

Fundamentos de robótica móvil

Guía de ejercicios de la Unidad 1

Gonzalo Perez Paina, David Gaydou, Diego Gonzalez Dondo

Ejercicios

1. Construir y grabar el ejemplo `blinky` (directorio `educiaa/blinky`)
2. GPIO con delay bloqueante:
 - Leer el estado de los pulsadores
 - TEC1 – `if(gpioRead(TEC1) == LOW) // Verificar esquemático`
 - TEC2 – completar
 - Cuando se encuentre presionado encender el LED correspondiente
 - Leer el estado de los pulsadores
 - LED1 – `gpioWrite(LED1, ON);`
 - LED2 – completar
 - Los LEDs deben quedar encendidos por 5s utilizando el retardo (delay) bloqueante de sAPI:
`delay(5000);`

Pasos a seguir:

- a) Copiar el directorio `blinky` a `ejercicio1`
 - b) Editar archivo `program.mk` para compilar este nuevo proyecto
 - c) Leer [API de la biblioteca sAPI](#)
 - d) Editar el archivos `src/main.c`
 - e) Construir y grabar
3. GPIO con delay no-bloqueante:
 - Ídem a ejercicio anterior pero con delay no bloqueante (delay de 2000ms)
 - Interpretar las funciones de delay no bloqueante (ver `sapi_delay.h` y `sapi_delay.c`)

```
delayInit(&retardo_TEC1, 2000);  
if( delayRead(&retardo_TEC1) ) { /* completar */ }
```

4. Manejo de motores (PWM) a nivel sAPI
 - Configurar GPIO de la señal de habilitación (enable) de motores
 - Configura el módulo de generación de señal PWM (Leer: [API de la biblioteca sAPI](#))
 - Poner el motor izquierdo al 100 % hacia atrás y motor derecho al 100

Descripción de pines:

- enable motor: TFILO

- PWM Izq: PWM4
 - PWM Der: PWM3
5. Codificar dentro de `main()` los siguientes comportamientos para el robot:
- Al presionar `TEC1` → Avance a máxima velocidad durante 2s
 - Al presionar `TEC2` → Retroceso a máxima velocidad por 2s
 - Al presionar `TEC3` → Giro en sentido antihorario a 50 % de velocidad máxima durante 4s
 - Al presionar `TEC4` → Giro en sentido horario a velocidad máxima durante 2s
6. Probar ejemplo de código del EduRoMAA para el manejo de la bocina. Analizar el código fuente.
7. Probar ejemplo de código del EduRoMAA para el manejo de los paragolpes. Analizar el código fuente.
8. Manejo de motores y paragolpes a nivel EduRoMAA
- Fijar la velocidad de ambos motores al 50 % hacia adelante durante 3s
 - Luego, fijar la velocidad hacia atrás al 100 % durante otros 3s
 - Si se presionan los paragolpes el robot debe moverse hacia adelante durante 2s a las siguientes velocidades:
 - Paragolpe izquierdo: motor izquierdo al 100 % y derecho al 50 %
 - Paragolpe derecho: motor izquierdo al 50 % y derecho al 100 %
 - Ambos paragolpes: ambos motores al 100 %
- Nota: utilizar las funciones `escribirMotorIzq()`, `escribirMotorDer()` y `escribirMotores()` de `motores.h` (en `eduromaa/lib/inc/motores.h`).
9. Ejercicio de integración
- Modificar el ejercicio 2 para que el robot describa una trayectoria cuadrada, desplazándose marcha atrás al 80 % de la velocidad máxima:
 - Todavía no tenemos parámetros importantes para la planificación de trayectorias ¿Qué método sencillo puede implementarse?
 - Detección de colisiones con los bumpers
 - Detener el robot y emitir un sonido de error utilizando la bocina hasta que se remueva el obstáculo.
 - ¿Resulta sencillo retomar la trayectoria con la alternativa implementada? ¿Por qué?
 - Cuando no se presentan obstáculos en la trayectoria:
 - ¿El robot termina en el mismo lugar donde empezó?
 - ¿Qué inconvenientes no tuvimos en cuenta?
10. Generación de camino circular
- Escribir un programa utilizando la biblioteca `eduromaa` de forma tal que el robot describa un círculo con velocidad angular positiva.
 - Realizar los ensayos y cálculos necesarios para determinar de forma aproximada:
 - ¿Cuál es la velocidad lineal y angular del robot?
 - ¿Cuales son las velocidades lineales y angulares de cada rueda?