

Sistema de Logging Geoposicional en Tiempo Real para Sistemas Empotrados

Autor:

Daniel Aguado Araujo

Directores:

Fernando Rincón Calle

Alberto Gómez de Dios

Índice

- Introducción
 - Marco de realización
 - Retos afrontados
- Objetivos
 - Sistema de Log
 - Comunicación con el GPS
 - Utilización del SO FreeRTOS
 - Integración del sistema
- Método de trabajo
 - Diseño Basado en Plataforma
 - Test-Driven Development

Marco de realización

- Este TFG forma parte del Proyecto Biker Assistant, de la empresa SmartSoC Solutions
- Consiste en la realización de un sistema electrónico que gestione y controle todos los subsistemas y comunicaciones de una motocicleta eléctrica



Marco de realización

- En el desarrollo de este proyecto intervienen los siguientes actores:



Objetivos del proyecto



Retosos afrontados

- Limitaciones del entorno de desarrollo
- Hardware impuesto por la empresa
- Hardware ofrecido sin soporte software
- Inestabilidad de los prototipos

Retos afrontados

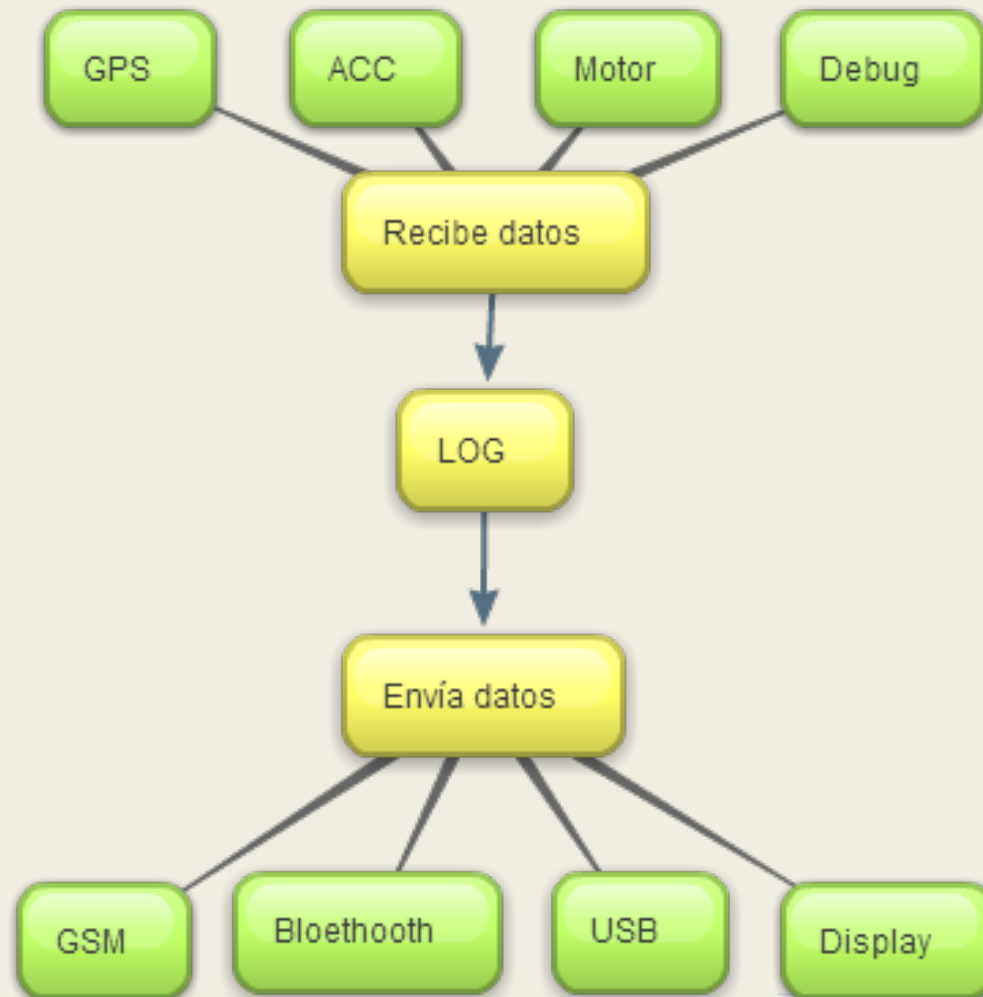
- Utilización de un prototipo hardware diferente al que se utilizará en el sistema final
- El prototipo no dispone de ningún componente de interacción con el usuario



Retos afrontados

- Necesidad de Tiempo Real
- Cantidad de datos almacenados
- Capacidad de la memoria disponible

Sistema de Log



Sistema de Log

Este sistema contiene los siguientes subsistemas:

- Driver de acceso a la memoria
- Sistema de archivos
- Aplicación de Log

Driver de acceso a memoria

Características comunes de la memoria en este tipo de sistemas:

- Palabra de 32 bits
- Operaciones de escritura de media, una o dos palabras y de lectura de hasta cuatro palabras
- No se puede reescribir sin antes realizar una operación de Erase o Borrado
- Las operaciones de Erase se hacen a nivel de sector físico completo

Driver de acceso a memoria

Características específicas:

- 1 MB de memoria
- El registro FLASH_SR contiene, en tiempo real, el estado de la última operación sobre la memoria
- 11 sectores físicos son de 16 KB, 64 KB y 128 KB

Driver de acceso a memoria

El fabricante proporciona una biblioteca de acceso a la memoria que nos ofrece la mayoría de las funcionalidades requeridas:

- FLASH_Lock
- FLASH_Unlock
- FLASH_EraseSector
- FLASH_ProgrammWord
- FLASH_GetStatus
- FLASH_WaitForLastOperation
- FLASH_ClearFlag

Sistema de Archivos

Principal reto:

- Los sistemas de archivos estándares y comerciales establecen el sector físico como unidad mínima de trabajo
- Tenemos únicamente 11 sectores físicos y demasiado grandes

Sistema de Archivos

Características del sistema de archivos implementado:

- Sector lógico de 16 bytes
- Acceso secuencial a los archivos
- Límite máximo de tamaño de archivo
- Límite máximo de número de archivos.

Sistema de Archivos

- Tabla de descriptores de archivo, con byte de validez y posiciones de inicio y fin de archivo
- Tamaño de un sector lógico

Validez	Nombre	Dirección de inicio	Dirección de fin
255	archivo	2048	2184
255	ex1.txt	4112	4140
255	1111111	4294967295	4294967295

Sistema de Archivos

Métodos de borrado:

- Borrar el Sistema de Archivos y crearlo de nuevo
- Crear un Sistema de Archivos en otro sector físico y:
 - mantener una copia del viejo hasta que queramos borrarlo
 - Hacer un backup del FS viejo en el nuevo

Sistema de Archivos

- f_mount
- f_mkfs
- f_open
- f_close
- f_read
- f_write
- f_lseek
- f_sync
- f_truncateStart
- f_getfree
- reset_sector
- change_sector

Comunicación con GPS

- El fabricante no proporciona biblioteca de acceso
- Se conecta a la placa a través de una USART
- Envía datos con el formato del estándar NMEA
- Los datos que se necesitan son los Datos Corregidos del Sistema de Posicionamiento Global, o GGA
- `$GPGGA,064951.000,2307.1256,N,12016.4438,E,1,8,0.95,39.9,M,17.8,M,,*65` (hasta 80 Bytes)

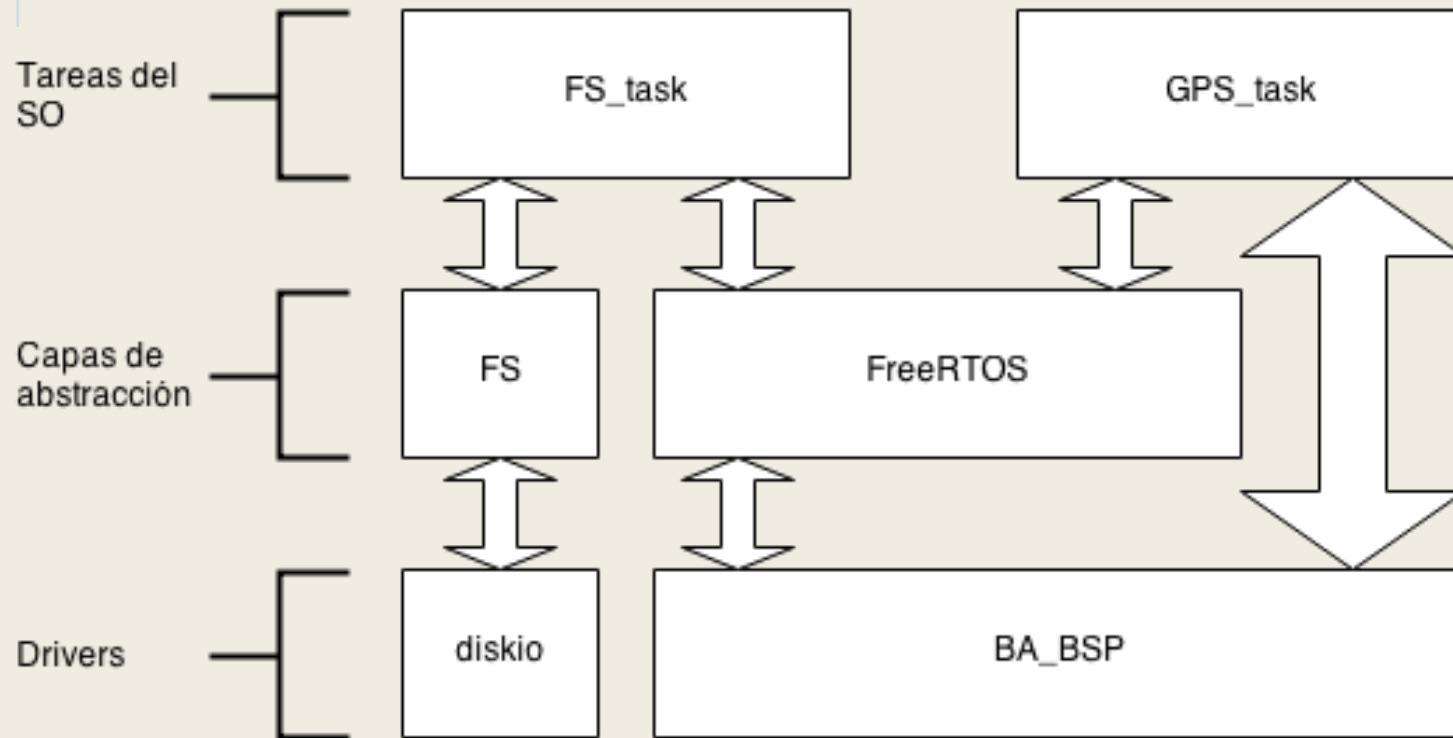
FreeRTOS

- Sistema de Tiempo Real
- Soporta multitud de arquitecturas
- GPL
- Cada función a realizar es una tarea del SO
- Cada tarea tiene su propio estado y su propia pila de datos, funcionando como programas independientes

FreeRTOS

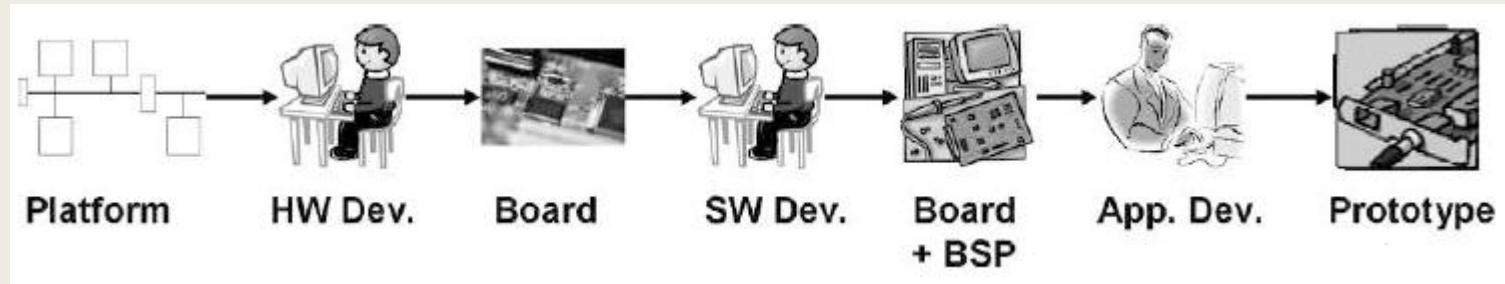
- Cada tarea tiene asignada una prioridad
- El planificador siempre ejecuta la tarea de mayor prioridad que esté preparada
- Solo puede ejecutarse una tarea a la vez
- Coordinación mediante semáforos, colas de mensajes e interrupciones

Sistema Final



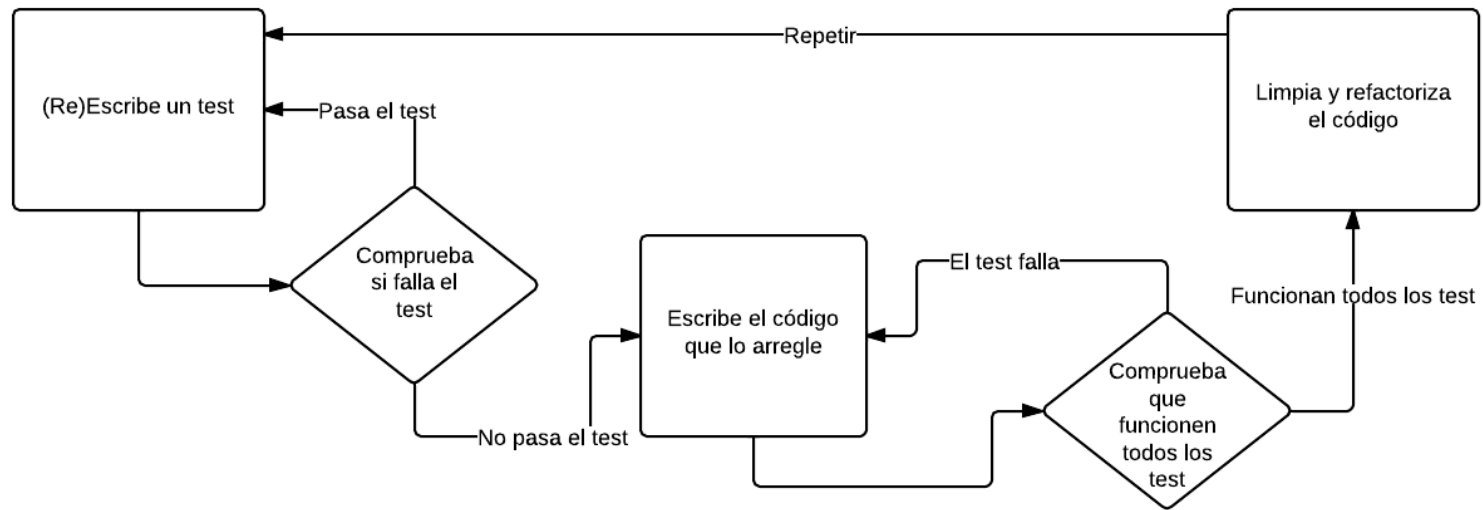
Método de trabajo

- Diseño basado en plataforma



Método de trabajo

- TDD: Desarrollo guiado por pruebas



Conclusiones

- Las restricciones de un sistema empotrado es lo que hacen difícil su programación
- No existe un entorno de desarrollo y pruebas estándar
- No tener componentes que ofrezcan interacción con el usuario ralentiza el desarrollo
- Poca documentación y de muy bajo nivel
- Utilizar estándares es bueno, pero no siempre es posible

Trabajo Futuro

- Cambiar la memoria o añadir memoria externa
- Estudiar la viabilidad de añadir Sistema de Archivos estándar sobre otro tipo de memorias
- Estudiar la viabilidad de un sistema de journaling para el Sistema de Archivos



¿Preguntas?