Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Proyecto:

Rastreador de Estrellas

Documentación

Asignatura: Laboratorio de Controladores y Microcontroladores Programables

Grupo: 407

Hora clase: N1 - N2 Jueves

Docente: Héctor Hugo Flores Moreno

Matrícula	Nombre	Carrera
1951068	Gerardo Daniel Lozano González	ITS
1952947	Heidi Pamela Martinez Martinez	ITS
1953575	Emmanuel Sánchez Aranda	ITS

Rastreador de Estrellas

¿Qué hace el proyecto?

El proyecto es un **"Rastreador de Estrellas"**. Su propósito principal es simular un dispositivo que muestra la ubicación de las estrellas más brillantes en el cielo nocturno, en tiempo real, basándose en la ubicación geográfica (latitud y longitud) y la hora del observador.

Resuelve el problema de "saber dónde mirar" para encontrar una estrella específica. Utiliza cálculos astronómicos para convertir las coordenadas celestes fijas de una estrella (Ascensión Recta y Declinación) en coordenadas locales (Altitud y Azimut), que indican qué tan alto sobre el horizonte y en qué dirección cardinal se encuentra la estrella.

Contexto y Alcance

Este es un proyecto de simulación académica para el laboratorio de "Controladores y Microcontroladores Programables" de la Facultad de Ingeniería Mecánica y Eléctrica (FIME) de la UANL.

El alcance del sistema está limitado al entorno de simulación Wokwi. No utiliza un módulo GPS real que se conecte a satélites, en su lugar, utiliza un "chip personalizado" (gps-neo6m.chip.c) que genera datos de ubicación y hora simulados. La salida es una visualización gráfica en una pantalla TFT, que muestra un mapa estelar simplificado con los puntos cardinales (N, S, E, W), los datos de coordenadas actuales y la posición de las estrellas con sus nombres.

¿Cómo funciona internamente?

Arquitectura General

El proyecto sigue una arquitectura de sistema embebido clásica de Entrada-Procesamiento-Salida, como se describe en el diagrama de bloques (imagen 1):

1. Bloque de Entrada (Módulo GPS NEO-6M Simulado):
    - Este es un chip personalizado de Wokwi (gps-neo6m.chip.c) que simula ser un receptor GPS.
    - Genera y transmite continuamente frases de texto estándar (NMEA), como $GPGGA y $GPRMC, a través de su pin TX.
    - Estas frases contienen datos simulados de latitud, longitud y hora, que varían ligeramente para simular movimiento.

2. Bloque de Procesamiento (Arduino Uno):
    - Es la unidad central del sistema que ejecuta la lógica principal definida en sketch.ino.
    - Recibe las frases NMEA del GPS en el pin D4 (RX) usando la librería SoftwareSerial.
    - Utiliza la librería TinyGPSPlus para decodificar (traducir) estas frases de texto en variables numéricas (latitud, longitud, hora).
    - El núcleo del procesamiento es la función raDecToAltAz. Esta función toma las coordenadas fijas de las estrellas (guardadas en la memoria PROGMEM del Arduino) y, usando la ubicación y hora del GPS, calcula su posición visible actual (Altitud y Azimut).
3. Bloque de Salida (Pantalla TFT ILI9341):
    - Actúa como la interfaz de salida visual.
    - El Arduino envía comandos de dibujo (coordenadas, formas y texto) a la pantalla a través de la interfaz de comunicación SPI.
    - Utiliza las librerías Adafruit_GFX y Adafruit_ILI9341 para mostrar el mapa estelar, los nombres de las estrellas, los puntos cardinales y los datos LAT/LON actuales.

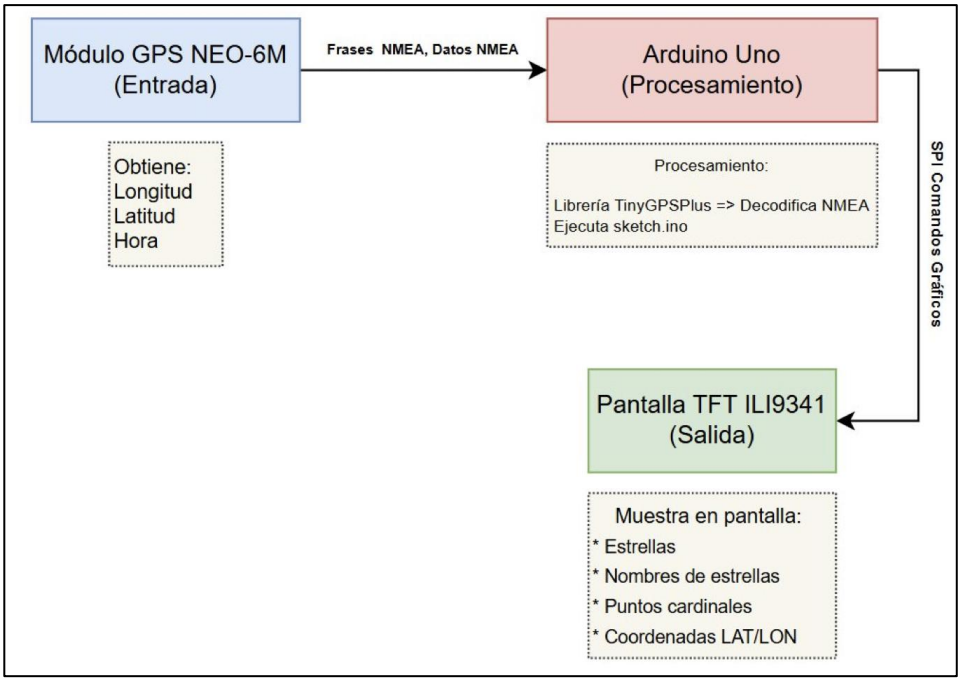

Imagen 1. Diagrama de Bloques

Estructura de Carpetas (Archivos del Proyecto)

En el entorno de Wokwi, el proyecto no usa una estructura de carpetas tradicional, sino un conjunto de archivos que definen el sistema:

- sketch.ino: El código principal (sketch) que se ejecuta en el Arduino Uno.
- diagram.json: Un archivo JSON que define qué componentes de hardware existen (Uno, GPS, LCD) y cómo están conectados (los cables).
- gps-neo6m.chip.c: El código fuente en C para el chip GPS personalizado, definiendo su lógica de simulación.
- gps-neo6m.chip.json: El archivo de configuración que le dice a Wokwi cómo definir el chip personalizado (nombre, autor, pines).
- libraries.txt: Un archivo de texto que le indica a Wokwi qué librerías de Arduino debe incluir automáticamente para que el proyecto compile.

Tecnologías Usadas

- Hardware (Simulado):
    - Arduino Uno (Microcontrolador)
    - Módulo GPS NEO-6M (Sensor de entrada)
    - Pantalla TFT ILI9341 (Salida visual)
- Software y Librerías:
    - Lenguaje Arduino (C++)
    - Lenguaje C (Para el chip personalizado)
    - Adafruit_GFX y Adafruit_ILI9341: Para el control de la pantalla TFT.
    - TinyGPSPlus: Para la decodificación de datos NMEA del GPS.
    - SoftwareSerial: Para crear un puerto serie por software y comunicarse con el GPS.
- Plataforma:
    - Wokwi: Simulador en línea de sistemas embebidos.

Comunicación de Módulos (Flujo de Datos)

El flujo de datos técnicos es el siguiente:

1. GPS a Arduino (UART): El pin TX del GPS simulado se conecta al pin D4 (RX de SoftwareSerial) del Arduino. Envía datos NMEA a 9600 baudios.
2. Arduino a GPS (UART): El pin D3 (TX de SoftwareSerial) del Arduino se conecta al pin RX del GPS. (Nota: En esta simulación, el Arduino solo *recibe* datos, por lo que esta conexión TX->RX no es funcionalmente crítica, pero está cableada).
3. Arduino a Pantalla (SPI): El Arduino controla la pantalla usando el protocolo SPI a través de pines de hardware específicos y pines digitales para el control:
    - D13 (SCK) -> lcd1:SCK
    - D11 (MOSI) -> lcd1:MOSI
    - D10 (CS) -> lcd1:CS
    - D9 (RST) -> lcd1:RST
    - D8 (D/C) -> lcd1:D/C

¿Cómo se usa o contribuye alguien más?

Poner el Proyecto en Marcha

Este proyecto es una simulación basada en web (Wokwi), por lo que no requiere instalación de software local, configuración de hardware físico ni compilación manual, sin embargo, si se llegara a necesitar este proyecto puede tener la siguiente instalación:

1. Instalar el IDE de Arduino.
2. Instalar las librerías desde el gestor:
   Adafruit_GFX
   Adafruit_ILI9341
   TinyGPSPlus

3. Cargar el código `sketch.ino` en tu Arduino.
4. Conectar el GPS y la pantalla según el pinout del sketch.
5. Verifica que el GPS obtenga señal.

Instalar Dependencias

Las dependencias de software (las librerías de Arduino) se gestionan automáticamente por Wokwi. El archivo libraries.txt le indica al simulador que debe incluir:

- Adafruit GFX Library
- Adafruit ILI9341
- TinyGPSPlus

Ejecutar la Simulación (Servidor Local)

Para ejecutar el proyecto:

1. Abre el proyecto en un navegador web usando el enlace:

   https://wokwi.com/projects/446194251655536641

2. Espera a que cargue el editor de Wokwi.
3. Presiona el botón verde de "Play" (Iniciar simulación) en la parte superior del editor.
4. La simulación comenzará:
   - El código sketch.ino se compilará y se cargará en el Arduino Uno virtual.
   - El chip GPS personalizado (gps-neo6m.chip.c) comenzará a ejecutarse y a enviar datos NMEA (puedes verlo en el monitor serial si lo configuras).
   - La pantalla TFT (ILI9341) se encenderá y, después de unos segundos, mostrará las coordenadas LAT/LON y el mapa estelar actualizándose.

¿Cómo Contribuir?

Siendo un proyecto de Wokwi, la contribución es sencilla:

1. Abre el enlace del proyecto.
2. Modifica los archivos según sea necesario (por ejemplo, puedes agregar más estrellas al arreglo stars[ ] en sketch.ino, o cambiar las coordenadas iniciales en gps-neo6m.chip.c).
3. Prueba tus cambios ejecutando la simulación.
4. Haz clic en el botón "Guardar" (Save) para guardar una nueva versión del proyecto (esto creará un "fork" o una copia bajo tu propia cuenta de Wokwi).
5. Comparte el nuevo enlace con los mantenedores del proyecto.

Diagrama de Arquitectura (Bloques)

El sistema utiliza una arquitectura de Entrada-Procesamiento-Salida, como se muestra en la imagen 1:

1. Entrada (GPS): El módulo GPS simulado envía frases NMEA (texto) con datos de ubicación/hora.
2. Procesamiento (Arduino): El Arduino recibe el texto, lo decodifica con TinyGPSPlus y calcula la posición de las estrellas (Altitud/Azimut).
3. Salida (TFT): El Arduino envía comandos de dibujo SPI a la pantalla para mostrar la información visualmente.

Diagramas de Flujo

Flujo Principal.

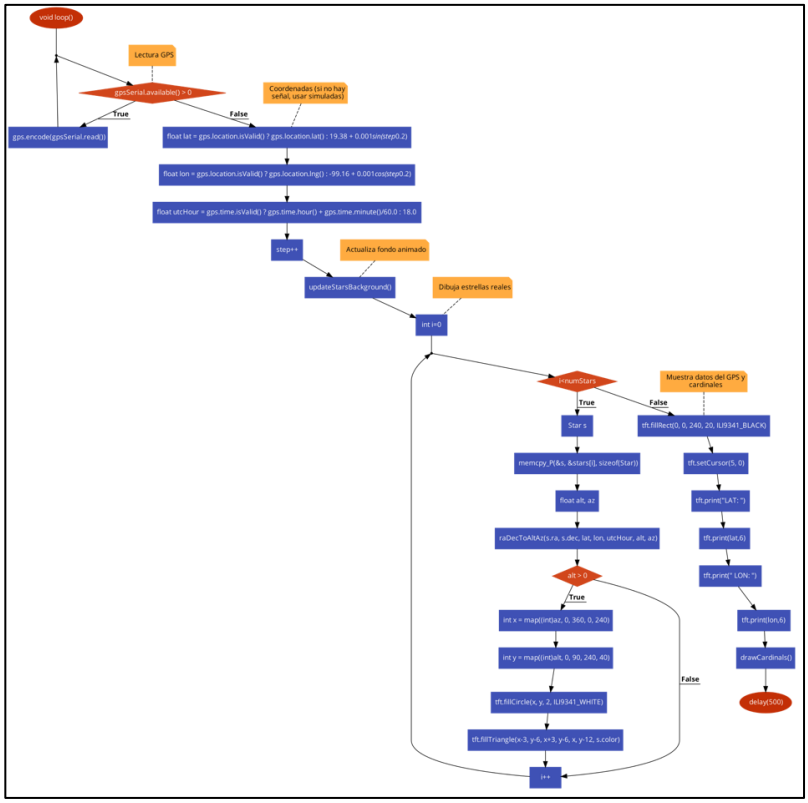

Imagen 2. Flujo Principal (loop)

El flujo principal coordina todas las operaciones del rastreador. Primero lee los datos del GPS y obtiene la latitud, longitud y hora. Si no hay señal, usa valores simulados. Luego actualiza el fondo animado, calcula la posición de cada estrella según la ubicación actual,

dibuja las que están visibles sobre el horizonte y finalmente muestra los datos en pantalla junto con los puntos cardinales.

Configuración Inicial.

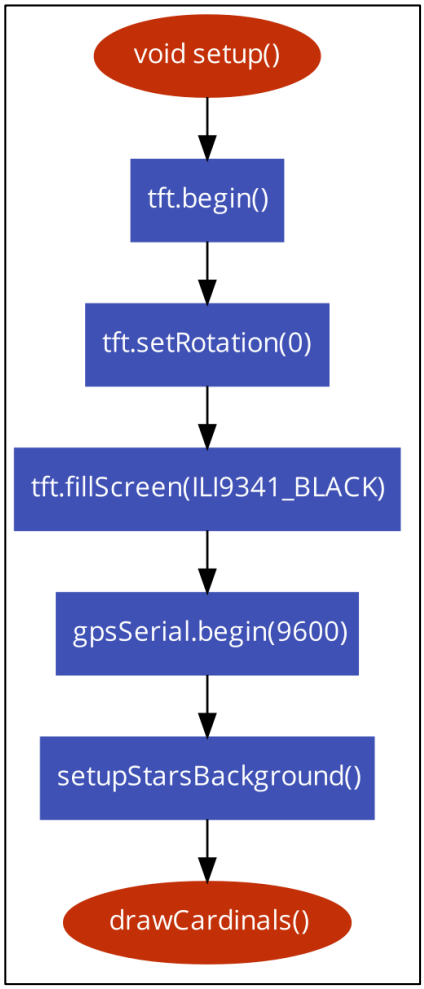

Imagen 3. Configuración inicial (setup)

Durante la etapa de configuración, se inicializan los componentes principales del sistema: la pantalla TFT, el módulo GPS y el fondo estelar. Además, se define la orientación de la pantalla, se limpia el contenido previo y se dibujan los puntos cardinales. Este proceso se ejecuta una sola vez al iniciar el dispositivo para dejar todo listo antes del bucle principal.

Actualización del fondo del cielo.

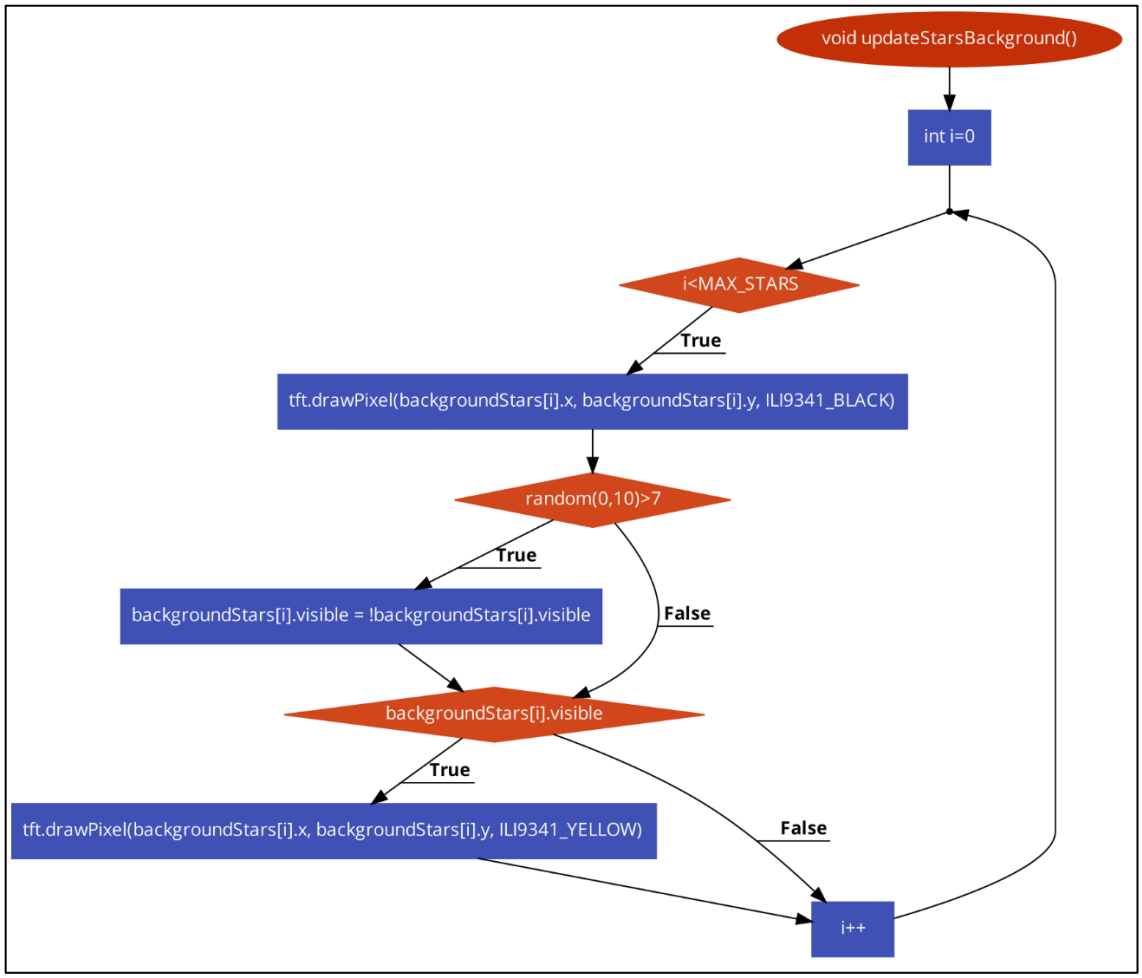

Imagen 4. Actualización del fonto de estrellas

Este flujo representa la animación del fondo del cielo. Recorre cada estrella del fondo, borra su píxel anterior y, con cierta probabilidad, cambia su estado de visibilidad para simular el parpadeo natural de las estrellas. Si está visible, se dibuja nuevamente en color amarillo. Es un proceso repetitivo que aporta dinamismo visual a la interfaz.

Coonversión de coordenadas.

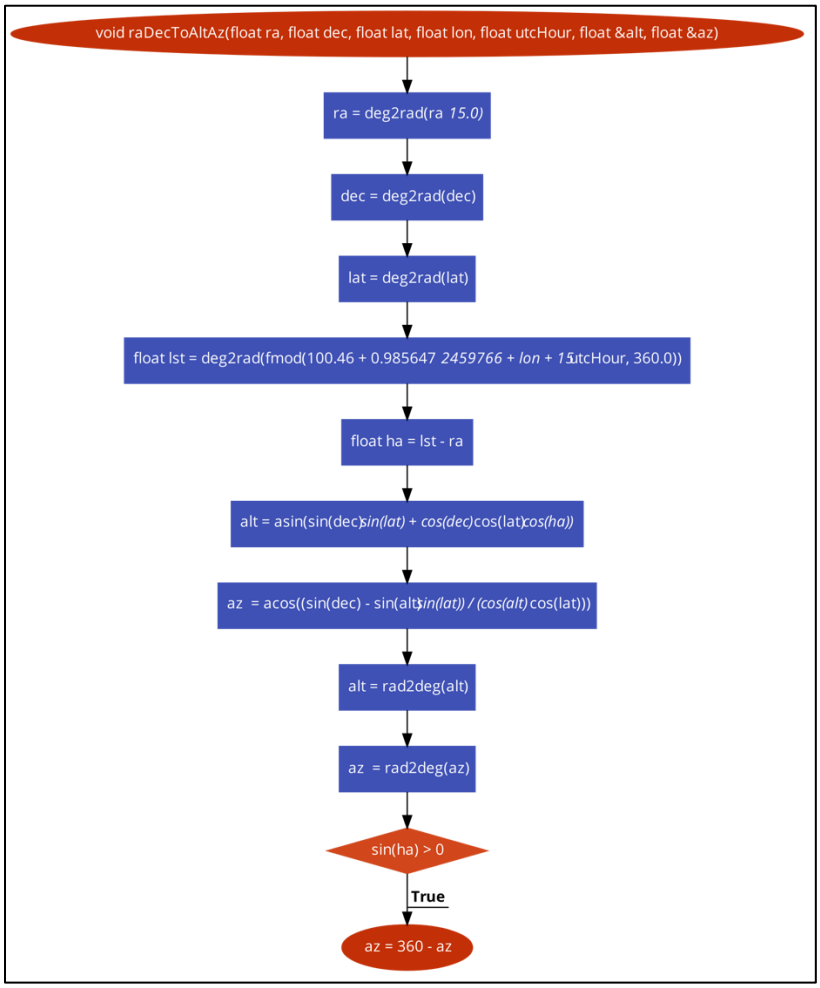

Imagen 5. Conversión de coordenadas

En esta función se realiza la conversión matemática de las coordenadas astronómicas ecuatoriales (ascensión recta y declinación) a coordenadas horizontales (altitud y acimut). Utiliza fórmulas trigonométricas y correcciones según la latitud, longitud y hora local. El resultado permite determinar la posición exacta de cada estrella en el cielo desde la ubicación del observador.

Diagrama de Secuencia

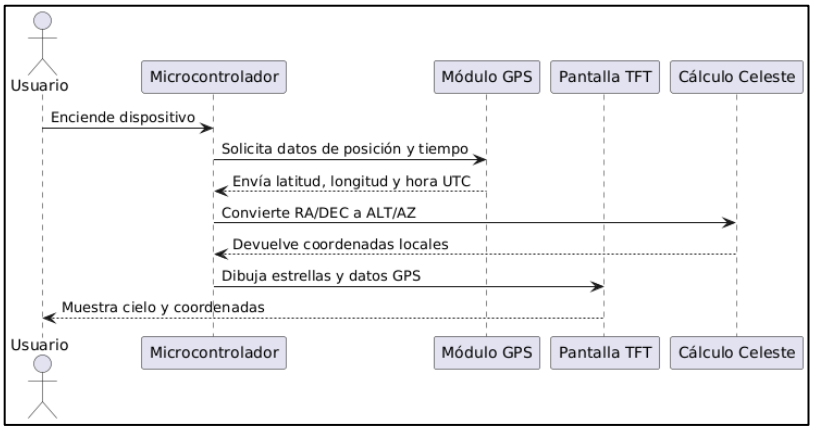

Imagen 6. Diagrama de Secuencia

Este diagrama ilustra el flujo de interacciones entre los componentes clave del sistema, repitiéndose dentro del loop() principal:

1. Entrada: El Módulo GPS envía un flujo de datos (frases NMEA) al Arduino Uno.
2. Procesamiento: El Arduino Uno se activa, decodifica los datos para obtener la latitud, longitud y hora. Luego, entra en un bucle para calcular la posición (Altitud/Azimut) de cada estrella.
3. Salida: Por cada estrella visible, el Arduino envía comandos de dibujo SPI a la Pantalla TFT. Finalmente, actualiza el texto de coordenadas en la pantalla y hace una breve pausa (delay) antes de reiniciar el ciclo.

Diagrama Pictórico (Conexiones)

Para el cableado detallado entre los componentes, consulta la imagen 2.

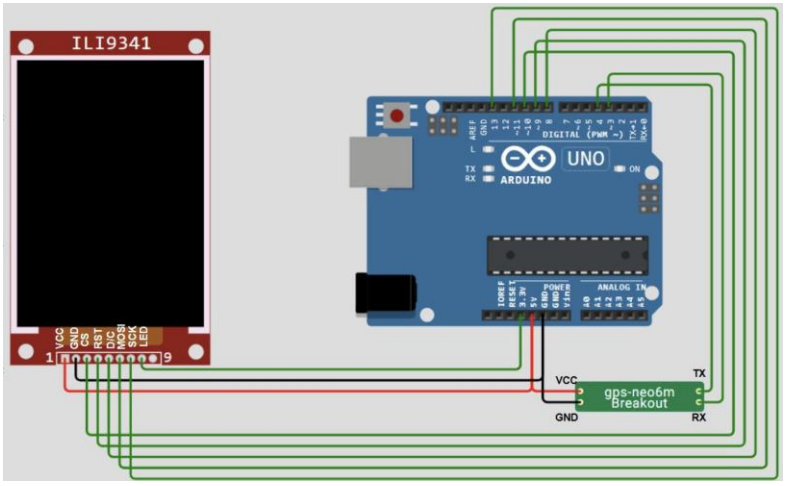

Imagen 7. Diagrama Pictórico

Resumen Técnico del Sistema

- **Comunicación GPS**: El Arduino (pin D4) escucha al GPS (pin TX) vía SoftwareSerial a 9600 baudios.
- **Comunicación Pantalla**: El Arduino controla la TFT vía SPI (pines D8 a D13).
- **Lógica Principal**: El loop() principal lee el GPS, y si hay datos válidos, recalcula la posición de todas las estrellas en la lista stars[] usando la función raDecToAltAz.
- **Simulación de GPS**: El chip gps-neo6m.chip.c se inicializa con coordenadas de la Ciudad de México (19.43 N, -99.13 W) y añade una pequeña variación aleatoria cada segundo para simular movimiento.

FAQ (Dudas Frecuentes)

Pregunta: ¿Por qué la pantalla está en negro o no muestra nada?

R: Asegúrate de haber presionado el botón "Play" en Wokwi. La simulación puede tardar unos segundos en inicializar la pantalla y recibir los primeros datos del GPS.

Pregunta: ¿Por qué las estrellas no se mueven?

R: Las estrellas sí se mueven, pero muy lentamente, al igual que en la vida real. La simulación actualiza las posiciones cada 500ms (delay(500)) 53, pero el cambio solo es notable si dejas la simulación corriendo por varios minutos o si modificas la hora (utcHour) 54 o la lógica de tiempo en el sketch.ino.

Pregunta: ¿Puedo usar esto con un GPS real?

R: Sí, El código en sketch.ino está listo para usarse. Solo necesitarías conectar un módulo GPS NEO-6M real a los pines 5V, GND, y conectar el pin TX del GPS al pin D4 del Arduino. El código gps.encode(gpsSerial.read()) 55 leerá los datos del hardware real exactamente igual que lee los datos simulados.

Referencias

Charting the Stars with Arduino - Wokwi ESP32, STM32, Arduino Simulator. (s. f.). https://wokwi.com/projects/446194251655536641

Guía de modelos Arduino y sus características | Arduino UNO | BricoGeek Lab. (2024, 13 noviembre). Guía Completa de Características y Documentación de los Modelos de Placas Arduino En un Solo Lugar. https://lab.bricogeek.com/tutorial/guia-de-modelos-arduino-y-sus-caracteristicas/arduino-uno

Hollanda Academy. (2024, 30 enero). ESP TFT ILI9341 Basic : Programming Adafruit ILI9341 TFT Display with GFX Library | Wokwi Simulation [Vídeo]. YouTube. https://www.youtube.com/watch?v=R3xkPtvCpd4

Hpmm. (s. f.). GitHub - HPMM2/Rastreador-de-Estrellas-con-Arduino: Este es un proyecto basado en Arduino UNO, un módulo GPS NEO-6M y una pantalla TFT ILI9341 simulado en Wokwi. Su propósito principal es simular un dispositivo que muestra la ubicación de las estrellas, en tiempo real, basándose en la ubicación geográfica (lat y lon) y la hora del observador. GitHub. https://github.com/HPMM2/Rastreador-de-Estrellas-con-Arduino

ILI9341.pdf. (s. f.). ILI9341 Documentation. https://cdn-shop.adafruit.com/datasheets/ILI9341.pdf

NEO-6 series. (2024, 22 febrero). U-blox. https://www.u-blox.com/en/product/neo-6-series

UNO R3. (s. f.). UNO R3 | Arduino Documentation. https://docs.arduino.cc/hardware/uno-rev3/

Vito Technology, Inc. (2024, 25 julio). Coordenadas celestes: guía definitiva. Star Walk. https://starwalk.space/es/news/celestial-coordinates