



TRABAJO PRACTICO N°3

MATERIA: SENSORES Y ACTUADORES

PROFESORES: ING. JORGE E. MORALES

TEC. SUP. GONZALO VERA

ALUMNA: MARIA CAROLINA NIS

AÑO 2022

Ejercicio N° 1


1) Que es un sensor generador

Se consideran sensores generadores aquellos que generan una señal eléctrica, a partir de la magnitud que miden, sin necesidad de una alimentación. Ofrecen una alternativa para medir muchas de las magnitudes ordinarias, sobre todo temperatura, fuerza y magnitudes afines. Pero, además, dado que se basan en efectos reversibles, están relacionados con diversos tipos de accionadores o aplicaciones inversas en general. Es decir, se pueden emplear para la generación de acciones no eléctricas a partir de señales eléctricas

2) Cuales son los tipos de sensores generadores

Los podemos clasificar como:

- * Sensores termoelectricos
- * Sensores piezoelectricos
- * Sensores piroelectricos
- * Sensores fotovoltaicos
- * Sensores electroquímicos



SENSORES GENERADORES			
Clasificación de sensores			
CLASIFICACIÓN	TIPO	DESCRIPCIÓN	EJEMPLO
Según la fuente de energía	Generadores (activos)	Utilizan la energía del medio donde miden	Sensor piezoelectrico
	Moduladores (pasivos)	Necesitan de una fuente externa de energía	Galga resistiva extensométrica
Según la señal de salida	Analógico	La señal de salida es continua en el tiempo	Sensor piezoelectrico
	Digital	La señal de salida es discreta en el tiempo	Codificador óptico

3) Mencione 5 características del sensor termopar

Los termopares tienen muchas ventajas y son, con diferencia, los sensores más frecuentes para la medida de temperaturas.

- Tienen un alcance de medida grande, no sólo en su conjunto, que va desde -270°C hasta 3000°C , sino también en cada modelo particular.
- Su estabilidad a largo plazo es aceptable y su fiabilidad elevada.
- Para temperaturas bajas tienen mayor exactitud que las RTD, y por su pequeño tamaño tienen velocidades de respuesta rápida, del orden de milisegundos.
- Son robustos, simples y tienen gran flexibilidad de utilización. Además, existen modelos de bajo precio adecuados para muchas aplicaciones.
- Al no necesitar excitación, no tienen problemas de autocalentamiento

4) Defina sensor piezoeléctrico y mencione 3 limitaciones

El efecto piezoeléctrico consiste en la aparición de una polarización eléctrica en un material al deformarse bajo la acción de un esfuerzo. Es un efecto reversible, de modo que al aplicar una diferencia de potencial eléctrico a un material piezoeléctrico, aparece una deformación. Estos fenómenos fueron descubiertos por Jacques y Pierre Curie en 1880. La piezoelectricidad está relacionada con la estructura cristalina de los materiales. Las propiedades piezoeléctricas se manifiestan en 20 de las 32 clases cristalográficas, aunque en la práctica se usan sólo unas pocas. Entre los materiales piezoeléctricos naturales, los de uso más frecuente son el cuarzo y la turmalina. En cuanto a las sustancias sintéticas, las que han encontrado más aplicación como materiales piezoeléctricos son las cerámicas.

La aplicación del efecto piezoeléctrico está sujeto a una serie de **limitaciones**.

- La resistencia eléctrica que presentan los materiales piezoeléctricos aunque es muy grande no es infinita. De modo que al aplicar un esfuerzo constante se genera inicialmente una carga que inevitablemente es drenada al cabo de un tiempo. Por lo tanto, no tienen respuesta en continua.

- Estos sensores presentan un pico en la respuesta para la frecuencia de resonancia. Por tanto, es preciso trabajar siempre a frecuencias muy inferiores a la de resonancia mecánica.
- La sensibilidad presenta derivas con la temperatura. Además, por encima de la temperatura de Curie (específica para cada material) desaparece el efecto piezoeléctrico.
- La impedancia de salida de estos sensores es muy alta, por lo que para medir la tensión de salida es preciso utilizar amplificadores con una impedancia de entrada enorme. Son los denominados amplificadores electrométricos.

Entre las **ventajas** de los sensores piezoeléctricos destacaremos las siguientes:

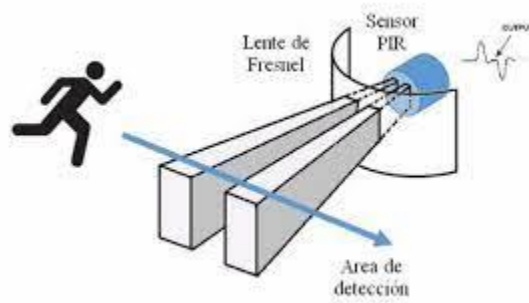
- Alta sensibilidad, obtenida muchas veces a bajo coste.
- Alta rigidez mecánica; las deformaciones experimentadas son inferiores a 1µm. Esta alta impedancia mecánica es conveniente para la medida de variables esfuerzo (fuerza, presión, etc)
- Pequeño tamaño y posibilidad de obtener dispositivos con sensibilidad unidireccional. Estas características hacen a este tipo de sensores especialmente adecuados para medir todo tipo de vibraciones. Por ejemplo una aplicación típica es como micrófono. También se utilizan mucho en la detección por ultrasonidos.
- Algunas de las aplicaciones más frecuentes de los sensores piezoeléctricos están en el campo de los ultrasonidos. Especial mención requieren (por ser muy numerosos) los dispositivos que utilizan técnicas de impulso-eco: detectores de grietas o anomalías en estructuras, detectores de presencia, equipos de ecografía, medidores de distancias, etc.

5) Explique el funcionamiento del sensor piroelectrico

Los sensores PIR, también llamados pirólicos, son un tipo de sensor óptico muy empleado para detectar el movimiento de personas, en especial en alumbrados automatizados.

Los módulos más comunes de este sensor se denominan HC-SR501. Consisten en 2 elementos fundamentales, un lente que hace que los rayos incidentes recaigan sobre un punto y el sensor PIR como tal. El sensor PIR tiene dos elementos sensibles a la luz, que al medirse el retardo de detección entre cada sensor, es posible calcular la distancia a la que se encuentra el objeto.

Así operan los elementos sensibles del módulo y la señal de salida



Los sensores PIR son de tipo ópticos, es decir, se basan en cambios en la radiación electromagnética para sensar el entorno. En lo particular, los sensores PIR trabajan en el rango de la luz infrarroja. Se les conoce como **piroeléctricos** debido a que los objetos que emiten mayor radiación infrarroja son aquellos cuya temperatura es mayor.

El modo en que detectan la luz es mediante cambios en la conductividad de los elementos semiconductores de los que se componen. Por ejemplo, el sensor RE200B tiene una ventana que expone el material semiconductor a la luz del ambiente, permitiendo que detecte y active un MOSFET interno.

6) Mencione los tipos de sensores fotovoltaicos y defina 2

Es un dispositivo electrónico que responde al cambio en la intensidad de la luz. Estos sensores requieren de un componente emisor que genera la luz, y un componente receptor que percibe la luz generada por el emisor. Todos los diferentes modos de sensado se basan en este principio de funcionamiento.

Están diseñados especialmente para la detección, clasificación y posicionado de objetos; la detección de formas, colores y diferencias de superficie, incluso bajo condiciones ambientales extremas.

Los sensores de luz se usan para detectar el nivel de luz y producir una señal de salida representativa respecto a la cantidad de luz detectada. Un sensor de luz incluye un transductor fotoeléctrico para convertir la luz a una señal eléctrica y puede incluir electrónica para condicionamiento de la señal, compensación y formateo de la señal de salida.

Existen tres tipos de sensores fotoeléctricos, los sensores por barrera de luz, reflexión sobre espejo o reflexión sobre objetos.

Tipos de sensores:

- Barrera de luz:

Las barreras tipo emisor-receptor están compuestas de dos partes, un componente que emite el haz de luz, y otro componente que lo recibe. Se establece un área de detección donde el objeto a detectar es reconocido cuando el mismo interrumpe el haz de luz.

Estos sensores operan de una manera precisa cuando el emisor y el receptor se encuentran alineados.

- Reflexión sobre espejo:

Tienen el componente emisor y el componente receptor en un solo cuerpo, el haz de luz se establece mediante la utilización de un reflector catadióptrico. El objeto es detectado cuando el haz formado entre el componente emisor, el reflector y el componente receptor es interrumpido. Debido a esto, la detección no es afectada por el color del mismo. La ventaja de las barreras réflex es que el cableado es en un solo lado, a diferencia de las barreras emisor-receptor que es en ambos lados.

- Reflexión sobre objeto:

La luz infrarroja viaja en línea recta, en el momento en que un objeto se interpone el haz de luz rebota contra este y cambia de dirección permitiendo que la luz sea enviada al receptor y el elemento sea censado, un objeto de color negro no es detectado ya que este color absorbe la luz y el sensor no experimenta cambios. Hay dos tipos de fotocélulas de reflexión sobre objeto, las de reflexión difusa y las de reflexión definida.

7) Que es un sensor electroquímico

Los sensores electroquímicos adecuados para determinar el contenido de oxígeno y los constituyentes nocivos del gas tales como CO, SO₂ o NO_x, funcionan basándose en el principio de la valoración potenciométrica sensible a los iones.

Los sensores están rellenos con un electrolito acuoso, específico para la tarea, en el que están dispuestos dos o tres electrodos, igualmente combinados específicamente, entre los que hay un campo eléctrico. Los sensores están sellados del exterior mediante membranas permeables al gas.

El diseño específico y el funcionamiento de los sensores difieren según el componente del gas a medir.

Ejercicio N°2

a) Implemente un control de velocidad, posición y sentido de giro utilizando un motor con encoder incremental, el control del motor se debe realizar con pwm.

Para el control de velocidad de motores de corriente directa se utiliza en general la técnica conocida como PWM ó 'Pulse Wide Modulation'. El método consiste en el switcheo rápido (alrededor de 20 khz) de la fuente de alimentación del motor, proporcionando a éste una potencia promedio controlada mediante el ancho de los pulsos.

Para tal fin, se utiliza típicamente un módulo especializado controlador de PWM, el cual a su vez es operado mediante un sistema microcontrolador.

Adicionalmente, estos módulos permiten también cambiar la dirección de giro del motor, por medio de un circuito conocido como 'Puente H'. Así se le conoce por su forma parecida a la letra H.

Básicamente es un circuito electrónico que permite invertir el sentido de la corriente directa en el motor, cambiando de esta forma su dirección de giro.

Codificación

```
#include <util/atomic.h>

#define ENCODER_A    2 // Amarillo
#define ENCODER_B    3 // Verde
#define BUTTON_MOD    4

// Pin del Potenciometro

const int pot = A0;
```

```

// Pines de Control Shield

const int E1Pin = 10;

const int M1Pin = 12;

const int E2Pin = 11;

const int M2Pin = 13;

//Variable global de posición compartida con la interrupción

volatile int theta = 0;

//Variable global de pulsos compartida con la interrupción

volatile int pulsos = 0;

unsigned long timeold;

float resolution = 374.22;

//Variable Global Velocidad

int vel = 0;

//Variable Global Posicion

int ang = 0;

//Variable Global MODO

bool modo = false;

//Estructura del Motor

typedef struct{

    byte enPin;

    byte directionPin;

}Motor;

//Creo el motor

const Motor motor = {E1Pin, M1Pin};

//Constantes de dirección del Motor

const int Forward = LOW;

const int Backward = HIGH;

void setup(){

    // set timer 1 divisor to 1024 for PWM frequency of 30.64 Hz

```



```

TCCR1B = TCCR1B & B11111000 | B00000101;

Serial.begin(9600);

//Encoders como entradas

pinMode(ENCODER_A, INPUT);

pinMode(ENCODER_B, INPUT);

//Pulsadores

pinMode(BUTTON_MOD, INPUT_PULLUP);

//Configura Motor

pinMode(motor.enPin, OUTPUT);

pinMode(motor.directionPin, OUTPUT);

//Configurar Interrupción

timeold = 0;

attachInterrupt(digitalPinToInterrupt(ENCODER_A), leerEncoder, RISING);
}

void loop(){

    float posicion;

    float rpm;

    int value, dir=true;

    //Lee el Valore del Potenciometro

    value = analogRead(pot);

    //Cambia de Modo Velocidad o Posición

    if(debounce(BUTTON_MOD)){

        modo = !modo;

        theta = 0;

    }

    if(modo){

        //Transforma el valor del Pot a velocidad

        vel = map(value, 0, 1023, 0, 255);

```

```

//Activa el motor dirección Forward con la velocidad
setMotor(motor, vel, false);

//Espera un segundo para el calculo de las RPM
if (millis() - timeold >= 1000)
{
    //Modifica las variables de la interrupción forma atómica
    ATOMIC_BLOCK(ATOMIC_RESTORESTATE){

        //rpm = float(pulsos * 60.0 / 374.22); //RPM

        rpm = float((60.0 * 1000.0 / resolution ) / (millis() - timeold) * pulsos);

        timeold = millis();

        pulsos = 0;
    }

    Serial.print("RPM: ");

    Serial.println(rpm);

    Serial.print("PWM: ");

    Serial.println(vel);
}
}

else{

    //Transforma el valor del Pot a ángulo

    ang = map(value,0,1023,0,360);

    //Modifica las variables de la interrupción forma atómica
    ATOMIC_BLOCK(ATOMIC_RESTORESTATE){

        posicion = (float(theta * 360.0 / resolution));

    }

    //Posiciona el ángulo con tolerancia +- 2

    if(ang > posicion+2){

        vel = 200;

        dir = true;
    }
}

```

```

    }

    else if(ang < posicion-2){

        vel = 200;

        dir = false;

    }

    else{

        vel = 0;

    }

    setMotor(motor, vel, dir);

}

}

//Función para dirección y velocidad del Motor

void setMotor(const Motor motor, int vel, bool dir){

    analogWrite(motor.enPin, vel);

    if(dir)

        digitalWrite(motor.directionPin, Forward);

    else

        digitalWrite(motor.directionPin, Backward);

}

//Función anti-rebote

bool debounce(byte input){

    bool state = false;

    if(! digitalRead(input)){

        delay(200);

        while(! digitalRead(input));

        delay(200);

        state = true;

    }

    return state;

```

```

}

//Función para la lectura del encoder
void leerEncoder(){

    //Lectura de Velocidad

    if(modo)

        pulsos++; //Incrementa una revolución


    //Lectura de Posición

    else{

        int b = digitalRead(ENCODER_B);

        if(b > 0){

            //Incremento variable global

            theta++;

        }

        else{

            //Decremento variable global

            theta--;

        }

    }

}

```

b) Explique que es el código gray y como se utiliza en los encoders absolutos

El **código Gray o código binario reflejado**, nombrado así en honor del investigador Frank Gray, es un sistema de numeración binario en el que dos números consecutivos difieren solamente en uno de sus dígitos.

De un valor en código Gray al siguiente solo hay un cambio de un bit, como puede apreciarse en la siguiente tabla. Se han incluido en ella las expresiones BCD, que ya son conocidas.

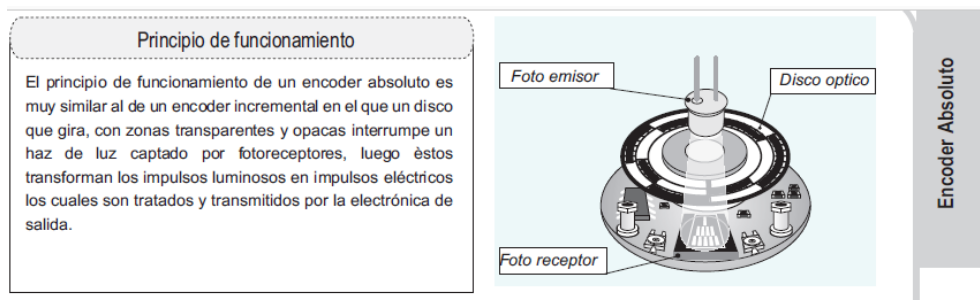
Observa también que los bits son simétricos respecto a la línea en la que se cambia de color.

Se puede ver que el código Gray es también **cíclico**, ya que cuando llegamos al número más alto, con un solo cambio de bit se vuelve al inicio.

El código Gray fue diseñado originalmente para prevenir señales falsas o viciadas de los relés que se empleaban en las primeras computadoras, y actualmente es usado para facilitar la corrección de errores en los sistemas de comunicaciones, tales como algunos sistemas de televisión por cable y la televisión digital terrestre.

En robótica, es utilizado principalmente en sistemas de posición (encoders), ya sea angular o lineal. Sus aplicaciones principales se encuentran en la industria y en robótica. Se suelen utilizar unos discos

codificados para dar la información de posición que tiene un eje en particular. Esta información se da en código GRAY.

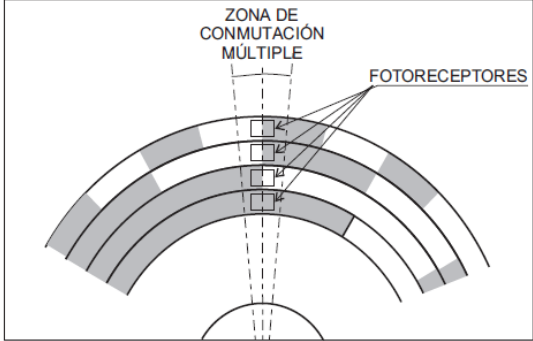


La codificación absoluta

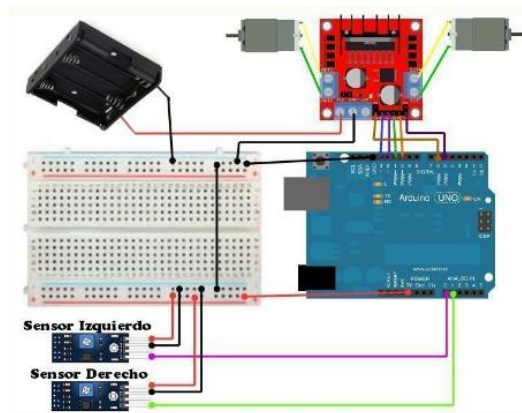
Respecto a los encoders incrementales, los encoders absolutos muestran importantes diferencias desde el punto de vista funcional. Mientras en los encoders incrementales la posición está determinada por el cómputo del número de impulsos con respecto a la marca de cero, en los encoders absolutos la posición queda determinada mediante la lectura del código de salida, el cual es único para cada una de las posiciones dentro de la vuelta. Por consiguiente los encoders absolutos no pierden la posición real cuando se corta la alimentación (incluso en el caso de desplazamientos), hasta un nuevo encendido (gracias a una codificación directa en el disco), la posición está actualizada y disponible sin tener que efectuar, como en el caso de los encoder incrementales la búsqueda del punto de cero.

Analicemos ahora el código de salida que se deberá utilizar para definir la posición absoluta. La elección más obvia es la del código binario, porque fácilmente puede ser manipulado por los dispositivos de control externos para la lectura de la posición, sin tener que efectuar particulares operaciones de conversión. En vista que el código se toma directamente desde el disco (que se encuentra en rotación) la sincronización y la captación de la posición en el momento de la variación entre un código y el otro se vuelve muy problemática. En efecto, si por ejemplo tomamos dos códigos binarios consecutivos como 7(0111) 8(1000), se nota que todos los bit del código sufren un cambio de estado: una lectura efectuada en el momento de la transición podría resultar completamente errónea porque es imposible pensar que las variaciones sean instantáneas y que se produzcan todas en el mismo momento. Debido a este problema se utiliza una variante del código binario: el código Gray, el cual tiene la particularidad que al pasar entre dos códigos consecutivos (o desde el último código al primero), uno sólo cambia su estado.

DECIMAL	BINARIO	GRAY
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000



c) Como implementaría el circuito de acondicionamiento de un fotodiodo utilizado para encontrar contraste en un auto robot seguidor de línea.



d) Implemente el controlador para un auto seguidor de línea utilizando el circuito del punto anterior.

1. `const int IN1 = 13;` // Pin digital
13- giro motor izquierdo
2. `const int IN2 = 12;` // Pin digital
12- giro motor izquierdo
- 3.

```

4.    // Definición de variables y
      constantes relacionadas con el motor
      derecho
5.    const int IN3 = 11;    // Pin digital
      11- giro motor izquierdo
6.    const int IN4 = 10;    // Pin digital
      10- giro motor izquierdo
7.
8.    // variables
9.    int lecturaSensorIzq;
10.   int lecturaSensorDer;
11.   const int sensorIzqPin = A1; // El
      sensor izq irá conectado al pin analógico
      A0
12.   const int sensorDerPin = A0
      ; // El sensor derecho irá conectado al
      pin analógico A1
13.
14.   void setup()
15.   {
16.       // Se declaran todos los pines como
      salidas
17.       // Pines asociados a los motores
18.       pinMode (IN1, OUTPUT);
19.       pinMode (IN2, OUTPUT);
20.       pinMode (IN3, OUTPUT);
21.       pinMode (IN4, OUTPUT);
22.       pinMode ( sensorIzqPin , INPUT) ;
23.       pinMode ( sensorDerPin , INPUT) ;
24.       Serial.begin(9600); // Inicio
25.   }
26.
27.   void loop()
28.   {
29.

```

```

30.      lecturaSensorIR(); // Lee sensores
      IR
31.      // Se analiza para hacer que el
      robot siga la línea negra
32.
33.      // Si el resultado de ambos sensores
      es 0 (zona blanca) el robot sigue se para
34.      if(lecturaSensorIzq == 0 &&
      lecturaSensorDer == 0)
35.      {
36.          robotParar(); // para
37.
38.      }
39.      // Si el izquierdo retorna 0 (zona
      blanca) y el derecho 1 (negra) el robot
      gira derecha
40.      if (lecturaSensorIzq == 0 &&
      lecturaSensorDer == 1)
41.      {
42.          robotDerecha();
43.          // gira a la derecha
44.
45.      }
46.      // Si el izquierdo retorna 1 (zona
      negra) y el derecho 0 (blanca) el robot
      gira izquierda
47.      if (lecturaSensorIzq == 1 &&
      lecturaSensorDer == 0)
48.      {
49.          robotIzquierda();
50.
51.      }
52.      // Si ambos sensores retornan 0
      (zona negra) el robot sigue recto

```



```

53.     if (lecturaSensorIzq == 1 &&
        lecturaSensorDer == 1)
54.     {
55.         robotAvance(); // El robot avanza
56.         Serial.println("robot avanza");
57.     }
58.
59. }
60. /*
61.     Función lecturaSensorIR: leerá el
        valor del sensor de infrarrojos TCRT5000
62.     y lo almacena en una variable. Dicho
        sensor retornará el valor 0 (LOW) si
63.     el sensor está en zona blanca y el
        valor 1 (HIGH) si el sensor está en zona
64.     negra.
65. */
66. void lecturaSensorIR()
67. {
68.     lecturaSensorIzq =
        digitalRead(sensorIzqPin); // Almacena la
        lectura del sensor izquierdo
69.     lecturaSensorDer =
        digitalRead(sensorDerPin); // Almacena la
        lectura del sensor derecho
70.
71.     Serial.println("El valor del sensor
        izquierdo es ");
72.     Serial.println(lecturaSensorIzq);
73.
74.     Serial.println("El valor del sensor
        derecho es ");
75.     Serial.println(lecturaSensorDer);
76.
77. }

```

```

78.  /*
79.      Función robotAvance: esta función
      hará que ambos motores se activen por lo
      que el robot avanzará hacia delante
80.  */
81.  void robotAvance()
82.  {
83.      // Motor izquierdo
84.      // Al mantener un pin HIGH y el otro
      LOW el motor gira en un sentido
85.      digitalWrite (IN1, HIGH);
86.      digitalWrite (IN2, LOW);
87.
88.      // Motor derecho
89.      // Al mantener un pin HIGH y el otro
      LOW el motor gira en un sentido
90.      digitalWrite (IN3, HIGH);
91.      digitalWrite (IN4, LOW);
92.  }
93.  /*
94.      Función robotRetroceso: esta función
      hará que ambos motores se activen a máxima
      potencia
95.      en sentido contrario al anterior por
      lo que el robot avanzará hacia atrás
96.  */
97.  void robotRetroceso()
98.  {
99.      // Motor izquierdo
100.     // Al mantener un pin LOW y el otro
      HIGH el motor gira en sentido contrario al
      anterior
101.     digitalWrite (IN1, LOW);
102.     digitalWrite (IN2, HIGH);
103.

```

```

104.    // Motor derecho
105.    // Al mantener un pin LOW y el otro
    HIGH el motor gira en sentido contrario al
    anterior
106.    digitalWrite (IN3, LOW);
107.    digitalWrite (IN4, HIGH);
108. }
109.
110. /*
111.     Función robotDerecha: esta función
    accionará el motor izquierdo y parará el
    derecho
112.     por lo que el coche girará hacia la
    derecha (sentido horario)
113. */
114. void robotDerecha()
115. {
116.     // Motor izquierdo
117.     // Se activa el motor izquierdo
118.     digitalWrite (IN1, HIGH);
119.     digitalWrite (IN2, LOW);
120.
121.     // Motor derecho
122.     // Se para el motor derecho
123.     digitalWrite (IN3, LOW);
124.     digitalWrite (IN4, HIGH);
125.
126. }
127. /*
128.     Función robotIzquierda: esta función
    accionará el motor derecho y parará el
    izquierdo por lo que el coche girará hacia
    la izquierda (sentido antihorario)
129. */
130. void robotIzquierda ()

```

```

131.  {
132.      // Motor izquierdo
133.      // Se para el motor izquierdo
134.      digitalWrite (IN1, LOW);
135.      digitalWrite (IN2, HIGH);
136.
137.      // Motor derecho
138.      // Se activa el motor derecho
139.      digitalWrite (IN3, HIGH);
140.      digitalWrite (IN4, LOW);
141.  }
142.  /*
143.      Función robotParar: esta función
        parará ambos motores
144.      por lo que el robot se parará.
145.  */
146.  void robotParar()
147.  {
148.      // Motor izquierdo
149.      // Se para el motor izquierdo
150.      digitalWrite (IN1, LOW);
151.      digitalWrite (IN2, LOW);
152.
153.      // Motor derecho
154.      // Se para el motor derecho
155.      digitalWrite (IN3, LOW);
156.      digitalWrite (IN4, LOW);
157.
158.  }

```