1.2.20. Encoders y Odometría

La odometría es utilizada en robótica para saber dónde está un robot midiendo el movimiento de sus ruedas. En esta actividad nos construiremos y programaremos un enconder para dotar de odometría a un robot de 2 ruedas basado en Arduino(como el Printbot que estamos utilizando en nuestras actividades).

Componentes

Para la realización de esta actividad utilizaremos el Robot Printbot o cualquier robot basado en Arduino UNO o compatible de dos ruedas con servos de rotación continua y 2 sensores de infrarrojo

- ZUM BT o Arduino UNO compatible
- 2 Sensores de infrarrojo digitales (en bloques separados). Si no tienes sensores infrarrojo en bloques separados (como ocurre en el PrintBot) puedes utilizar los sensores LDR en su lugar, pero el funcionamiento será peor ya que el LDR no lee tan bien la diferencia negro/blanco del disco del encoder
- 2 servos de rotación continua
- Dos discos impresos con franjas blanco-negro como los de la Figura 1.2.20-1. Te puedes fotocopiar esta página o bajar el pdf con estos discos de nuestra página web www.automaticayrobotica.es para poder imprimirlos en tu impresora.

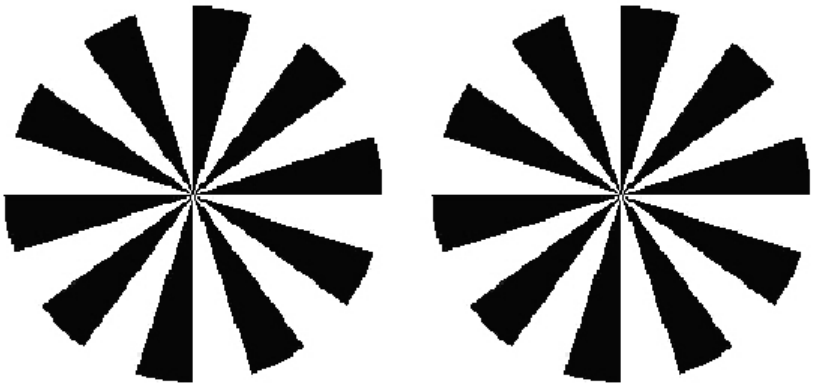

Figura 1.2.20-1 Discos para enconder de las ruedas

Para montar el encoder primero deberemos desmontar las dos ruedas de nuestro robot y colocar cada sensor de infrarrojos de manera que el sensor quede enfrentado a la rueda lo más cercano posible pero sin tocarla (Figura 1.2.20-2.1). En el caso del Printbot hemos optado por sujetarlo con una simple goma elástica, pero puedes también intentar atornillarlo al chasis. Pon cinta de doble cara en la parte interna de cada rueda (Figura 1.2.20-2.2) y pega los discos (Figura 1.2.20-2.3). Vuele a colocar las ruedas a los servos de rotación continua. En nuestra web también encontrarás instrucciones de como colocar los encoders en otros robots, como el robot Renacuajo de BQ.

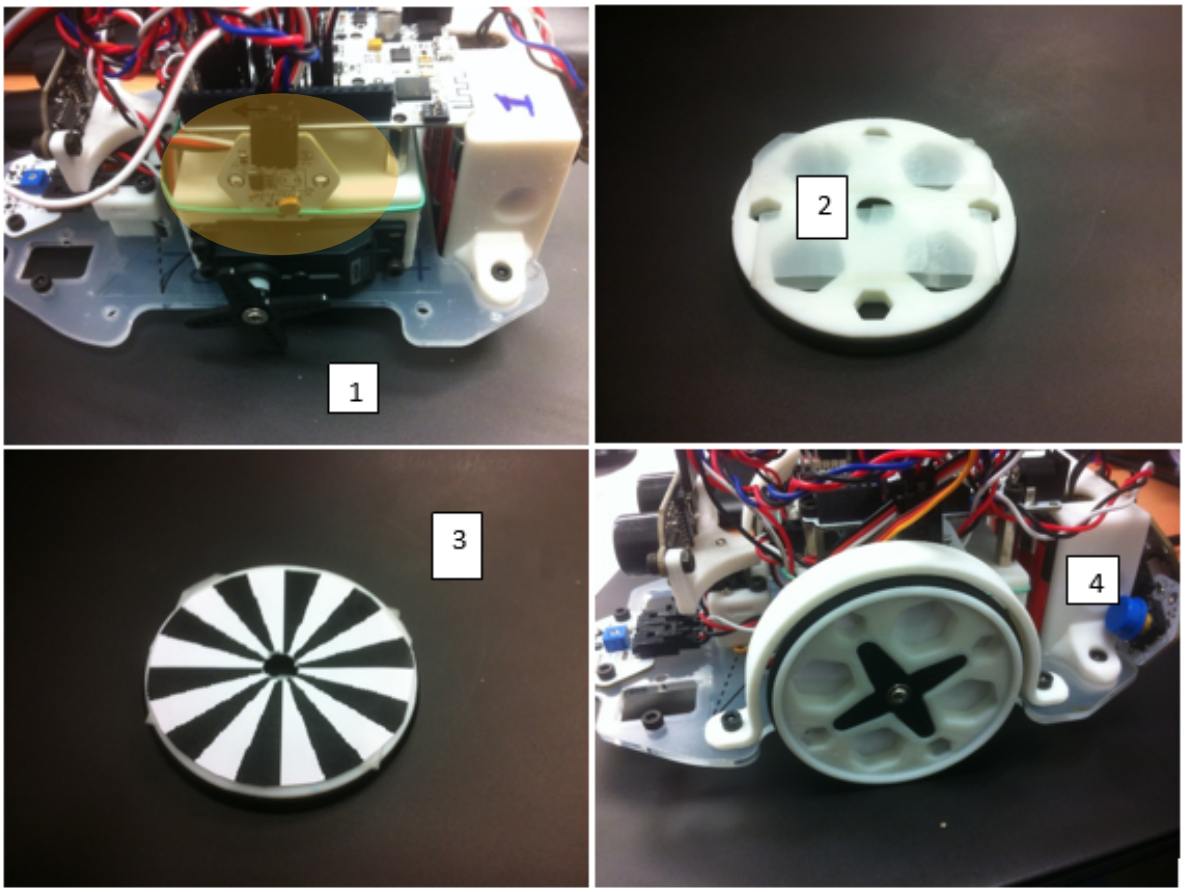

Figura 1.2.20-2 Montaje de un encoder en un robot de 2 ruedas (Printbot)

Conexionado

Realizaremos el siguiente conexionado entre los componentes y la tarjeta Arduino.

- Infrarrojo Izquierdo→ Pin 3 Arduino
- Infrarrojo Derecho→ Pin 2 Arduino
- Motor Izquierdo→ Pin 9 Arduino
- Motor Derecho→ Pin 6 Arduino

Programación

Para proceder a programar es inevitable que expliquemos antes unas sencillas ecuaciones que nos van a permitir saber cómo se mueve un robot en función de cuanto giran sus ruedas.

Primero debemos saber que cada vez que el encoder pase de leer blanco a leer negro generará una cuenta (es decir, vamos a contar cuantas veces pasa de negro a blanco, cosa que ocurre a medida que la rueda va girando). Si el robot se mueve, en un tiempo ***t*** el **encoder** izquierdo genera un cambio de ***Ni*** cuentas, y el derecho *de* ***Nd*** cuentas.

Utilizaremos un factor de conversión ***f*** entre cuentas y desplazamiento definido por la ecuación:

$$f = \pi D / R$$

donde ***R*** es la resolución del ***encoder*** (en cuentas por vuelta) y ***D*** el diámetro de la rueda (la resolución del disco de la Figura 1.2.20-1 es 20, y en el caso de PrintBot, el diámetro de la rueda es 6cm).

La distancia recorrida (L) por cada rueda será:

$$Li = f * Nd$$
$$Ld = f * Nd$$

De esta manera, el centro del robot ***C*** se habrá desplazado:

$$LC = (Li + Ld) / 2.$$

Además, podemos saber cuánto ha girado el robot sobre su eje con la siguiente ecuación:

$$\Delta\theta = (Ld - Li) / B$$

donde ***B*** es la distancia entre ruedas (siendo 9cm en el caso del Printbot).

Por último, si ***x(0), y(0), θ(0)*** eran las coordenadas iniciales del robot, la nuevas coordenadas (posición y orientación) del robot una vez se ha movido serán:

$$\theta(t) = \theta(0) + \Delta\theta$$
$$x(t) = x(0) + LC \cos \theta(t)$$
$$y(t) = y(0) + LC \text{ sen } \theta(t)$$

Estas ecuaciones se traducen en siguiente código Arduino que te puedes descargar de nuestra página web (debido a la complejidad del código en esta actividad no utilizaremos programación por bloques):

```
#include <SoftwareSerial.h>
#include <Servo.h>

int LeftSensor;
int RightSensor;

int cont_izq;
int cont_der; //contador de pulsos
int estado_der; //estado 0 es negro 1 es blanco
int estado_izq;


Servo servo_6; //motor rueda izquierda
Servo servo_9;//motor rueda derecha

void setup(){


  Serial.begin(19200);
  delay(1000);

  pinMode(2, INPUT); //sensor rueada izquierda
  pinMode(3, INPUT);//sensor rueda derecha
```

```
  cont_der = 0; //contador de pulsos
  cont_izq = 0;
  estado_der = 0; //estado 0 es negro 1 es blanco
  estado_izq = 0;

  servo_6.attach(6);
  servo_9.attach(9);


   }
void Forward(){
   servo_6.write(0);
    servo_9.write(180);

}

void Motor_Stop(){
    servo_6.write(90);
    servo_9.write(90);
}


void odometria(){

  //CALCULO LA POSICION X,Y y ORIENTACION THETA

  float  pi= 3.1416;
  float  diametro=5.70; //cm
  float  resolucion= 20; //ticks por vuelta del encoder
  float  dist_entre_ruedas=8.5;//cm
  float  f= pi*diametro/resolucion; //FACTOR DE CONVERSION
  float  dist_rueda_izq=f*cont_izq;
  float  dist_rueda_der=f*cont_der;
  float  dist_centro_robot=(dist_rueda_izq + dist_rueda_der)/2;
  float  theta= (dist_rueda_izq - dist_rueda_der)/dist_entre_ruedas;
  float  x=dist_centro_robot*cos(theta);
  float  y=dist_centro_robot*sin(theta);

 //MUESTRO POR EL SERIE LA POSICION X Y

 Serial.print("x = ");
 Serial.println(x);
 Serial.print("y = ");
 Serial.println(y);

 Serial.print("cont_izq = ");
 Serial.println(cont_izq);
 Serial.print("cont_der = ");
 Serial.println(cont_der);

   delay(1000);
}

void loop(){

   if( millis() < 4000){ // la ejecucion de movimientos durara 4 segundos

    if( millis() < 2000)  Forward(); //tiempo en el que se mueven los motores
    if( millis() > 2000)   Motor_Stop(); //tiempo en el que espero que termine
de moverse el robot

  LeftSensor = digitalRead(2);          // Read value from left sensor
```

```
  RightSensor = digitalRead(3);          // Read value from right sensor

  if((LeftSensor==1) & (estado_izq==0)){ //cambio de estado a 0
     cont_izq=cont_izq+1;
    estado_izq=1;
  }
  if((LeftSensor==0) & (estado_izq==1)){ //cambio de estado a 1
    cont_izq=cont_izq+1;
    estado_izq=0;
  }
  if((RightSensor==1) & (estado_der==0)){ //cambio de estado a 0
     cont_der=cont_der+1;
    estado_der=1;
  }
 if((RightSensor==0) & (estado_der==1)){ //cambio de estado a 1
    cont_der=cont_der+1;
    estado_der=0;
 }
 }
 else{ //una vez que han pasado 4 segundos muestro el resultado

  Motor_Stop();

  odometria();

   }
}
```

Conecta el robot al puerto al USB y carga el programa. Luego abre un monitor serie y enciende el robot. El robot se moverá durante un tiempo y luego se parará. Pinta con un lápiz el punto de partida (con la orientación) y el punto de llegada (con la orientación) y comprueba que los valores que te envía el robot por el puerto serie de lo que se ha movido corresponden con lo que has medido en la realidad (véase Figura 1.2.20-3)

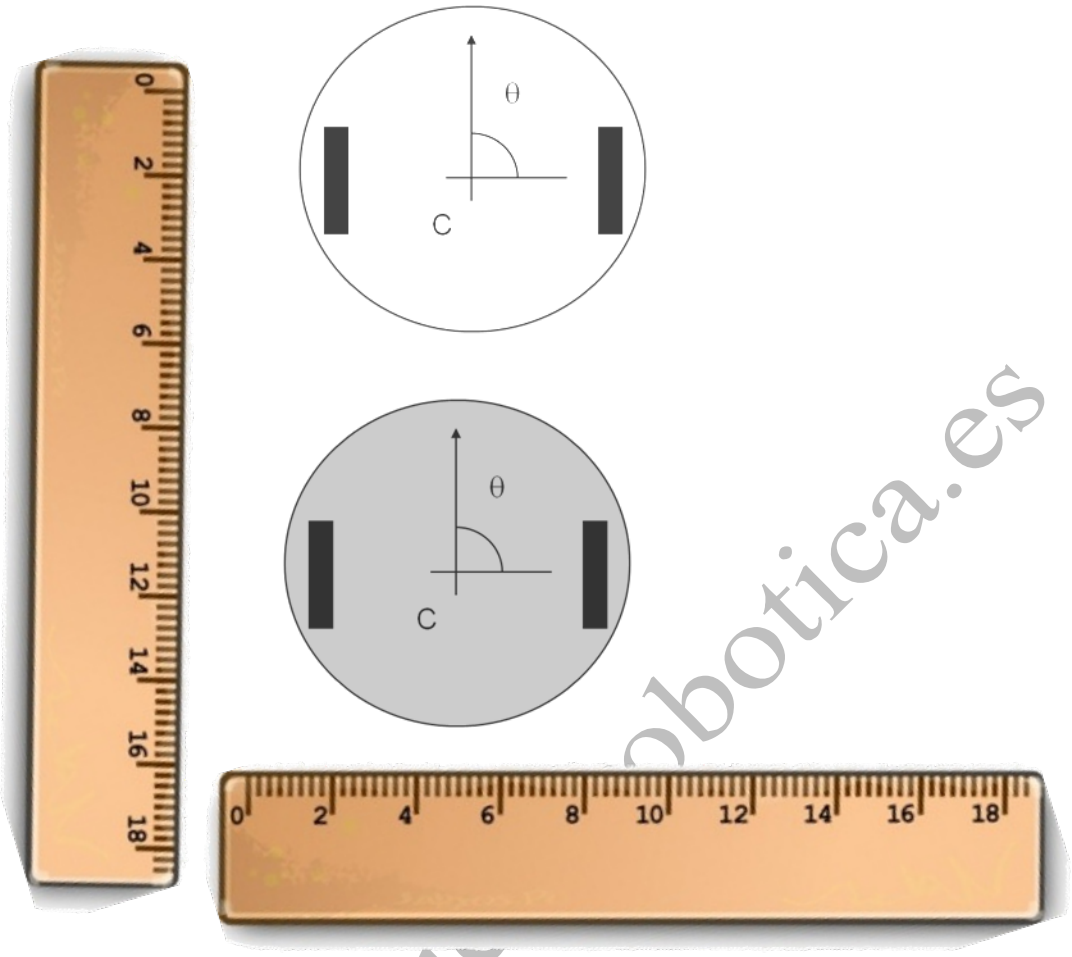

Figura 1.2.20-3 Comprobación de la odometría de un robot

Como verás existe un error (esperemos que pequeño) entre lo que el robot cree que se ha movido y lo que realmente se mueve. Esto es debido a varias razones, como pueden ser la poca precisión del sensor o los deslizamientos de las ruedas. En los robots reales la odometría funciona mejor, pero aun así tiene fallos que hace que no se utilice como único medio para saber cuánto se mueve un robot, sino que se utiliza con otros sistemas como por ejemplo sensores laser, GPS, etc.