

Práctica final: Competición de mini-sumo

Andrés Casasola Domínguez
Juan Manuel Vázquez Jiménez
Asignatura: Microbótica
Febrero 2020

I. RESUMEN

En este proyecto se va a desarrollar un robot de dos ruedas para competir en luchas de mini-sumo. En la figura 1, se puede ver una imagen del robot.

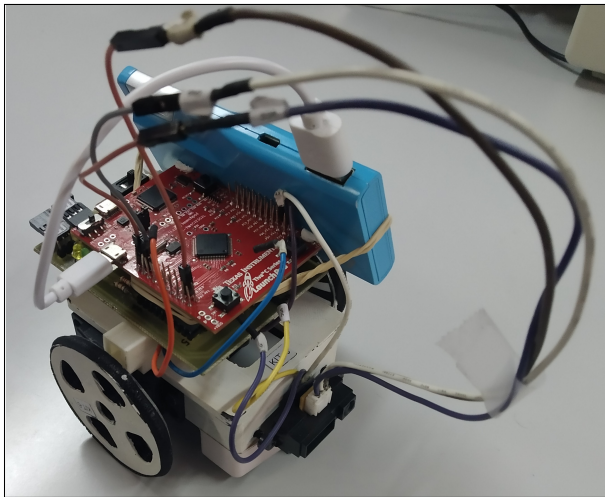


Fig. 1. Imagen del robot montado

II. CARACTERISTICAS

El robot utiliza dos sensores para moverse por la pista sin salirse, detectando obstáculos y reaccionando ante ellos. El sensor que detecta cuando se está saliendo de pista, es un encoder y el que detecta obstáculos, es el sensor de distancia Sharp GP2Y0A41SK0F. Además, el robot tiene una batería acoplada en el lateral con masilla moldeable y un par de elásticos. Tanto el encoder como el sensor de distancia están posicionados en la cara frontal del robot, por lo tanto, el robot siempre se desplazará hacia delante.

III. JERARQUIA DE FICHEROS

Para la implementación se han creado librerías que abstraen el código main del bajo nivel de los timers para el PWM o para el ADC, por ejemplo.

De esta forma, mediante el uso de algunas funciones sencillas e intuitivas, se puede cambiar el estado de movimiento y sensado del robot. En la figura 2 se puede ver un diagrama en el que se representa la jerarquía de ficheros.

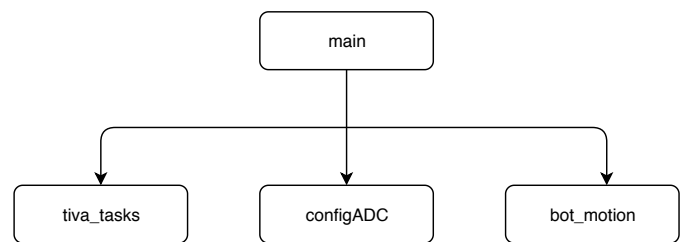


Fig. 2. Jerarquía de los ficheros mas representativos en el proyecto.

IV. COMPORTAMIENTO

El comportamiento del robot es puramente reactivo, es decir, el robot siempre se desplazará hacia delante a la espera de eventos, cuando detecta que se sale de pista, realiza un giro de 180 grados, cuando detecta un obstáculo realiza una maniobra de flanqueo para posicionarse detrás del obstáculo y continua hacia delante. Si el robot comienza una maniobra de flanqueo y detecta que se sale de pista, aborta la maniobra y realiza un giro de 180 grados.

La traducción en cuanto al funcionamiento del sistema consiste en una espera continua de interrupciones. Las fuentes de interrupciones pueden ser o un flanco en el sensor encoder o un dato proveniente del ADC. Cada interrupción es tratada por una tarea, por tanto, existen dos tareas, una para procesar los cambios en el encoder y otra para procesar los datos del ADC. Es importante aclarar que la tarea del encoder activa o bloquea la tarea del sensor, de esta forma, no salirse de pista es siempre el objetivo prioritario.

V. ESPECIFICACIONES

Las especificaciones cumplidas por este diseño son de nivel 1, comportamiento reactivo.

VI. DIAGRAMAS DE FLUJO DEL CODIGO

En la figura 3, se puede ver el diagrama de flujo del fichero principal, *main.c*.

En la figura 4, se puede ver la función de configuración, *config()*.

En la figura 5, se puede ver la tarea del encoder de mayor prioridad, *ReactiveTask()*.

En la figura 6, se puede ver la tarea del ADC de menor prioridad, *SensorTask()*.

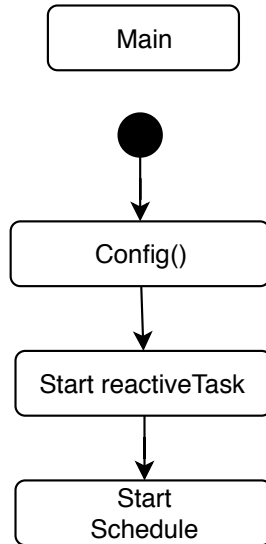


Fig. 3. Diagrama de flujo del fichero *main.c*.

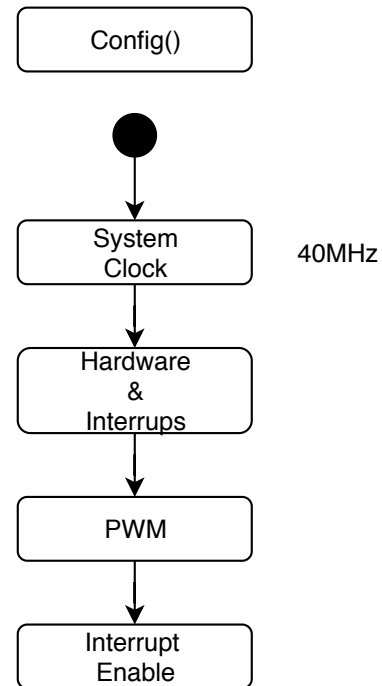


Fig. 4. Diagrama de flujo de la función de configuración *config()*.

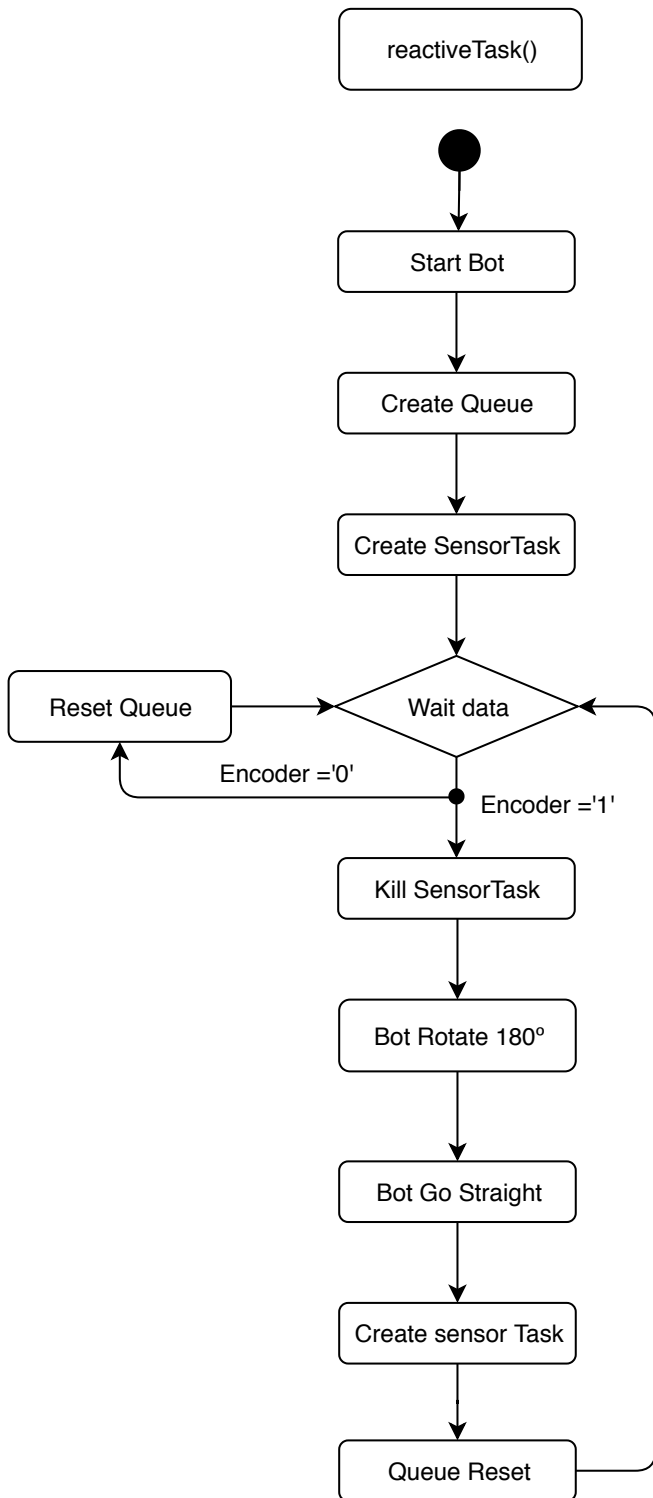


Fig. 5. Diagrama de flujo de la tarea *ReactiveTask()*.

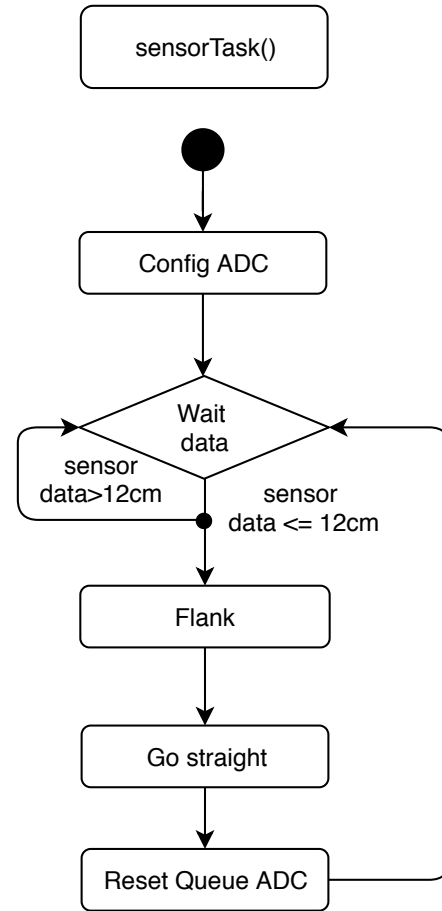


Fig. 6. Diagrama de flujo de la tarea *SensorTask()*.