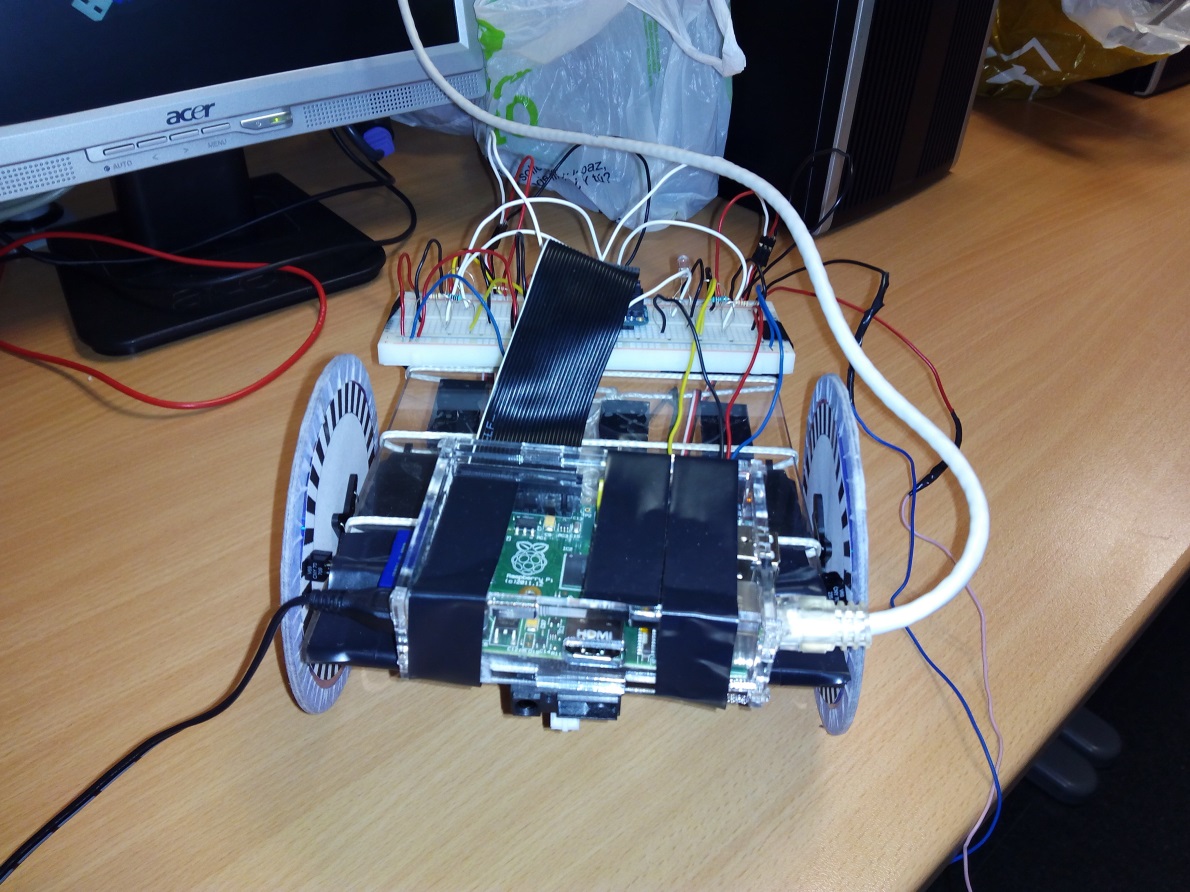
Práctica 4 – Construcción del Robot – 2ª Parte

**2015**

Luis Javier Cabrera Sagbay, Santiago Gómez Muñoz, Ángel Luis Ortiz Folgado

Grupo 9

21/01/2015



Índice

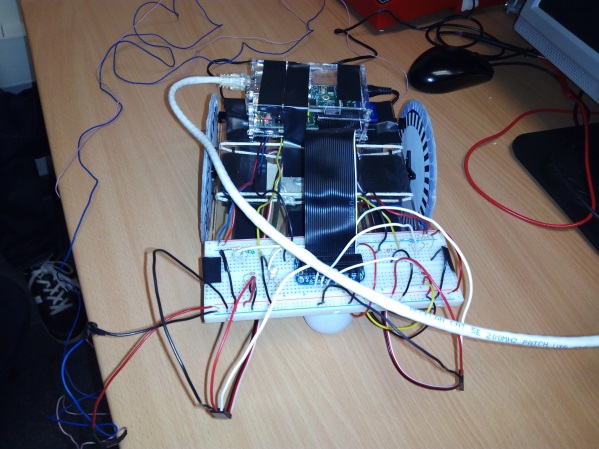
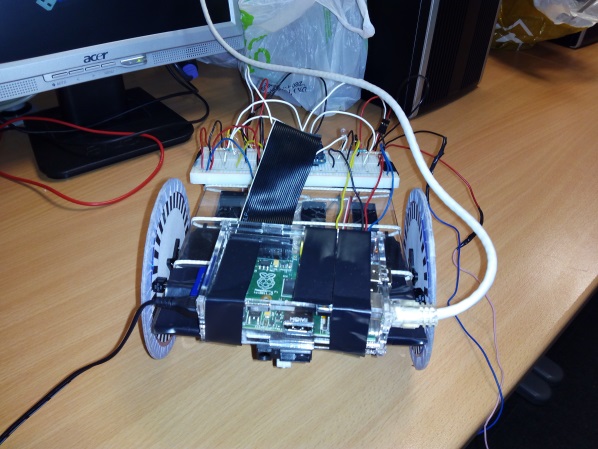
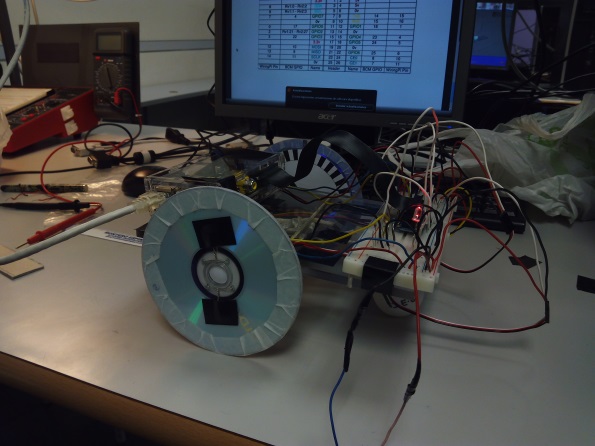
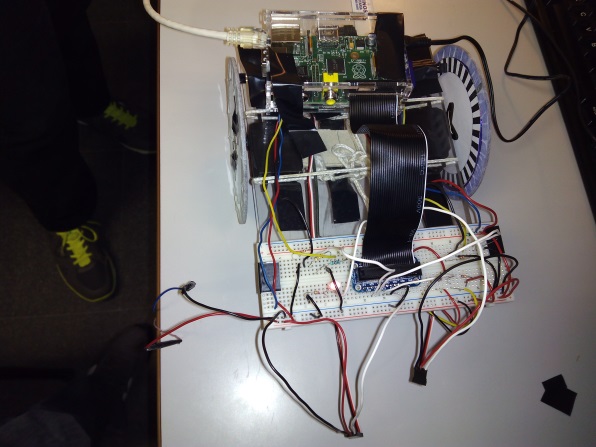
[**Construcción del robot**](#Apartado1)**3**

[**Código para las pruebas del robot**](#Apartado2)**4**

[**Video del funcionamiento del robot**](#Apartado3)**11**

1. **Construcción del robot.**

Las siguientes imágenes muestran el montaje final del robot, acorde con el guion de la práctica. Este cuenta con dos motores, tres sensores, uno que funcionará como sensor de proximidad y los otros dos son los encoders ópticos de las ruedas y un led para indicar el estado.



1. **Código para las pruebas del robot.**

# Experimento 2. Realización de tareas complejas

**2.1.- Seguimiento de trayectorias complejas.**

En esta parte lo que tenemos que hacer es que dado un array en el cual las posiciones pares sirven para avanzar o retroceder y las posiciones impares para girar a la izquierda o a la derecha. Para ello hemos realizado cuatro funciones con las correspondientes acciones que debe realizar el robot.

/\*

\*

\* Apartado 1. Práctica 4 Parte 2

\* Grupo 9

\*

\*/

#include<stdio.h>

#include<wiringPi.h>

#include<softPwm.h>

#define motor1 0 //Pin 17

#define motor2 1 //Pin 18

#define encoder1 5 //Pin 24

#define encoder2 6 //Pin 25

#define sensor 4 //Pin 23

#define led 3 //Pin 22

#define cinco\_cm 4 //Es el número de pulsos que tienen que recoger los encoders para desplazarse 5 centímetros

#define diez\_gr 6 //Es el número de pulsos que tienen que recoger los encoders para girar diez grados

void giraIzq**(**int grados**)** **{**

int valor1 **=** 0**,** valor2 **=** 0**,** cont1 **=** 0**,** cont2 **=** 0**;**

**while** **(**cont1 **<=** **(**grados **/** 10**)** **\*** diez\_gr **&&** cont2 **<=** **(**grados **/** 10**)** **\*** diez\_gr**)** **{**

softPwmWrite**(**motor1**,** 5**);**

softPwmWrite**(**motor2**,** 5**);**

valor1 **=** digitalRead**(**encoder1**);**

valor2 **=** digitalRead**(**encoder2**);**

printf**(**"Cont1: %d\n"**,** cont1**);**

printf**(**"Cont2: %d\n"**,** cont2**);**

// Si es blanco se incrementa el cont1.

**if** **(**valor1 **==** 0**)**

cont1**++;**

delay**(**4.6**);**

// Si es blanco se incrementa el cont2.

**if** **(**valor2 **==** 0**)**

cont2**++;**

delay**(**4.6**);**

**}**

**}**

void giraDer**(**int grados**)** **{**

int valor1 **=** 0**,** valor2 **=** 0**,** cont1 **=** 0**,** cont2 **=** 0**;**

**while** **(**cont1 **<=** **(**grados **/** -10**)** **\*** diez\_gr **&&** cont2 **<=** **(**grados **/** --10**)** **\*** diez\_gr**)** **{**

softPwmWrite**(**motor1**,** 30**);**

softPwmWrite**(**motor2**,** 30**);**

valor1 **=** digitalRead**(**encoder1**);**

valor2 **=** digitalRead**(**encoder2**);**

printf**(**"Cont1: %d\n"**,** cont1**);**

printf**(**"Cont2: %d\n"**,** cont2**);**

// Si es blanco se incrementa el cont1.

**if** **(**valor1 **==** 0**)**

cont1**++;**

delay**(**4.6**);**

// Si es blanco se incrementa el cont1.

**if** **(**valor2 **==** 0**)**

cont2**++;**

delay**(**4.6**);**

**}**

**}**

void avanza**(**int distancia**)** **{**

int valor1 **=** 0**,** valor2 **=** 0**,** cont1 **=** 0**,** cont2 **=** 0**;**

**while** **(**cont1 **<=** **(**distancia **/** 5**)** **\*** cinco\_cm

**&&** cont2 **<=** **(**distancia **/** 5**)** **\*** cinco\_cm**)** **{**

softPwmWrite**(**motor1**,** 5**);**

softPwmWrite**(**motor2**,** 30**);**

valor1 **=** digitalRead**(**encoder1**);**

valor2 **=** digitalRead**(**encoder2**);**

printf**(**"Cont1: %d\n"**,** cont1**);**

printf**(**"Cont2: %d\n"**,** cont2**);**

digitalWrite**(**led**,** LOW**);**

// Si es blanco se incrementa el cont1.

**if** **(**valor1 **==** 0**)**

cont1**++;**

delay**(**20**);**

// Si es blanco se incrementa el cont2.

**if** **(**valor2 **==** 0**)**

cont2**++;**

delay**(**20**);**

**}**

**}**

void retrocede**(**int distancia**)** **{**

int valor1 **=** 0**,** valor2 **=** 0**,** cont1 **=** 0**,** cont2 **=** 0**;**

**while** **(**cont1 **<=** **(**distancia **/** -5**)** **\*** cinco\_cm

**&&** cont2 **<=** **(**distancia **/** -5**)** **\*** cinco\_cm**)** **{**

softPwmWrite**(**motor1**,** 30**);**

softPwmWrite**(**motor2**,** 5**);**

valor1 **=** digitalRead**(**encoder1**);**

valor2 **=** digitalRead**(**encoder2**);**

printf**(**"Cont1: %d\n"**,** cont1**);**

printf**(**"Cont2: %d\n"**,** cont2**);**

// Si es blanco se incrementa el cont1.

**if** **(**valor1 **==** 0**)**

cont1**++;**

delay**(**20**);**

// Si es blanco se incrementa el cont2.

**if** **(**valor2 **==** 0**)**

cont2**++;**

delay**(**20**);**

**}**

**}**

int main**(**void**)** **{**

int mov**[]** **=** **{** 40**,** 45**,** 40**,** **-**45**,** 40 **};**

int l\_mov **=** 5**;**

int i **=** 0**;**

printf**(**"Apartado 1\n"**);**

// Esperamos 10s antes de iniciar el programa.

delay**(**10000**);**

// Inicializamos los puertos.

wiringPiSetup**();**

// Esto crea un pin PWM controlado por software.

softPwmCreate**(**motor1**,** 0**,** 50**);**

softPwmCreate**(**motor2**,** 0**,** 50**);**

// Configuramos los sensores del robot.

pinMode**(**encoder1**,** INPUT**);**

pinMode**(**encoder2**,** INPUT**);**

pinMode**(**sensor**,** INPUT**);**

pinMode**(**led**,** OUTPUT**);**

**for** **(**i **=** 0**;** i **<** l\_mov**;** i**++)** **{**

int lect **=** mov**[**i**];**

**if** **(**i **%** 2 **==** 0**)** **{** // Entra si es una posición par de mov

**if** **(**lect **>** 0**)** **{** // Avance

avanza**(**lect**);**

**}** **else** **{** // Retrocede

retrocede**(**lect**);**

**}**

**}** **else** **{** // Entra si es una posicion impar de mov

**if** **(**lect **>** 0**)** **{** //Gira izquierda

giraIzq**(**lect**);**

**}** **else** **{** //Gira derecha

giraDer**(**lect**);**

**}**

**}**

**}**

**return** 0**;**

**}**

**2.2.- Detección de un obstáculo y esquiva de éste.**

En esta parte tenemos que darle una posición de origen y una posición de destino, y si se encuentra un obstáculo entra en una rutina de esquiva. En nuestro caso cuando el robot detecta un obstáculo entra en modo esquiva (encendiendo el led) y para ello gira a la izquierda avanza 20 centímetros y al final vuelve a girar a la derecha para dejarlo en la posición que estaba antes. Después de salir del modo esquiva avanza lo que tenía que recorrer para llegar al destino.

/\*

\*

\* Apartado 2. Práctica 4 Parte 2

\* Grupo 9

\*

\*/

#include<stdio.h>

#include<wiringPi.h>

#include<softPwm.h>

#define motor1 0 //Pin 17

#define motor2 1 //Pin 18

#define encoder1 5 //Pin 24

#define encoder2 6 //Pin 25

#define sensor 4 //Pin 23

#define led 3 //Pin 22

#define cinco\_cm 4 //Es el número de pulsos que tienen que recoger los encoders para desplazarse 5 centimetros

#define diez\_gr 6 //Es el número de pulsos que tienen que recoger los encoders para girar diez grados

void giraIzq**(**int grados**)** **{**

int valor1 **=** 0**,** valor2 **=** 0**,** cont1 **=** 0**,** cont2 **=** 0**;**

**while** **(**cont1 **<=** **(**grados **/** 10**)** **\*** diez\_gr **&&** cont2 **<=** **(**grados **/** 10**)** **\*** diez\_gr**)** **{**

softPwmWrite**(**motor1**,** 5**);**

softPwmWrite**(**motor2**,** 5**);**

valor1 **=** digitalRead**(**encoder1**);**

valor2 **=** digitalRead**(**encoder2**);**

printf**(**"Cont1: %d\n"**,** cont1**);**

printf**(**"Cont2: %d\n"**,** cont2**);**

// Si es blanco se incrementa el cont1.

**if** **(**valor1 **==** 0**)**

cont1**++;**

delay**(**4.6**);**

// Si es blanco se incrementa el cont2.

**if** **(**valor2 **==** 0**)**

cont2**++;**

delay**(**4.6**);**

**}**

**}**

void giraDer**(**int grados**)** **{**

int valor1 **=** 0**,** valor2 **=** 0**,** cont1 **=** 0**,** cont2 **=** 0**;**

**while** **(**cont1 **<=** **(**grados **/** 10**)** **\*** diez\_gr **&&** cont2 **<=** **(**grados **/** 10**)** **\*** diez\_gr**)** **{**

softPwmWrite**(**motor1**,** 30**);**

softPwmWrite**(**motor2**,** 30**);**

valor1 **=** digitalRead**(**encoder1**);**

valor2 **=** digitalRead**(**encoder2**);**

printf**(**"Cont1: %d\n"**,** cont1**);**

printf**(**"Cont2: %d\n"**,** cont2**);**

// Si es blanco se incrementa el cont1.

**if** **(**valor1 **==** 0**)**

cont1**++;**

delay**(**4.6**);**

// Si es blanco se incrementa el cont2.

**if** **(**valor2 **==** 0**)**

cont2**++;**

delay**(**4.6**);**

**}**

**}**

void esquivaObstaculo**()** **{**

int valor1 **=** 0**,** valor2 **=** 0**,** cont1 **=** 0**,** cont2 **=** 0**;**

// Paramos el robot.

softPwmWrite**(**motor1**,** 15**);**

softPwmWrite**(**motor2**,** 15**);**

// Encendemos el LED.

digitalWrite**(**led**,** HIGH**);**

// Esperamos 1 segundo.

delay**(**1000**);**

giraIzq**(**90**);**

**while** **(**cont1 **<=** **(**20 **/** 5**)** **\*** cinco\_cm **&&** c2 **<=** **(**20 **/** 5**)** **\*** cinco\_cm**)** **{**

softPwmWrite**(**motor1**,** 5**);**

softPwmWrite**(**motor2**,** 30**);**

valor1 **=** digitalRead**(**encoder1**);**

valor2 **=** digitalRead**(**encoder2**);**

// Si es blanco se incrementa el cont1.

**if** **(**valor1 **==** 0**)**

cont1**++;**

delay**(**20**);**

// Si es blanco se incrementa el cont2.

**if** **(**valor2 **==** 0**)**

cont2**++;**

delay**(**20**);**

**}**

giraDer**(**90**);**

// Paramos el robot.

softPwmWrite**(**motor1**,** 15**);**

softPwmWrite**(**motor2**,** 15**);**

delay**(**1000**);**

// Apagamos el LED.

digitalWrite**(**led**,** LOW**);**

**}**

void avanza**(**int distancia**)** **{**

int valor1 **=** 0**,** valor2 **=** 0**,** cont1 **=** 0**,** cont2 **=** 0**;**

**while** **(**cont1 **<=** **(**distancia **/** 5**)** **\*** cinco\_cm

**&&** cont2 **<=** **(**distancia **/** 5**)** **\*** cinco\_cm**)** **{**

softPwmWrite**(**motor1**,** 5**);**

softPwmWrite**(**motor2**,** 30**);**

**while** **(**cont1 **<=** **(**distancia **/** 5**)** **\*** cinco\_cm

**&&** cont2 **<=** **(**distancia **/** 5**)** **\*** cinco\_cm

**&&** digitalRead**(**sensor**)** **!=** 1**)** **{**

valor1 **=** digitalRead**(**encoder1**);**

valor2 **=** digitalRead**(**encoder2**);**

printf**(**"Cont1: %d\n"**,** cont1**);**

printf**(**"Cont2: %d\n"**,** cont2**);**

// Si es blanco se incrementa el cont1.

**if** **(**valor1 **==** 0**)**

cont1**++;**

delay**(**20**);**

// Si es blanco se incrementa el cont2.

**if** **(**valor2 **==** 0**)**

cont2**++;**

delay**(**20**);**

**}**

esquivaObstaculo**();**

**}**

**}**

int main**(**void**)** **{**

int p\_ini**[]** **=** **{** 0**,** 0**,** 1 **};** // Primer valor x, segundo valor y y tercer valor indica la orientacion 1 norte, 2 oeste, 3 sur y 4 este

int p\_fin**[]** **=** **{** 80**,** 80**,** 1 **};**

int p\_act**[]** **=** **{** 0**,** 0**,** 1 **};**

printf**(**"Apartado 2\n"**);**

// Esperamos 10s antes de iniciar el programa.

delay**(**10000**);**

// Inicializamos los puertos.

wiringPiSetup**();**

// Esto crea un pin PWM controlado por software.

softPwmCreate**(**motor1**,** 0**,** 50**);**

softPwmCreate**(**motor2**,** 0**,** 50**);**

// Configuramos los sensores del robot.

pinMode**(**encoder1**,** INPUT**);**

pinMode**(**encoder2**,** INPUT**);**

pinMode**(**sensor**,** INPUT**);**

pinMode**(**led**,** OUTPUT**);**

/\* El siguiente bucle se ejecuta mientras el robot no llegue a su destino.

\*/

**while** **(**p\_act**[**0**]** **!=** p\_fin**[**0**]** **&&** p\_act**[**1**]** **!=** p\_fin**[**1**])** **{**

**if** **(**p\_act**[**2**]** **!=** 4**)** **{**

**switch** **(**p\_act**[**2**])** **{**

**case** 1**:**

giraDer**(**90**);**

**break;**

**case** 2**:**

giraDer**(**180**);**

**break;**

**case** 3**:**

giraIzq**(**90**);**

**break;**

**}**

p\_act**[**2**]** **=** 4**;**

**}**

avanza**(**p\_fin**[**0**]** **-** p\_act**[**0**]);**

p\_act**[**0**]** **+=** p\_fin**[**0**]** **-** p\_act**[**0**];**

**if** **(**p\_act**[**2**]** **!=** p\_fin**[**2**])** **{**

**switch** **(**p\_fin**[**2**])** **{**

**case** 1**:**

giraIzq**(**90**);**

**break;**

**case** 2**:**

giraDer**(**180**);**

**break;**

**case** 3**:**

giraDer**(**90**);**

**break;**

**}**

p\_act**[**2**]** **=** p\_fin**[**2**];**

**}**

avanza**(**p\_fin**[**1**]** **-** p\_act**[**1**]);**

p\_act**[**1**]** **+=** p\_fin**[**1**]** **-** p\_act**[**1**];**

**}**

**return** 0**;**

**}**

1. **Video del funcionamiento del robot.**



Para reproducirlo hacer clic sobre el icono.