

# Medidor de Velocidad Promedio

## Introducción

Este proyecto se implementa un sistema de medición de vueltas y visualización con LEDs y display de 7 segmentos sobre el kit STM32F4DISCOVERY (STM32F407G-DISC1) y la placa de expansión SD Board.

La idea es registrar los tiempos por vuelta de un auto, calcular la velocidad promedio y mostrar resultados en el display, aplicando lo visto de temporización y un modelo de control basado en *Máquinas de estados jerárquicas de Harel* (Fig. 1).

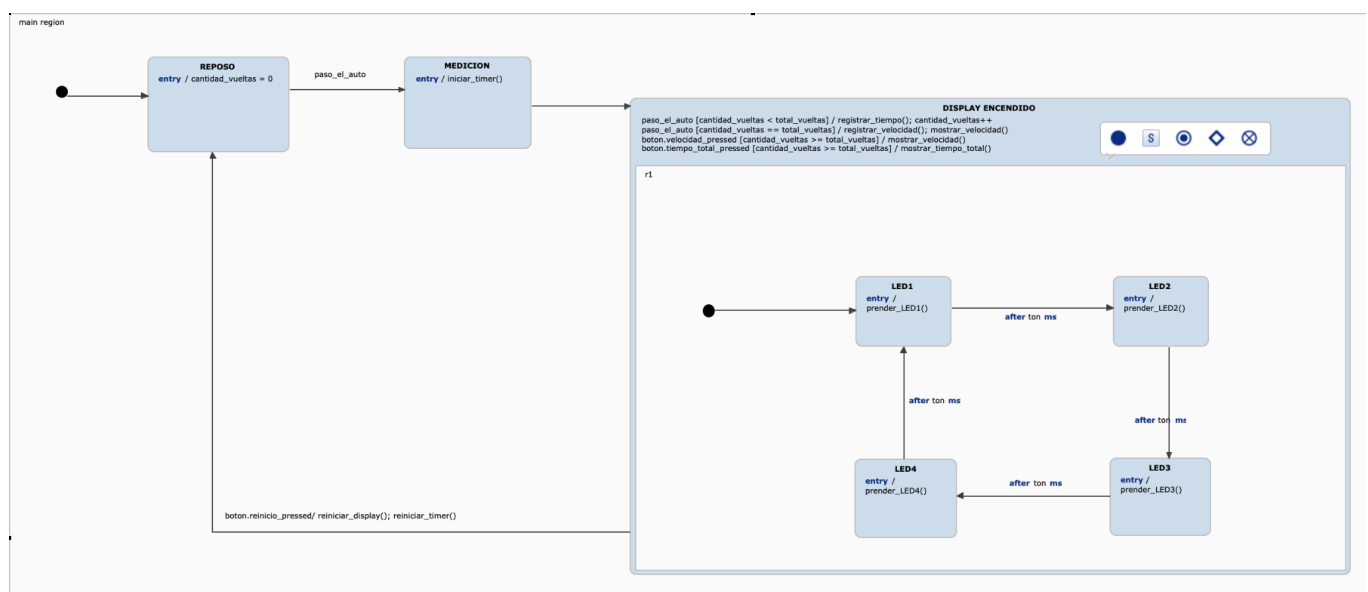


Figura 1. Modelo de control

En la fig. 1 se observa que el sistema parte en REPOSO, luego en el primer paso del objeto por el sensor lleva a MEDICIÓN, y ahí se inicia el temporizador para medir el tiempo por vuelta.

Por cada vuelta, se llaman operaciones para registrar el tiempo (mostrando además el tiempo en los LEDs del display por cada evento) Cuando el objeto alcanza las vueltas definidas, se habilita la visualización y en ese momento el usuario puede alternar entre velocidad promedio y tiempo total.

Dentro de MEDICIÓN corre una submáquina de LEDs, la cual es la responsable multiplexar el tiempo (LED1→LED2→LED3→LED4) (la cual posteriormente será temporizada en el HW).

Luego al detectar un evento de que se pulsó el botón Reset, se reinicia el display, y se manda la señal para reiniciar el timer, para luego volver al estado de REPOSO, el cual resetea variables y deja el timer listo.

El diagrama de flujo con la idea general, se puede ver en este [link](#).

## Arquitectura del sistema

Se implementó usando el kit STM32F4DISCOVERY (STM32F407G-DISC1) y la placa de expansión SDBoard.

Para mantener independiente el tiempo de la transición de estados con respecto a la medición del auto, se separaron los tiempos en dos temporizadores:

- Un timer TIM14 con IRQ. Con temporizador periódico para implementar por HW, la transición de estados compuestos para la submáquina de encendido de LEDs
- Un timer TIM12, para tomar el tiempo cada vez que se dispara el sensor. Se limitó los TICKs con PSC=15999, ARR = 65535, para obtener una fclk = 1Khz, y así obtener un tick de 1ms.

Se usó un sensor infrarrojo (IR) HW 201, configurado en PA7 con una interrupción de flanco de subida, para detectar cuando el objeto deje de reflejar.

Para mostrar los tiempos de cada vuelta, la velocidad promedio y el tiempo total se usó un display de 7 segmentos, multiplexado con temporización periódica con el con ISR del TIM14.

Y para alternar lo que se ve en el display, se usó tres pulsadores con interrupciones en PA2, PA4 y PA5 con flanco descendente, los cuales tienen estas funciones:

- Botón SW1: muestra la velocidad promedio en el display
- Botón SW2: muestra el tiempo total en el display
- Botón SW3: reinicia la medición, y se resetea el display

Se aplicó un filtro antirrebotes por SW, con la técnica de debounce, el cual consiste en aplicar una ventana de inhibición temporal en las interrupciones, en este caso se usó tanto para el sensor IR, como para los pulsadores.

Ademas se habilitó las interrupciones en el NVIC de la placa:

- EXT12, EXT14 y EXTI[9:5] habilitadas (para PA2/PA4/PA5 (pulsadores) y PA7 (Sensor IR)).
- TIM14 global interrupt habilitada, para el timer periódico

## Medición y cálculo

Cuando se detecta el auto por primera vez se inicializa el temporizador, para después por cada detección válida del sensor IR, se genere una interrupción, la cual en la implementación en el HW se toma el tiempo del timer TIM2, y se almacena en un array que almacena tantos tiempos como vueltas tenga la carrera (además irse mostrando en el display en cada vuelta).

Finalmente, cuando se llega a la cantidad de vueltas, al detectar el sensor IR se calcula la velocidad promedio de la siguiente forma:

$$v = \frac{L}{T_{Promedio}},$$
$$T_{Promedio} = \frac{\sum_{i=0}^N t_i}{N}$$

- La longitud  $L$  de la pista (dada por la cátedra)
- Cada tiempo  $t_i$  se obtiene del array con el historial de tiempos almacenados
- La cantidad de vueltas  $N$  de la carrera

## **Problemas encontrados y soluciones aplicadas**

### **Mezcla en el temporizador entre la transición de los LEDs del display y la medición.**

Una de las complicaciones a la hora de implementar funcionalidades en HW, fue que se necesitaba un timer que interrumpiera cada cierto periodo para mantener funcionando la máquina de estados para encender las luces. Pero por otro lado necesitaba tener un temporizador dedicado a tomar el tiempo cada vez que pasara el objeto, incluso que tuviese la opción de reiniciar.

Se resolvió configurando timers independientes: **TIM14 con IRQ** para la transición de los LEDs del display y **TIM12 libre** para cronometraje.

### **Doble disparo por rebotes.**

Para evitar el efecto del rebote, el cual activa las interrupciones muchas veces en un período muy corto, se implementó un debounce por SW en las ISR de EXTI tanto para el sensor como los pulsadores), se aplica ventana de inhibición descarta flancos si no pasó un mínimo de tiempo.

### **Detecciones erróneas en el sensor.**

En el caso de que el objeto quede en el sensor, la idea es que no marque el inicio del temporizador, u otra vuelta (según sea el caso). Para solucionar eso se configuró en el PIN del sensor (PA7) con flanco de subida, para que se detecte el momento en que el objeto deja de reflejar, esto combinado con el antirrebote disminuyó falsos positivos.

### **Parpadeo del display de 7-segmentos.**

Se producía un parpadeo debido al multiplexado para cada dígito del LED, para solucionarlo se pudo ajustar el tiempo de transición de los estados de transición de LED, y configurar el dato en la implementación del timer **TIM14** en el HW, manteniendo un período constante y actualizando el buffer de dígitos fuera de la ISR.

## **Propuestas de mejora**

### **Punto flotante en el display**

Para facilitar las pruebas, se muestra el tiempo de cada vuelta en el display en segundos lo cual implica un redondeo (aunque si registro el valor completo en el array de tiempos (en ms), cuando se calcula la velocidad, se lo hace con los datos sin redondeo, por lo cual se puede llegar a percibir la diferencia respecto al redondeo mostrado en cada vuelta. Una mejora sería agregar un punto flotante en el display de 7 segmentos.