



COCHE TELEDIRIGIDO

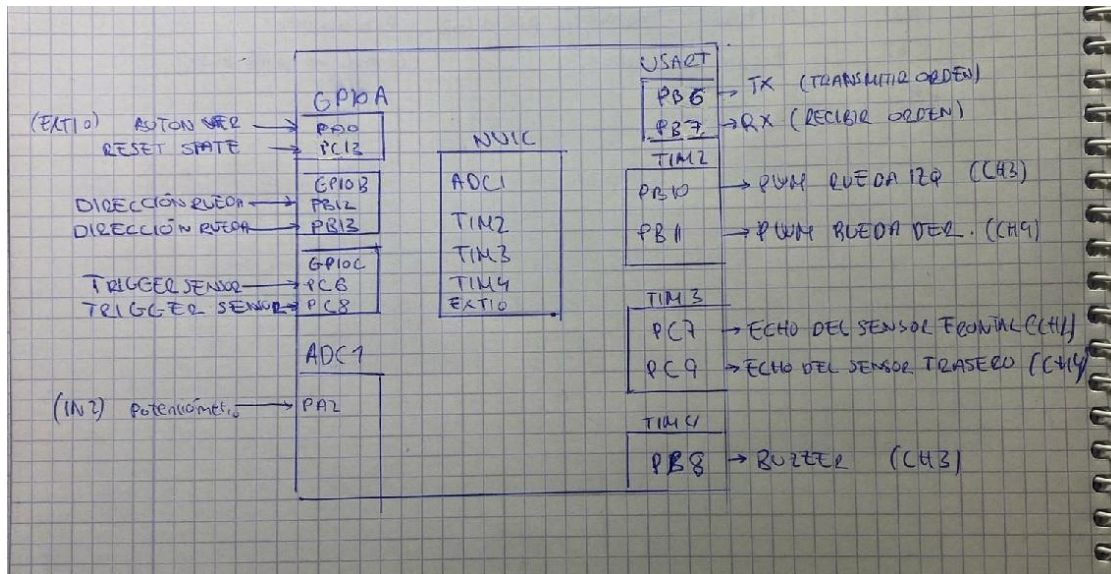
SISTEMA BASADO EN LA PLACA BASE: STML32



INDÍCE

1. Diagrama en bloques.....	2
2. Esquemáticos y descripción de las conexiones, mostrando la placa utilizada y los componentes..	3
3. Breve explicación sobre los periféricos del microcontrolador utilizados y su configuración básica.	4
4. Qué interrupciones (IRQ) se han utilizado y la funcionalidad lograda por sus rutinas de servicio de interrupción (ISR).	5
5. Diagramas de flujo de las ISRs desarrolladas.	8

1. Diagrama en bloques



2. Esquemáticos y descripción de las conexiones, mostrando la placa utilizada y los componentes

Pin	Función	
A0	Entrada digital + interrupción	Botón USER (interrumpe con EXTI0)
A2	Entrada analógica (ADC1_IN2)	Potenciómetro para velocidad
B8	Función alternativa (TIM4_CH3)	Control de sonido (buzzer)
B10	Función alternativa (TIM2_CH3)	PWM rueda izquierda
B11	Función alternativa (TIM2_CH4)	PWM rueda derecha
B12	Salida digital	Dirección rueda izquierda
B13	Salida digital	Dirección rueda derecha
C6	Salida digital	Trigger del sensor frontal
PC7	Función alternativa (TIM3_CH2)	Echo del sensor frontal
PC8	Salida digital	Trigger del sensor trasero
PC9	Función alternativa (TIM3_CH4)	Echo del sensor trasero
PB6	TX	Transmitir comandos USART
PB7	RX	Recibir órdenes USART

3. Breve explicación sobre los periféricos del microcontrolador utilizados y su configuración básica.

3.1. Motores y Ruedas

Básicamente los componentes que hacen que el coche se pueda mover hacia todas las direcciones. Los pines utilizados son los siguientes: PB10 y PB11 configurados como función alternativa(10), se usa un timer PWM para cada rueda (TIM2_CH3 y CH4) que controla la velocidad mediante los pulsos del timer. También se usan los pines PB12 y PB13 en este caso como digital output(01) que se encargan de del cambio de dirección y los giros dependiendo si se activan o se desactivan.

3.2. Botón User

El botón User lo utilizamos para dar comienzo al programa, el pin PA0 es el que se encarga de su configuración como digital input(00) y también usa interrupciones EXTI[0] (0000)

3.3. Potenciómetro

El potenciómetro se encarga de regular la potencia suministrada, es decir regula cual será la velocidad máxima dependiendo de su configuración, para este periférico se usa el pin PA2 con configuración analógica(11), que usa un ADC que hace que lea su valor cada 2 segundos y también usa interrupciones. (NVIC->ISER[0])

3.4. Ultrasonidos

Usamos dos ultrasonidos uno en la parte delantera y otra en la trasera, resumidamente se encargan de detectar obstáculos a cierta distancia del coche y avisar de su presencia. Se dividen en dos partes:

Para el Trigger usamos los pines PC6 Y PC8 (delantero y trasero) configurados como digital output (00), envía pulsos cada 10 microsegundos para esto se usa un timer en modo TOC (TIM4_CH1).

Para el Echo usamos los pines PC7 y PC8 (delantero y trasero) configurados como función alternativa (10), y lo que hace es recibir las señales del sensor, se usan timers en modo TIC (TIM3_CH2 y CH4) para medir el tiempo que tarda en recibir la señal desde que se envía y así calcular la distancia con el objeto

3.5. Zumbador

El zumbador (altavoz) se encarga de emitir sonidos intermitentes cuando los ultrasonidos detectan un objeto a menos de 30 centímetros y uno sonido continuo cuando está a menos de 5 centímetros. Usamos el pin PB8 en función alternativa (10) y un temporizador en modo TOC (TIM4_CH3) que hace que el zumbador suene cada 50ms.

3.6. USART

EL USART lo usamos para conectar el coche con nuestros dispositivos móviles mediante bluetooth, los pines usados son el PB6 y PB7 en función alternativa (10) que envían (PB6 TX) y reciben (PB7 RX) los datos.

4. Qué interrupciones (IRQ) se han utilizado y la funcionalidad lograda por sus rutinas de servicio de interrupción (ISR).

En cuanto a las IRQ que hemos usado para desarrollar el proyecto, son:

- EXTI0_IRQHandler
- ADC1_IRQHandler
- TIM3_IRQHandler
- TIM4_IRQHandler

4.1. void EXTI0_IRQHandler(void)

En un primer momento el coche va a empezar a avanzar hacia delante e irá detectando los obstáculos, pero en el momento que toquemos el botón USER.

Este método comprueba el flanco de subida en la línea 0 ,es decir, cuando pulsemos el botón. En este momento el coche invertirá el sentido de su marcha y activando los triggers del mismo sentido pero en cambio desactivando los del sentido opuesto.

4.2. void ADC1_IRQHandler(void)

Para el proyecto usamos el ADC para controlar los valores de la velocidad máxima mediante los rangos del potenciómetro dividido en 10 secciones.

Cuando el ADC termina de realizar una conversión (de analógico a digital), esta función se ejecuta automáticamente si la interrupción está habilitada, para ello, verificamos si la conversión ha terminado viendo el bit End Of Conversion del SR.

En el caso de que haya finalizado la conversión, guardamos ese valor del registro de datos(DR) y así poder adaptar la velocidad a la que va el coche.

4.3. void TIM3_IRQHandler(void)

Para los sensores de ultrasonido usamos para el delantero el canal 2 del Tim3 y para el trasero el canal 4 del Tim3 , para capturar el tiempo de llegada de señales (pulso reflejado por el obstáculo) y gracias a ello, calcular la distancia a la que se encuentra el objeto.

Ambos sensores , delantero y trasero, funcionan de la misma manera y se activan y desactivan dependiendo del sentido de la marcha, si el coche va hacia delante funciona el sensor delantero y el trasero se desactiva y viceversa.

En el caso de que estemos esperando el primer flanco de la señal del echo, se activa el modo de recepción para ver el flanco de bajada que es cuando vuelve la señal.

En el caso de que ya haya llegado la señal, se vuelve al modo de esperar a que llegue un flanco de subida de la señal de echo.

El valor capturado es la diferencia entre el tiempo actual guardado en CCRy(2 para el delantero y 4 para el trasero) y un tiempo previo del delay cuando se genera el trigger.

Por último limpiamos el flag cada canal.

4.4. void TIM4_IRQHandler(void)

Utilizamos diferentes canales del Tim4 para este hito:

- Tim4 ch1 para poner el trigger a 10us.
- Tim4 ch2 para medir el valor del potenciómetro cada 2 segundos.
- Tim4 ch3 para el buzzer.

Si se activa la interrupción del ch1 y el trigger estaba emitiendo y le llega la señal de echo de 500us, pasa a estar leyendo la señal y si el sensor estaba leyendo ,pasa a que el trigger emita la señal de 10us.Por último hay que limpiar el flag de la interrupción del canal 1.

En el caso de que se active la interrupción del canal 2, hacemos que avise cada 1ms contando 2000 avisos. Cuando avise 2000 veces habrán pasado 2 segundos.

Si el coche está en el modo de buscar una salida debido a un obstáculo, se cuentan las veces del giro de este.

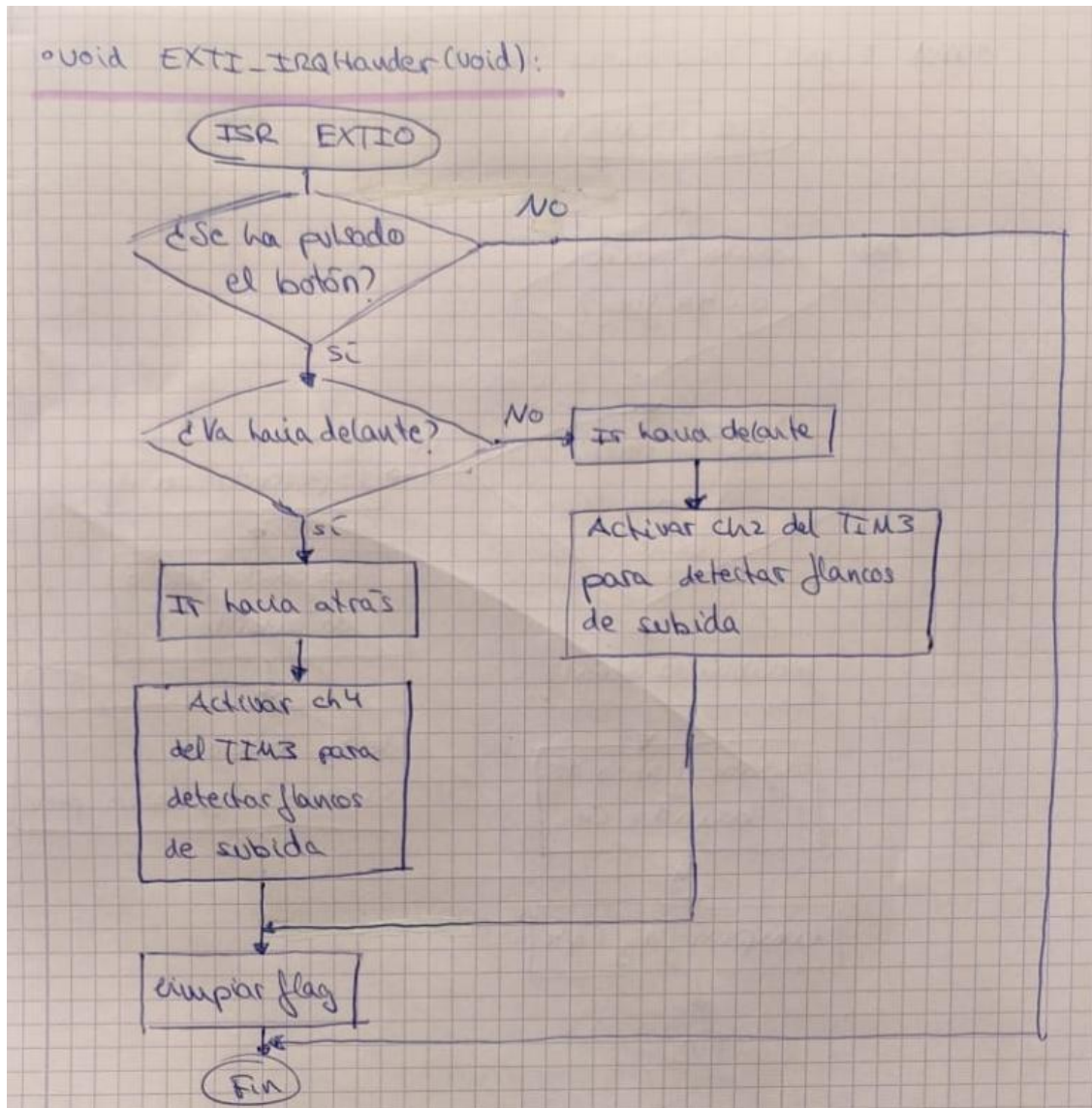
Cuando haya 2000 avisos de 1ms, es decir, 2 segundos ,se lee el valor del potenciómetro. Por último tenemos que limpiar el flag de la interrupción del canal 2.

En el último caso , en el que se activa la interrupción del canal 3 y hacemos que el zumbador suene cada 50ms.

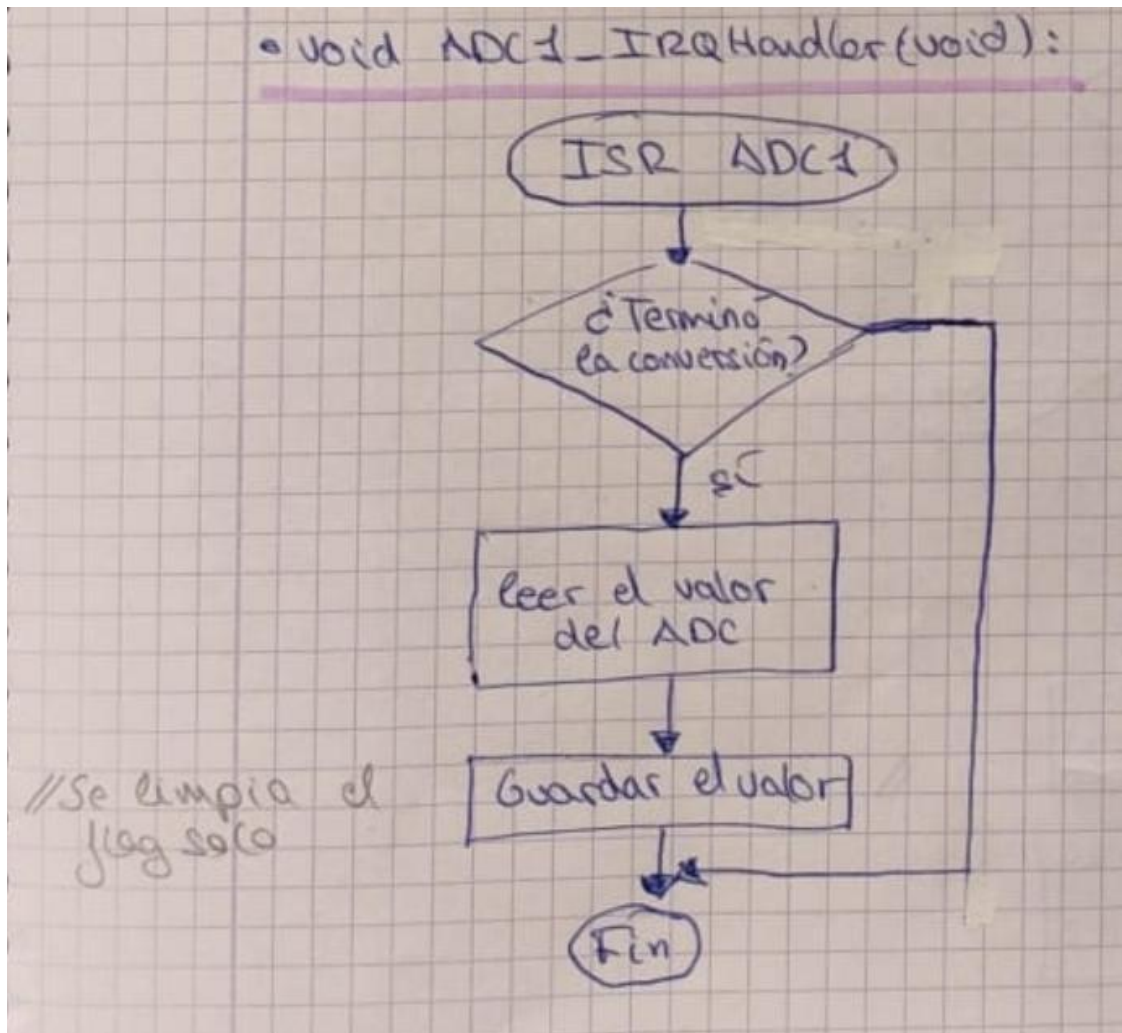
Como en todos los casos , acabamos limpiando el flag de la interrupción en este caso del canal 3.

5. Diagramas de flujo de las ISRs desarrolladas.

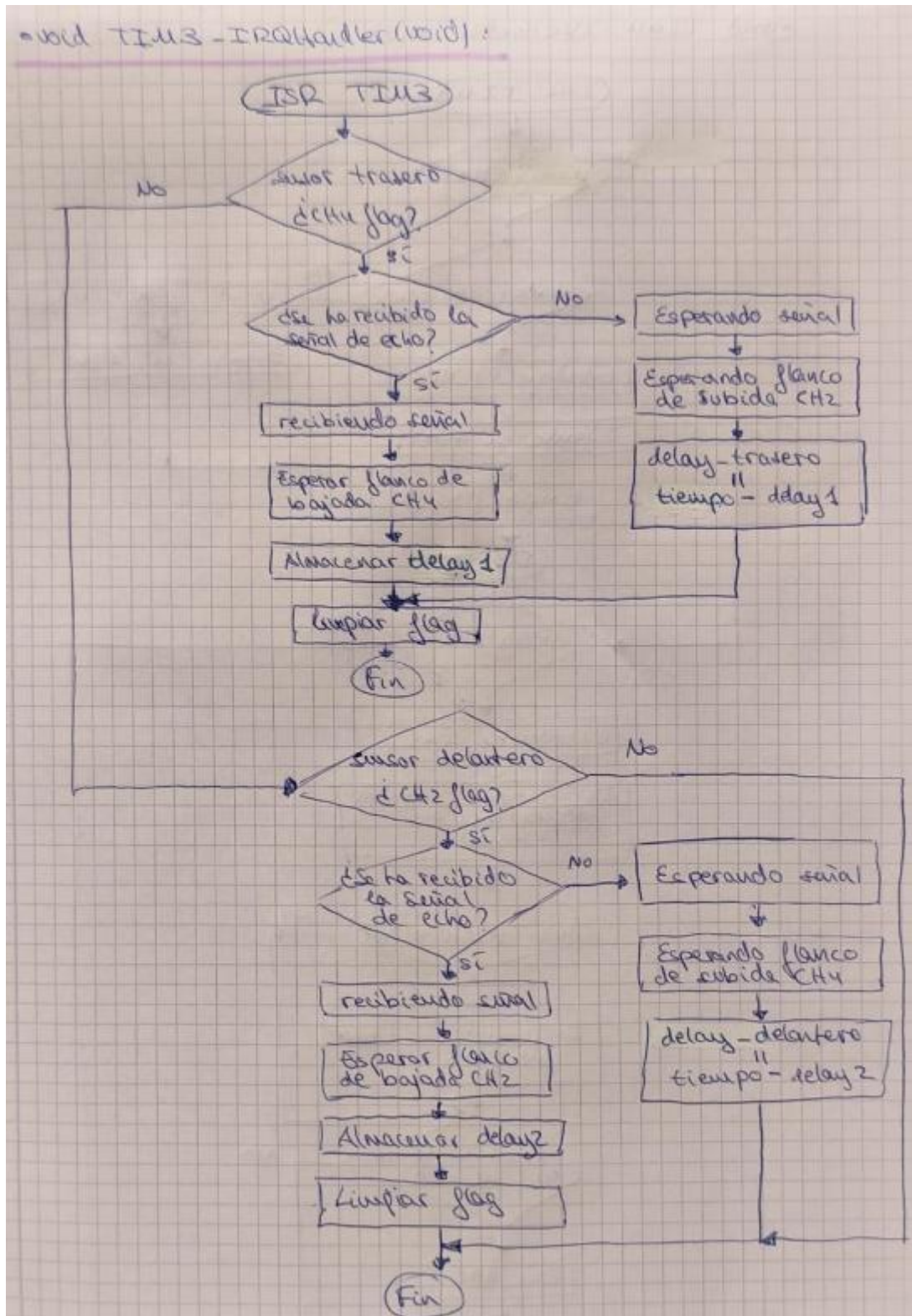
5.1. EXTIO



5.2. ADC1



5.3. TIM3



5.4. TIM4

