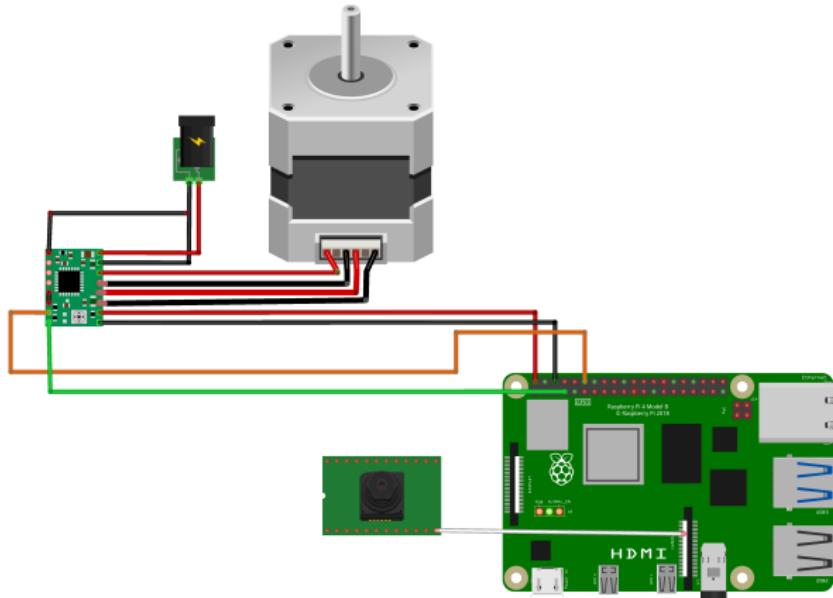


Figura 9: Circuito topológico.

Conexiones en el sistema (diagrama eléctrico)

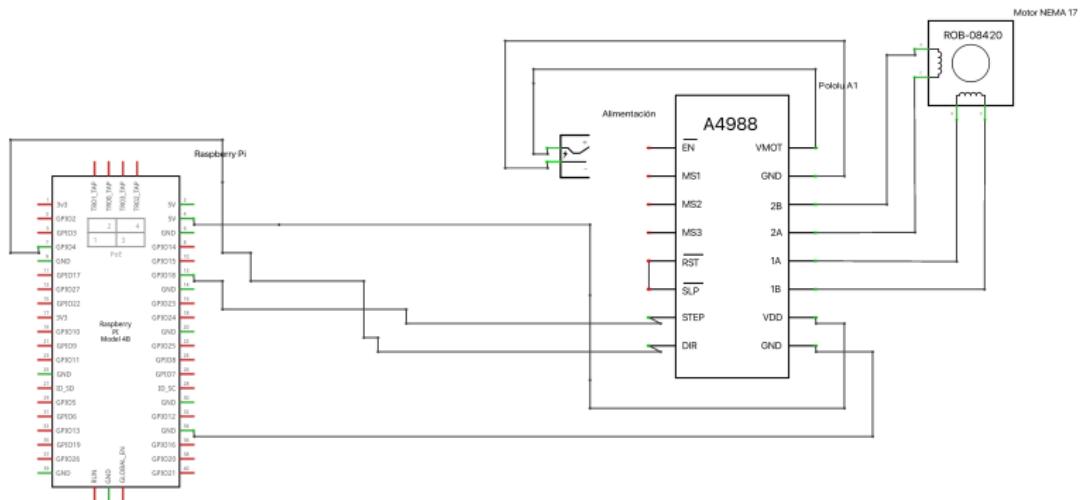


Figura 10: Circuito esquemático.

Diseños en 3D

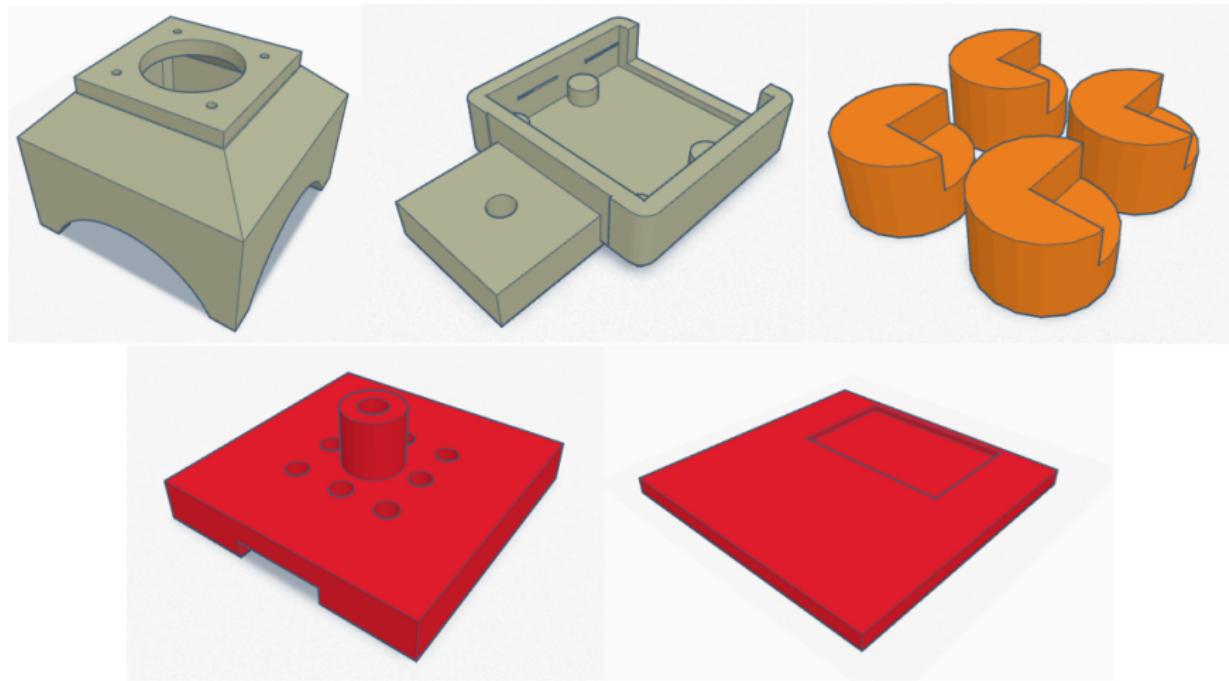


Figura 11: Diseños en 3D.

Implementación física

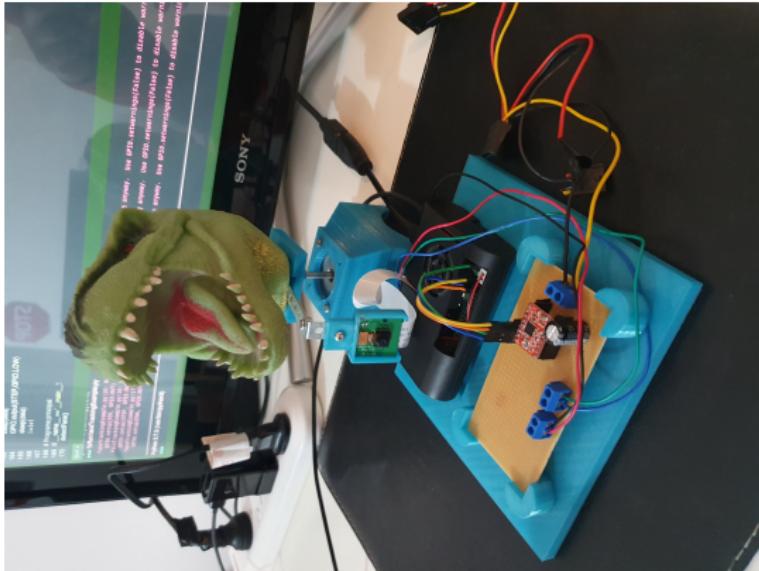


Figura 12: Prototipo físico.

Recomendaciones finales

- Trabajar en la comunicación del equipo de trabajo desde el comienzo para asentar las bases con las que se va a trabajar.
- Rellenar las lagunas de conocimientos antes de avanzar con la realización del proyecto, sino se generan disparidades que después afectan al desarrollo del mismo.
- Plantear objetivos de acuerdo al tiempo disponible, para evitar incompletitudes o exigir mucho en poco tiempo al acercarse fechas definitorias.
- Hacer análisis en tiempo continuo de los sistemas a implementar, para luego llevarlos a su análogo de la vida real, que es el discreto, partiendo del muestreo implementado en los microcontroladores.
- Elegir desde un inicio un único lenguaje de programación, para poder luego juntar código que se haya hecho por separado, y permitir la división de tareas que refieran al desarrollo de un software específico.

Conclusiones

A pesar de los contratiempos se puede concluir que el camino transitado en el desarrollo del seguidor de rostro fue todo menos turbulento. Los objetivos se modificaron y se cumplieron en tiempo y forma. Se solucionaron los problemas presentados a nivel de software y hardware, lo que permitió elaborar una solución satisfactoria. Entonces es posible argumentar que el proceso de aprendizaje y de desempeño en la elaboración del proyecto fue lo suficientemente bueno, y se vio representado en el correcto funcionamiento del dispositivo.

Por último, se agradece la disposición del docente para evaluar positivamente las cosas que funcionaban, corregir las cosas que no y siempre estar abierto a la comunicación tanto presencial como de forma remota. También agradecer a los miembros del equipo por tener la capacidad de adaptarse a los cambios para superar los obstáculos y por mantenerse unidos como grupo.

Gracias por su atención