

DESARROLLO DE SOFTWARE PARA SISTEMAS EMPOTRADOS

Práctica 3: Uso básico y configuración de motores

Héctor Pérez
Michael González

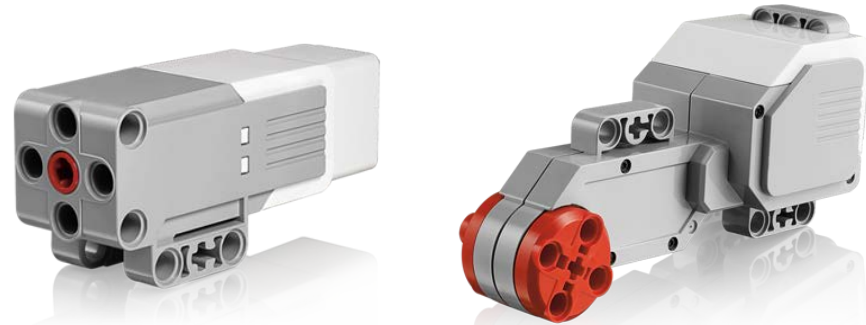


Objetivos

- Uso básico de motores
 - desarrollo de aplicaciones sencillas en C que nos permitan manejar motores
- Configuración de motores
 - obtención y ajuste de parámetros característicos en una aplicación particular
- Ejecución de aplicaciones planificadas por prioridades
 - configuración de atributos de los threads

Introducción

- Motores EV3



- dispositivos que transforma energía eléctrica en energía mecánica
- proporciona retroalimentación para un control preciso de posición y velocidad
 - sensor de rotación integrado
- Posibles usos:
 - coches, cintas transportadoras, brazos robóticos, etc.

Introducción

- Gestión de dispositivos en el kernel *ev3dev*



/sys/class/tacho-motor/motor<N>

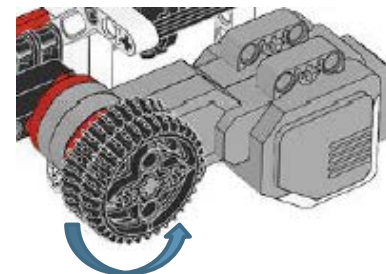
<i>Fichero</i>	<i>Descripción</i>
commands	Listado de los comandos aceptados
stop_commands	Listado de los comandos de parada aceptados
count_per_rot	Número de unidades en una rotación completa del motor
position	Posición actual del motor en unidades del sensor de rotación
speed	Velocidad del motor en unidades del sensor de rotación por segundo
duty_cycle	Porcentaje de tiempo en un intervalo en el que se transmite energía al motor
state	Estado actual del motor (running, ramping, holding o stalling)
polarity	Sentido de giro
*_sp	Valores finales de las variables asociadas al terminar de ejecutar el comando adecuado

Desarrollo

- Uso básico y configuración de motores
 - seguir el montaje del robot proporcionado en Moodle
 - incorporarlo al montaje realizado en la práctica anterior
 - ejecutar la aplicación sencilla proporcionada
 - indicar brevemente qué realiza el código propuesto
 - compilar mediante el *sistema de desarrollo cruzado*
 - se proporciona el script ***reset_motors.sh*** por si se pierde el control de los motores
 - modificar la aplicación proporcionada para que el motor realice una rotación completa
 - razonar la elección del comando utilizado
 - configurar el motor para que opere en los siguientes modos de parada: *coast*, *brake* y *hold*.
 - indicar la diferencia de funcionamiento entre ellos

Desarrollo

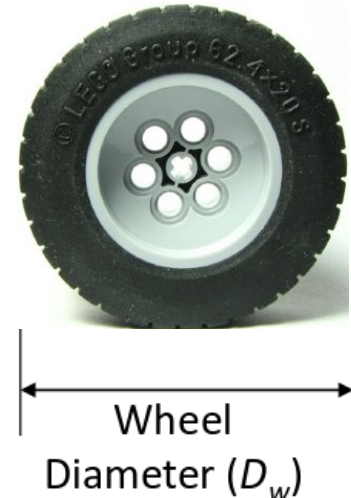
- Uso básico y configuración de motores
 - el control de velocidad puede realizarse mediante 2 métodos:
 - *ev3_set_duty_cycle_sp* (rango 0-100%)
 - *ev3_set_speed_sp* (rango 0-900)
 - desarrollar una aplicación que gire de forma continua el motor durante 15 segundos
 - configurar el *duty_cycle* para que opere al 25%
 - ejecutar la aplicación y estimar la velocidad media de giro
 - configurar la aplicación para que utilice la velocidad estimada en vez del *duty_cycle* y ejecutar la aplicación
 - para ambos métodos, aplicar una pequeña resistencia al engranaje para simular una carga pesada
 - indicar las posibles diferencias entre ambos métodos



Desarrollo

- Uso básico y configuración de motores
 - modificar la aplicación proporcionada para que la cinta transportadora desplace un objeto y pare en cada engranaje:
 - posteriormente, el objeto debe volver a su posición original
 - indicar la configuración del motor elegida
 - calcular la longitud de movimiento:
 - por cada rotación completa, la distancia recorrida viene dada por

$$L = \pi \cdot D_w$$



Desarrollo

- Ejecución de aplicaciones planificadas por prioridades
 - crear un thread para controlar el movimiento del motor implementado en el apartado anterior
 - además, debe indicar en voz alta la posición de la cinta cada vez que se pare
 - por ejemplo, con el siguiente texto: *"My current position is XXX"*
 - puede utilizarse las funciones **speak** de la librería *ev3c-addons.a*
 - requiere añadir las siguientes librerías externas al proceso de compilación: libespeak, libsonic, libasound, libm, libpthread
 - crear otro thread que añada carga al sistema
 - puede utilizarse las funciones **load** de la librería *ev3c-addons.a*

Desarrollo

- Ejecución de aplicaciones planificadas por prioridades
 - ejecutar la aplicación con privilegios de administrador
 >> *sudo ./practica3*
 - configurar el uso de una política de planificación basada en prioridades fijas (*SCHED_FIFO*)
 - ejecutar la aplicación en los siguientes escenarios y razonar los resultados obtenidos
 - la duración de la carga puede limitarse a 10 segundos

Prioridad del thread motor	Low	High	Low	High	Low	High
Prioridad del thread load	High	Low	High	Low	High	Low
Porcentaje de carga	10	10	85	85	95	95