

# Practica 2: Caracterización y uso de sensores en el Skybot(II)<sub>v4</sub>

Ignacio Herrero Reder

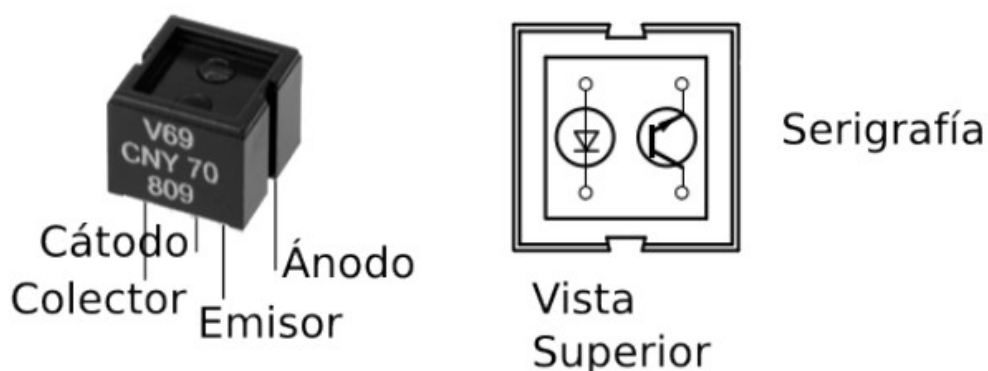
23 de Octubre de 2017

## Introducción

En esta parte de la práctica estudiaremos el conexionado de todos los sensores de nuestro robot, ya que vamos a comenzar a utilizar también los sensores CNY70, por lo que se harán necesarias tomas de alimentación extras y circuitos de polarización.

Una vez establecida la conexión de los sensores, acoplaremos a las ruedas del robot unos discos rayados en blanco y negro para permitir la medida de fracciones de vuelta, construyendo nuestro propio encoder. Mediante este encoder podremos realizar tareas de medición de la odometría de nuestro robot y controlar aspectos como la velocidad o la distancia recorridas.

Los CNY70 son sensores que integran un led emisor y un fototransistor infrarrojos en un único encapsulado. Esta configuración permite que la luz emitida por el led sea captada por el fototransistor si rebota en un objeto a poca distancia. Si un objeto es de color oscuro, probablemente absorba las frecuencias infrarrojas, de forma que para el CNY-70 será como si no se detectara nada o el obstáculo estuviese muy lejos.

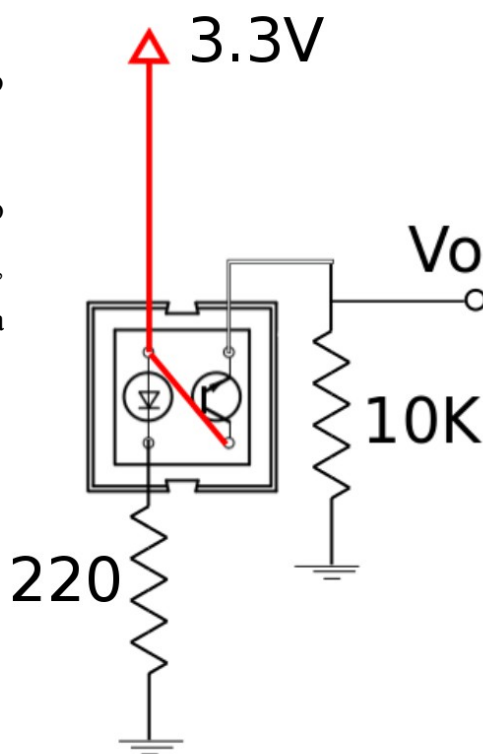
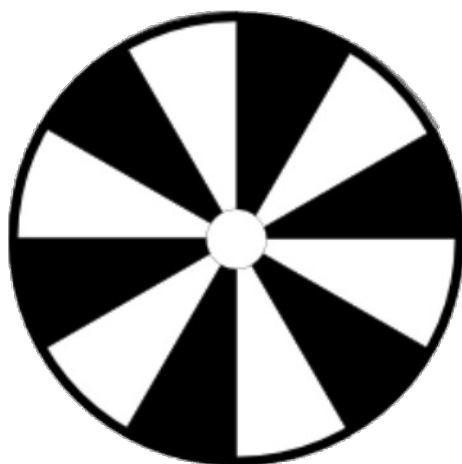


Es posible utilizarlo como un sensor analógico, pero nosotros nos ceñiremos a detectar blanco/negro. Para ello, recurriremos a la siguiente configuración:

Cuando el fototransistor esté muy cerca de un objeto reflectante (claro) tendremos un nivel alto.

Por contra, cuando el objeto absorba el infrarrojo (oscuro) tendremos un nivel bajo.

En esta práctica, utilizaremos un disco coloreado, como el del dibujo inferior, en el interior de la rueda del robot, para contar las transiciones blanco/negro y así, obtener la odometría.



En el laboratorio disponemos de sensores CNY70 ya contruidos, en los que las conexiones y resistencias están integradas en el propio cable, de forma que solo es necesario conectar sus 3 entradas a las respectivas de 3.3V, GND, y Vo. También disponemos de sensores sin polarización por si alguien desea implementar su propia placa de conexión de sensores.

## Ejercicio 1. Construcción/Adaptación del circuito de sensores.

Si consideramos sensores analógicos SHARP como los de la primera parte de la práctica, necesitamos, para conectar los sensores, 1 señal de 5V y 1 de tierra(GND) (por cada SHARP); 1 de 3.3V y una de tierra(GND) (por cada CNY70); y 1 pin de entrada para la lectura de los datos en cada sensor. En la parte delantera, la placa de potencia tiene un array de 10 pines donde son accesibles todas las alimentaciones del robot, varios pines de tierra, así como cuatro pines de entrada al microcontrolador (ver [diagrama de conexiones](#)). En principio se consideró una conexión fija a estos pines de los sensores (los CNY70 a PA2 y PA3; y 1 SHARP a PE5, que incluye canal de

entrada al ADC. No obstante, experiencias de cursos pasados en los que malas conexiones llevaban a la destrucción de dichos pines nos han llevado a considerar una mayor libertad de elección en los pines donde conectar los sensores. En todo caso, se necesitarán suficientes señales de alimentación y tierra para los sensores considerados, por lo que es posible que haya que añadir una pequeña placa de expansión de alimentaciones. Por ahora, para usar únicamente un SHARP (5V) y dos sensores CNY70(3.3V) nos basta con las tomas de alimentación que proporciona la propia TIVA + BOOSTERPACK (2 de 3.3V; 2 de 5V; y 6 de Tierra).

Elige un par de pines de E/S digital y conecta allí las salidas de datos de los sensores CNY70. Conecta el resto de señales a los pines de alimentación correspondiente.

- Para comprobar que los sensores funcionan correctamente, entrega un código que conmute los leds verde y rojo según las transiciones blanco/negro de los dos CNY70.

## Ejercicio 2. Construcción y prueba del encoder.

Una vez conectados los sensores CNY70m hemos de instalar en las ruedas los discos blanco/negro para contar las vueltas. En la última página dispones de varias configuraciones, para diferentes resoluciones. Hemos de recortar el disco e insertarlo en la cara interior de las ruedas. A continuación fijaremos el CNY70 en el interior, cerca de la rueda, de manera que sea capaz de leer las transiciones de blanco/negro.

Material necesario para este ejercicio:

- Tijeras para recortar los discos
- Blu-tac -o pegamento- para fijar los discos al interior de la rueda (proporcionado por nosotros)

Aunque no es indispensable, se recomienda fijar previamente el disco blanco/negro a un material un poco más rígido – como el cartón- antes de pegarla a la rueda<sup>1</sup>.

Cuando esté montado el sistema, realiza el siguiente código para verificar que todo funciona correctamente:

- Entrega un código que cuente los pulsos del encoder de cada rueda mediante interrupciones, y conmute cada led cuando se complete una vuelta completa de la rueda.
- **Nota:** como los PWM de la TIVA van conectados a los LEDS *on-board* de la placa, no es posible usar estos LEDS a la vez que se usen los motores. Por ello, si vas a integrar esta práctica con la de control de motores, **deberás añadir LEDS externos a tu placa TIVA -es muy fácil, pero pregunta a los profesores si tienes dudas-**

---

<sup>1</sup> Algunos estudiantes de años pasados han conseguido mejores resultados plastificando las hojas donde se imprimen ruedas antes de recortarlas e instalarlas.

## Ejercicio 3 – Comandos de movimiento del Skybot

Ahora que somos capaces de contar el desplazamiento de cada rueda, es posible conocer el desplazamiento del robot. En nuestro caso, para un robot diferencial con ruedas de radio  $R$ , la ecuación que describe la distancia recorrida ( $D$ ), en función de los ángulos recorridos por cada rueda ( $\theta_l$  y  $\theta_r$ ) es:

$$D = \frac{R}{2} \cdot (\theta_l + \theta_r)$$

Asimismo, el ángulo girado por el robot, en función de los ángulos girados por cada rueda, y la separación entre ruedas ( $l$ ) es:

$$\theta = \frac{R}{l} \cdot (\theta_l - \theta_r)$$

Utilizando los *encoders* instalados en el ejercicio anterior, diseña un código con dos funciones (o puedes usar tareas si usas FreeRTOS, adelantando ya trabajo de la práctica final):

- `mover_robot(c)`: para desplazarse  $c$  centímetros (adelante o atrás).
- `girar_robot(g)`: para girar  $g$  grados en cualquier sentido.

Vamos a considerar que los ángulos que han de recorrer ambas ruedas son los mismos ( $\theta_l$  y  $\theta_r$ ), para simplificar las ecuaciones. Además, ten en cuenta que la precisión de estos movimientos dependerá en gran medida de la resolución de la rueda del *encoder* que utilizaste, por lo que podrías necesitar instalar uno más sensible.

- Entrega un código que haga al robot trazar un rectángulo de 12 x 18 cm. Sube **también** un video al canal YOUTUBE de la asignatura con el nombre *Practica2BEj3-GrupoX* mostrando al robot trazando el rectángulo.

## Ejercicio 4 – Reflexión sobre aplicación de los sensores a la práctica final (mini-sumo)

- Lee el documento [Normativa minisumo estandar Liga Nacional Robótica de Competición](#)

[v1.1](#), proporcionado en la página WEB de la asignatura.

- Entrega un ensayo corto razonando qué número y tipo de sensores necesarias/podrías usar para la construcción del microbot de mini-sumo, y una descripción cualitativa de como los situarías/utilizarías en el microbot.

## **Ejercicio 5 (Opcional y voluntario<sup>2</sup>) Control de velocidad en lazo cerrado**

Ahora es posible determinar la velocidad real a la que se mueven las ruedas de nuestro robot y por tanto, corregirlas. Utilizando el método de Ziegler-Nichols, determina las constantes proporcional, integral y derivada que permiten controlar la velocidad de cada motor. Este es un proyecto complejo, pero permite resolver los problemas que algunos habéis detectado cuando el robot “escora”. Si decides abordarlo, lo mas cómodo seria que pudieses mandar información del microcontrolador al PC (velocidad medida por el *encoder*, por ejemplo), para poder realizar los cálculos *offline*. Puedes utilizar el código de la practica 1 en la que intercambiamos información entre el MSP y el PC para controlar el motor.

- Entrega una carpeta comprimida con el proyecto realizado, así como un documento con los valores obtenidos y su calculo. El programa ha de permitir fijar la velocidad de los motores utilizando un PID, cuyas constantes se han calculado previamente.

Como referencia, algunos proyectos que utilizan el control por PID para regular un PWM:

- Let's Make robots: [PID control](#)
- TI design contest: [PID Control Tutor as a Levitator](#)

---

2 Este ejercicio está planteado solo para aquellos que quieran “jugar” un poco mas con esta práctica y tengan el tiempo suficiente para ello!! Si estás apurado de tiempo por esta u otras asignaturas NO LO HAGAS!! No es necesario hacerlo para obtener los 1.5p asignados a este apartado; en todo caso te proporcionaría PUNTUACIÓN EXTRA