

INGENIERIA INFORMATICA
Escuela Politécnica Superior
Universidad Autónoma De Madrid

Ay!Bot

Tarea 4

12/8/2019

Índice de Contenidos

1. Introducción.....	2
2. Planteamiento del problema	3
3. Solución del problema.....	5
4. Código final	7
5. Simulación del circuito	9

Lista de Tablas y Figuras

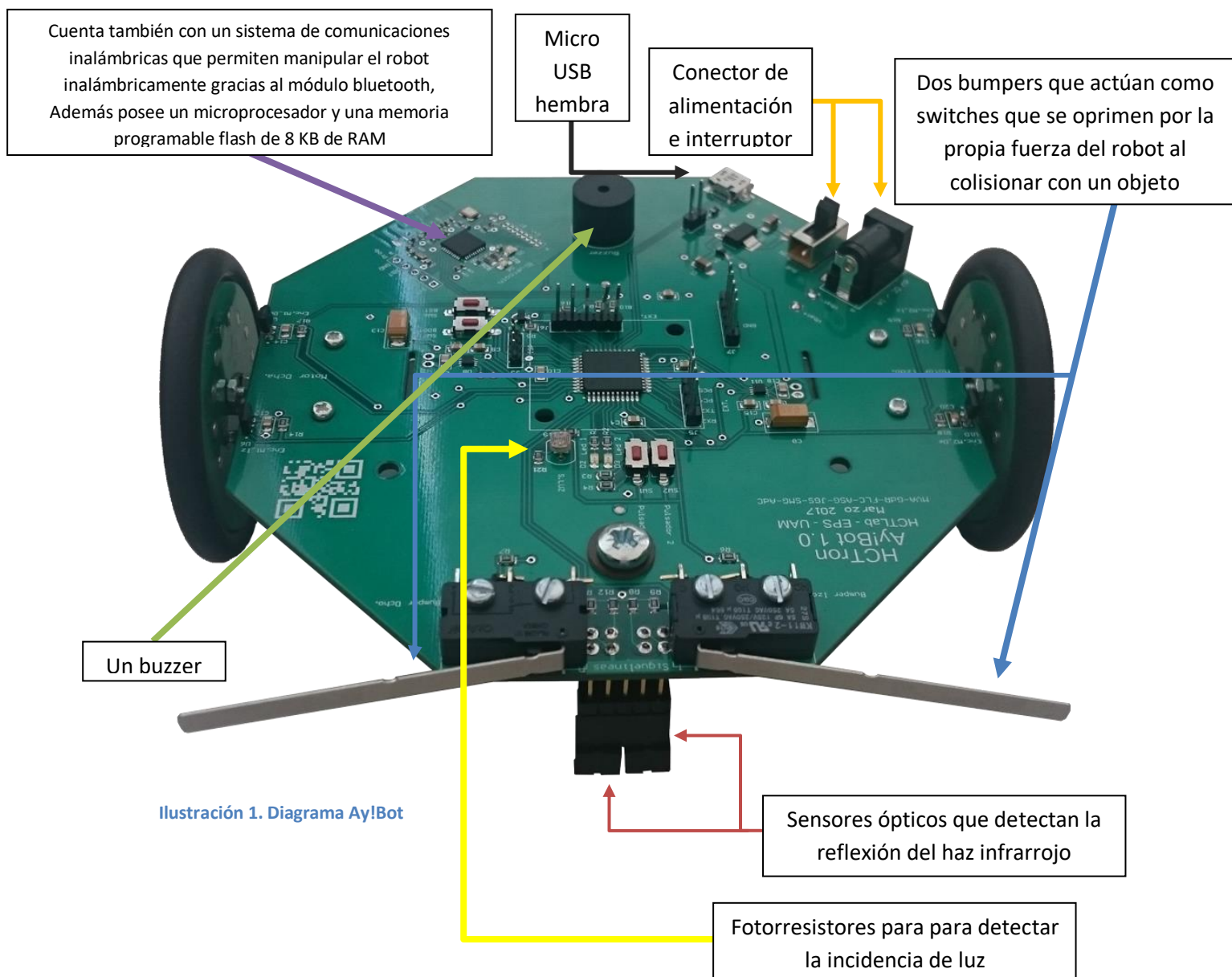
Ilustración 1. Diagrama Ay!Bot.....	2
Ilustración 2. Diagrama reflexión de la luz en los colores	3
https://medium.com/@a20183390_21926/isaac-newton-reflexi%C3%B3n-y-refracci%C3%B3n-de-la-luz-2fb8052fdd79	
Ilustración 3. Resolución circuito 1	9
Ilustración 4. Resolución circuito 2	9

1. Introducción

En esta práctica el objetivo es programar el robot para que sea capaz de resolver unos circuitos siguiendo la línea negra.

El robot en cuestión se trata del Ay!Bot, el cual cuenta con diferentes sensores que nos permitirán realizar con éxito la práctica.

Además de los fotorresistores con los que cuenta para para detectar la incidencia de luz, cuenta entre otros con:



2. Planteamiento del problema

El ejercicio planteado consiste en seguir diferentes circuitos siguiendo siempre la línea negra.

Para ello emplearemos los sensores ópticos que detectan la reflexión de luz y que nos vendrán muy bien para la práctica ya que el color negro se observa en cuerpos que absorben todos los colores de la luz y por tanto no se produce reflexión, lo que emplearemos para saber si se encuentra dentro de la línea que debe seguir, o en caso contrario, se podrá saber en qué dirección debe girar para que vuelva al circuito.

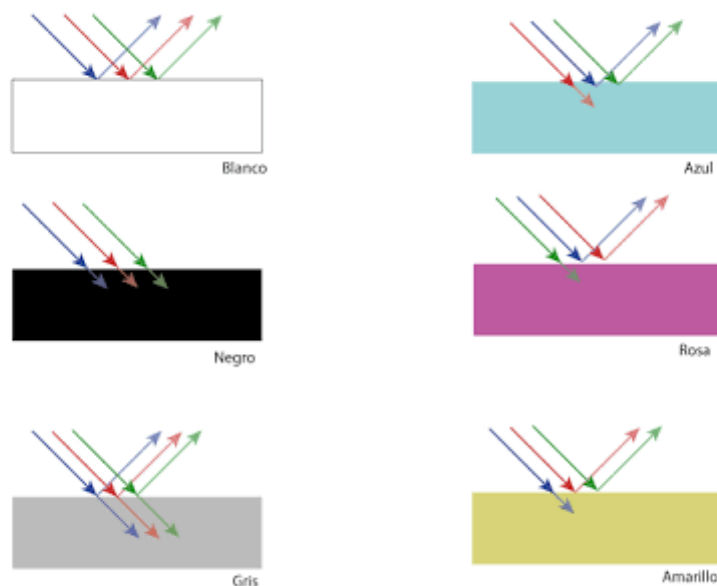


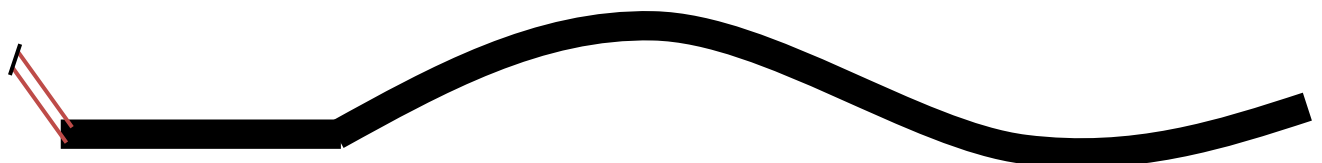
Ilustración 2. Diagrama reflexión de la luz en los colores

Ahora falta saber cómo plantear el problema a través de la información que recibimos gracias a los sensores.

Supongamos que se desea superar el siguiente circuito, las líneas rojas representan los fototransistores.

Lo primero que debemos plantearnos es la posición inicial de los mismos; supongamos que ambos inician sobre la línea.

Es obvio en este caso que mientras ambos se encuentren en la línea debe ir recto.



Es decir, mientras ambos sensores registren un 1 (la luz incide en el color negro y por tanto no se produce la reflexión).

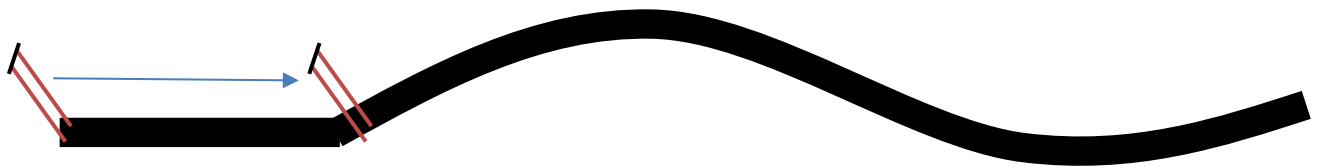
```
if(SensoresLuz_led_izquierdo() == 1 && SensoresLuz_led_derecho() == 1){  
  
  Motores_avanza("potencia del motor");  
  
  _delay_ms("tiempo en ms de delay");  
  
}
```

El robot seguirá recto hasta que se produzca un cambio en el valor de retorno de los sensores;

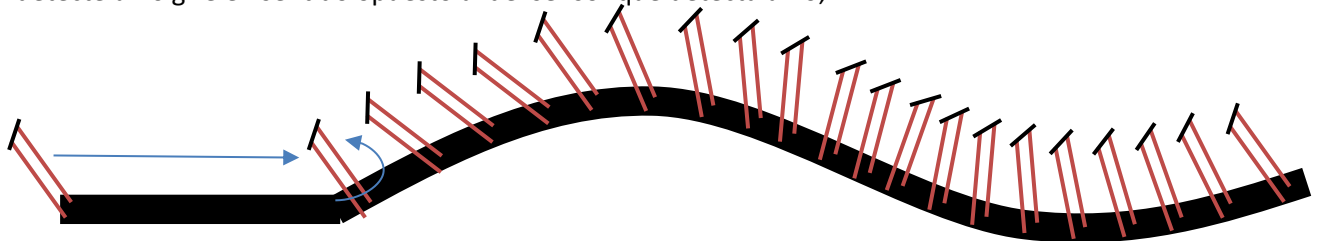
Siguiendo nuestro ejemplo, el primer cambio que se produce es que el sensor derecho llega al blanco lo que significa que, si tiene lugar la reflexión de luz, por lo que, para que el robot siga avanzando tiene que girar para que ambos vuelvan a registrar 1

```
else if (SensoresLuz_led_izquierdo() == 1 && SensoresLuz_led_derecho() == 0){  
  
  Motores_gira_izq(40);  
  
  _delay_ms(1);  
  
}
```

El caso derecha es idéntico al caso izquierda cambiando el valor de los sensores y la dirección a la que girar.



Lo que significa que debemos programar el robot de forma que cuando uno de los sensores detecte un 0 gire en sentido opuesto al del sensor que detecta un 0;



No obstante, este método tiene un problema y es que no tiene definido que hacer en caso de encontrarse con un giro de 90°.

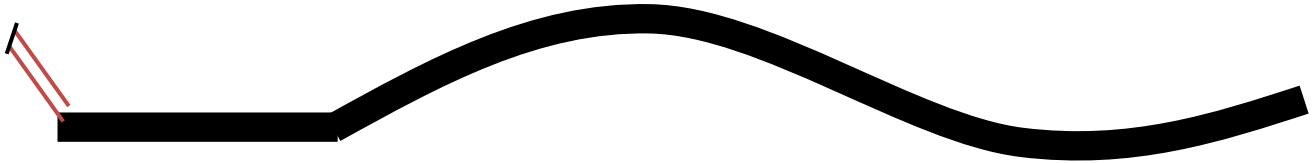
El problema se podría resolver calculado en tiempo que tarda en hacer los noventa grados (definiendo una función o una macro con el tiempo estimado en hacer el giro a una determinada potencia), que realiza los noventa grados en una dirección y en caso de mantener ambos sensores a 0, vuelva a la posición original y gire esos noventa grados a la otra dirección.

No obstante, sigue sin ser la solución más óptima pues puede ocurrir que en un circuito con las líneas más finas y curvas empinadas se pierda y detecte ambos sensores a 0.

3. Solución del problema

Como hemos visto, colocando el robot en un inicio con ambos sensores detectando la línea negra llegamos a un problema por lo que probaremos cambiando la posición inicial de los sensores;

En lugar de colocar ambos sobre la línea, colocaremos un sensor sobre la misma y otro fuera.



De esta forma empleamos un sensor para saber cuándo ir recto y otra para saber si tiene que girar y en función de la combinación de ambos determinar el sentido del giro.

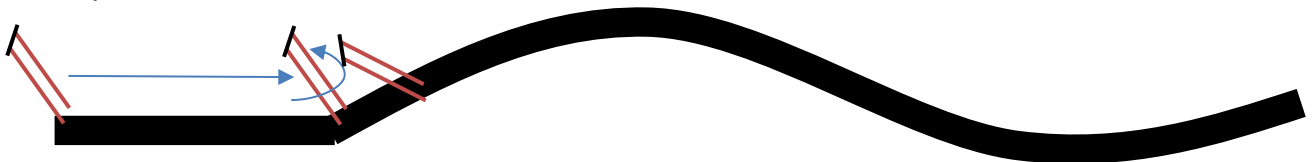
Viendo el diagrama, es obvio que mientras que la situación el dibujo se mantenga el robot debe ir recto, es decir, mientras se cumpla que el sensor de luz izquierdo se mantenga a 0 y el derecho a 1, el robot seguirá recto.

```
else if(SensoresLuz_led_izquierdo() == 0 && SensoresLuz_led_derecho() == 1){  
    Motores_avanza(20);  
    _delay_ms(1);  
}
```

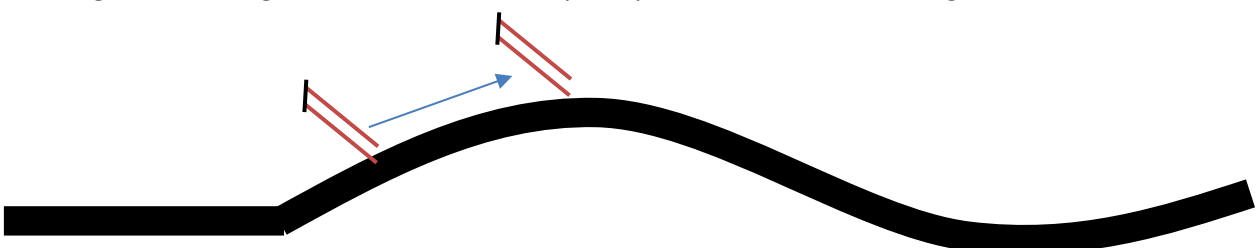


El primer cambio que se aprecia en los sensores ocurre en la primera curva en la que ambos sensores pasan a valer a 1, en esta situación, el robot debe girar para volver al caso anterior y seguir avanzando.

```
else if(SensoresLuz_led_izquierdo() == 1 && SensoresLuz_led_derecho() == 1){  
    Motores_gira_izq(40);  
    _delay_ms(1);  
}
```



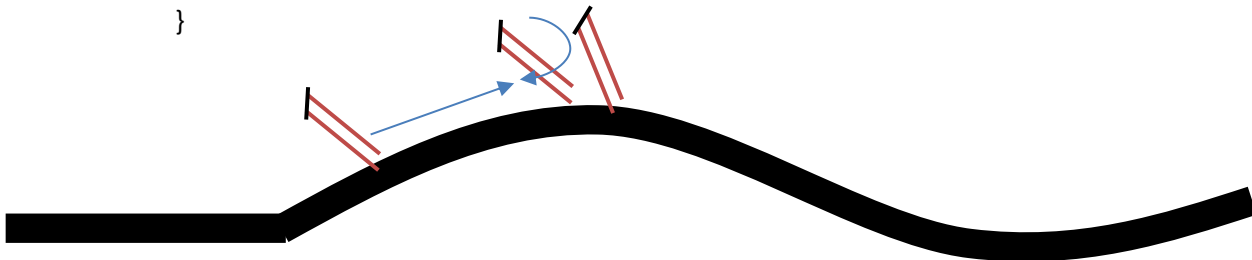
Siguiendo el diagrama, el último cambio que experimenta el robot es el siguiente;



Siguiendo el mismo razonamiento, el robot en esta ocasión debe girar hasta que nuevamente el robot vuelva al estado en el que avanza;

Es decir, debe girar hasta que un sensor derecho vuelva a estar en 1;

```
if(SensoresLuz_led_izquierdo() == 0 && SensoresLuz_led_derecho() == 0){  
    Motores_gira_der(40);  
    _delay_ms(1);  
}
```



Con este último cambio el robot es perfectamente capaz de realizar el circuito sin problemas. El código resultante es el siguiente:

Para que funcione sin importar el sensor que este dentro, se declara una función que sirve para saber cual es el sensor que se encuentra dentro, y en función de cual de los dos sea, retornará un valor u otro.

4. Código final

```
#include "ayBot/ConfigPlaca.h"

int main (void){
    int n=0;
    iniciarRobot(false);//
    do{
        if(SensoresLuz_led_izquierdo() == 0 && SensoresLuz_led_derecho()
== 1){
            n=1;
        }
        else if(SensoresLuz_led_izquierdo() == 1 &&
SensoresLuz_led_derecho() == 0){
            n=2
        }
    }while(n==0);
    if(n==1){
        do{
            if(SensoresLuz_led_izquierdo() == 0 && SensoresLuz_led_derecho()
== 0){
                Motores_gira_der(40);
                _delay_ms(1);
            }
            else if(SensoresLuz_led_izquierdo() == 0 &&
SensoresLuz_led_derecho() == 1){
                Motores_avanza(20);
                _delay_ms(1);
            }
        }
        else if(SensoresLuz_led_izquierdo() == 1 &&
SensoresLuz_led_derecho() == 0){
            Motores_gira_izq(40);
            _delay_ms(1);
        }
    }
    else{
        Motores_gira_izq(40);
        _delay_ms(1);
    }
}
```



```
    }  
  }while(true);  
}  
if(n==2){  
  do{  
    if(SensoresLuz_led_izquierdo() == 0 && SensoresLuz_led_derecho()  
== 0){  
      Motores_gira_izq(40);  
      _delay_ms(1);  
    }  
    else if(SensoresLuz_led_izquierdo() == 0 &&  
SensoresLuz_led_derecho() == 1){  
      Motores_gira_der(40);  
      _delay_ms(1);  
    }  
    else if(SensoresLuz_led_izquierdo() == 1 &&  
SensoresLuz_led_derecho() == 0){  
      Motores_avanza(20);  
      _delay_ms(1);  
    }  
    else{  
      Motores_gira_der(40);  
      _delay_ms(1);  
    }  
  }while(true);  
}  
  
return 0;  
}
```

5. Simulación del circuito

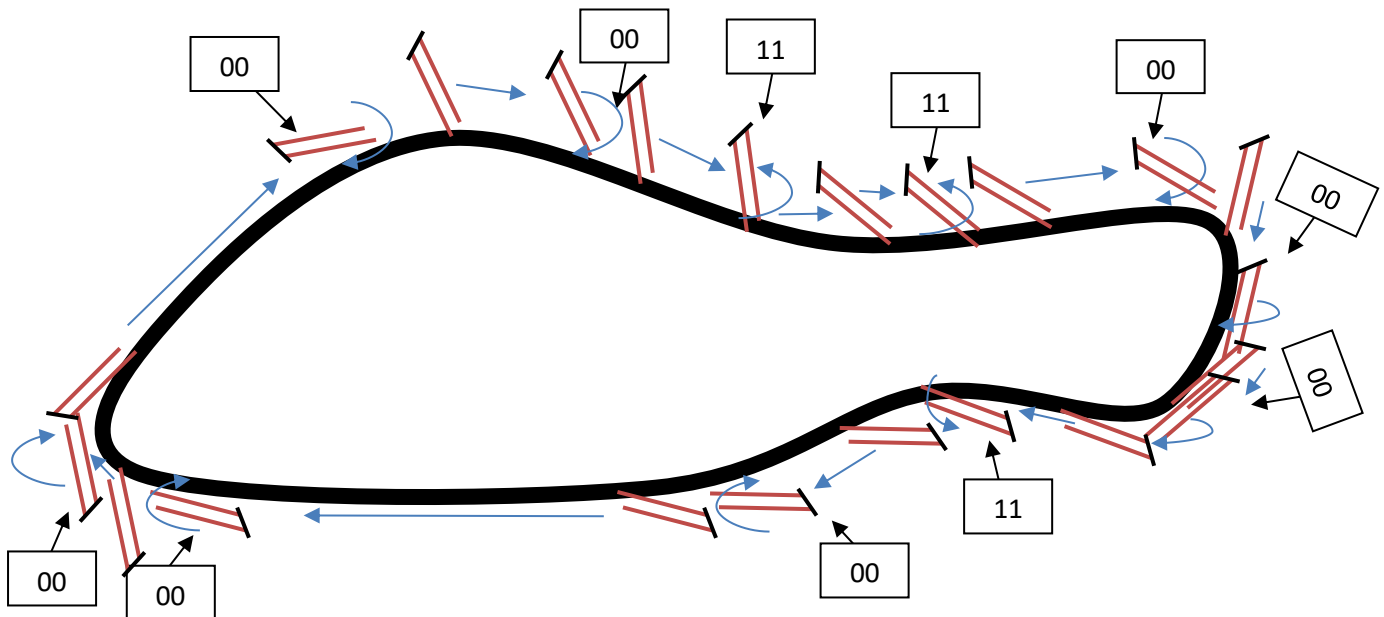


Ilustración 3. Resolución circuito 1

Entrada (sensor izq/sensor derecha)	00	01	10	11
Salida	derecha	recto	izquierda	izquierda

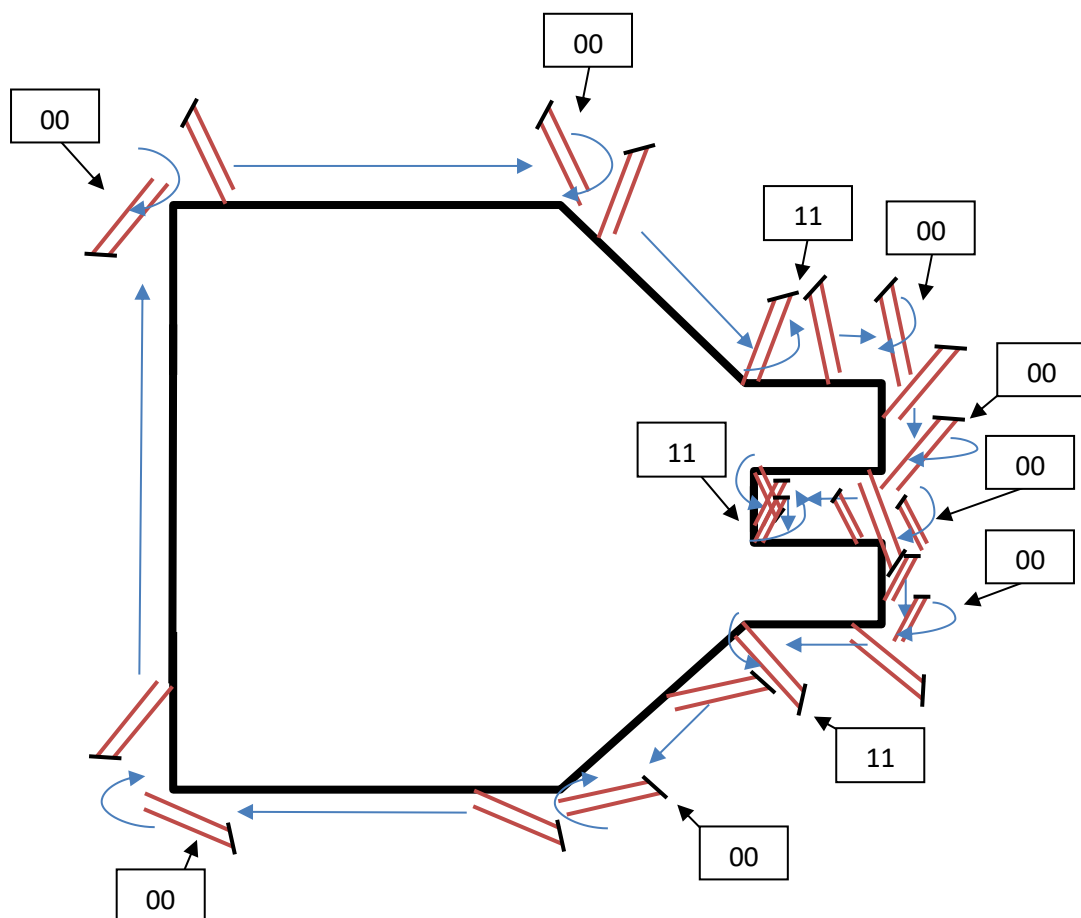


Ilustración 4. Resolución circuito 2

[FINAL DE DOCUMENTO]