



Universidad Nacional de Colombia

Departamento de Eléctrica y Electrónica.
FACULTAD DE INGENIERA - SEDE BOGOTÁ

INFORME FINAL. SPACE LORA.

Electronica Digital 1

Autores:

Sergio Andres Bolaños Penagos
Johan Camilo Patiño Mogollon

Profesor:

Johan Sebastian Eslava Garzon

Julio 2025
Bogotá, Colombia

Índice general

0.1. Introducción	2
0.2. Arquitectura Implementada	2
0.2.1. Diagrama General del Sistema	2
0.2.2. Diagrama de Bloques en Quartus	3
0.2.3. Análisis del Baud Rate Generator	3
0.2.4. Diseño del UART Receiver	4
0.3. Testeos.	4
0.3.1. Modulo GPS.	4
0.3.2. ESP32-LoRa	5
0.4. Uso de Inteligencia Artificial	6
0.5. Conclusiones	6

0.1. Introducción

En la actualidad, los sistemas de localización y seguimiento de activos han adquirido una relevancia crítica en aplicaciones como la logística, la gestión de flotas, la seguridad personal y la trazabilidad industrial. Soluciones comerciales como el Apple AirTag ofrecen una localización precisa mediante tecnologías como Bluetooth y redes colaborativas, pero su uso está restringido a ecosistemas cerrados y dependientes de infraestructura propietaria. Esta condición limita significativamente su escalabilidad, personalización y aplicación en contextos académicos o de desarrollo comunitario donde se priorizan plataformas abiertas y de bajo costo.

Asimismo, existen soluciones de rastreo GPS basadas en LoRaWAN, como las ofrecidas por Digital Matter o AbeeWay, que permiten comunicación a larga distancia sin necesidad de infraestructura celular. Sin embargo, estas también operan bajo esquemas de licenciamiento cerrado, dificultando su adaptación a proyectos que requieren modificar o estudiar sus protocolos internos. En ese contexto, surge la necesidad de diseñar una alternativa abierta, flexible y completamente documentada que permita a investigadores, estudiantes y desarrolladores replicar y escalar un sistema IoT de rastreo geográfico.

Con este objetivo, se propone el desarrollo de un prototipo denominado **SpaceLoRa**, concebido como un sistema de localización IoT implementado sobre una FPGA Cyclone IV. El proyecto se enmarca en la asignatura de Electrónica Digital I y persigue tanto fines académicos como de validación técnica de una arquitectura abierta para adquisición y transmisión de coordenadas geográficas. El sistema está compuesto por los siguientes elementos:

- Un receptor GPS NEO-6MV2 encargado de recibir sentencias NMEA, que contienen datos de posicionamiento global como latitud, longitud y hora.
- Una FPGA Cyclone IV que actúa como núcleo de procesamiento, extrayendo en tiempo real las coordenadas GPS a través de una comunicación UART, almacenando y procesando los datos mediante una lógica diseñada en Verilog HDL.
- Un módulo LoRa LilyGo TTGO LoRa32 (basado en SX1276 + ESP32), encargado de transmitir los datos empaquetados mediante radiofrecuencia de baja potencia a distancias de hasta varios kilómetros en condiciones óptimas.
- Un nodo receptor LoRa que visualiza los datos enviados, sirviendo como interfaz de monitoreo remoto.

Finalmente, la arquitectura del sistema ha sido estructurada de forma modular, permitiendo un desarrollo escalonado y validación por etapas. Esto incluye bloques dedicados al control de reloj (baud rate), recepción UART, almacenamiento temporal mediante FIFO, parsing de sentencias NMEA, y empaquetamiento eficiente para transmisión LoRa. La integración de estos módulos ha sido respaldada por pruebas unitarias y simulaciones que aseguran su correcto funcionamiento en condiciones reales.

0.2. Arquitectura Implementada

El sistema propuesto, denominado **SpaceLoRa**, implementa una arquitectura embebida orientada al rastreo geográfico mediante una plataforma totalmente abierta basada en FPGA. El objetivo principal es adquirir sentencias GPS en formato NMEA, procesarlas en hardware mediante módulos digitales y transmitirlos a un nodo remoto utilizando la tecnología LoRa a través de un microcontrolador ESP32. La arquitectura fue diseñada en lenguaje Verilog, simulada en entornos digitales y desplegada en una FPGA Cyclone IV.

0.2.1. Diagrama General del Sistema

A continuación se presenta el diagrama de arquitectura del sistema, el cual muestra los principales bloques funcionales desarrollados y su flujo de datos:

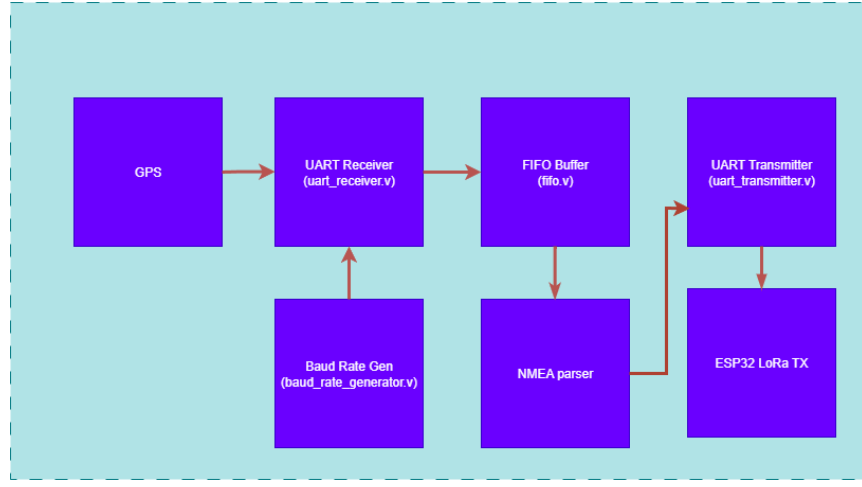


Figura 1: Arquitectura funcional del sistema SpaceLoRa.

Descripción de bloques

1. **GPS (NEO-6MV2):** Dispositivo encargado de generar información geográfica en formato estándar NMEA 0183. Transmite continuamente datos como latitud, longitud, velocidad y tiempo mediante interfaz UART.
2. **Baud Rate Generator (baud_rate_generator.v):** Bloque digital que genera una señal de sincronización basada en la frecuencia del sistema para muestrear correctamente los bits recibidos por la UART. Su salida define el pulso de habilitación para el receptor UART.
3. **UART Receiver (uart_receiver.v):** Captura la señal serial del GPS y reconstruye los bytes válidos. Implementa una máquina de estados que sincroniza, detecta bits de inicio y parada, y almacena cada byte decodificado en un bus de datos.
4. **FIFO Buffer (fifo.v):** Cola digital que almacena temporalmente los bytes recibidos del GPS. Actúa como desacoplador entre la velocidad de adquisición UART y el procesamiento posterior, previniendo pérdida de datos.
5. **NMEA Parser:** Módulo encargado de interpretar el contenido de las sentencias NMEA, extrayendo únicamente la información relevante como latitud y longitud. Filtra las tramas deseadas y transforma los datos a un formato más ligero para su transmisión.
6. **UART Transmitter (uart_transmitter.v):** Reenvía la información procesada al ESP32 en formato serial, asegurando compatibilidad con su interfaz UART. Este módulo convierte los datos binarios en flujo UART nuevamente.
7. **ESP32 LoRa TX:** Microcontrolador encargado de recibir los datos UART desde la FPGA y transmitirlos a través del protocolo LoRa. Este componente proporciona conectividad inalámbrica de largo alcance en bandas ISM (e.g., 915 MHz).

0.2.2. Diagrama de Bloques en Quartus

Al sintetizar los módulos en Quartus Prime, se obtiene un esquema lógico del diseño implementado. En la siguiente figura se muestra el diagrama general donde se evidencian las conexiones entre los módulos UART, FIFO y controladores internos.

0.2.3. Análisis del Baud Rate Generator

Uno de los módulos clave para la correcta recepción UART es el generador de baud rate. Su función es generar un `tick` de habilitación que divide la frecuencia del reloj principal para obtener una frecuencia de muestreo ajustada al protocolo UART. En este caso, para una entrada de 50 MHz, el módulo genera un pulso cada 5208 ciclos, correspondiente a 9600 baudios.

Los datos recibidos contenían coordenadas de latitud y longitud. Sin embargo, se observó que el número de decimales presentados por defecto era limitado (dos cifras decimales), lo cual redujo la precisión del posicionamiento. Esto generó un error aproximado de hasta 400 metros con respecto a la ubicación real detectada por herramientas como Google Maps. Este comportamiento se atribuye al formato y resolución de los datos extraídos directamente desde las sentencias NMEA sin procesamiento adicional.

Br. Cdad. Universitaria

Teusaquillo, Bogotá

[4.639184, -74.088811](#)

Figura 4: Posición aproximada que se obtuvo.

Br. Cdad. Universitaria

Teusaquillo, Bogotá

[4.636448, -74.091286](#)

Figura 5: Posición real en la que se tomaron los datos.

0.3.2. ESP32-LoRa

Para probar la comunicación inalámbrica, se configuraron dos módulos LilyGo TTGO LoRa32 (ESP32 + SX1276), uno como transmisor de datos y otro como receptor. La configuración inicial se realizó mediante el entorno Arduino IDE, estableciendo parámetros como frecuencia, ancho de banda y potencia de transmisión de acuerdo con las recomendaciones del fabricante.

Según el datasheet, estos módulos pueden alcanzar un rango de hasta 5 km en condiciones ideales (línea de vista sin obstáculos). En las pruebas realizadas en un entorno urbano con obstrucciones moderadas (principalmente vegetación), se logró una comunicación estable y continua hasta aproximadamente 1 km de distancia. A pesar de no alcanzar el máximo teórico, se evidenció una conexión robusta, sin pérdida de paquetes ni desconexiones notables, lo cual valida el correcto funcionamiento del sistema bajo condiciones reales. conexión fuerte y estable.



Figura 6: ESP32 LoRa - Emisor

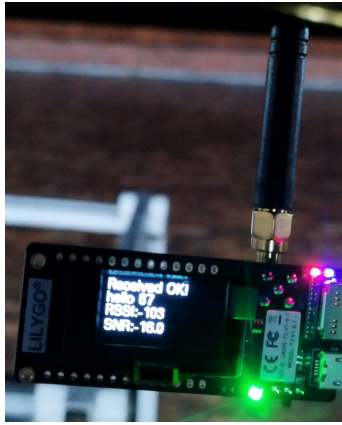


Figura 7: ESP32 LoRa - Receptor

0.4. Uso de Inteligencia Artificial

El uso de herramientas basadas en inteligencia artificial (IA) se ha vuelto una práctica común en entornos académicos y profesionales, facilitando la búsqueda de información, la generación de ideas y la depuración de código. En el desarrollo del presente proyecto, estas tecnologías jugaron un papel complementario en distintas etapas del proceso.

Durante la fase inicial de investigación y formulación teórica, se utilizó la herramienta Gemini, desarrollada por Google, con el objetivo de obtener referencias bibliográficas, definiciones técnicas y contexto general sobre protocolos como NMEA, LoRa y UART. Una de sus principales ventajas fue la capacidad de proveer enlaces directos a las fuentes consultadas, lo que resultó útil para validar y ampliar la información encontrada.

En contraste, para la fase de diseño e implementación en Verilog, se recurrió principalmente al uso de ChatGPT, desarrollado por OpenAI. Esta IA fue empleada para sugerencias de estructura de código, depuración de errores comunes, revisión de módulos como UART, FIFO y generación de reloj, así como para mejorar la documentación técnica del proyecto. Si bien su aporte fue significativo para resolver dudas puntuales y acelerar el proceso de codificación, hubo casos en los que la solución final requirió la intervención directa de la docente de laboratorio, la profesora Diana Maldonado, debido a la naturaleza específica del hardware o problemas de síntesis en Quartus.

En conclusión, el uso de inteligencia artificial se consolidó como una herramienta de apoyo que complementó el trabajo técnico, sin reemplazar la validación manual, las pruebas de laboratorio ni el acompañamiento docente necesario para el desarrollo exitoso del sistema.

0.5. Conclusiones

1. **Importancia del trabajo colaborativo:** La complejidad del proyecto, que involucró múltiples módulos digitales y protocolos de comunicación, dejó en evidencia la necesidad constante de apoyo entre compañeros, docentes y herramientas complementarias como las inteligencias artificiales. Enfrentar una problemática de esta magnitud requirió combinar conocimientos, dividir tareas de forma eficiente y recurrir a fuentes de conocimiento más experimentadas para resolver obstáculos específicos.
2. **Gestión del tiempo y cumplimiento de cronogramas:** Aunque el proyecto fue inicialmente propuesto para ser entregado como parte del reto de la materia, los retrasos en etapas intermedias llevaron a replantear la fecha de entrega. Esto evidenció la importancia de cumplir los cronogramas establecidos, anticipar posibles bloqueos técnicos y distribuir mejor la carga de trabajo para evitar acumulaciones en la etapa final.
3. **Relevancia de pruebas en condiciones reales:** Si bien se realizaron pruebas preliminares de comunicación y precisión, es evidente que se requieren escenarios más diversos para evaluar

la robustez del sistema. El comportamiento del GPS y la calidad del enlace LoRa pueden variar considerablemente según el entorno, por lo que se recomienda realizar más pruebas de campo en futuros desarrollos para asegurar su adaptabilidad.

4. **Desafíos en el diseño de sistemas con GPS:** El desarrollo de un sistema funcional de localización implica coordinar correctamente múltiples capas de hardware y software. Cualquier fallo en la captura, procesamiento o transmisión de datos afecta directamente la calidad del sistema. Este proyecto permitió comprender de forma práctica la dificultad de integrar tecnologías como GPS y LoRa, destacando la necesidad de precisión tanto en el diseño digital como en la implementación física.

Bibliografía

- [1] Apple Inc. (2021). *AirTag*. Recuperado de <https://www.apple.com/airtag/>
- [2] Actility & AbeeWay. (2023). *AbeeWay LoRaWAN Asset Tracker*. Recuperado de <https://www.abeway.com/>
- [3] Digital Matter. (2023). *LoRaWAN GPS Tracking Devices*. Recuperado de <https://www.digitalmatter.com/>
- [4] Juan R., Patiño J., Bolaños S. (2025). *Propuesta Proyecto – Airtag IoT con FPGA: Localización GPS y Transmisión LoRa* [PDF]. Universidad Nacional de Colombia.
- [5] LilyGo. (2020). *TTGO LoRa32 Series Development Board*. Recuperado de <https://www.lilygo.cc/products/ttgo-lora32-series>
- [6] National Marine Electronics Association. (2020). *NMEA 0183 Standard*. Recuperado de <https://www.nmea.org/>
- [7] Semtech Corporation. (2015). *LoRa Modulation Basics*. Recuperado de <https://www.semtech.com/lora>
- [8] Bolaños, S. (2025). *esp-lora32-fpga-cyclone-IV* [Repositorio GitHub]. Recuperado de <https://github.com/sergiosinlimites/esp-lora32-fpga-cyclone-IV.git>