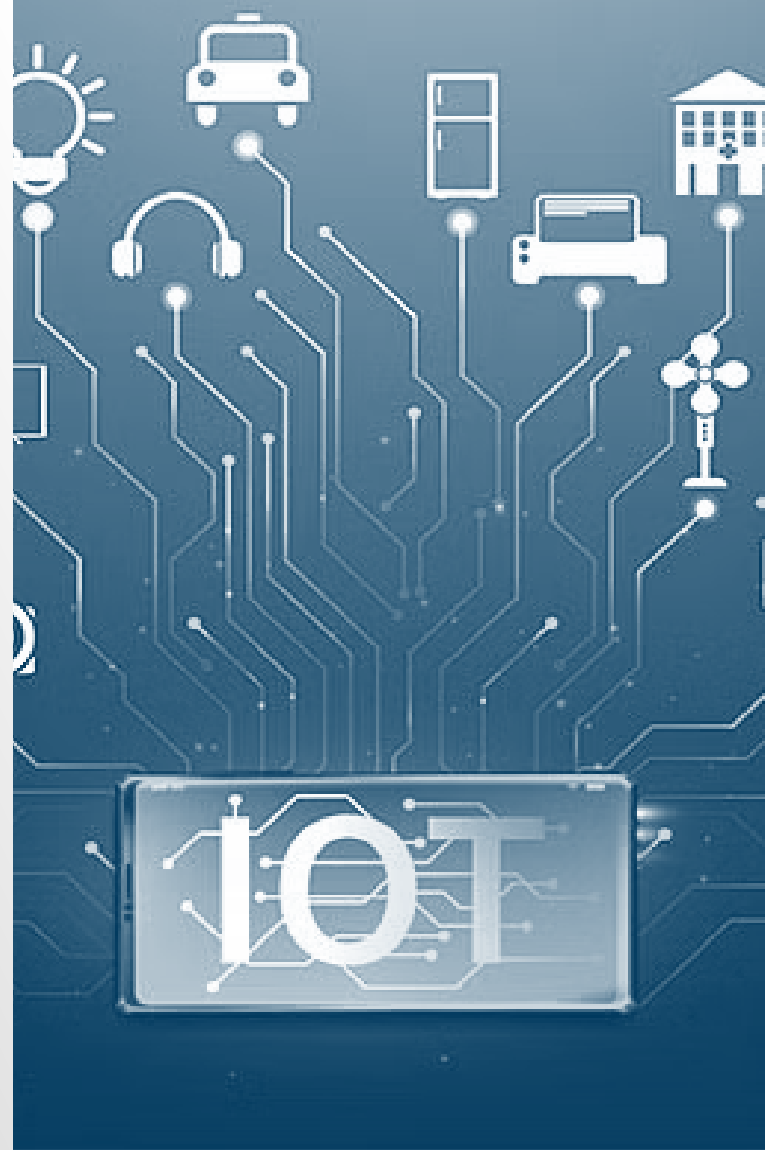


Simulacion de GPS Tracker con Display OLED y ESP32



Instituto Superior Politecnico Cordoba
Grupo 5
Tecnatura superior en Telecomunicaciones



Contents

Introducción	3
Funcionamiento.....	4
<i>Función setup():</i>	4
<i>Función loop():</i>	5
<i>Función displayCoordinates():</i>	5
<i>Función displayMessage():</i>	6
Librerías.....	6
Conclusiones	6

Introducción

Este documento tiene como propósito proporcionar una explicación detallada de la simulación del proyecto del GPS Tracker en la plataforma Wowki, así como de las funciones implementadas en el código y la razón por la cual no fue posible utilizar un módulo GPS físico en dicha plataforma. Además, se abordará el objetivo de realizar una simulación de un GPS dentro del código, así como su importancia para llevar a cabo una prueba de hardware y código simulado que permita evaluar su funcionamiento y obtener una idea clara de cómo abordar el proyecto al implementarlo en el entorno físico.



Este proyecto busca proporcionar una solución de seguimiento y localización a través de un sistema de GPS. Sin embargo, debido a limitaciones técnicas y la falta de disponibilidad de un módulo GPS físico compatible con la plataforma, se decidió implementar una simulación de GPS dentro del código.

La simulación se basa en la generación de coordenadas GPS aleatorias, las cuales son utilizadas para simular la ubicación y otros datos asociados, como la altitud, la velocidad y el

número de satélites al que se encuentra conectado. Estos valores son generados de manera aleatoria y actualizados en intervalos regulares dentro del código.

El propósito de esta simulación es realizar una prueba de hardware y código simulado que nos permita evaluar el funcionamiento del GPS Tracker en la plataforma Wowki. A través de la simulación,

podemos obtener una idea clara de cómo se comportaría el sistema en el entorno físico y qué resultados podríamos esperar al implementarlo.

Al realizar esta simulación, podemos identificar posibles problemas o mejoras en el diseño del proyecto, así como familiarizarnos con las funcionalidades y limitaciones del código antes de llevarlo a la práctica. Esto nos brinda la oportunidad de optimizar el sistema y tomar decisiones informadas al abordar la implementación física del GPS Tracker.

En resumen, este documento proporcionará una explicación detallada de la simulación del proyecto del GPS Tracker en la plataforma Wowki, describirá las funciones implementadas en el código y justificará la necesidad de simular un GPS dentro del código debido a la falta de disponibilidad de un módulo GPS físico compatible. Asimismo, se destacará el propósito de la simulación para realizar una prueba de hardware y código simulado que nos permita evaluar su funcionamiento y obtener una visión clara de cómo abordar el proyecto al llevarlo a cabo en el entorno físico.

Funcionamiento

Función *setup()*:

La función *setup()* se encarga de realizar la configuración inicial del sistema antes de que comience el ciclo principal. A continuación, se describe en detalle el proceso llevado a cabo por esta función:

- **Inicialización de la comunicación I2C:** Se utiliza el método *Wire.begin()* para inicializar la comunicación I2C, que permite la comunicación entre el microcontrolador y otros dispositivos periféricos, en este caso, la pantalla OLED.
- **Inicialización de la pantalla OLED:** Se utiliza el método *display.begin(SSD1306_SWITCHCAPVCC, 0x3C)* para inicializar la pantalla OLED con una resolución de 128x64 píxeles. La comunicación con la pantalla se realiza a través de la biblioteca Adafruit_SSD1306.
- **Configuración del tamaño y color del texto:** Se utiliza el método *display.setTextSize(1)* para establecer el tamaño del texto en 1, lo que representa un tamaño pequeño y adecuado para la pantalla OLED. Además, se utiliza el método *display.setTextColor(SSD1306_WHITE)* para configurar el color del texto en blanco.
- **Configuración del pin del LED:** Se utiliza la función *pinMode(LED_PIN, OUTPUT)* para configurar el pin correspondiente al LED integrado como salida. Esto permite controlar el encendido y apagado del LED para indicar el estado de la conexión y búsqueda de señal.
- **Generación de coordenadas GPS aleatorias:** Se utiliza la función *randomSeed(analogRead(0))* para inicializar el generador de números aleatorios. Esto se hace utilizando el valor de una lectura analógica del pin 0 como semilla para generar números aleatorios más precisos. A continuación, se generan valores aleatorios para la latitud, longitud, altitud, velocidad y número de satélites utilizando la función *random()*. Estos valores aleatorios representan una simulación de las coordenadas GPS.
- **Mostrar mensaje "Conectando" durante la conexión:** Se utiliza la función *displayMessage("Conectando", 1000, 3000, true)* para mostrar en la pantalla OLED el mensaje "Conectando" durante un período de tiempo determinado. Este mensaje indica que se está estableciendo la conexión del GPS. Esto no es mas que para simular.

- **Mostrar mensaje "Buscando Satélites" durante la búsqueda de señal:** Se utiliza la función `displayMessage("Buscando Satelites", 1000, 4000, true)` para mostrar en la pantalla OLED el mensaje "Buscando Satélites" durante un período de tiempo determinado. Este mensaje indica que el sistema está en búsqueda de señal de los satélites GPS. Esto no es mas que para simular.
- **Encendido del LED al terminar de buscar satélites y conectar:** Se utiliza la función `digitalWrite(LED_PIN, HIGH)` para encender el LED integrado en el microcontrolador. Esto se realiza después de establecer la conexión y finalizar la búsqueda de satélites.
- **Mostrar mensaje "Conectado" y las coordenadas GPS:** Se utiliza la función `displayMessage("Conectado", 2000, false)` para mostrar en la pantalla OLED el mensaje "Conectado" durante un período de tiempo determinado. Este mensaje indica que la conexión del GPS se ha establecido correctamente. Esto no es mas que para simular.

A continuación, se llama a la función `displayCoordinates()` para mostrar las coordenadas GPS generadas aleatoriamente en la pantalla y en la consola serial.

Función `loop()`:

La función `loop()` es el ciclo principal del programa y se ejecuta de forma repetida. A continuación, se describe en detalle el proceso llevado a cabo por esta función:

- **Generación de nuevas coordenadas GPS aleatorias:** Se utiliza la función `random()` para generar nuevos valores aleatorios para la latitud, longitud, altitud, velocidad y número de satélites. Estos valores simulan un cambio en la posición y otros parámetros del GPS.
- **Actualización de la pantalla con las nuevas coordenadas:** Se llama a la función `displayCoordinates()` para actualizar la pantalla OLED con las nuevas coordenadas GPS generadas.
- **Retraso de 2 segundos antes de generar nuevas coordenadas:** Se utiliza la función `delay(2000)` para introducir un retraso de 2 segundos antes de generar nuevas coordenadas. Esto permite simular una frecuencia de actualización de las coordenadas.

Función `displayCoordinates()`:

La función `displayCoordinates()` se encarga de mostrar las coordenadas GPS en la pantalla OLED y en la consola serial. A continuación, se describe en detalle el proceso llevado a cabo por esta función:

- **Actualización de la pantalla con las coordenadas GPS:** Se utiliza el método `display.clearDisplay()` para borrar el contenido previo de la pantalla OLED. A continuación, se utilizan los métodos `display.setCursor()` y `display.println()` para imprimir las etiquetas y los valores de las coordenadas GPS en la pantalla OLED.
- **Mostrar los datos en la consola serial:** Se utilizan los métodos `Serial.println()` y `Serial.print()` para imprimir las etiquetas y los valores de las coordenadas GPS en la consola serial. Esto permite monitorear los datos generados por el GPS simulado.

Función *displayMessage()*:

La función *displayMessage()* se encarga de mostrar un mensaje en la pantalla OLED con la opción de mostrar puntos animados. A continuación, se describe en detalle el proceso llevado a cabo por esta función:

- **Mostrar un mensaje en la pantalla con la opción de mostrar puntos animados:** Se utiliza el método *display.setTextSize()* para establecer el tamaño del texto. A continuación, se utiliza el método *display.setTextColor()* para configurar el color del texto. Se muestra el mensaje en la pantalla utilizando el método *display.print()*.
- **Uso de parámetros como el mensaje, la duración y la opción de mostrar puntos:** Esta función recibe como parámetros el mensaje a mostrar, la duración del mensaje en milisegundos y un valor booleano que indica si se deben mostrar puntos animados. Estos parámetros permiten personalizar el mensaje y su apariencia en la pantalla OLED.
- **Actualización de los puntos en cada iteración:** Si se selecciona la opción de mostrar puntos animados, se utiliza una variable *dotCount* para realizar un seguimiento del número de puntos que se deben mostrar. En cada iteración, se actualiza esta variable para alternar entre la visualización de diferentes cantidades de puntos, creando así una animación visual en la pantalla OLED.

Con el desarrollo de estas funciones y la simulación de GPS dentro del código, se logra obtener una representación visual de las coordenadas GPS generadas aleatoriamente en la plataforma Wowki. Esto permite realizar pruebas de hardware y código simulado para evaluar el funcionamiento del sistema y obtener información valiosa para futuras implementaciones físicas del proyecto.

Librerías

Las librerías utilizadas en el código del GPS simulado son las siguientes:

- **Wire:** Esta librería proporciona funciones para la comunicación I2C (Inter-Integrated Circuit) utilizada para conectar dispositivos electrónicos en el proyecto.
- **Adafruit_GFX:** Es una librería que permite dibujar formas geométricas y texto en pantallas OLED y otros dispositivos gráficos compatibles.
- **Adafruit_SSD1306:** Esta librería proporciona funciones específicas para controlar pantallas OLED basadas en el controlador SSD1306. Permite inicializar la pantalla, mostrar texto y gráficos, y realizar otras operaciones relacionadas con la pantalla.

Conclusiones

Las funciones desempeñan un papel fundamental en la estructura y funcionalidad del código del GPS simulado. A través de ellas, se logra una organización modular del código, lo que facilita su comprensión, mantenimiento y reutilización. Cada función cumple con una tarea específica y bien definida, lo que facilita su depuración y modificación en caso de ser necesario.

La función *setup()* se encarga de realizar la inicialización y configuración de los componentes utilizados, como la comunicación I2C y la pantalla OLED. Además, genera coordenadas GPS aleatorias y muestra mensajes de conexión y búsqueda de satélites.

La **función *loop()*** es el núcleo del programa, donde se generan nuevas coordenadas GPS aleatorias, se actualiza la pantalla con los nuevos valores y se introduce un retraso para simular la frecuencia de actualización.

La **función *displayCoordinates()*** se encarga de mostrar las coordenadas GPS en la pantalla OLED y en la consola serial. Esto permite visualizar y monitorear los datos generados por el GPS simulado.

La **función *displayMessage()*** permite mostrar mensajes personalizados en la pantalla OLED, con la opción de animar puntos. Esto brinda una retroalimentación visual al usuario durante diferentes etapas del programa.

En cuanto a la modularidad y mantenibilidad del código, el uso de funciones facilita la comprensión y organización del mismo. Cada función se encarga de una tarea específica, lo que permite una fácil identificación y corrección de errores. Además, si en el futuro se requiere modificar o ampliar alguna funcionalidad, es más sencillo hacerlo de forma aislada sin afectar otras partes del código.