

ROBÓTICA

INTRODUCCIÓN Y PRINCIPIOS BÁSICOS

Sensor ultrasónico

Modalidad: Individual

Valor porcentual: 5%

Puntos: 20 pts

Objetivos: Comprender el uso del condicional if, la función tone de Arduino y el funcionamiento del sensor ultrasónico

Recursos necesarios para realizar la actividad:

- Tener instalado el programa Arduino en su PC
- Haber estudiado la unidad 8
- 1 cable USB mini B
- El robot completo
- El speaker de 8 omh
- 1 cable female to female

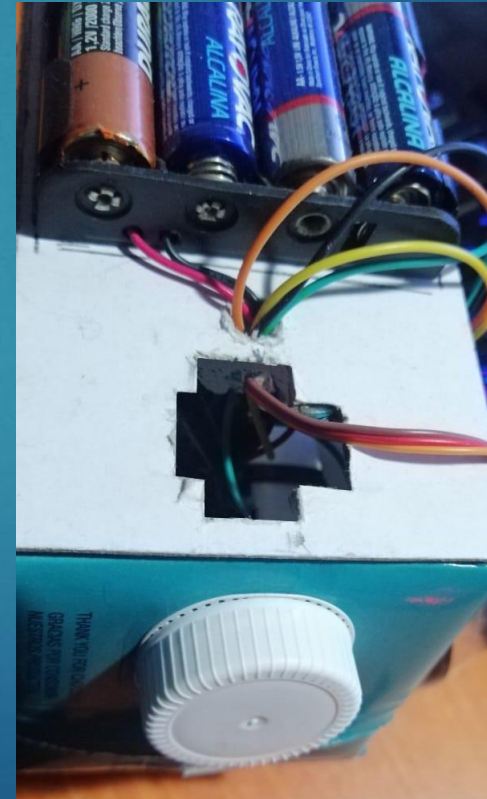
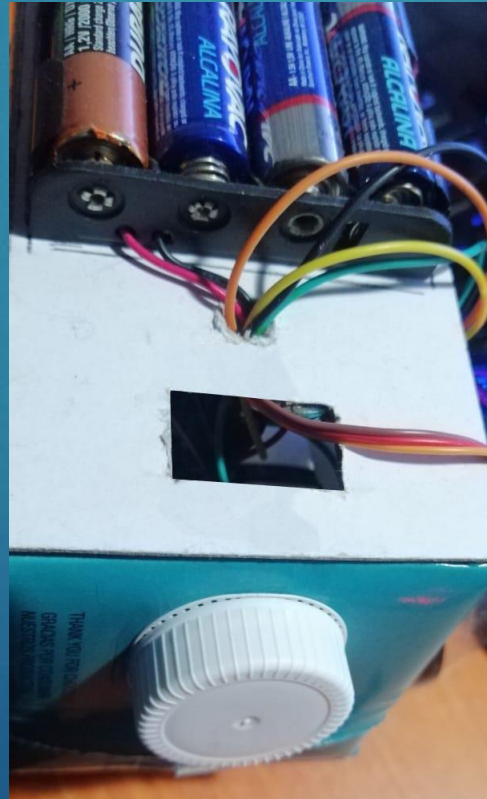
Instrucciones para realizar la Actividad :

- Lea primero toda la práctica.
- Ejecute los pasos de acuerdo a lo solicitado.
- Entregue las evidencias de aprendizaje (Fotografías del trabajo hecho) por medio del enlace de entrega.

Primera parte (Ajuste del servo motor).

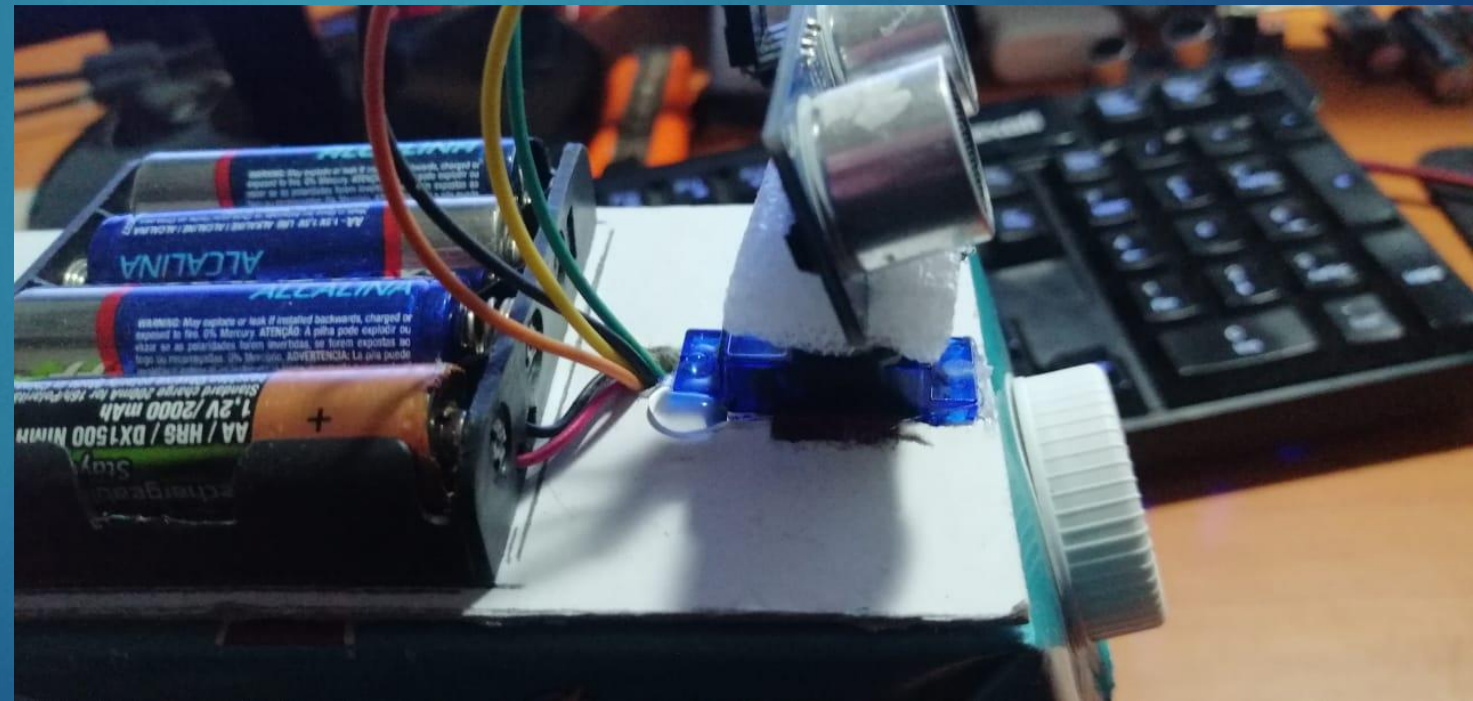
Para continuar nuestro aprendizaje con el robot debemos hacer un pequeño ajuste: cambiar de posición el servo motor, que habíamos colocado de manera horizontal con respecto al frente del robot, pero por ser un servo motor de 180° , hay que colocarlo de manera vertical con respecto al frente del robot, de la siguiente manera:

- 1- Apartamos temporalmente el servo motor de su lugar.
- 2- Cortamos con un cutter el nuevo agujero:



Ajuste del servo motor

Colocaremos de nuevo el servo motor en su nuevo lugar, y la fijaremos con silicón.

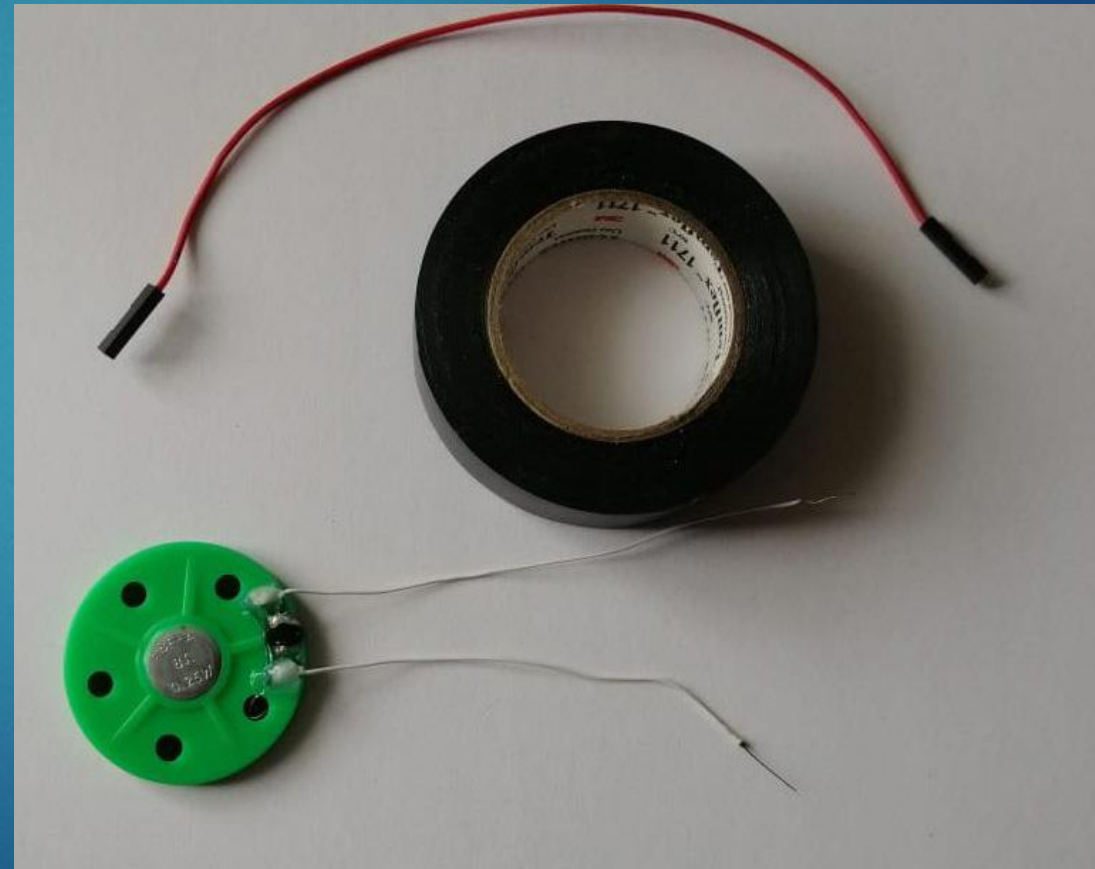


Segunda parte (Colocar el parlante dentro del robot).

Debemos conectar el parlante a la placa, esto para configurar algunas alertas de sonido, de acuerdo a nuestra conveniencia.

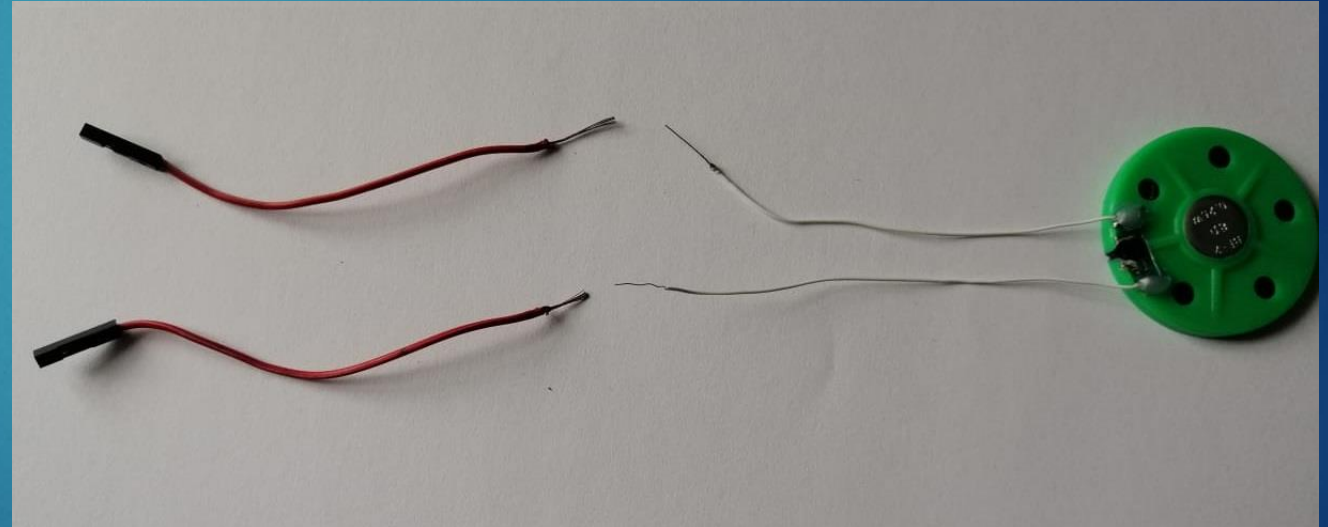
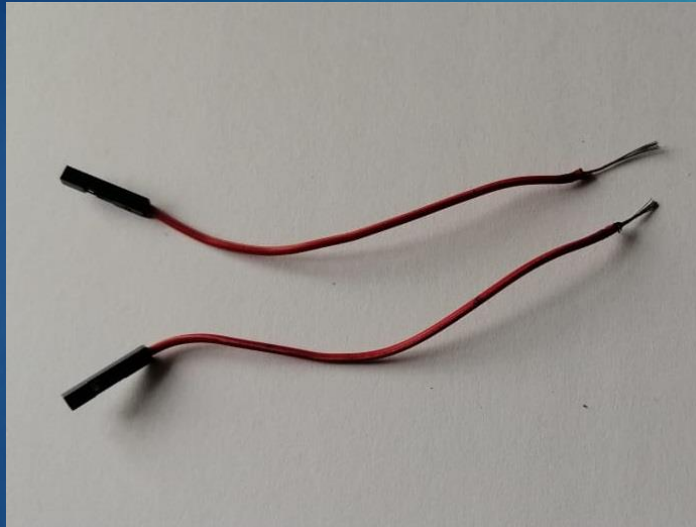
Para realizar este procedimiento necesitaremos:

- 1- Cable female to female
- 2- Un speaker
- 3- Cinta aislante negra (tape)



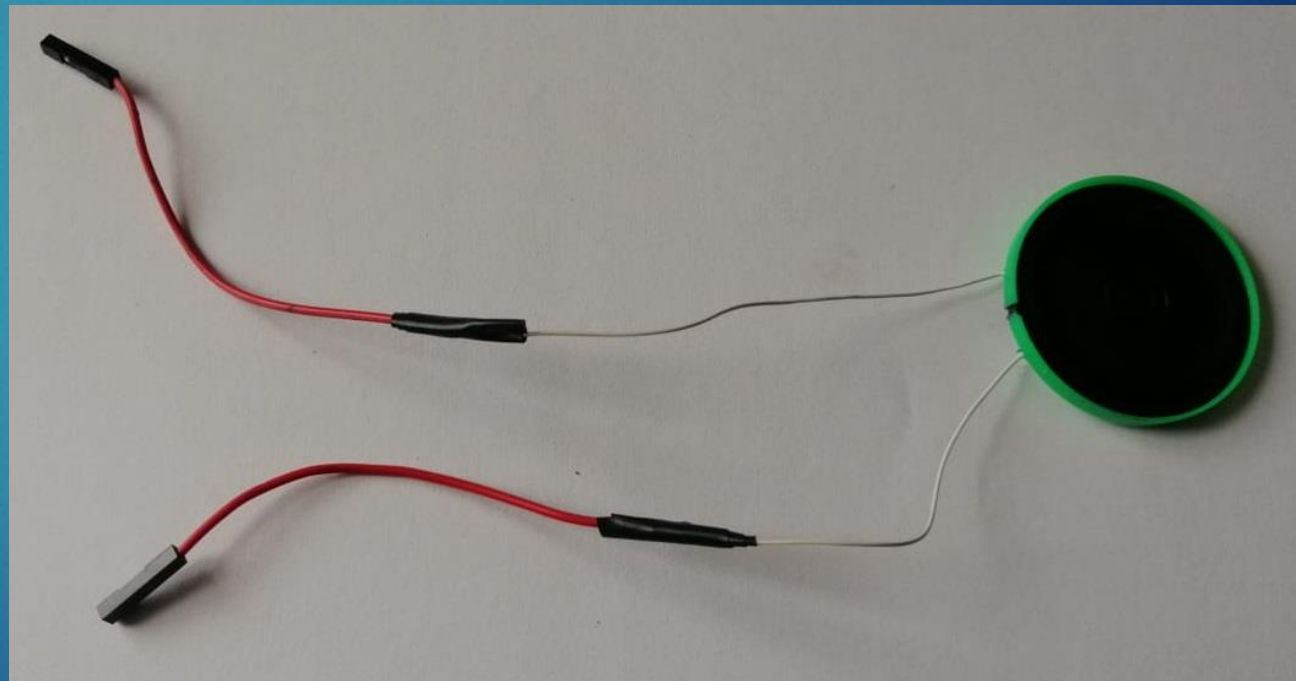
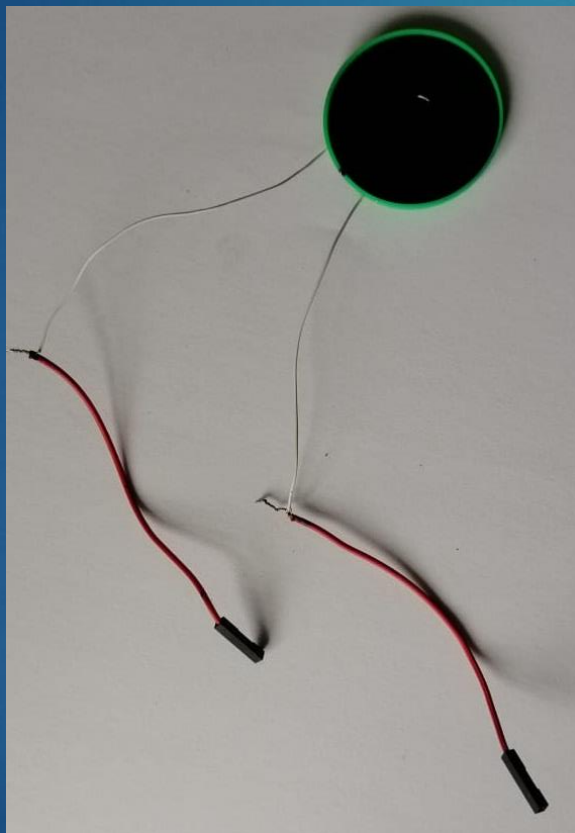
Colocar el parlante dentro del robot

Cortaremos el cable female to female a la mitad, y removeremos el plástico de los extremos de ambos cables, para poder emparejar el cobre de los cables con los del speaker:



Colocar el parlante dentro del robot

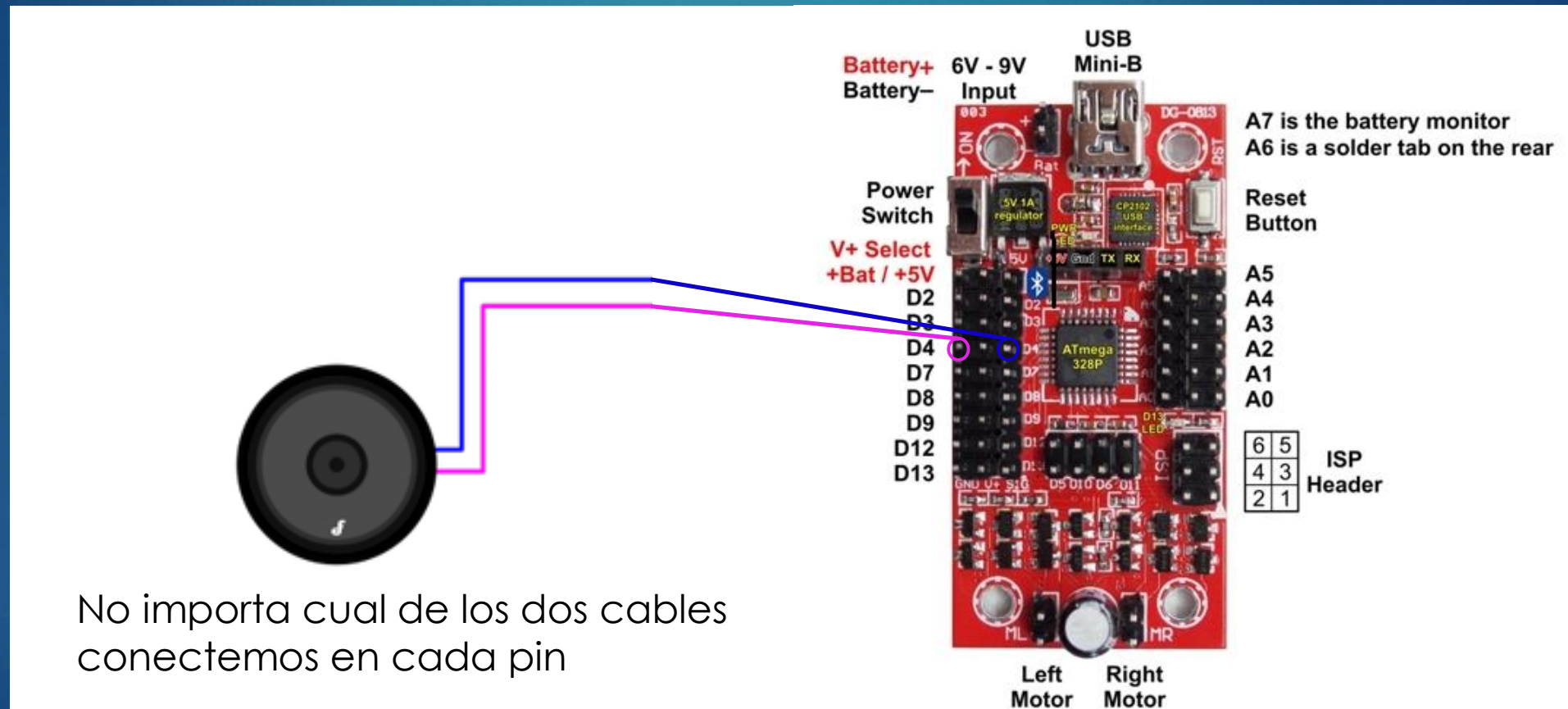
Aseguraremos los extremos de los cables “enrollándolos” y cubriéndolos con la cinta aislante, para evitar cortos:





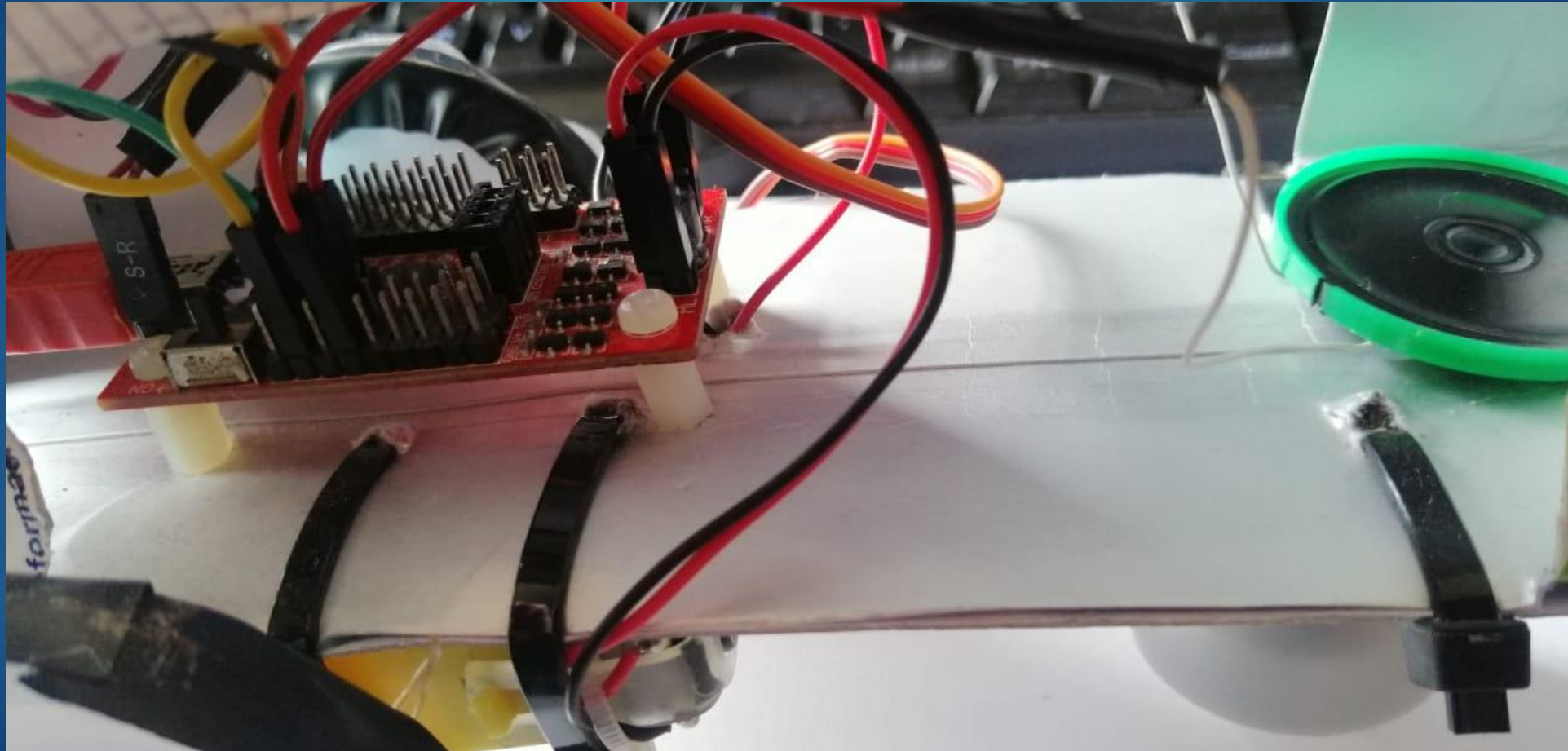
Colocar el parlante dentro del robot

El speaker lo conectaremos al pin digital 4 de la placa, un de los cables al Ground, y el otro al pin de 5 voltios:



Colocar el parlante dentro del robot

El speaker quedaría conectado así, podemos asegurarlo al chasis, para que no quede suelto, con una gota de silicón,



Programar sonidos

Crearemos un pequeño programa de Arduino, con el único fin de probar el funcionamiento del parlante o speaker:

- 1- Cree una carpeta con el nombre tone
- 2- Cree un nuevo programa de Arduino, guárdelo con en la carpeta tone con el nombre tone.
- 3- Escriba las siguientes instrucciones:

```
tone
1 void setup() {
2
3 }
4 void loop() {
5 sonido();
6 }
7
8 void sonido(){
9 tone(4,392,200) ; //Se declara que al pin 4, se le asigna al tono un valor de 392 (Sol) por 200 milésimas de segundo
10 delay(500) ;//Ejecuta la acción por 500 milésimas de segundo
11 tone(4,294,200) ; //Se declara que al pin 4, se le asigna al tono un valor de 294 (Re) por 200 milésimas de segundo
12 delay(500) ; //Ejecuta la acción por 500 milésimas de segundo
13 }
```

4- Si el robot está conectado por medio del cable USB a la computadora, podemos subir el programa a la placa, si todo está bien, deberá de emitir un sonido repetitivo. En conclusión, ya sabemos como hacer para que el robot emita una alarma.

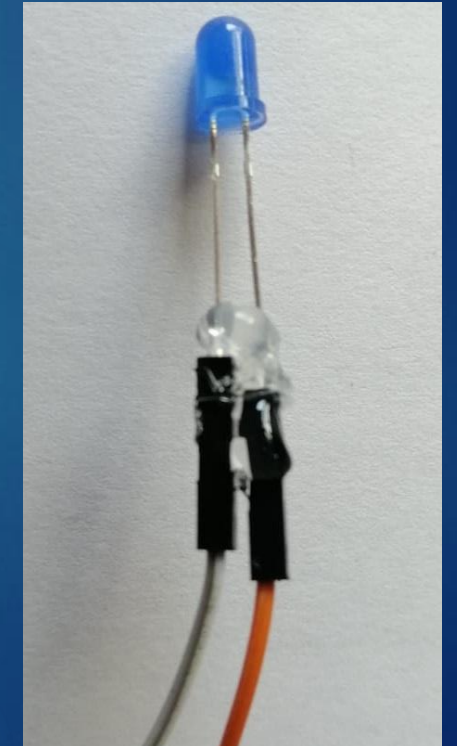
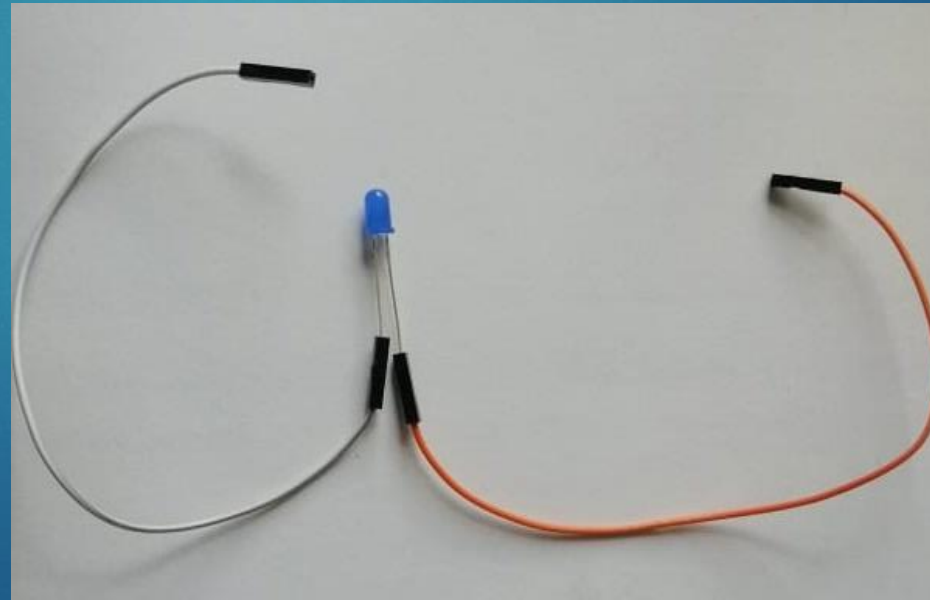
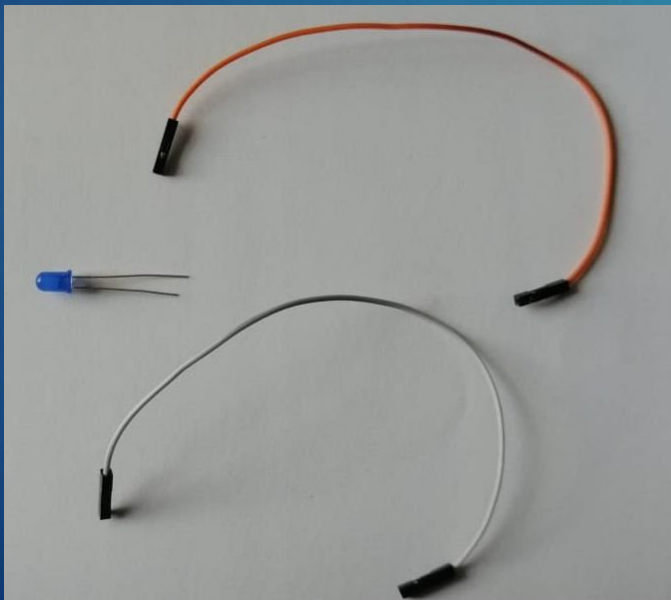
Colocando las luces led en el robot

Vamos a colocar tres luces led en el robot, lo primero que debemos hacer es preparar las luces, conectando sus pines a dos cables female to female, para poder unirlos a la placa.

Tenemos que fijarnos bien en cual cable colocamos el pin positivo, para no equivocarnos al conectar la luz a la placa.

El color del led y de los cables no es importante, en este ejemplo el pin más largo (El positivo) le colocaremos el cable color naranja, y al pin pequeño el cable blanco.

Para que la luz no se suelte de los cables, la aseguraremos con una gota de silicón.



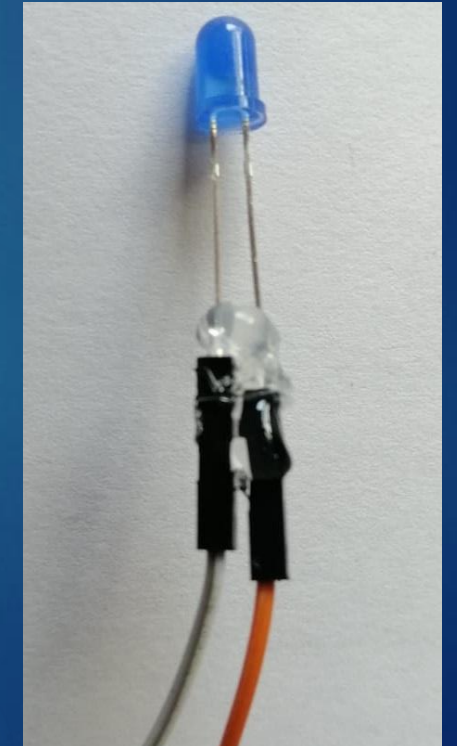
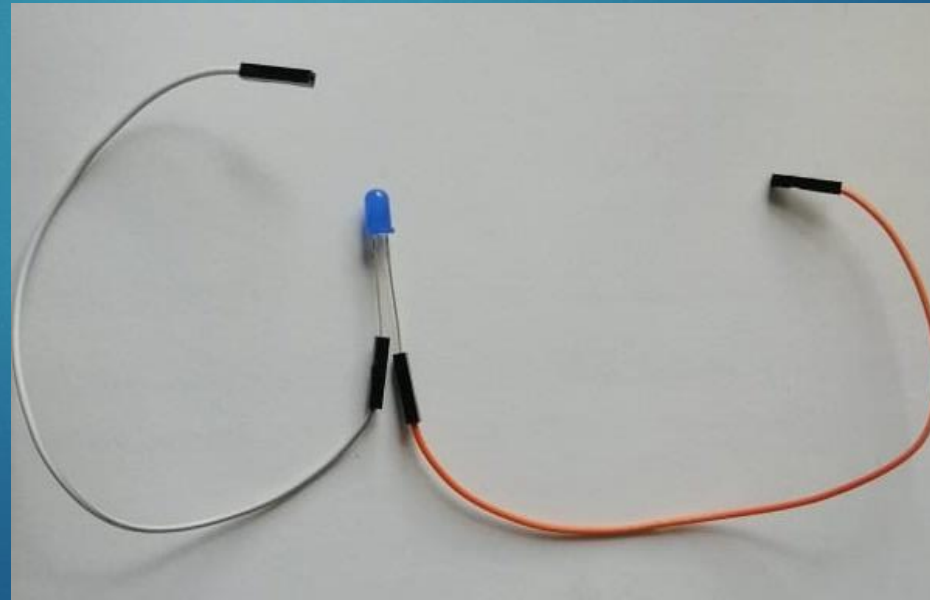
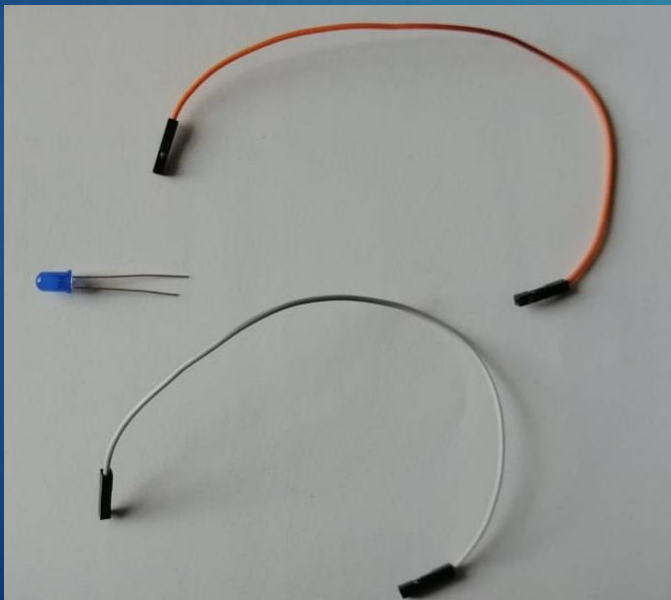
Colocando las luces led en el robot

Vamos a colocar tres luces led en el robot, lo primero que debemos hacer es preparar las luces, conectando sus pines a dos cables female to female, para poder unirlos a la placa.

Tenemos que fijarnos bien en cual cable colocamos el pin positivo, para no equivocarnos al conectar la luz a la placa.

El color del led y de los cables no es importante, en este ejemplo el pin más largo (El positivo) le colocaremos el cable color naranja, y al pin pequeño el cable blanco.

Para que la luz no se suelte de los cables, la aseguraremos con una gota de silicón.



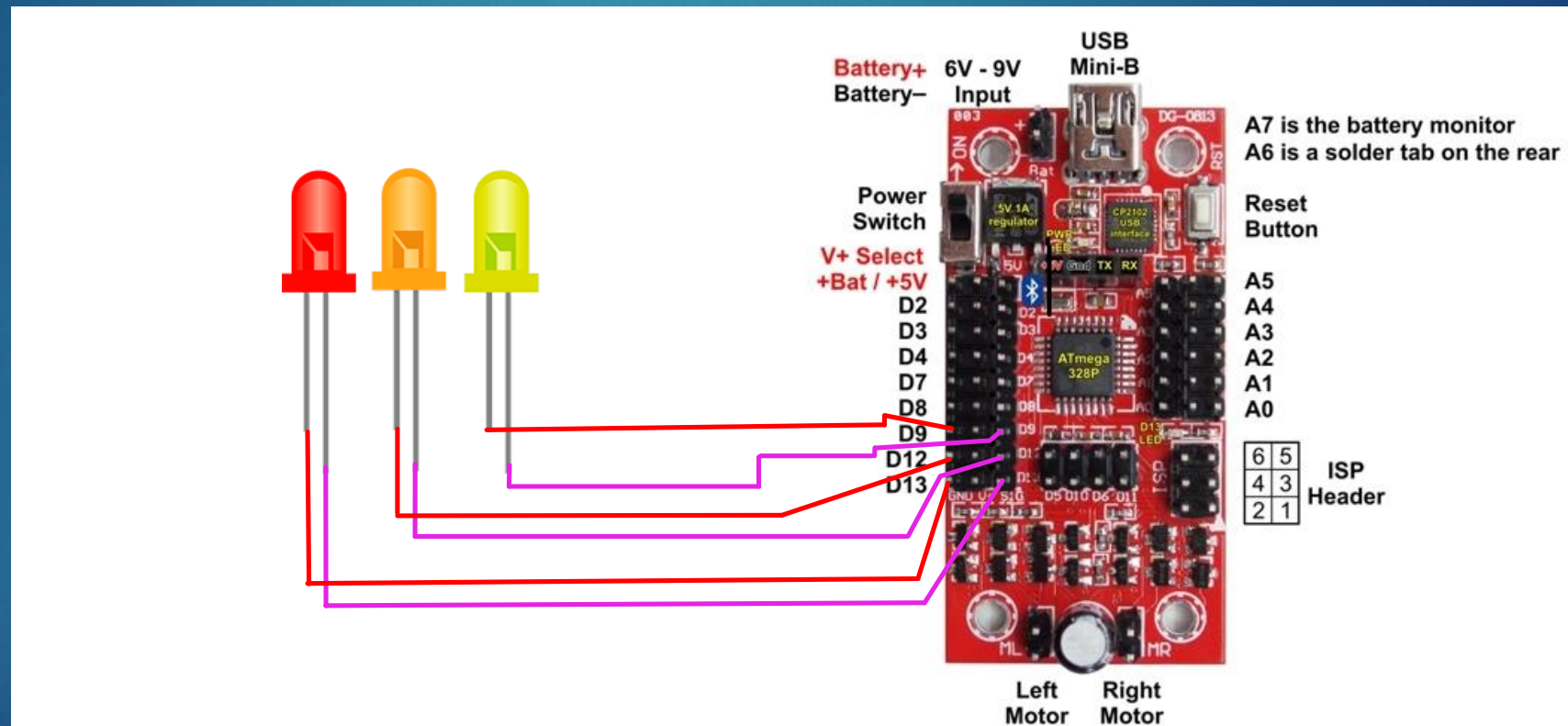
Colocando las luces led en el robot

Podemos fijar las tres luces en cualquier lugar del robot, en este ejemplo, se colocaron justo detrás del holder de las baterías con una gota de silicón cada luz.



Colocando las luces led en el robot

Las luces están debidamente conectadas en los pines 13, 12 y 9



Actualizando el programa con la función de sonido

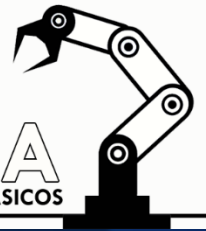
Necesitamos agregar al programa del laboratorio anterior, las funciones de encender luces y activar el sonido, para hacerlo haremos lo siguiente:

- 1- Crear una nueva carpeta que se llame laboratorio8
- 2- Crear un documento nuevo de Arduino y guardarlo en la carpeta laboratorio8 con el nombre laboratorio8
- 3- Abrir el programa del laboratorio 7
- 4- Seleccionar y copiar todas las instrucciones que escribimos y probamos en el laboratorio anterior
- 5- Pegar lo que copiamos en el archivo del laboratorio8

El programa hasta este momento estaría así:

```
Laboratorio8
1 int velocidadDerecha=100;
2 int velocidadIzquierda=100;
3
4 void setup() {
5   pinMode(10,OUTPUT);
6   pinMode(5,OUTPUT);
7   pinMode(11,OUTPUT);
8   pinMode(6,OUTPUT);
9   pinMode(12, OUTPUT);
10  pinMode(13, OUTPUT);
11  pinMode(9, OUTPUT);
12 }
13 void loop() {
14   velocidadDerecha=100;
15   velocidadIzquierda=100;
16   adelante();
17   delay(3000);
18   giro();
19   delay(1000);
20   reversa();
21   delay(2000);
22   detener();
23   delay(2000);
24 }
25
26 //Función para ir hacia adelante
27 void adelante()
28 {
29   digitalWrite(10, HIGH);
30   digitalWrite(11, HIGH);
31   analogWrite(5, velocidadDerecha);
32   analogWrite(6, velocidadIzquierda);
33   digitalWrite(12, HIGH);
34   digitalWrite(13, LOW);
```

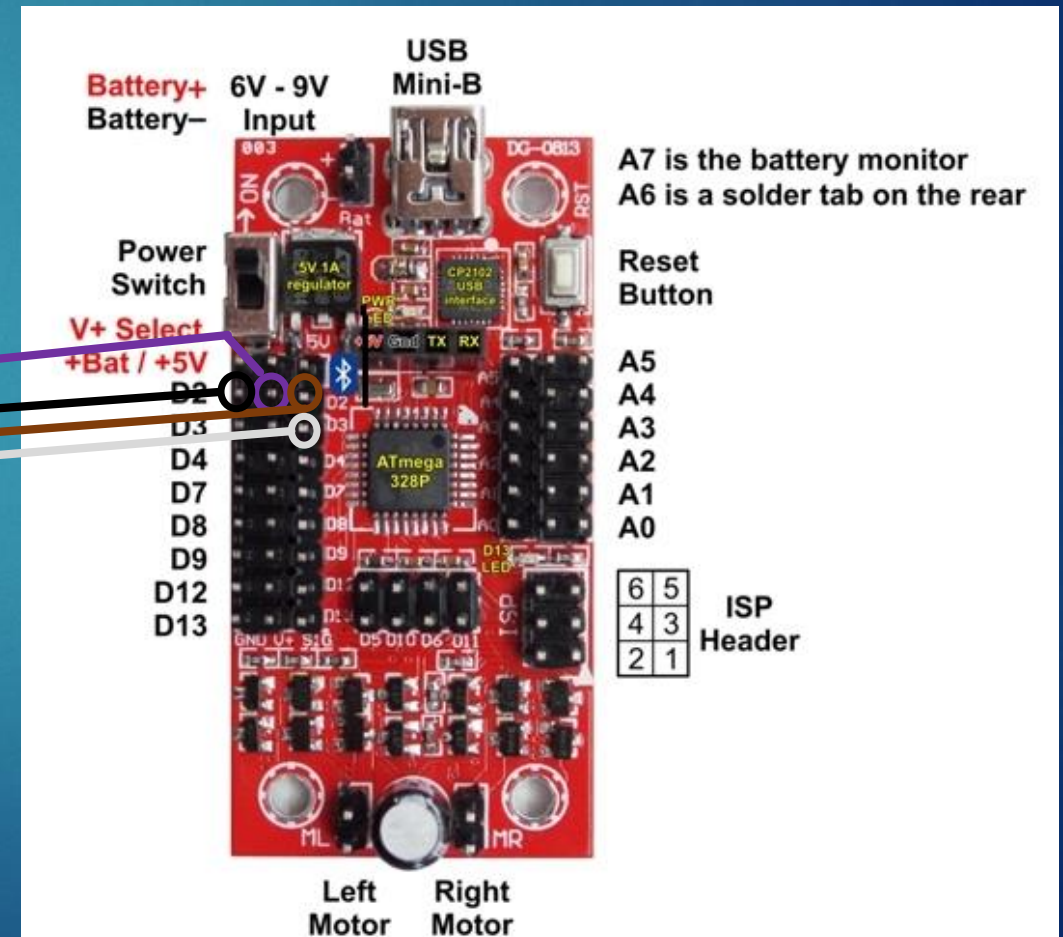
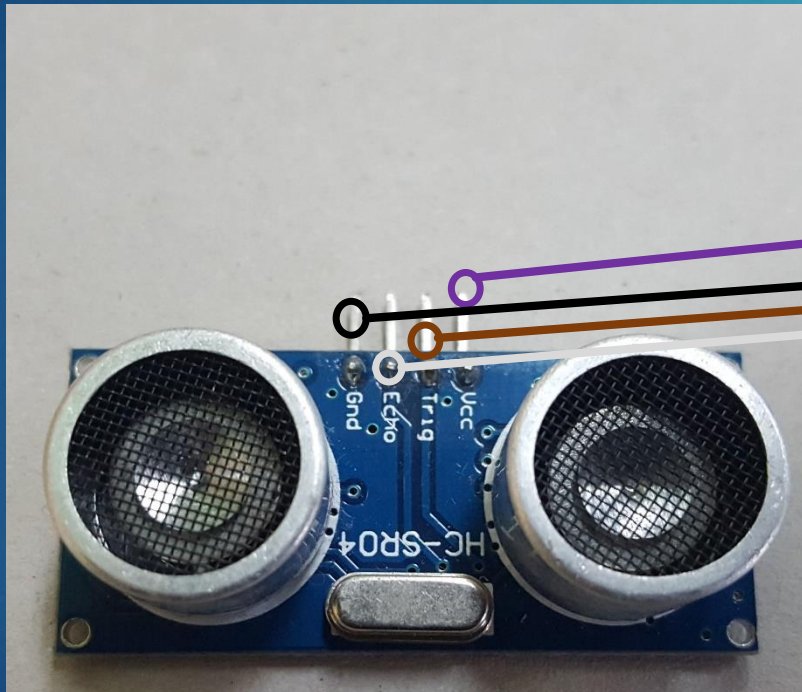
```
35   digitalWrite(9, LOW);
36 }
37 //Función para ir hacia atrás
38 void reversa()
39 {
40   digitalWrite(10, LOW);
41   digitalWrite(11, LOW);
42   analogWrite(5, velocidadDerecha);
43   analogWrite(6, velocidadIzquierda);
44   digitalWrite(12, LOW);
45   digitalWrite(13, HIGH);
46   digitalWrite(9, LOW);
47 }
48 //Función para girar
49 void giro()
50 {
51   digitalWrite(10, HIGH);
52   digitalWrite(11, LOW);
53   analogWrite(5, velocidadDerecha);
54   analogWrite(6, velocidadIzquierda);
55   digitalWrite(12, LOW);
56   digitalWrite(13, LOW);
57   digitalWrite(9, HIGH);
58 }
59 //Función para detener
60 void detener(){
61   velocidadDerecha=0;
62   velocidadIzquierda=0;
63   analogWrite(5, velocidadDerecha);
64   analogWrite(6, velocidadIzquierda);
65   digitalWrite(9, HIGH);
66   digitalWrite(12, LOW);
67   digitalWrite(13, LOW);
68 }
```



Agregaremos la función sonido, que ya habíamos probado en el programa anterior, colocaremos esta función entre la función loop() y la función adelante()

```
Laboratorio8 $
7  pinMode(11,OUTPUT);
8  pinMode(6,OUTPUT);
9  pinMode(12, OUTPUT);
10 pinMode(13, OUTPUT);
11 pinMode(9, OUTPUT);
12 }
13 void loop() {
14   velocidadDerecha=100;
15   velocidadIzquierda=100;
16   adelante();
17   delay(3000);
18   giro();
19   delay(1000);
20   reversa();
21   delay(2000);
22   detener();
23   delay(2000);
24 }
25 void sonido() {
26   tone(4,392,200) ;
27   delay(500) ;
28   tone(4,294,200) ;
29   delay(500) ;
30 }
31 //Función para ir hacia adelante
32 void adelante()
33 {
34   digitalWrite(10, HIGH);
35   digitalWrite(11, HIGH);
36   analogWrite(5, velocidadDerecha);
37   analogWrite(6, velocidadIzquierda);
38   digitalWrite(12, HIGH);
39   digitalWrite(13, LOW);
40   digitalWrite(9, LOW);
41 }
```

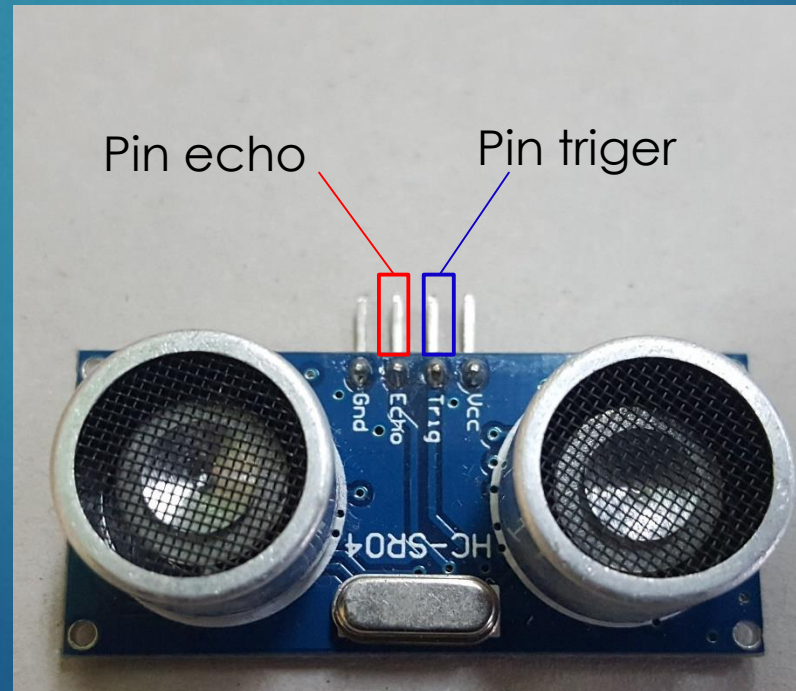

Continuaremos con nuestro robot, configurando la programación del sensor ultrasónico, este sensor ya está debidamente conectado, se encuentra configurado de la siguiente manera:

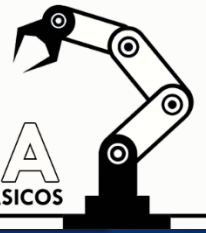




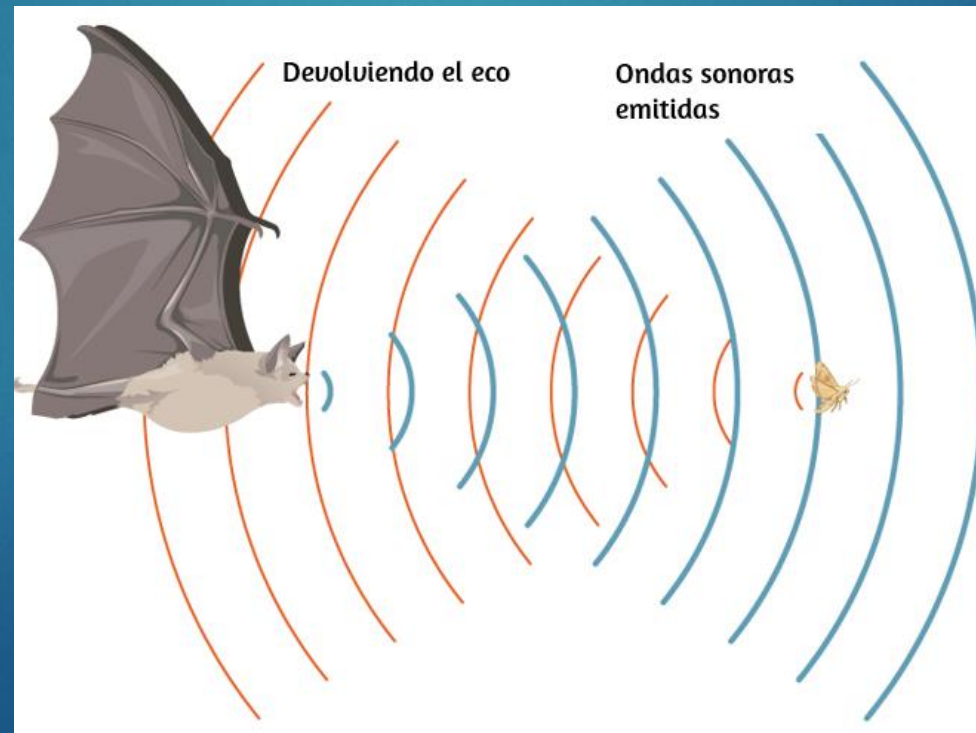
Los pasos para crear la función que puede detectar obstáculos que se encuentren frente al sensor ultrasónico es la siguiente:

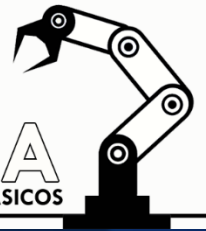
1- En el setup, declararemos los pines que vamos a utilizar con el sensor analógico, usaremos el pin 2 para el pin trigger y el pin 3 para el pin echo



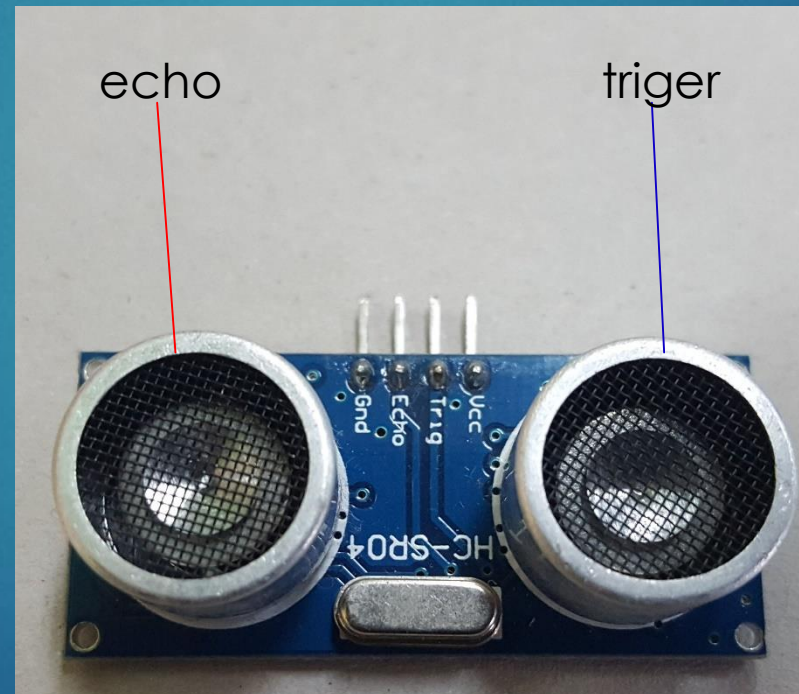


El sensor ultrasónico funciona con el mismo principio que algunas especies de murciélagos, estos animales emiten con su hocico un sonido ultrasónico, es un chillido que no podemos oír los humanos, este sonido se desplaza por el espacio y rebota al chocar con objetos y obstáculos, el murciélago escucha el eco que ocasiona el sonido al rebotar contra los objetos, y así puede saber que objetos lo rodean y como evitarlos.





En el sensor ultrasónico, una de las dos bocinas emite el sonido, es la que se llama “trigger” y la otra bocina “escucha” y detecta el eco, esta se llama “echo”



Agregaremos en la función setup() la declaración de los pines 2 y 3. El pin 2 para el emisor, ósea el pin trigger, y el pin 3 para el receptor: el pin echo.

El pin 2 al ser emisor se declara como salida (OUTPUT)

El pin 3 por ser receptor se declara como entrada (INPUT)

Laboratorio8 §

```
1 int velocidadDerecha=100;
2 int velocidadIzquierda=100;
3
4 void setup() {
5     pinMode(2, OUTPUT);
6     pinMode(3, INPUT);
7     pinMode(10, OUTPUT);
8     pinMode(5, OUTPUT);
9     pinMode(11, OUTPUT);
10    pinMode(6, OUTPUT);
11    pinMode(12, OUTPUT);
12    pinMode(13, OUTPUT);
13    pinMode(9, OUTPUT);
14 }
15 void loop() {
```

En el setup() agregaremos también la instrucción Serial.begin(9600) para activar el monitor serie, y poder monitorear el comportamiento del sensor

Laboratorio8 §

```
1 int velocidadDerecha=100;
2 int velocidadIzquierda=100;
3
4 void setup() {
5     Serial.begin(9600);
6     pinMode(2, OUTPUT);
7     pinMode(3, INPUT);
8     pinMode(10, OUTPUT);
9     pinMode(5, OUTPUT);
10    pinMode(11, OUTPUT);
11    pinMode(6, OUTPUT);
12    pinMode(12, OUTPUT);
13    pinMode(13, OUTPUT);
14    pinMode(9, OUTPUT);
15 }
```


Después de la función `loop()` y antes de la función `sonido()` escribiremos la función para el sensor ultrasónico, es la siguiente:

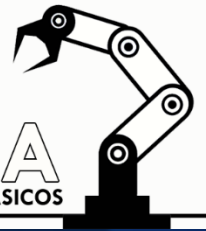
```
16 void loop() {
17   velocidadDerecha=100;
18   velocidadIzquierda=100;
19   adelante();
20   delay(3000);
21   giro();
22   delay(1000);
23   reversa();
24   delay(2000);
25   detener();
26   delay(2000);
27 }
28 int sensorUltrasonico()
29 {
30   long duracion, distancia;
31   digitalWrite(2, LOW);
32   delayMicroseconds(2);
33   digitalWrite(2, HIGH);
34   delayMicroseconds(10);
35   digitalWrite(2, LOW);
36   duracion = pulseIn(3, HIGH, 2000);
37   distancia = duracion / 58;
38   if(distancia == 0){
39     distancia = 999;
40   }
41   return distancia;
42 }
43 void sonido(){
44   tone(4,392,200) ;
45   delay(500) ;
46   tone(4,294,200) ;
47   delay(500) ;
48 }
```



La función realmente es muy sencilla, podemos leer que hace cada una de las líneas de la función en los comentarios siguientes:

```
28 int sensorUltrasonico()
29 {
30     long duracion, distancia; //Se declaran dos variables de tipo long
31     digitalWrite(2, LOW); //El pin 2 controla el trigger, el emisor. Inicia apagado (LOW)
32     delayMicroseconds(2); //por 2 milésimas de segundo
33     digitalWrite(2, HIGH); //Se inicia el trigger que como sabemos es el emisor y está conectado al pin 2
34     delayMicroseconds(10); //Se mantiene encendido por 10 milésimas
35     digitalWrite(2, LOW); //Se apaga el trigger, este comportamiento lo que hace es emitir sonidos en "pulsos" Ejemplo: bit - bit - bit - bit
36     duracion = pulseIn(3, HIGH, 2000); //A la variable duración se le agrega la función pulseIn que mediante el pin 3 (Echo) Registrará los pulsos dentro de esta variable
37     distancia = duracion / 58; //A la variable distancia se le asigna el valor de la división del dato que recoge el pin 3 echo, y lo divide entre 58, se obtiene un número
38     //que representa la distancia en centímetros.
39     if(distancia == 0){ //Este condicional transforma el valor 0 en 999, ya que 0 es un valor que con la lógica que vamos a usar significa que el robot colisionó con un obstáculo.
40         distancia = 999;
41     }
42     return distancia; //Gracias a la instrucción "return" otra variable puede "recoger el valor que se obtiene de la función sensorUltrasonico()
43 }
```

Recordemos que no es necesario escribir los comentarios, estos los escribí a manera de explicación únicamente.



En la función loop() borraremos todas las instrucciones que habíamos escrito:

```
Laboratorio8$  
1 int velocidadDerecha=100;  
2 int velocidadIzquierda=100;  
3  
4 void setup() {  
5     Serial.begin(9600);  
6     pinMode(2, OUTPUT);  
7     pinMode(3, INPUT);  
8     pinMode(10, OUTPUT);  
9     pinMode(5, OUTPUT);  
10    pinMode(11, OUTPUT);  
11    pinMode(6, OUTPUT);  
12    pinMode(12, OUTPUT);  
13    pinMode(13, OUTPUT);  
14    pinMode(9, OUTPUT);  
15 }  
16 void loop() {  
17  
18  
19  
20 }  
21 int sensorUltrasonico()  
22 {  
23     long duracion, distancia;  
24     digitalWrite(2, LOW);  
25     delayMicroseconds(2);  
26     digitalWrite(2, HIGH);  
27     delayMicroseconds(10);  
28     digitalWrite(2, LOW);  
29     duracion = pulseIn(3, HIGH, 2000);  
30     distancia = duracion / 58;  
31     if(distancia == 0){  
32         distancia = 999;  
33     }  
34     return distancia;  
35 }
```


Lo primero que haremos es escribir las instrucciones que nos permitan ver que esta registrando el sensor.

Para lograrlo, agregaremos una variable global en la línea 3, de tipo número entero

```
Laboratorio8 $  
1 int velocidadDerecha=100;  
2 int velocidadIzquierda=100;  
3 int registro;  
4 void setup() {  
5     Serial.begin(9600);  
6     pinMode(2, OUTPUT);  
7     pinMode(3, INPUT);  
8     pinMode(10, OUTPUT);  
9     pinMode(5, OUTPUT);  
10    pinMode(11, OUTPUT);  
11    pinMode(6, OUTPUT);  
12    pinMode(12, OUTPUT);  
13    pinMode(13, OUTPUT);  
14    pinMode(9, OUTPUT);  
15 }  
16 void loop() {  
17     velocidadDerecha=100;  
18  
19  
20 }
```

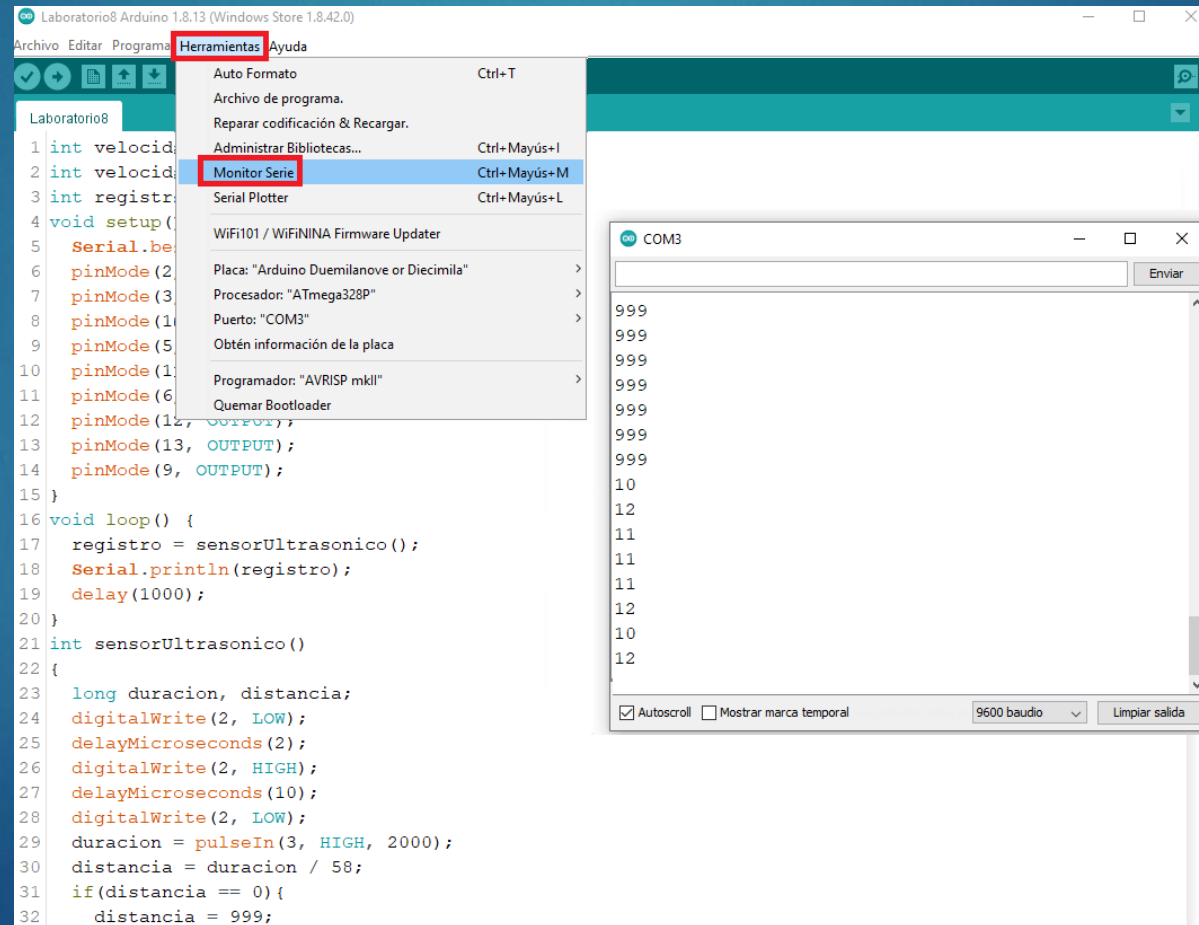
En la función `loop()` agregaremos por el momento 3 instrucciones.

- 1- En la primera instrucción, cargaremos el resultado de la función `ultrasonico()` en la variable llamada `registro`.
- 2- En la segunda instrucción enviaremos el valor de la variable “registro” al monitor serie.
- 3- En la tercera instrucción, mediante la sentencia `delay()` haremos que las dos últimas instrucciones se ejecuten por un segundo, y luego se repetirán de nuevo.

Escriba entonces las siguientes instrucciones en el `loop()`

```
16 void loop() {  
17     registro = sensorUltrasonico(); //Recibe el valor arrojado por el sensor  
18     Serial.println(registro);  
19     delay(1000);  
20 }
```

Podemos subir el programa a la placa, y sin desconectar el cable USB, activaremos el monitor serie, podremos colocar nuestra mano en diferentes distancias frente al sensor ultrasónico, en el monitor serie veremos cuantos centímetros separan nuestra mano del sensor. El sensor puede registrar entre los 3 centímetros, hasta los 20 centímetros aproximadamente.



The screenshot shows the Arduino IDE interface. The 'Herramientas' (Tools) menu is open, and the 'Monitor Serie' option is highlighted. The serial monitor window is also open, showing the output of the program. The code in the background is as follows:

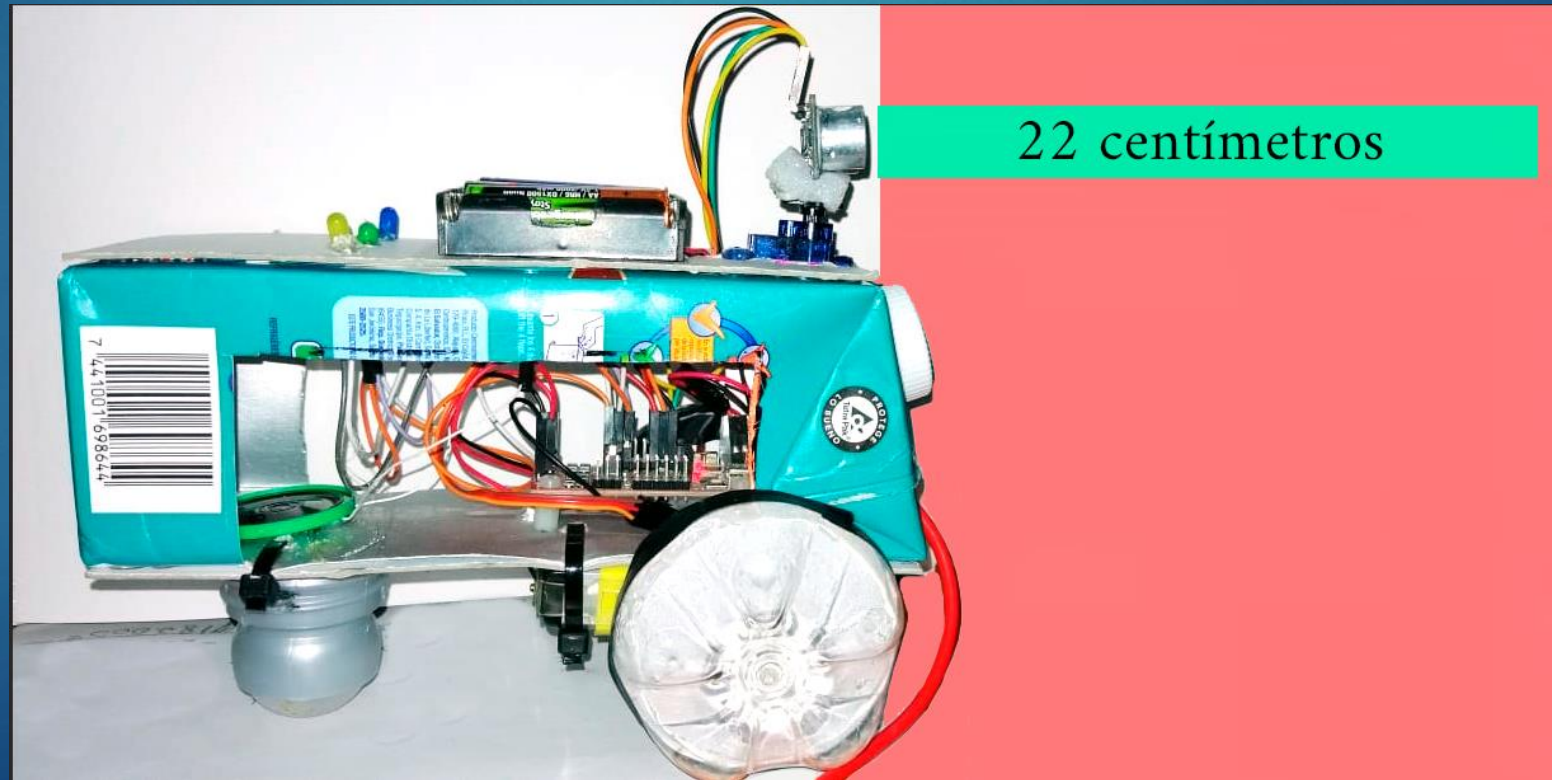
```
1 int velocidad;
2 int velocidad;
3 int registro;
4 void setup() {
5   Serial.begin(9600);
6   pinMode(2, INPUT);
7   pinMode(3, INPUT);
8   pinMode(10, INPUT);
9   pinMode(5, INPUT);
10  pinMode(11, INPUT);
11  pinMode(6, INPUT);
12  pinMode(12, OUTPUT);
13  pinMode(13, OUTPUT);
14  pinMode(9, OUTPUT);
15 }
16 void loop() {
17   registro = sensorUltrasonico();
18   Serial.println(registro);
19   delay(1000);
20 }
21 int sensorUltrasonico()
22 {
23   long duracion, distancia;
24   digitalWrite(2, LOW);
25   delayMicroseconds(2);
26   digitalWrite(2, HIGH);
27   delayMicroseconds(10);
28   digitalWrite(2, LOW);
29   duracion = pulseIn(3, HIGH, 2000);
30   distancia = duracion / 58;
31   if(distancia == 0){
32     distancia = 999;
```

The serial monitor output shows the following values:

```
999
999
999
999
999
999
999
10
12
11
11
11
12
10
12
```


Ahora, por medio del uso de estructuras condicionales si, programaremos nuestro robot para que al avanzar, y usando el sensor ultrasónico, pueda detectar obstáculos que tenga al frente y evitarlos.

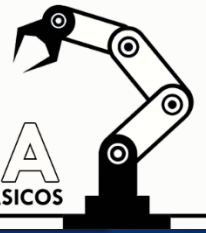
Tenemos que tener en cuenta que el rango que detecta el sensor es muy limitado, y se restringe a un máximo de 22 centímetros y en área del sensor, en la siguiente imagen se representa el área de detección de color verde:



En la función loop() podemos escribir el siguiente programa:

Laboratorio8

```
1 int velocidadDerecha=100;
2 int velocidadIzquierda=100;
3 int registro;
4 void setup() {
5     Serial.begin(9600);
6     pinMode(2, OUTPUT);
7     pinMode(3, INPUT);
8     pinMode(10, OUTPUT);
9     pinMode(5, OUTPUT);
10    pinMode(11, OUTPUT);
11    pinMode(6, OUTPUT);
12    pinMode(12, OUTPUT);
13    pinMode(13, OUTPUT);
14    pinMode(9, OUTPUT);
15 }
16 void loop() {
17     registro = sensorUltrasonico(); //Recibe el valor arrojado por el sensor
18     Serial.println(registro);
19     if (registro < 10)
20     {
21         giro();
22         delay(500);
23     }
24     else//sino
25     {
26         adelante(); //...ejecute el programa caminar hacia adelante
27     }
28 }
29 int sensorUltrasonico()
30 {
31     long duracion, distancia;
32     digitalWrite(2, LOW);
```



Y por el momento agregaremos a la función giro la ejecución de la función giro(), para que suene la alarma cuando el robot detecta algún obstáculo a 10 centímetros o menos.

```
Laboratorio8
51 void adelante()
52 {
53     digitalWrite(10, HIGH);
54     digitalWrite(11, HIGH);
55     analogWrite(5, velocidadDerecha);
56     analogWrite(6, velocidadIzquierda);
57     digitalWrite(12, HIGH);
58     digitalWrite(13, LOW);
59     digitalWrite(9, LOW);
60 }
61 //Función para ir hacia atrás
62 void reversa()
63 {
64     digitalWrite(10, LOW);
65     digitalWrite(11, LOW);
66     analogWrite(5, velocidadDerecha);
67     analogWrite(6, velocidadIzquierda);
68     digitalWrite(12, LOW);
69     digitalWrite(13, HIGH);
70     digitalWrite(9, LOW);
71 }
72 //Función para girar
73 void giro()
74 {
75     sonido();
76     digitalWrite(10, HIGH);
77     digitalWrite(11, LOW);
78     analogWrite(5, velocidadDerecha);
79     analogWrite(6, velocidadIzquierda);
80     digitalWrite(12, LOW);
81     digitalWrite(13, LOW);
82     digitalWrite(9, HIGH);
83 }
```


En las siguientes imágenes pueden visualizar el programa completo conforme al laboratorio 8:

Laboratorio8 §

```
1 int velocidadDerecha=100;
2 int velocidadIzquierda=100;
3 int registro;
4 void setup() {
5     Serial.begin(9600);
6     pinMode(2, OUTPUT);
7     pinMode(3, INPUT);
8     pinMode(10, OUTPUT);
9     pinMode(5, OUTPUT);
10    pinMode(11, OUTPUT);
11    pinMode(6, OUTPUT);
12    pinMode(12, OUTPUT);
13    pinMode(13, OUTPUT);
14    pinMode(9, OUTPUT);
15 }
16 void loop() {
17     registro = sensorUltrasonico();
18     Serial.println(registro);
19     if (registro < 10)
20     {
21         giro();
22         delay(500);
23     }
24     else//sino
25     {
26         adelante();
27     }
28 }
29 int sensorUltrasonico()
30 {
31     long duracion, distancia;
32     digitalWrite(2, LOW);
33     delayMicroseconds(2);
34     digitalWrite(2, HIGH);
```

```
35     delayMicroseconds(10);
36     digitalWrite(2, LOW);
37     duracion = pulseIn(3, HIGH, 2000);
38     distancia = duracion / 58;
39     if(distancia == 0){
40         distancia = 999;
41     }
42     return distancia;
43 }
44 void sonido(){
45     tone(4,392,200) ;
46     delay(500) ;
47     tone(4,294,200) ;
48     delay(500) ;
49 }
50 //Función para ir hacia adelante
51 void adelante()
52 {
53     digitalWrite(10, HIGH);
54     digitalWrite(11, HIGH);
55     analogWrite(5, velocidadDerecha);
56     analogWrite(6, velocidadIzquierda);
57     digitalWrite(12, HIGH);
58     digitalWrite(13, LOW);
59     digitalWrite(9, LOW);
60 }
61 //Función para ir hacia atrás
62 void reversa()
63 {
64     digitalWrite(10, LOW);
65     digitalWrite(11, LOW);
66     analogWrite(5, velocidadDerecha);
67     analogWrite(6, velocidadIzquierda);
68     digitalWrite(12, LOW);
```

```
69     digitalWrite(13, HIGH);
70     digitalWrite(9, LOW);
71 }
72 //Función para girar
73 void giro()
74 {
75     sonido();
76     digitalWrite(10, HIGH);
77     digitalWrite(11, LOW);
78     analogWrite(5, velocidadDerecha);
79     analogWrite(6, velocidadIzquierda);
80     digitalWrite(12, LOW);
81     digitalWrite(13, LOW);
82     digitalWrite(9, HIGH);
83 }
84 //Función para detener
85 void detener(){
86     velocidadDerecha=0;
87     velocidadIzquierda=0;
88     analogWrite(5, velocidadDerecha);
89     analogWrite(6, velocidadIzquierda);
90     digitalWrite(9, HIGH);
91     digitalWrite(12, LOW);
92     digitalWrite(13, LOW);
93 }
```



Evidencias de aprendizaje

- 1- Tome un vídeo corto de su robot ejecutando el programa laboratorio 8, recuerde colocar obstáculos que estén a la altura del sensor, para que el robot pueda detectarlos.
- 2- Coloque en una carpeta el vídeo, junto con la carpeta llamada Laboratorio8 con el programa de Arduino Laboratorio8
- 3- Comprima la carpeta con el vídeo y el programa en zip
- 4- Entregue por medio del enlace del Laboratorio 8

Laboratorio práctico 8 Valor 5%

Nombre del estudiante:

Rubro	Valor	Ptos. Obtenidos
Entrega el vídeo y el programa en la carpeta zip solicitados en la fecha indicada.	5pts	
Utiliza el buzón correspondiente para subir su tarea.	2 pts	
El robot emite un sonido al girar	5	
El Programa cumple lo esperado: detecta objetos cerca del sensor y los evita.	8	
Total	20	

Comentarios: