



**UANL**

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



**FIME**

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN**  
**FACULTAD DE INGENIERIA MECANICA Y ELECTRICA**

**Lab. Controladores y Microcontroladores**

**Docente: Ing. Héctor Hugo Flores Moreno**

**Actividad documentación**

| Nombre                          | Matrícula | Carrera |
|---------------------------------|-----------|---------|
| David Maximiliano Reza Zavala   | 1911460   | ITS     |
| Mirna Guadalupe Herrera Pérez   | 1904929   | ITS     |
| Alan Gabriel Vazquez Espronceda | 1854242   | ITS     |

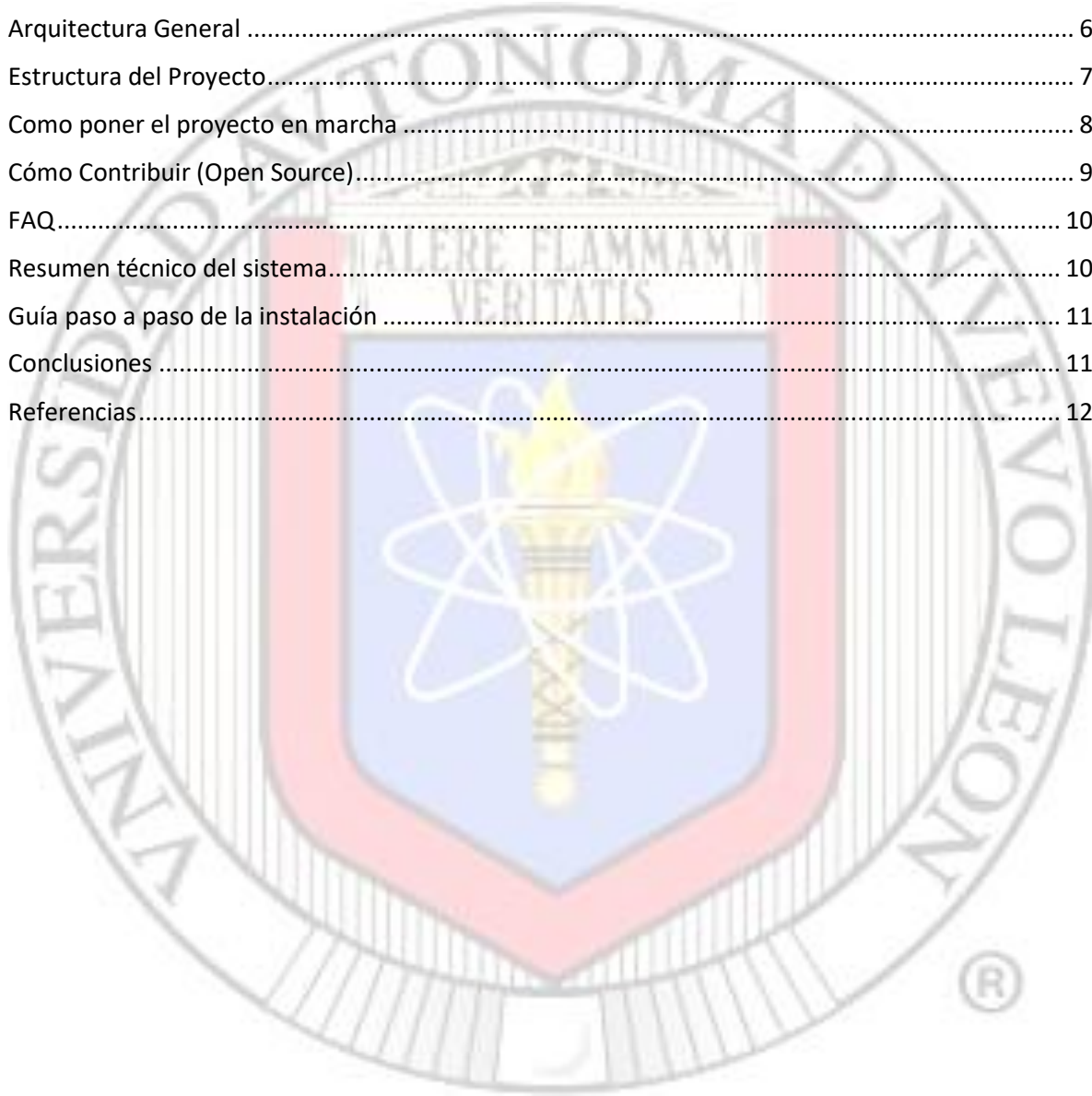
**Frecuencia: Jueves      Horas: N5-N6**

**Grupo:409      Salón: 9103**

## INDICE

### Contenido

|  |    |
|--|----|
| INTRODUCCION .....                       | 3  |
| ¿Qué Hace el Proyecto? .....             | 4  |
| ¿Como Funciona? .....                    | 5  |
| Arquitectura General .....               | 6  |
| Estructura del Proyecto.....             | 7  |
| Como poner el proyecto en marcha .....   | 8  |
| Cómo Contribuir (Open Source).....       | 9  |
| FAQ.....                                 | 10 |
| Resumen técnico del sistema.....         | 10 |
| Guía paso a paso de la instalación ..... | 11 |
| Conclusiones .....                       | 11 |
| Referencias.....                         | 12 |



## INTRODUCCION

### “El Rastreador Estelar y el Desafío de la Rotación Terrestre”

La astrofotografía, la disciplina de capturar imágenes del cosmos, se enfrenta a un desafío fundamental impuesto por las leyes de la física: la rotación de la Tierra. Aunque imperceptible en la vida cotidiana, este movimiento angular hace que las estrellas, galaxias y nebulosas se desplacen lentamente en el campo de visión de un telescopio o cámara. Este fenómeno, conocido como movimiento aparente de las estrellas, limita drásticamente el tiempo de exposición de las fotografías. Si una cámara permanece estática durante más de unos pocos segundos, el resultado no serán puntos de luz nítidos, sino **largas y antiestéticas estelas** luminosas.

Para **contrarrestar este efecto** y lograr las exposiciones prolongadas necesarias para revelar los detalles y colores de los objetos de cielo profundo, los astrónomos y aficionados recurren a los **Rastreadores Estelares**. Estos dispositivos son monturas motorizadas de precisión que giran exactamente a la misma velocidad angular que la Tierra, pero en sentido contrario, manteniendo al objetivo celeste perfectamente centrado. El corazón de cualquier *star tracker* no es su mecánica, sino la **precisión de su algoritmo de control motorizado**.

Este proyecto se enfoca precisamente en el núcleo programático y lógico de un rastreador estelar. El objetivo principal es diseñar y simular el control de un motor paso a paso para replicar la velocidad sideral de la Tierra, el tiempo exacto que tarda el planeta en completar una rotación.

Utilizaremos el entorno de simulación Tinkercad y el lenguaje C/C++ de Arduino para construir un prototipo virtual.

La **elección de un motor** y la plataforma Arduino no es casual. Los motores paso a paso permiten un control digital preciso mediante pulsos eléctricos, lo que los convierte en la herramienta ideal para replicar la minúscula y constante corrección de movimiento requerida.

## Rastreador de estrellas

### ¿Qué Hace el Proyecto?

#### Problema Resuelto:

El proyecto resuelve el problema de la **"estela de estrellas"** en la astrofotografía de larga exposición causado por la rotación de la Tierra. Dado que el eje de rotación de la Tierra permanece fijo respecto al cielo (día sideral), cualquier cámara estática verá a las estrellas moverse lentamente.

Este proyecto simula el mecanismo de un *Star Tracker*, un dispositivo que **contrarresta con precisión la rotación terrestre** mediante un motor de control. Permite a los desarrolladores y aficionados a la electrónica entender la lógica de control motriz (frecuencia de pulsos y retardo) necesaria para el seguimiento sideral.

#### Propósito, Contexto y Alcance:

- **Propósito:** Simular la programación del movimiento motorizado necesario para el seguimiento astronómico con precisión sideral.
- **Contexto:** Es un prototipo educativo, enfocado en el **control de motores** y la **programación en tiempo real** utilizando el entorno de Arduino (C/C++).
- **Alcance:** El proyecto **solo** abarca la simulación del **motor y su controlador** para lograr la velocidad sideral. **No** incluye la alineación polar, la detección de estrellas (visión artificial) o el hardware físico fuera del entorno de simulación de Tinkercad.





## ¿Como Funciona?

El núcleo del proyecto es un algoritmo de control de motor. La Tierra completa una rotación sideral en aproximadamente 86,164 segundos. Para que un star tracker compense este movimiento, su motor de seguimiento debe dar una vuelta completa exactamente en ese mismo lapso.

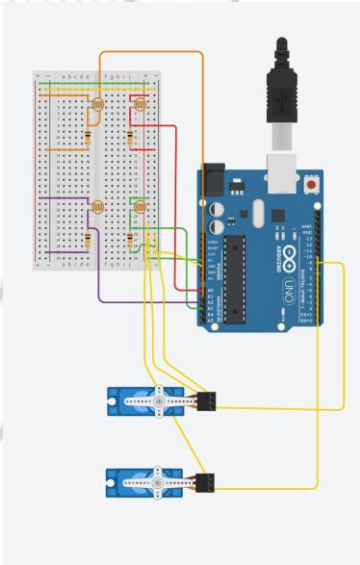
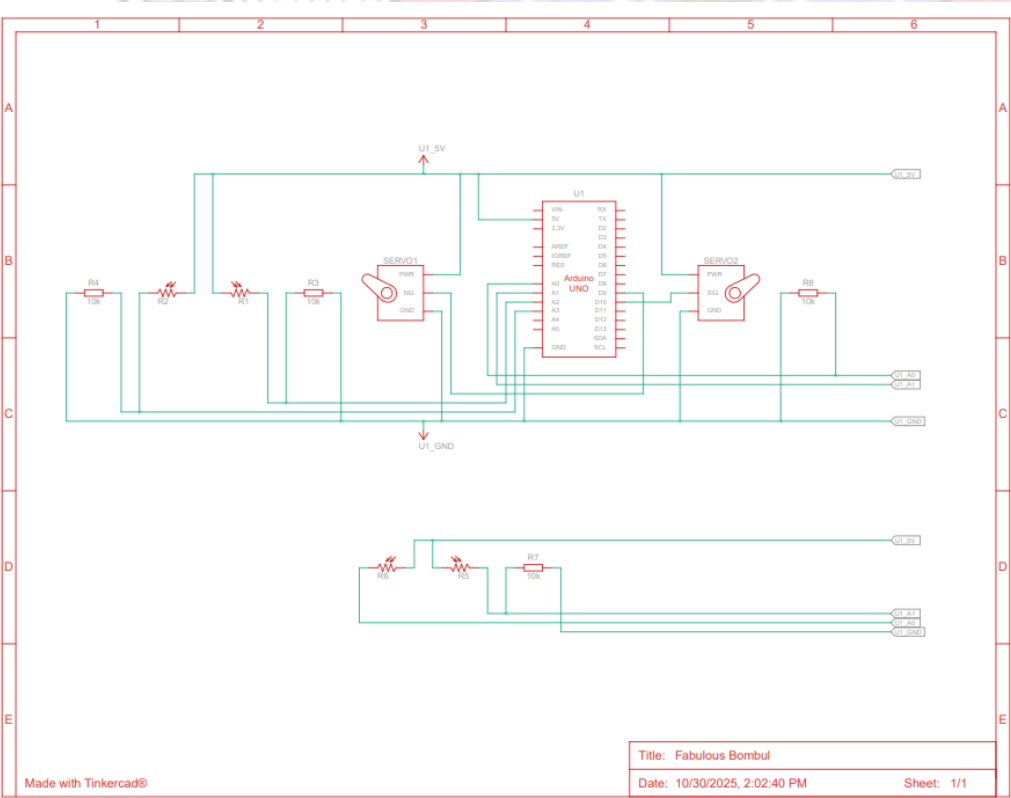
El proyecto simula este proceso mediante los siguientes pasos clave:

1. Cálculo de Precisión: El código C/C++ de Arduino calcula la frecuencia de pulsos necesaria. Esto se logra dividiendo el tiempo total sideral en 86164 segundos, entre el número total de pasos por revolución de nuestro motor 2048
2. El resultado es el tiempo de retardo que debe existir entre cada pulso que activa el motor.
3. Aceleración de Simulación: Dado que el retardo real es de aproximadamente 42 segundos por paso (demasiado lento para observar), la simulación aplica un factor de aceleración de 1000 x. Esto reduce el tiempo total de una revolución de 86164 segundos a unos 86 segundos. haciendo el movimiento observable en Tinkercad.

**Control y Ejecución:** Usando la librería Stepper.h, el Arduino establece la velocidad en RPM (basada en la velocidad acelerada). La **función** ejecuta una secuencia continua de pulsos a través del driver a los pines del motor.

| Módulo      | Componente Principal        | Función  | Tecnología/Lenguaje      |
|-------------|-----------------------------|--|--------------------------|
| Controlador | Arduino UNO                 | Ejecuta el algoritmo de seguimiento sideral. Calcula el delay exacto entre pulsos. | C / C++ (Arduino Sketch) |
| receptor    | Fotoceldas                  | Amplifica la señal y le da secuencia al motor                                      | Simulación en Tinkercad  |
| Actuador    | Motor Servomotor (Simulado) | Gira a la velocidad calculada, representando el movimiento de seguimiento.         | Simulación en Tinkercad  |

# Arquitectura General



## Estructura del Proyecto

/star-tracker-simulator

- | — README.md # Documentación principal
- | — SIMULACION/
  - | | — star\_tracker\_sketch.ino # Código fuente C/C++ (el sketch de Arduino)
  - | | — tinkercad\_link.txt # Enlace público a la simulación en Tinkercad
- | — DOCUMENTACION\_TECNICA.md # Detalles internos, diagramas y FAQs

### Cómo se comunican los módulos, decisiones técnicas, diagrama de arquitectura o flujo.

- **Tecnologías Implementadas:** Tinkercad, Arduino, C/C++
- **Decisiones técnicas:** Se utilizaron estas herramientas ya que contábamos con todos los componentes para simular el proyecto. Además Tinkercad deja trabajar en conjunto al equipo de manera en línea lo cual ayudo a obtener un mejor resultado en la resolución de el proyecto.
- **Diagrama por Bloques:**

| Block | descripción                  | Contexto  |
|-------|------------------------------|---|
| A-B   | Inicio/configuración Inicial | Programa arranca y ejecuta funcion  |
| C-D   | Definir parámetros           | Se definen las constantes de hardware, como los Pasos Por Revolución (PPR) del motor y la Relación de Reducción (engranajes). Se calcula el total de pasos para una |

|     |                             |   |
|-----|-----------------------------|---|
|     |                             | revolución del eje de salida  |
| E   | Calcular el retardo         | Se divide el tiempo de un Día Sideral, entre los pasos totales. Este es el tiempo de espera real por pulso  |
| F-G | Aplicar aceleración         | Se introduce el <b>Factor de Aceleración</b> ) para hacer observable el movimiento. Este nuevo retardo se convierte a la velocidad requerida en RPM para configurar la librería |
| H   | Establecer velocidad motora | La función configura la frecuencia de los pulsos que se enviarán al motor.  |
| I-J | Bloque de seguimiento       | La función se ejecuta constantemente. El comando envía la secuencia de pulsos al motor, manteniendo el movimiento de seguimiento a la velocidad calculada en el paso G.         |
| K-L | Fin del proceso             | El bucle continúa indefinidamente hasta que la simulación se detiene manualmente.   |

## Como poner el proyecto en marcha

### ¿Cómo se Usa o Contribuye Alguien Más?

El proyecto está diseñado para ser un recurso educativo y de prototipado.

### Requisitos e Instalación

No se requiere instalación de librerías externas o dependencias de *software*, ya que la simulación se realiza en línea.

- **Requisito:** Una cuenta gratuita en **Tinkercad**.
- **Dependencias de Librería (C/C++):** Se usa la librería estándar de Arduino:



(ya incluidas en el IDE de Arduino y disponible en Tinkercad).

## Cómo Poner el Proyecto en Marcha (Entorno Local/Simulado)

### 1. Clonar el Repositorio:

Bash

```
git clone https://github.com/tu_usuario/star-tracker-simulator.git  
cd star-tracker-simulator
```

### 2. Acceder a la Simulación:

- Abre el archivo SIMULACION/tinkercad\_link.txt.
- Copia y pega el enlace en tu navegador para ver la simulación interactiva.

### 3. Ejecutar la Simulación:

- Dentro de Tinkercad, haz clic en el botón **"Iniciar simulación"** (Start Simulation).
- El motor comenzará a girar a la velocidad acelerada.

### 4. Verificación del Funcionamiento:

- Abre el **"Monitor Serie"** en la parte inferior de la ventana de código.
- Verifica los mensajes de inicio y los mensajes de "Revolución simulada completada"

## Cómo Contribuir (Open Source)

¡Las contribuciones son bienvenidas! Específicamente, mejoras en la precisión del cálculo o la implementación de modos de paso alternativos.

1. **Fork** el repositorio.
2. Crea una nueva rama (git checkout -b feature/nueva-logica).
3. Implementa tus cambios en el archivo SIMULACION/star\_tracker\_sketch.ino.
4. Realiza *commits* claros y descriptivos.
5. Envía una solicitud de extracción (*Pull Request*) a la rama main.

## FAQ

### ¿Necesito un driver real?

*Sí, en el hardware real (fuera de Tinkercad), necesitarías un driver como el ULN2003 o un driver A4988 para un motor bipolar, ya que el Arduino no puede suministrar suficiente corriente directamente.*

### ¿Es esto preciso para astrofotografía?

*La lógica de cálculo es precisa. Sin embargo, la precisión real depende de la calidad del motor y la reducción mecánica (holgura) del mecanismo físico.*

### ¿Qué pasaría si quito el factor de aceleración?

*El motor giraría tan lento que no se notaría el movimiento durante días. El código es funcional, pero impráctico para la simulación visual.*

## Resumen técnico del sistema

El proyecto Star Tracker Simulator (STS) es una implementación de control de movimiento de precisión diseñada para replicar el mecanismo de un rastreador estelar (star tracker) en un entorno virtual. El objetivo central es demostrar la lógica algorítmica necesaria para contrarrestar la rotación de la Tierra.

| Módulo      | Componente Principal        | Función  | Tecnología/Lenguaje      |
|-------------|-----------------------------|--|--------------------------|
| Controlador | Arduino UNO                 | Ejecuta el algoritmo de seguimiento sideral. Calcula el delay exacto entre pulsos. | C / C++ (Arduino Sketch) |
| receptor    | Fotoceldas                  | Amplifica la señal y le da secuencia al motor                                      | Simulación en Tinkercad  |
| Actuador    | Motor Servomotor (Simulado) | Gira a la velocidad calculada, representando el movimiento de seguimiento.         | Simulación en Tinkercad  |

## Guía paso a paso de la instalación

- No se requiere instalación de librerías externas o dependencias de *software*, ya que la simulación se realiza en línea.
- **Requisito:** Una cuenta gratuita en **Tinkercad**.
- **Dependencias de Librería (C/C++):** Se usa la librería estándar de Arduino: (ya incluida en el IDE de Arduino y disponible en Tinkercad).

## Conclusiones

**Mirna Guadalupe Herrera Perez 1904929:** La elección de Tinkercad Circuits como plataforma para la simulación resultó ser un acierto pedagógico y técnico crucial. Este entorno permitió aislar el complejo problema del control de precisión motriz de las inevitables dificultades y costos de montar un hardware físico. Al simular el Arduino, y el motor, el proyecto se centró completamente en la lógica del código C/C++, que era el objetivo principal. Pudimos probar y depurar la compleja relación entre los pasos del motor, la relación de reducción mecánica y el tiempo sideral, sin preocuparnos por fallos de cableado o limitantes de corriente. Esto subraya la importancia de la simulación en la ingeniería de prototipos, permitiendo a los desarrolladores concentrarse en la arquitectura del software antes de comprometerse con el hardware.

**Alan Gabriel Vázquez Espronceda 1854242:** Este proyecto demostró con éxito la implementación de un sistema de control de bucle abierto para el seguimiento estelar. En un bucle abierto, el microcontrolador envía comandos sin recibir retroalimentación del sistema sobre la posición real de las estrellas. La precisión del *star tracker* depende enteramente de la exactitud de los cálculos de *timing* y de la estabilidad de la velocidad del motor. Esto tiene implicaciones significativas para la escalabilidad y el costo del proyecto.

**David Maximiliano Reza Zavala 1911460:** La simplicidad del bucle abierto, basado solo en la constante de tiempo sideral, mantiene el sistema económico y fácil de construir. Sin embargo, también establece sus límites: la precisión a largo plazo dependería de una alineación polar perfecta y de la ausencia de errores mecánicos (como el *periodic error* en el tornillo sin fin), que un sistema de bucle cerrado (que usaría sensores ópticos) corregiría automáticamente. En resumen, el STS valida el diseño de bajo costo y alta accesibilidad, mostrando que, con una programación

precisa, un motor básico y un controlador de código abierto se puede lograr la funcionalidad principal de la astrofotografía de seguimiento.

## Referencias

Autodesk. (s. f.). *Tinkercad Circuits: Simulación en línea*. Recuperado de <https://www.tinkercad.com/circuits>

Bilbao, F. (2019). *Control electrónico de motores: Análisis y diseño de sistemas paso a paso*. Marcombo.

Arduino. (s. f.). *Referencia del Lenguaje Arduino (C++)*. Recuperado de <https://www.arduino.cc/reference/en/>

