

MarchONs

Manuel Velásquez, Julian Sánchez, María del Mar Arbeláez

Objetivos del producto

El Smartwatch MarchONs busca proveer conteo de algunos datos básicos orientados a deportistas novatos o a usuarios aficionados que deseen saber más sobre su actividad física.

A través de monitores y sensores, permite al usuario saber calorías consumidas y distancia recorrida así como ritmo cardíaco y pasos caminados, incluyéndose también la funcionalidad básica de un reloj.



Diagrama de Bloques

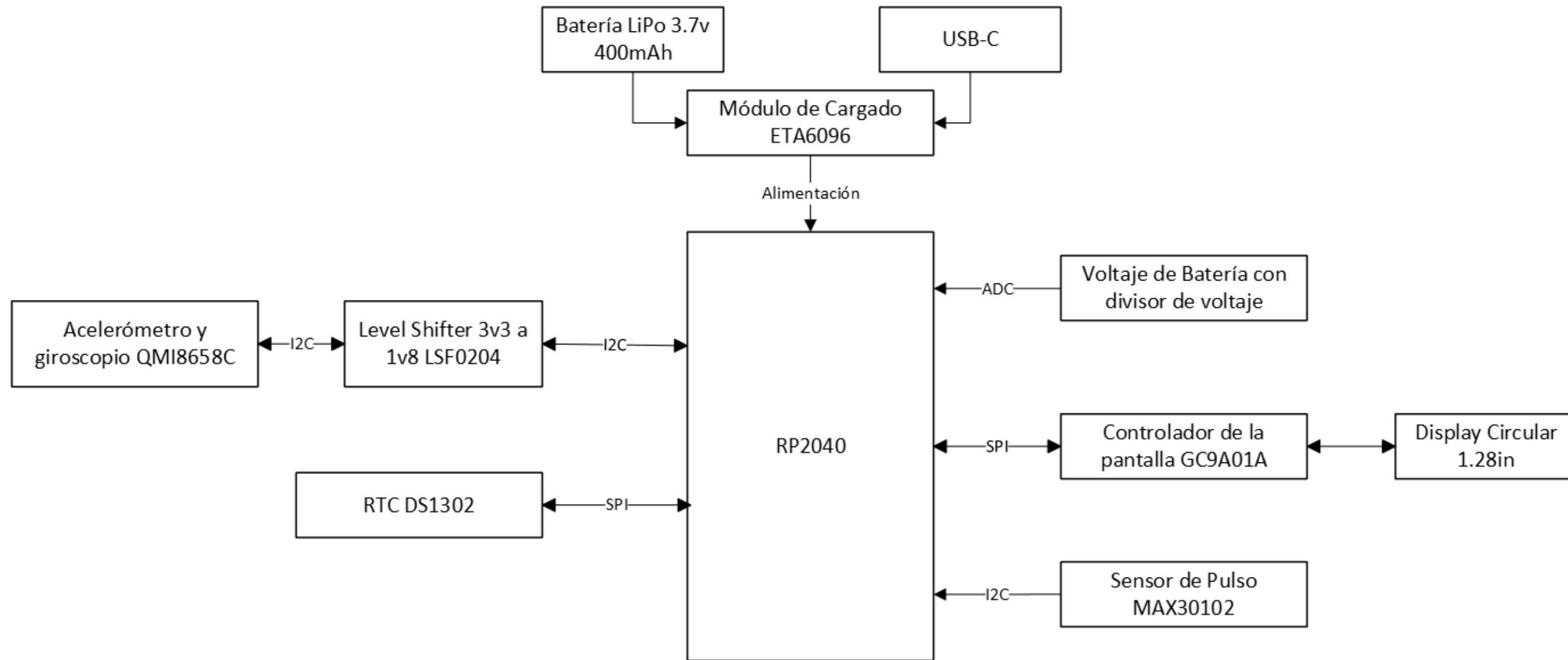


Diagrama de Bloques
MarchONs

Estructura de software

El software del producto está dividido en varios archivos y carpetas para facilitar la lectura del código y su utilización.

Se hace uso de una carpeta de código fuente y una carpeta de headers, con ambas incluyendo subcarpetas para cada utilidad.

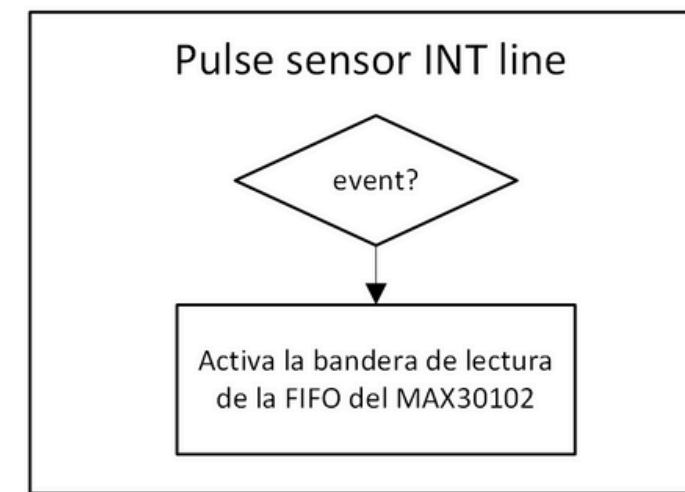
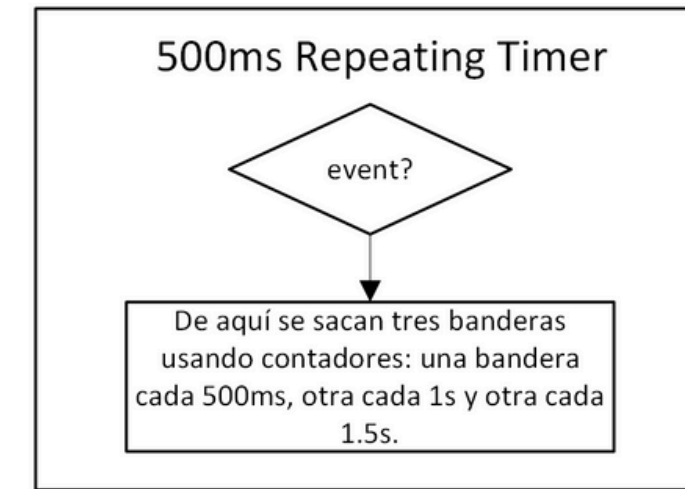
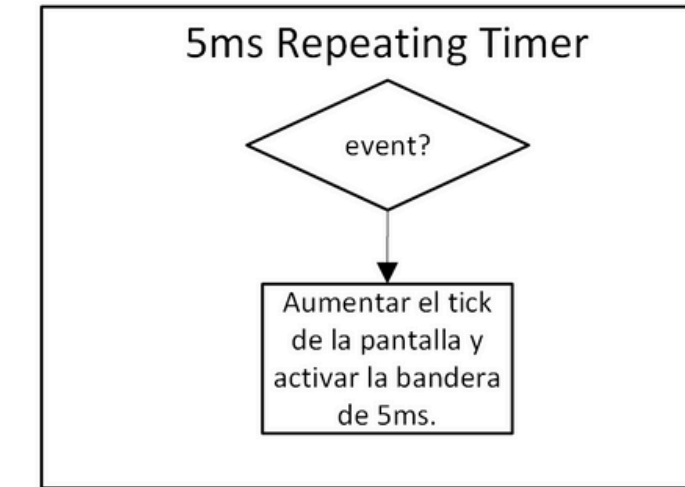
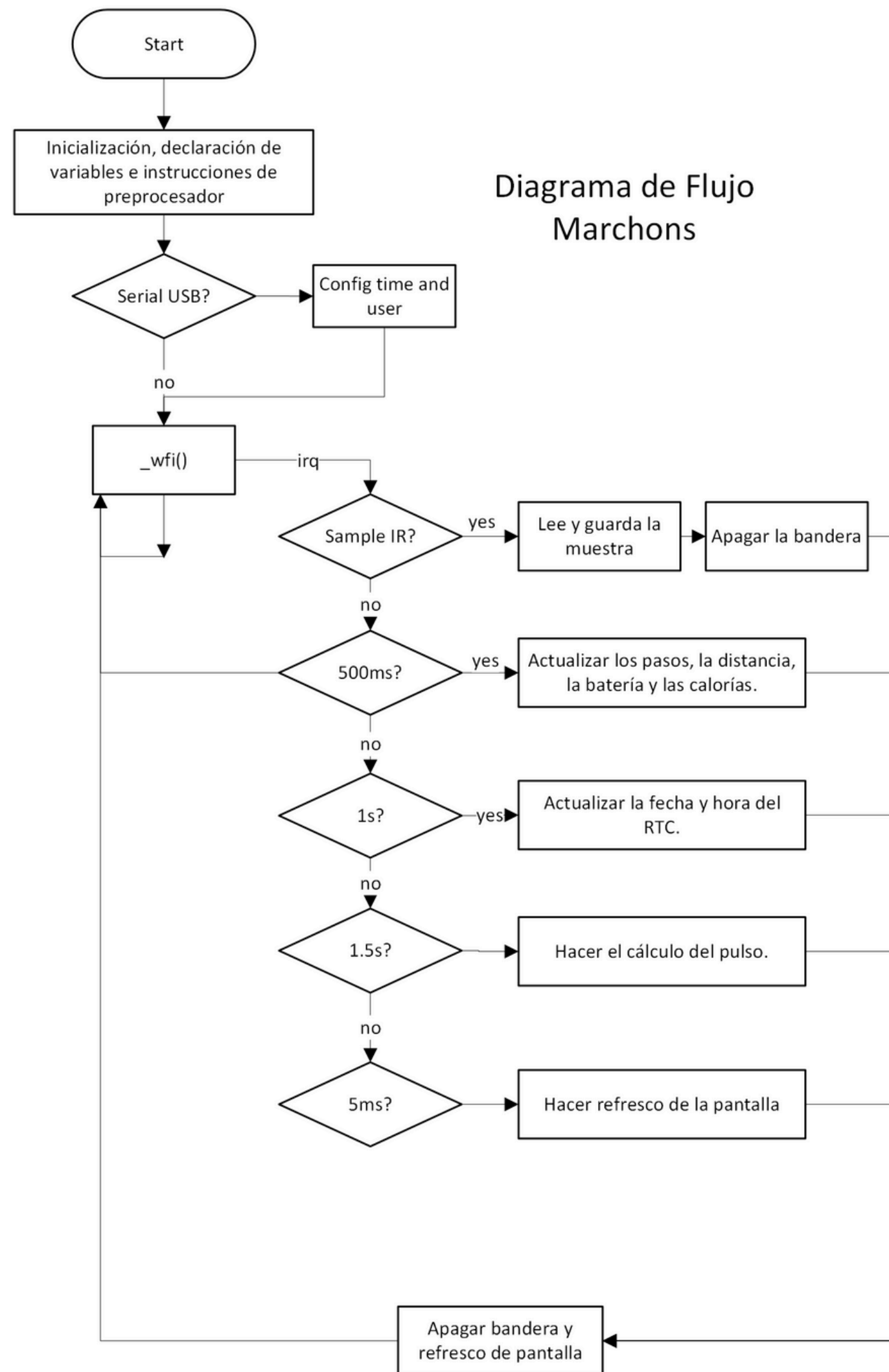
Se dividen en drivers, librerías y funciones que llaman a las dos anteriores para una tarea en específico, o para diferentes tareas como es el caso de los drivers de comunicación I2C y SPI.



Flujo del programa

El programa funciona de la forma interrupt más polling, haciendo uso de banderas de software que ejecutarán funciones una vez se detecten interrupciones ya sea por módulos, alarmas, comunicaciones, etc. El manejo se hace a través de la unión de cada funcionalidad por modulo escrita a la pantalla que se muestra cada 5 milisegundos.





Partes del sistema

IMU

Se comunica por I2C a la RP2040, tras pasar por un level-shifter. Permite no solo la lectura de la aceleración sino también de la medida de pasos a través de la configuración de thresholds.

BAT

Este módulo permite la lectura del estado de la batería a través de una señal analógica. Se hace uso del ADC para leer su estado. El circuito integrado también se hace cargo de la carga de la batería y de su uso de forma autónoma y no requiere ser programado.

LCD

Para el control de la pantalla se hace uso del DMA, iniciando cada vez que se escriba por SPI el cual es el modo de comunicación del módulo controlador (GC9A01A). Una vez sean cargados los objetos LVGL con sus respectivos valores se escriben en pantalla cada 5 milisegundos.

Partes del sistema



RTC

Se utiliza el DS1302 para la tener la fecha y hora en el reloj además de tener una memoria RAM para el guardado de datos. Se comunica con SPI, como el módulo no diferencia MISO y MOSI, fue necesario separarlos con una resistencia.



Pulse

Este módulo se encarga del tomar la medida generada por su LED IR, guardando datos en su FIFO, que es lo que se lee mediante la comunicación I2C. Estos datos se procesan en software para calcular el pulso.

Funcionales

- **Procesamiento de datos:** se hace principalmente en el cálculo de pulso, pues es suavizar la señal, generar una ventana y analizarla para encontrar el pulso.
- **Alimentación:** Batería con protección contra sobre-descargas y carga por USB C a través del PMIC incluido.
- **Interfaz de Usuario:** Pantalla LCD que muestra los datos de interés. El reloj es configurable cuando se inicia conectado a un puerto serial USB.

No Funcionales

- **Tiempo de respuesta:** Al rededor de 5 milisegundos entre cada refresco de la los objetos LGVL pantalla. Mayor para la actualización de las variables presentadas
- **Usabilidad:** Manual de funciones incluido.
- **Autonomía de la batería:** Aproximadamente una hora y media de funcionamiento autónomo.

Análisis Requisitos

Presupuesto

Elemento	Valor (\$COP)
Placa MCU RP2040 con LCD redondo y sensor de acelerómetro y giroscopio	93.415
RTC DS1302 Mini	5.593
Sensor de ritmo cardíaco MAX30102	19.000
Batería Litio-Polímero 3.7v 500mAh	16.800
Headers 1.27mm x4	16.123
Total	150.931





Gracias