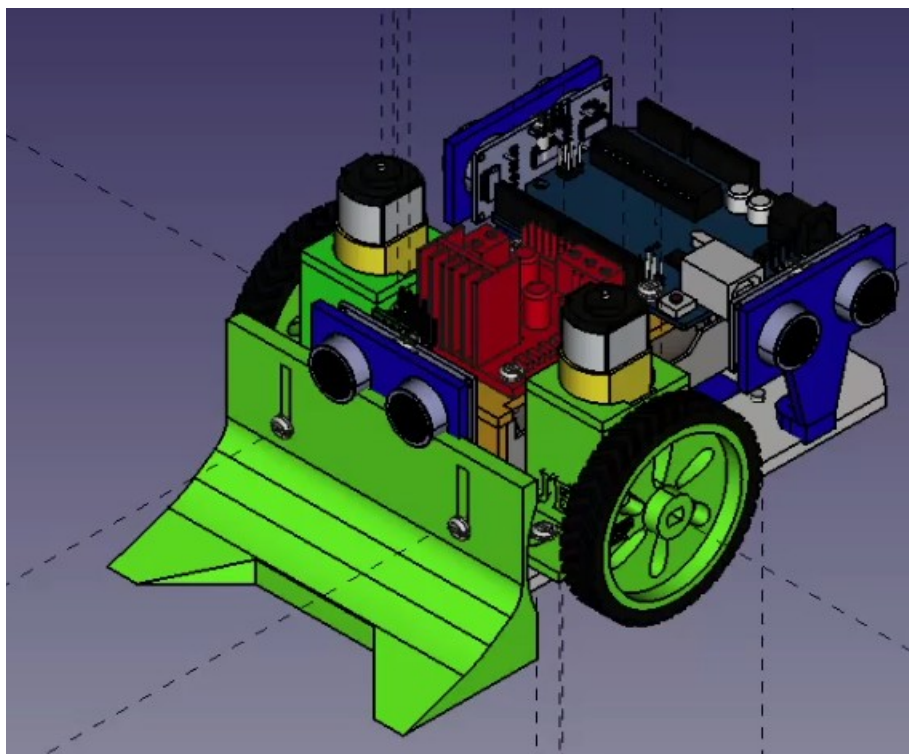

Manual de robot laberinto con arduino



Autor: Profesor Palacios José Emanuel

Año: 2022

Institución: Escuela de Educación Técnico Profesional N° 602 "Gral San Martín"

www.eetp602.edu.ar



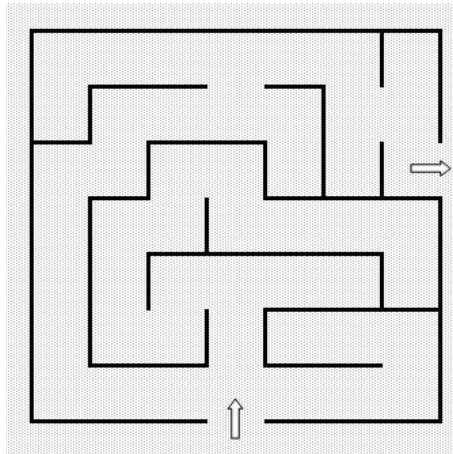
Manual de Robot Laberinto con Arduino

Contenido

Introducción: Robotlaberinto.....	3
Materiales.....	3
Interruptor on/off y alimentación.....	4
Sensor ultrasonico hc-sr04.....	5
Módulo ultrasónico HC-SR04 funcionando.....	5
Control de Motores.....	13
Funcionamiento del puente H.....	15
Programación para controlar el puente h.....	17
Lógica de movimiento	19
Montaje de los componentes.....	23
Lógica para la programación de la aplicación final.....	32
Diagrama de conexión final.....	38
 Bibliografía:.....	 40

Introducción: Robotlaberinto.

En este manual desarrollaremos un robotlaberinto funcional básico, que se espera sufra modificaciones por quien lee este manual. Existen diferentes tipos de laberintos pero si esta es tu primera vez programando un robot que resuelve laberintos lo mejor será empezar por algo "sencillo". Nuestro laberinto tendrá su entrada y salida en el perímetro y no contendrá islas ni lazos cerrados. Puede haber divergencia de caminos y callejones sin salida. El siguiente ejemplo de laberintos podría servir. **Nuestro robot debe ser capaz de resolverlo.**



Materiales.

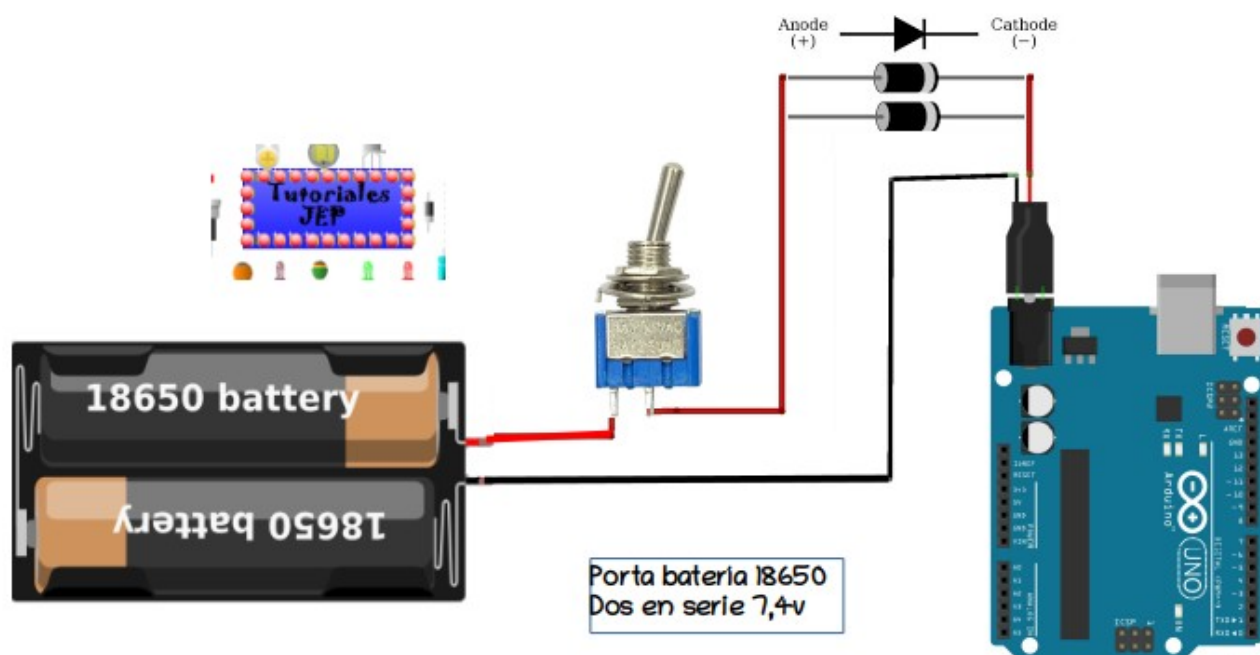
- 1x Arduino uno
- 1x Chasis que cumpla con las normas de la categoría (se proporcionaran planos de ejemplo).
- 2x motoreductor arduino
- Rueda para motoreductor arduino (se proporciona plano para su diseño 3d).
- shield doble puente h l298n.
- Porta baterías 18650 x2.
- Baterías recargables tipo 18650 3.7v 5000 Mah
- Plug de alimentación DC 2.1mm
- 30 tornillos cabeza tanque de 1/8 de 10mm con tuerca.
- 2 tornillos cabeza tanque de 1/8 de 30mm con tuerca (para la pala).
- 2 tornillo m2 X 15mm(diametro2mm) para sujetar ruedas.
- 20 conector dupont macho hembra 10cm
- 20 conector dupont macho macho10cm
- Cable 1mm rojo y negro, 50cm.
- Tecla on/Off.
- 3 sensor ultrasonico hc-sr04
- 4 Diodo rectificador 1n4007
- Cinta aisladora o termocontraible (de 3 a 1mm).
- Estaño .
- Piezas del diseño de chasis (archivos stl para descargar <https://youtu.be/j2guSepUVq8>)



Manual de Robot Laberinto con Arduino

Interruptor on/off y alimentación

Es conveniente agregar un interruptor para la alimentación del circuito, en serie a un diodo rectificador 1N4007 (para proteger el circuito en caso de conectar las baterías con la polaridad incorrecta), de acuerdo al consumo es la cantidad de diodos 1N4007 que conectaremos ya que cada diodo soporta 1A, en este caso conectaremos 2 diodos en paralelo para que soporten 2A, de necesitar que soporte mas amperaje incrementar la cantidad de diodos o colocar uno que soporte dicho amperaje.



Sensor ultrasonico hc-sr04

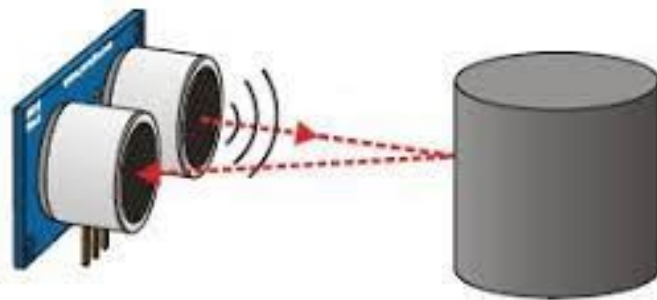
El Sensor de Distancia Ultrasónico HC-SR04 es una solución popular y de bajo costo para la función de medición de distancia sin contacto. Es capaz de medir distancias de 2 cm a 400 cm con una precisión de aproximadamente 3 mm. Este módulo incluye transmisor ultrasónico, receptor ultrasónico y su circuito de control.

El módulo HC-SR04 tiene 4 pines:

- **VCC** - 5V, + ive de la fuente de alimentación
- **TRIG** - Pin del gatillo
- **ECHO** - Pin Echo
- **GND** - - de la fuente de alimentación

Los pines TRIG y ECHO se pueden usar para conectar este módulo con una unidad de microcontrolador. Estos son los pines de salida de entrada TTL (0 - 5V).

Módulo ultrasónico HC-SR04 funcionando

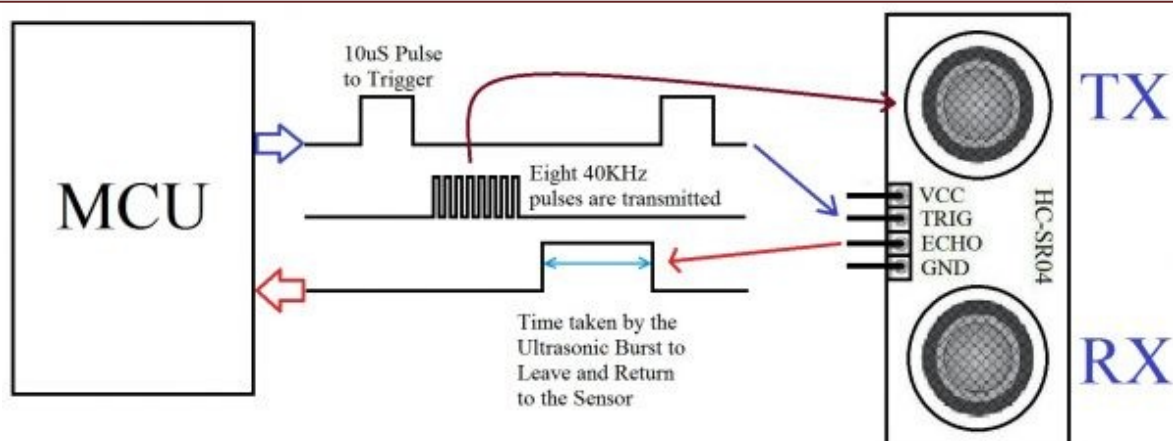


$$\text{Tiempo} = 2 * (\text{Distancia} / \text{Velocidad})$$

$$\text{Distancia} = \text{Tiempo} \cdot \text{Velocidad} / 2$$

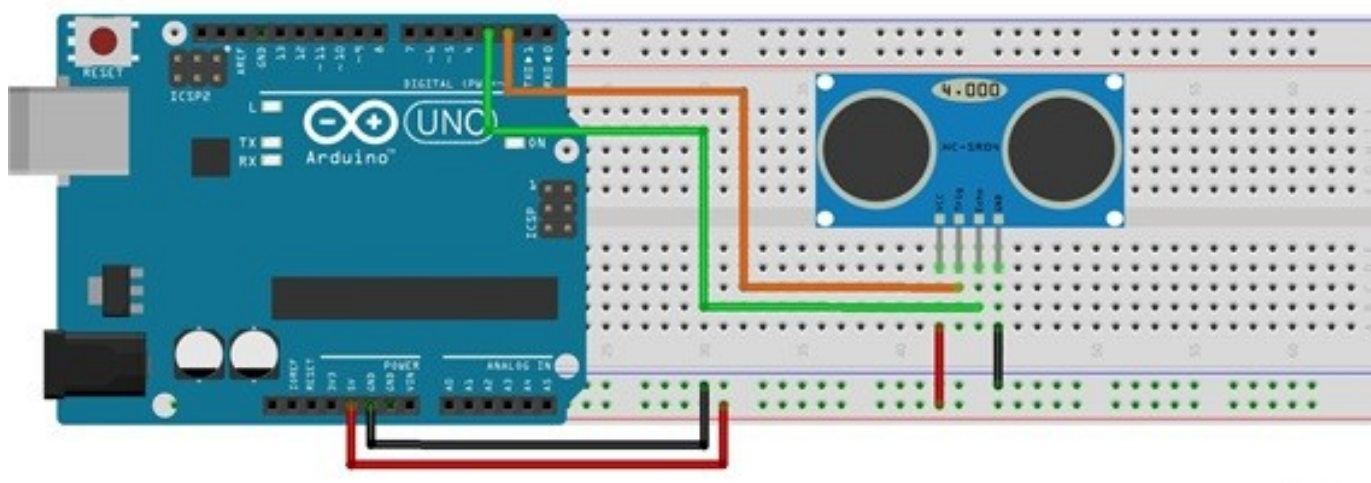
- Proporcione la señal TRIGGER, al menos 10 μ S de pulso de nivel alto (5V).
- El módulo transmitirá automáticamente ocho ráfagas ultrasónicas de 40 kHz.
- Si hay un obstáculo en frente del módulo, reflejará la explosión ultrasónica.
- Si la señal está de regreso, la salida ECHO del sensor estará en estado ALTO (5 V) durante un tiempo de envío y recepción de la explosión ultrasónica. El ancho del pulso varía de aproximadamente 150 μ S a 25mS y, si no se detecta ningún obstáculo, el ancho del pulso del eco será de aproximadamente 38 ms.

Manual de Robot Laberinto con Arduino



Empezamos insertando el sensor ultrasónico en un protoboard y con cables hacemos las siguientes conexiones:

- Trigger del sensor al pin 2 del arduino
- Echo del sensor al pin 3 del arduino

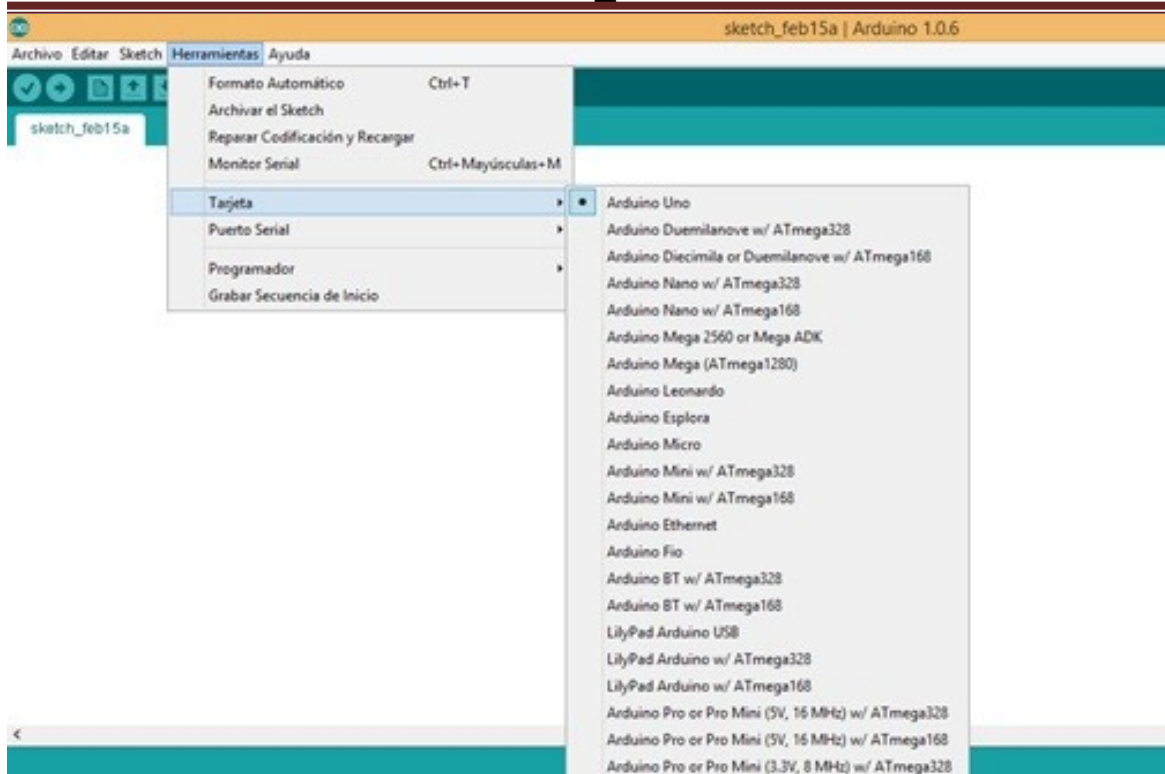


Todas las conexiones se realizan con el Arduino Apagado (desconectado de la PC o de cualquier fuente externa):

Ahora abrimos el entorno de programación de Arduino, en Herramientas -> Tarjeta, y seleccionamos el modelo de placa Arduino que estemos utilizando. Nosotros seleccionaremos Arduino Uno.



Manual de Robot Laberinto con Arduino



Una vez configurado el IDE, empezamos a programar nuestro sketch, explicaremos paso a paso el código para sea más comprensible:

Primero configuramos los pines y la comunicación serial a 9800 baudios.

```
const int Trigger = 2; // enmascaro al pin 2 con el nombre de Trigger del sensor
const int Echo = 3; // enmascaro al pin 3 con el nombre de Echo del sensor

void setup() {
  Serial.begin(9600); // inicializamos la comunicación
  pinMode(Trigger, OUTPUT); // pin como salida
  pinMode(Echo, INPUT); // pin como entrada
  digitalWrite(Trigger, LOW); // Inicializamos el pin con 0
}
```

Ahora en el bucle void loop() empezamos enviando un pulso de 10us al Trigger del sensor.

```
digitalWrite(Trigger, HIGH); // ponemos el pin trigger en 5v
delayMicroseconds(10); // Esperamos 10us
digitalWrite(Trigger, LOW); // ponemos el pin trigger en 0v
// con esto Enviamos un pulso de 10us
```

Seguidamente recibimos el pulso de respuesta del sensor por el pin Echo, para medir el pulso usamos la función `pulseIn(pin, value)`

Manual de Robot Laberinto con Arduino

```
t=pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
```

La variable t , tiene el tiempo que dura en llegar el eco del ultrasonido, el siguiente paso es calcular la distancia entre el sensor ultrasónico y el objeto.

Partimos de la siguiente formula:

$$Velocidad = \frac{distancia\ recorrida}{tiempo}$$

Donde *Velocidad* es la velocidad del sonido 340m/s, pero usaremos las unidades en cm/us pues trabajaremos en centímetros y microsegundos, **tiempo** es el tiempo que demora en llegar el ultrasonido al objeto y regresar al sensor, y la *distancia recorrida* es dos veces la distancia hacia el objeto, reemplazando en la formula tenemos:

$$\frac{340m}{s} \times \frac{1s}{1000000us} \times \frac{100cm}{1m} = \frac{2d}{t}$$

$$d(cm) = \frac{t(us)}{59}$$

Finalmente enviamos serialmente el valor de la distancia y terminamos poniendo una pausa de 100ms, que es superior a los 60ms recomendado por los datos técnicos del sensor



Manual de Robot Laberinto con Arduino

A continuación se muestra el código completo del programa para testear un (1) sensor ultrasonico.

Prueba de un (1) sensor ultrasonico HC-SR04

```

/*****
/* Autor: Palacios José Emanuel docente de la EETPN°602          */
/* Programa:Prueba de un(1) sensor  ultrasonico  HC-SR04          */
/* Año:2022                */
*****/
const int Trigger = 2;    // enmascaro al pin 2 con el nombre de Trigger del sensor
const int Echo = 3;      // enmascaro al pin 3 con el nombre de Echo del sensor
void setup() {
  Serial.begin(9600); //iniciaizamos la comunicación
  pinMode(Trigger, OUTPUT); //pin como salida
  pinMode(Echo, INPUT); //pin como entrada
  digitalWrite(Trigger, LOW); //Inicializamos el pin con 0
}

void loop()
{
  // VARIABLES PARA USAR CON EL SENSOR ULTRASONICO_1*****
  long t; //tiempo que demora en llegar el eco
  long d; //distancia en centimetros

  // ENVIO PULSO PARA EL SENSOR ULTRASONICO _ 1
  digitalWrite(Trigger, HIGH); // ponemos el pin trigger en 5v
  delayMicroseconds(10); //Esperamos 10us
  digitalWrite(Trigger, LOW); // ponemos el pin trigger en 0v
  //con esto Enviamos un pulso de 10us PARA EL SENSOR ULTRASONICO_1

  t=pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
  d=t/59; //calculamos la distancia en cm

  Serial.print("Distancia: ");
  Serial.print(d); //Enviamos serialmente el valor de la distancia
  Serial.print("cm");
  Serial.println();
  delay(100); //Hacemos una pausa de 100ms
}

```

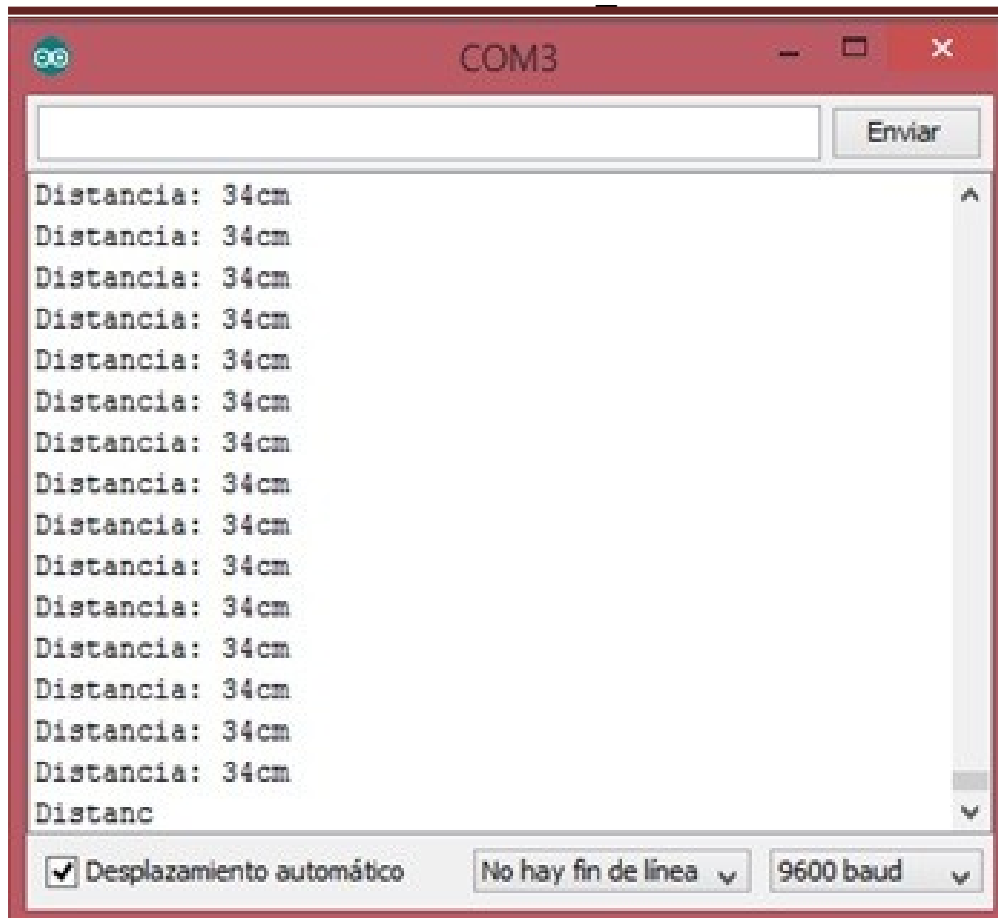
Conecte el Arduino Uno y cargue el programa.

Después de esto el Arduino y sensor ya deben estar trabajando, para poder visualizar los datos vaya a herramientas y habrá el monitor serial.

En el monitor serial le aparecerán los valores de la distancia que censa el HC-SR04, ponga un objeto al frente y varíe su distancia respecto al sensor y verifique que la distancia mostrada en el monitor serial sea la correcta.



Manual de Robot Laberinto con Arduino





Manual de Robot Laberinto con Arduino

A continuación se muestra el código completo del programa para testear dos (2) sensores ultrasonicos.

Prueba de dos (2) sensores ultrasonicos HC-SR04

```

/*****
/* Autor: Palacios José Emanuel docente de la EETPN°602          */
/* Programa:Prueba de dos(2) sensores ultrasonicos HC-SR04      */
/* Año:2022                                                       */
*****/
const int Trigger = 2;    // enmascaro al pin 2 con el nombre de Trigger del sensor 1
const int Echo = 3;      // enmascaro al pin 3 con el nombre de Echo del sensor 1

const int Trigger2 = 4;   // enmascaro al pin 4 con el nombre de Trigger2 del sensor 2
const int Echo2 = 5;     // enmascaro al pin 5 con el nombre de Echo2 del sensor 2

void setup() {
  Serial.begin(9600); //iniciailzamos la comunicación
  pinMode(Trigger, OUTPUT); //pin como salida
  pinMode(Echo, INPUT); //pin como entrada
  digitalWrite(Trigger, LOW); //Inicializamos el pin con 0
  pinMode(Trigger2, OUTPUT); //pin como salida
  pinMode(Echo2, INPUT); //pin como entrada
  digitalWrite(Trigger2, LOW); //Inicializamos el pin con 0
}

void loop()
{
  // VARIABLES PARA USAR CON EL SENSOR ULTRASONICO_1*****
  long t; //tiempo que demora en llegar el eco
  long d; //distancia en centimetros

  // ENVIO PULSO PARA EL SENSOR ULTRASONICO _ 1
  digitalWrite(Trigger, HIGH); // ponemos el pin trigger en 5v
  delayMicroseconds(10); //Esperamos 10us
  digitalWrite(Trigger, LOW); // ponemos el pin trigger en 0v
  //con esto Enviamos un pulso de 10us PARA EL SENSOR ULTRASONICO_1

  t=pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
  d=t/59; //calculamos la distancia en cm

  Serial.print("Distancia: ");
  Serial.print(d); //Enviamos serialmente el valor de la distancia
  Serial.print("cm");
  Serial.println();
  delay(100); //Hacemos una pausa de 100ms

  // VARIABLES PARA USAR CON EL SENSOR ULTRASONICO_2*****
  long t2; //tiempo que demora en llegar el eco
  long d2; //distancia en centimetros

  // ENVIO PULSO PARA EL SENSOR ULTRASONICO _ 2
  digitalWrite(Trigger2, HIGH); // ponemos el pin trigger en 5v
  delayMicroseconds(10); //Esperamos 10us
  digitalWrite(Trigger2, LOW); // ponemos el pin trigger en 0v
  //con esto Enviamos un pulso de 10us PARA EL SENSOR ULTRASONICO_2

  t2=pulseIn(Echo2, HIGH); //obtenemos el ancho del pulso
  d2=t2/59; //calculamos la distancia en cm

```



Manual de Robot Laberinto con Arduino

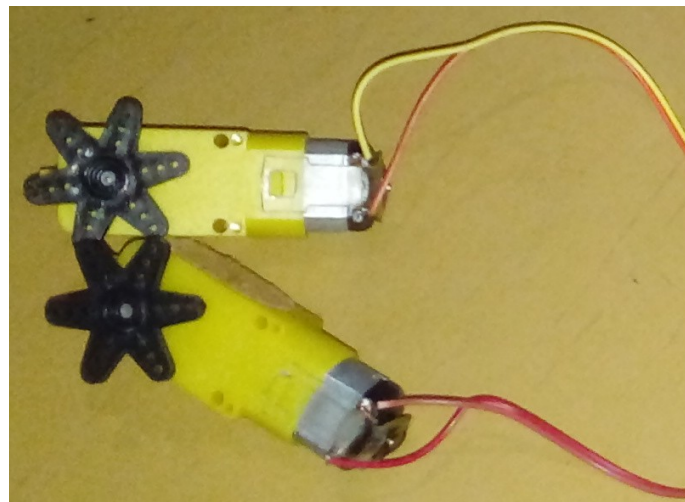
```
Serial.print("Distancia2: ");  
Serial.print(d2);           //Enviamos serialmente el valor de la distancia 2  
Serial.print("cm");  
Serial.println();  
delay(100);                 //Hacemos una pausa de 100ms  
}
```

Control de Motores

Nos queda por probar el movimiento de los motores, estos funcionan con un voltaje de corriente directa, para controlar el giro y la velocidad se usará un driver de motores.

Primero soldaremos los cables a los motores.

Se deben soldar los cables de manera perpendicular al motor de modo que la longitud del motor sea lo más corta posible (ahora que se ha incluido el cable).



Una vez soldados los motores debemos conectarlos al driver.

Shiel doble puente h l298n.





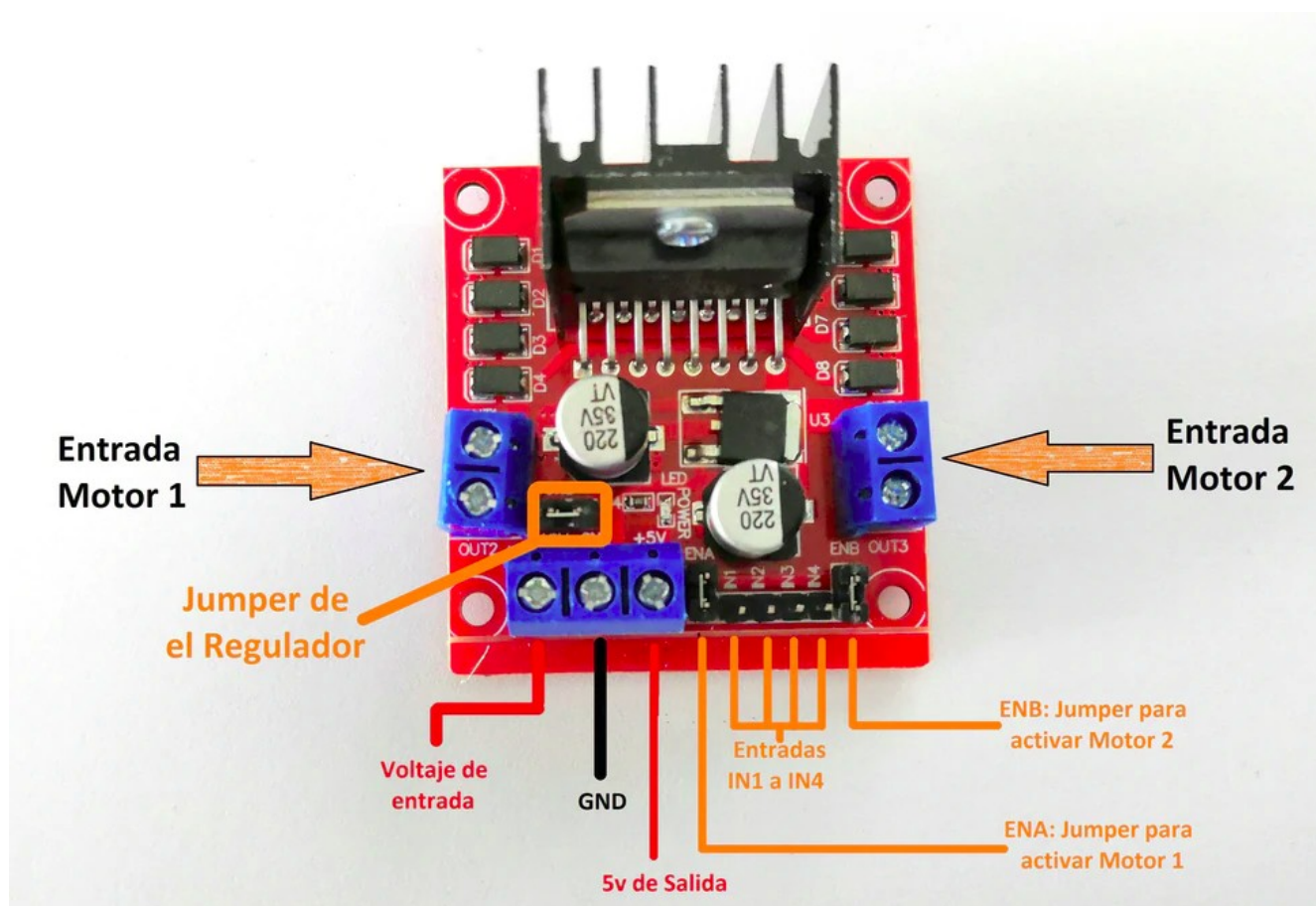
Manual de Robot Laberinto con Arduino

Este módulo es el complemento ideal para proyectos de robótica y Router CNC. Permite controlar hasta 2 motores de corriente continua o un motor paso a paso bipolar. También permite controlar un motor paso a paso unipolar configurado como bipolar de forma muy sencilla y eficaz.

Características

- Voltaje de alimentación, mínimo de 5 V. Posee dos entradas, una de 5V para controlar la parte lógica y otra para alimentar las salidas al motor, que pueden ser de 5V o más.
- La tarjeta tiene la opción de habilitar un regulador LM7805 integrado en ella para alimentar la parte lógica con lo que se puede alimentar la tarjeta con 12V por ejemplo.
- Corriente máxima 2 Amperios.
- Posee 6 entradas de control (ver tabla de control)
- Admite entradas de señal PWM para el control de velocidad.
- Dimensiones: 43 mm x 23,9 mm x 43 mm.
- Salidas: para 2 motores de DC o para un motor bipolar paso a paso.

Funcionamiento del puente H



El voltaje de entrada puede ser desde 3v hasta 35v, aun lado de la entrada de voltaje contamos con el GND

La tercera es la salida lógica de 5v la cual se utiliza de la siguiente manera:

- Si el jumper se encuentra puesto se activará el regulador de tensión y tendremos una salida de 5V, que podremos usar para darle potencia a otro componente.
- Si quitamos el jumper desactivaremos el regulador, necesitaríamos alimentar el módulo, Tendríamos que ingresar 5v por la salida lógica para que el modulo funcione.
- **¡PRECAUCION!** Si ingresamos 5v por la parte lógica con el jumper puesto podríamos causar daño a la tarjeta.
- El regulador solo nos funciona si ingresamos un voltaje desde 5v hasta 12v, Si queremos ingresar un voltaje mayor tendríamos que quitar el jumper y alimentar el modulo con 5v con una fuente externa.

Las salidas de los motores 1 y 2 nos dan energía para mover los motores. Hay que tomar en cuenta las polaridades de los motores para que queden girando hacia adelante. En caso de que un motor este girando al lado contrario solo es necesario invertir los cables.

Los dos pines IN1 e IN2 sirven para controlar el sentido de giro en el motor 1, y los dos pines IN3 e IN4 el sentido de giro del motor 2.



Manual de Robot Laberinto con Arduino

Funcionan de la siguiente manera:

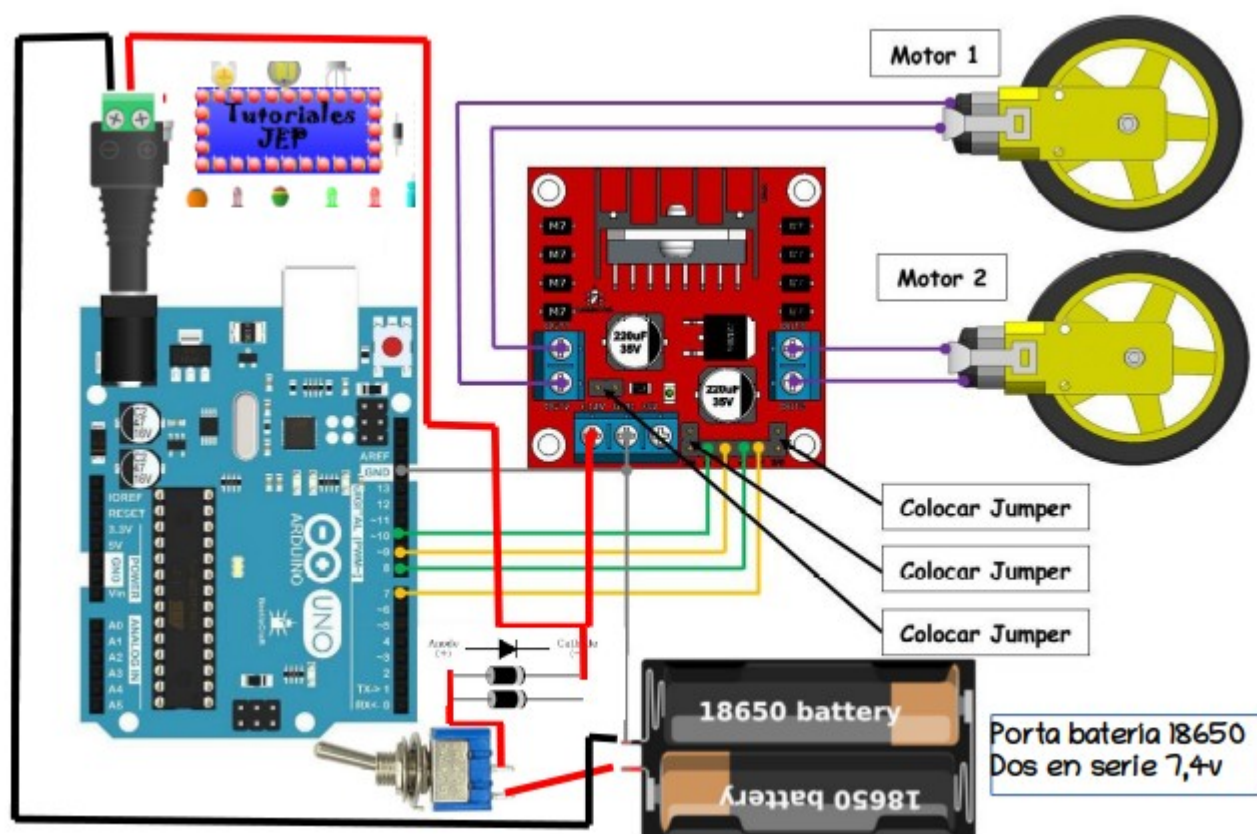
Si IN1 le llegan 5v y al IN2 0v, el motor 1 gira en un sentido, y si IN1 le llegan 0v y a IN2 5v lo hace en el sentido contrario.

Sería lo mismo con los pines IN3 e IN4 para el motor 2.

Para controlar la velocidad a la que van a girar los motores tenemos que retirar los jumpers en ENA y ENB. Los tenemos que conectar a dos salidas PWM de nuestro Arduino con el cual enviaremos una señal con un valor entre 0 y 255 para que controle la velocidad de giro. Si no quitamos los jumpers, los motores estarán girando a la misma velocidad siempre(que es el caso que se explica en este tutorial)

La siguiente imagen muestra las conexiones a realizar, para los pines ENA y ENB se dejan con los jumpers pero si quieren controlar la velocidad podemos utilizar cualquier conexión PWM en el arduino quitando los jumpers.

Conexiones



Programación para controlar el puente h

A continuación tenemos un código de ejemplo para controlar los motores:

```
/* **** */
/* Autor: Palacios José Emanuel docente de la EETPN°602 */
/* Programa:Prueba motores */
/* Año:2022 */
/* **** */

// Conexiones puente h
//motor 1
int IN1=10; // enmascaro al pin 10 con el nombre de IN1
int IN2=9; // enmascaro al pin 9 con el nombre de IN2

//motor 2
int IN3=8; // enmascaro al pin 8 con el nombre de IN3
int IN4=7; // enmascaro al pin 7 con el nombre de IN4

// ***** Inicialización *****
void setup() {
    pinMode(IN1, OUTPUT); // configuro el pin donde conecte IN1 como salida
    pinMode(IN2, OUTPUT); // configuro el pin donde conecte IN2 como salida
    pinMode(IN3, OUTPUT); // configuro el pin donde conecte IN3 como salida
    pinMode(IN4, OUTPUT); // configuro el pin donde conecte IN4 como salida
}

//***** programa principal *****
void loop() {

//giro motor 1 sentido horario
digitalWrite(IN1,HIGH); //envió 5V al pin donde conecte IN1
digitalWrite(IN2,LOW); //envió 0V al pin donde conecte IN2
delay(1000); // Temporiza un segundo (1s = 1000ms)

//giro motor 1 sentido anti horario
digitalWrite(IN1,LOW); //envió 0V al pin donde conecte IN1
digitalWrite(IN2,HIGH); //envió 5V al pin donde conecte IN2
delay(1000); // Temporiza un segundo (1s = 1000ms)

//motor 1 DETENIDO
digitalWrite(IN1,LOW); //envió 0V al pin donde conecte IN1
digitalWrite(IN2,LOW); //envió 0V al pin donde conecte IN2
delay(1000); // Temporiza un segundo (1s = 1000ms)

//giro motor 2 sentido horario
digitalWrite(IN3,HIGH); //envió 5V al pin donde conecte IN3
digitalWrite(IN4,LOW); //envió 0V al pin donde conecte IN4
delay(1000); // Temporiza un segundo (1s = 1000ms)

//giro motor 2 sentido antihorario
digitalWrite(IN3,LOW); //envió 0V al pin donde conecte IN3
digitalWrite(IN4,HIGH); //envió 5V al pin donde conecte IN4
delay(1000); // Temporiza un segundo (1s = 1000ms)

//motor 2 DETENIDO
digitalWrite(IN3,LOW); //envió 0V al pin donde conecte IN3
```



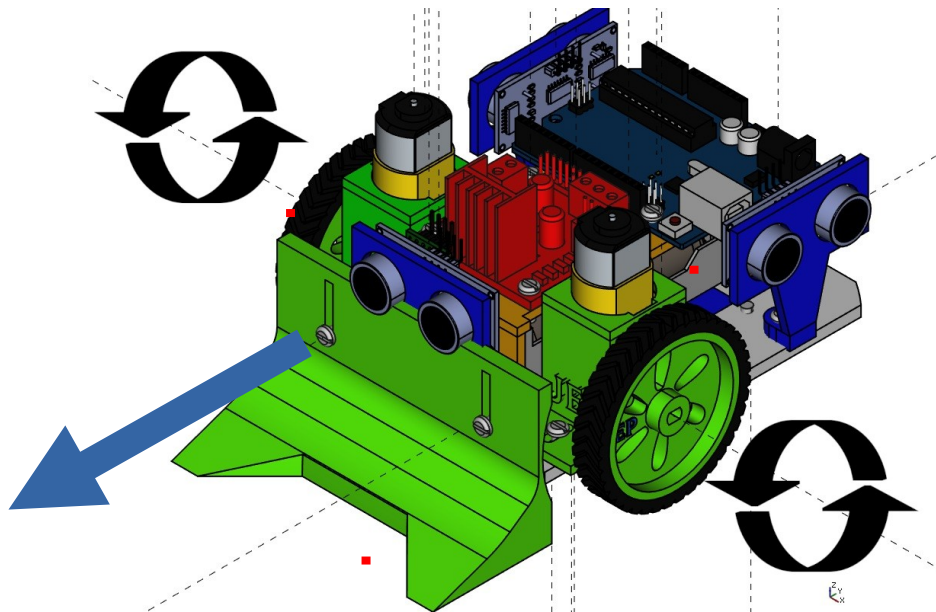
Manual de Robot Laberinto con Arduino

```
digitalWrite(IN4,LOW); //envío 0V al pin donde conecte IN4  
delay(1000); // Temporiza un segundo (1s = 1000ms)  
}
```

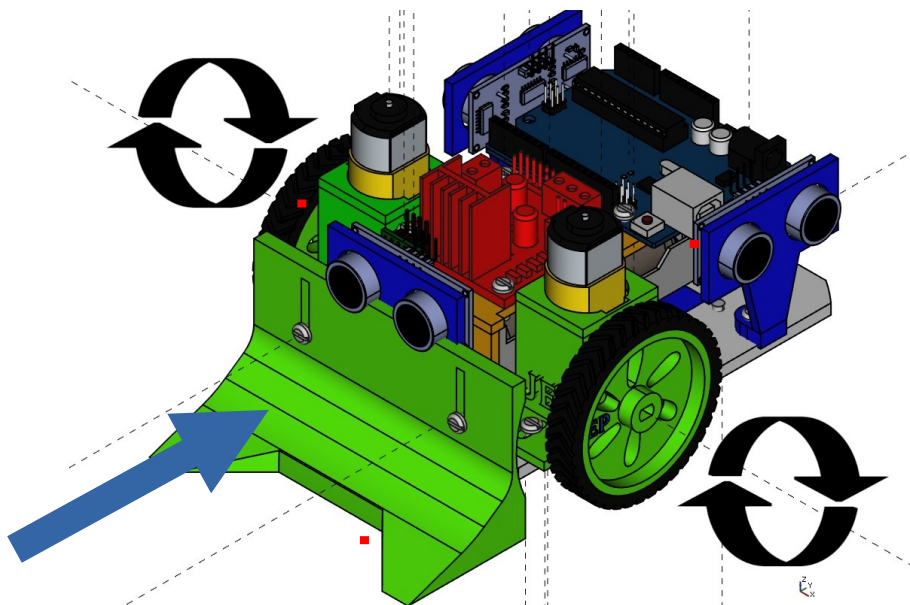

Lógica de movimiento

La estructura que hemos elegido para nuestro robot nos permitirá realizar movimientos hacia Adelante, hacia Atrás, giros a la Derecha, ala Izquierda y sobre sí mismo.

Movimiento hacia delante: Se hacen girar los dos motores en la misma dirección (sentido horario), esto provoca un movimiento recto hacia delante de la estructura.



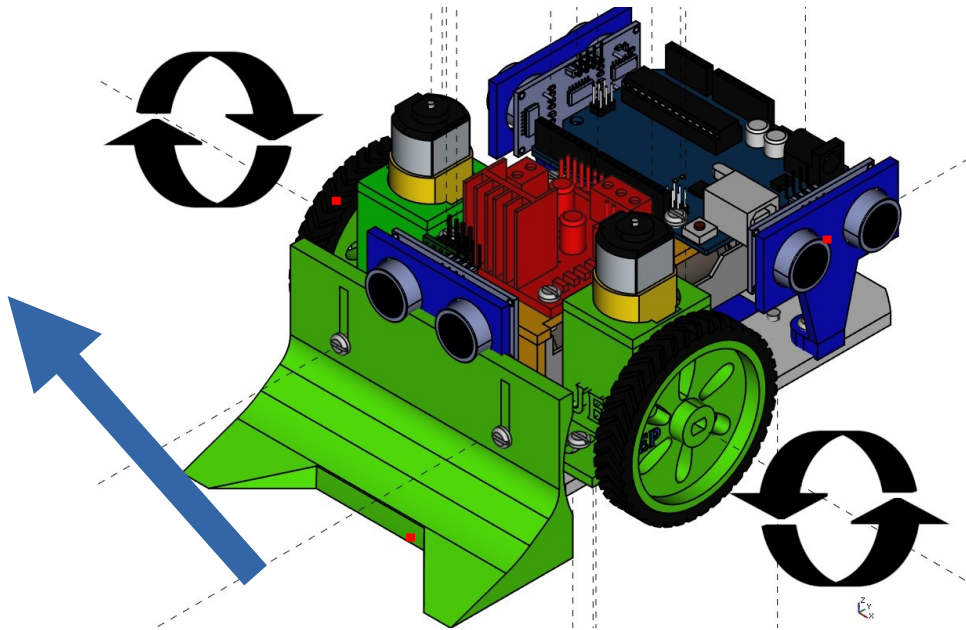
Movimiento hacia atrás: Se hacen girar los dos motores en la misma dirección (antihorario), esto provoca un movimiento recto hacia atrás de la estructura.



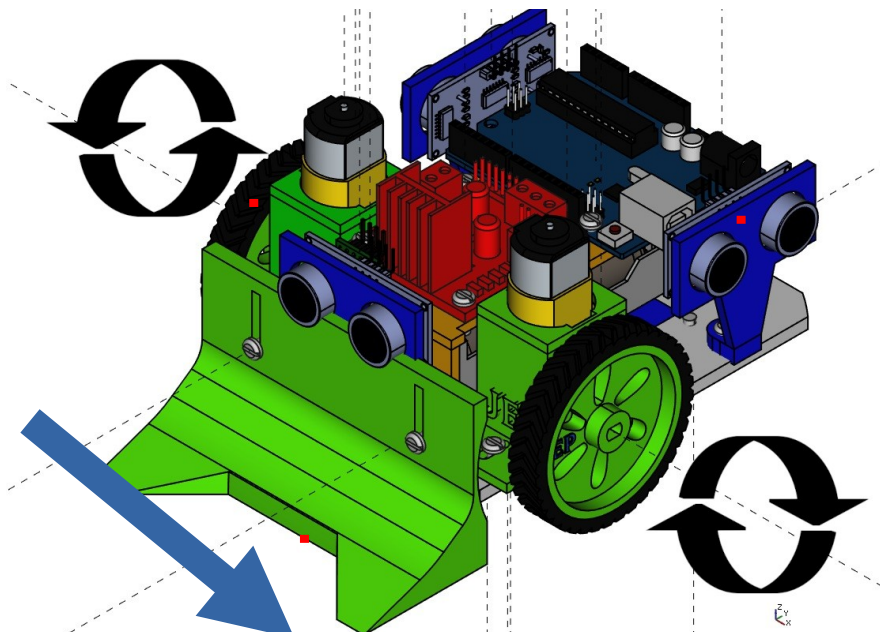


Manual de Robot Laberinto con Arduino

Movimiento hacia la derecha: Se hace girar el motor izquierdo hacia delante (horario) y el motor de la Derecha hacia atrás (antihorario). Esto provoca un movimiento de giro a la Derecha de la estructura.



Movimiento hacia la izquierda: Se hace girar el motor de la Izquierda hacia atrás (antihorario) y el motor de la Derecha hacia delante (horario), esto provoca un movimiento de giro a la Izquierda de la estructura.





Manual de Robot Laberinto con Arduino

El código de Arduino quedaría de la siguiente manera:

```

/*****
/* Autor: Palacios José Emanuel docente de la EETPN°602          */
/* Programa:movimiento de robot                                */
/* Año:2022                                                    */
*****/
// Conexiones puente h
//motor 1_izquierdo
int IN1=10; // enmascaro al pin 10 con el nombre de IN1
int IN2=9; // enmascaro al pin 9 con el nombre de IN2

//motor 2_derecho
int IN3=8; // enmascaro al pin 8 con el nombre de IN3
int IN4=7; // enmascaro al pin 7 con el nombre de IN4

void setup()
{
    pinMode(IN1, OUTPUT); // configuro el pin donde conecte IN1 como salida
    pinMode(IN2, OUTPUT); // configuro el pin donde conecte IN2 como salida
    pinMode(IN3, OUTPUT); // configuro el pin donde conecte IN3 como salida
    pinMode(IN4, OUTPUT); // configuro el pin donde conecte IN4 como salida
}

void loop()
{
    // voy hacia adelante por un segundo

    //giro motor 1 sentido anti horario
    digitalWrite(IN1,LOW); //envió 0V al pin donde conecte IN1
    digitalWrite(IN2,HIGH); //envió 5V al pin donde conecte IN2
    //giro motor 2 sentido horario
    digitalWrite(IN3,HIGH); //envió 5V al pin donde conecte IN3
    digitalWrite(IN4,LOW); //envió 0V al pin donde conecte IN4
    delay(1000); // demora de un segundo

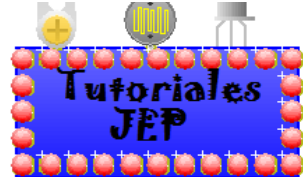
    //voy hacia atrás por un segundo

    //giro motor 1 sentido horario
    digitalWrite(IN1,HIGH); //envió 5V al pin donde conecte IN1
    digitalWrite(IN2,LOW); //envió 0V al pin donde conecte IN2
    //giro motor 2 sentido antihorario
    digitalWrite(IN3,LOW); //envió 0V al pin donde conecte IN3
    digitalWrite(IN4,HIGH); //envió 5V al pin donde conecte IN4
    delay(1000); // demora de un segundo

    //voy a la derecha con ambas ruedas por un segundo

    //giro motor 1 sentido anti horario
    digitalWrite(IN1,LOW); //envió 0V al pin donde conecte IN1
    digitalWrite(IN2,HIGH); //envió 5V al pin donde conecte IN2
    //giro motor 2 sentido antihorario
    digitalWrite(IN3,LOW); //envió 0V al pin donde conecte IN3
    digitalWrite(IN4,HIGH); //envió 5V al pin donde conecte IN4
    delay(1000); // demora de un segundo

```



Manual de Robot Laberinto con Arduino

```
// voy a la izquierda con ambas ruedas por un segundo

//giro motor 1 sentido horario
digitalWrite(IN1,HIGH); //envió 5V al pin donde conecte IN1
digitalWrite(IN2,LOW); //envió 0V al pin donde conecte IN2
//giro motor 2 sentido horario
digitalWrite(IN3,HIGH); //envió 5V al pin donde conecte IN3
digitalWrite(IN4,LOW); //envió 0V al pin donde conecte IN4
delay(1000); // demora de un segundo

// me detengo por un segundo

//motor 1 DETENIDO
digitalWrite(IN1,LOW); //envió 0V al pin donde conecte IN1
digitalWrite(IN2,LOW); //envió 0V al pin donde conecte IN2
//motor 2 DETENIDO
digitalWrite(IN3,LOW); //envió 0V al pin donde conecte IN3
digitalWrite(IN4,LOW); //envió 0V al pin donde conecte IN4
delay(1000); // demora de un segundo

// voy a la derecha con una rueda por un segundo

//giro motor 1 sentido anti horario
digitalWrite(IN1,LOW); //envió 0V al pin donde conecte IN1
digitalWrite(IN2,HIGH); //envió 5V al pin donde conecte IN2
delay(100); // demora de 10milisegundos
//motor 1 DETENIDO
digitalWrite(IN1,LOW); //envió 0V al pin donde conecte IN1
digitalWrite(IN2,LOW); //envió 0V al pin donde conecte IN2
delay(1000); // demora de un segundo

//voy a la izquierda con una rueda por un segundo

//giro motor 2 sentido horario
digitalWrite(IN3,HIGH); //envió 5V al pin donde conecte IN3
digitalWrite(IN4,LOW); //envió 0V al pin donde conecte IN4
delay(100); //demora de 10milisegundos
//motor 2 DETENIDO
digitalWrite(IN3,LOW); //envió 0V al pin donde conecte IN3
digitalWrite(IN4,LOW); //envió 0V al pin donde conecte IN4
delay(1000); // demora de un segundo

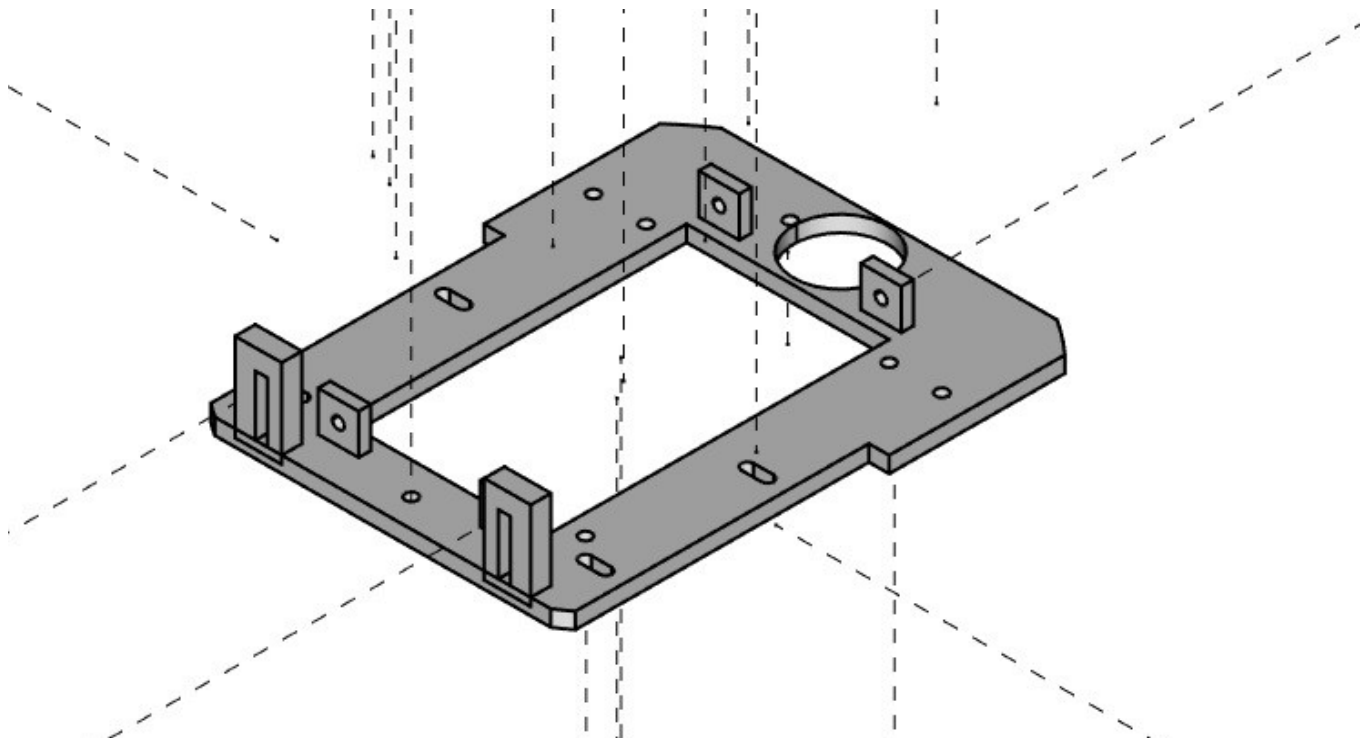
}
```

Con este programa comprobamos los movimientos del robot. Se espera que el lector corrija cualquier error de notar alguno.

Montaje de los componentes.

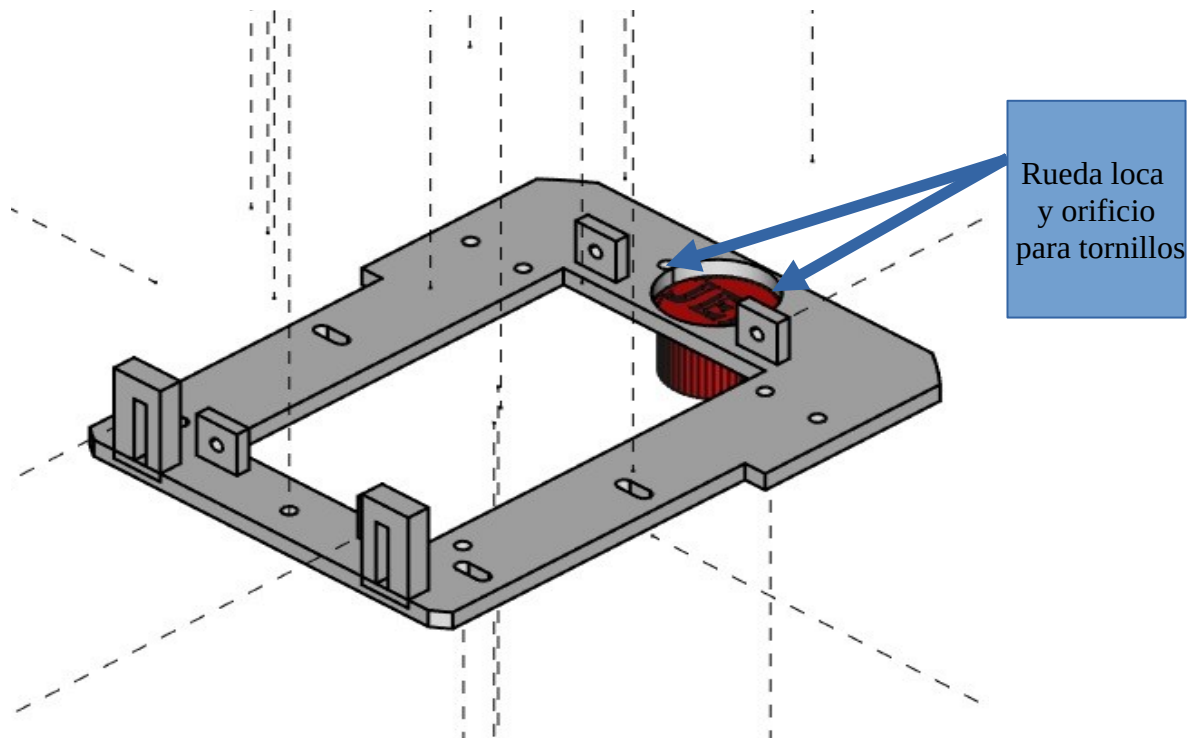
Cuando ya se han probado todos los componentes por separado lo siguiente es montar todo el circuito electrónico, sensores, motores y control en el chasis. Recomendamos el tornillos.

1) Partiremos por la base impresa en 3d

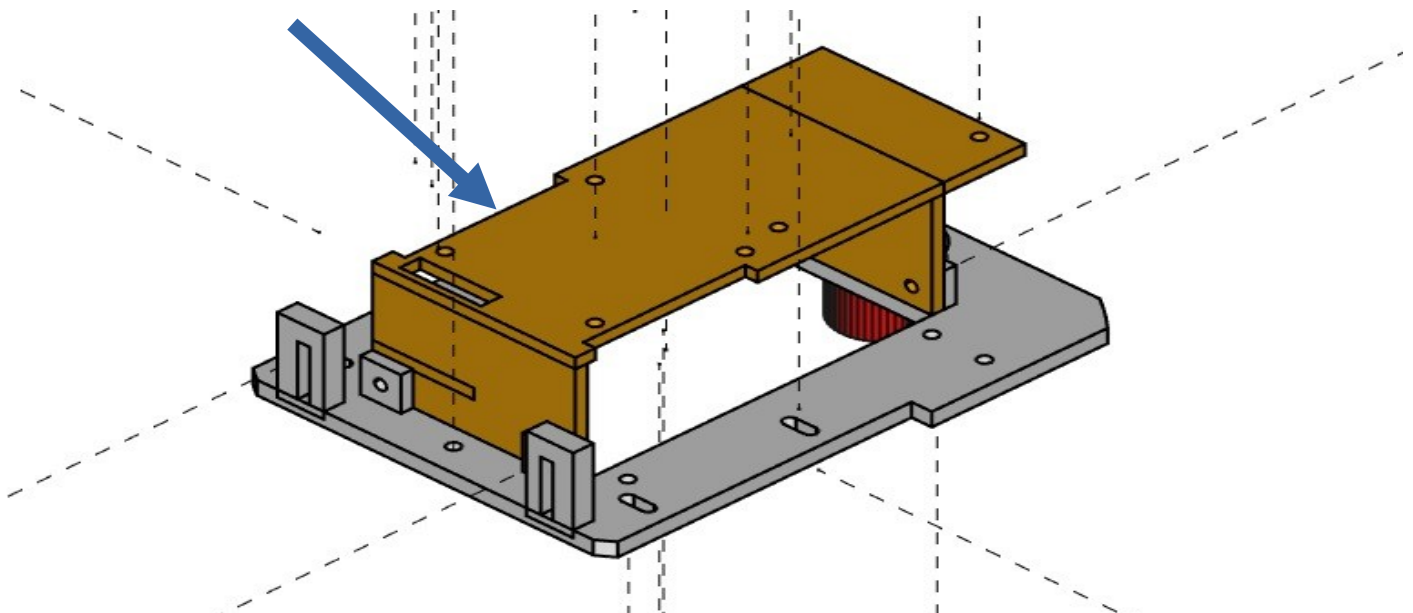


Manual de Robot Laberinto con Arduino

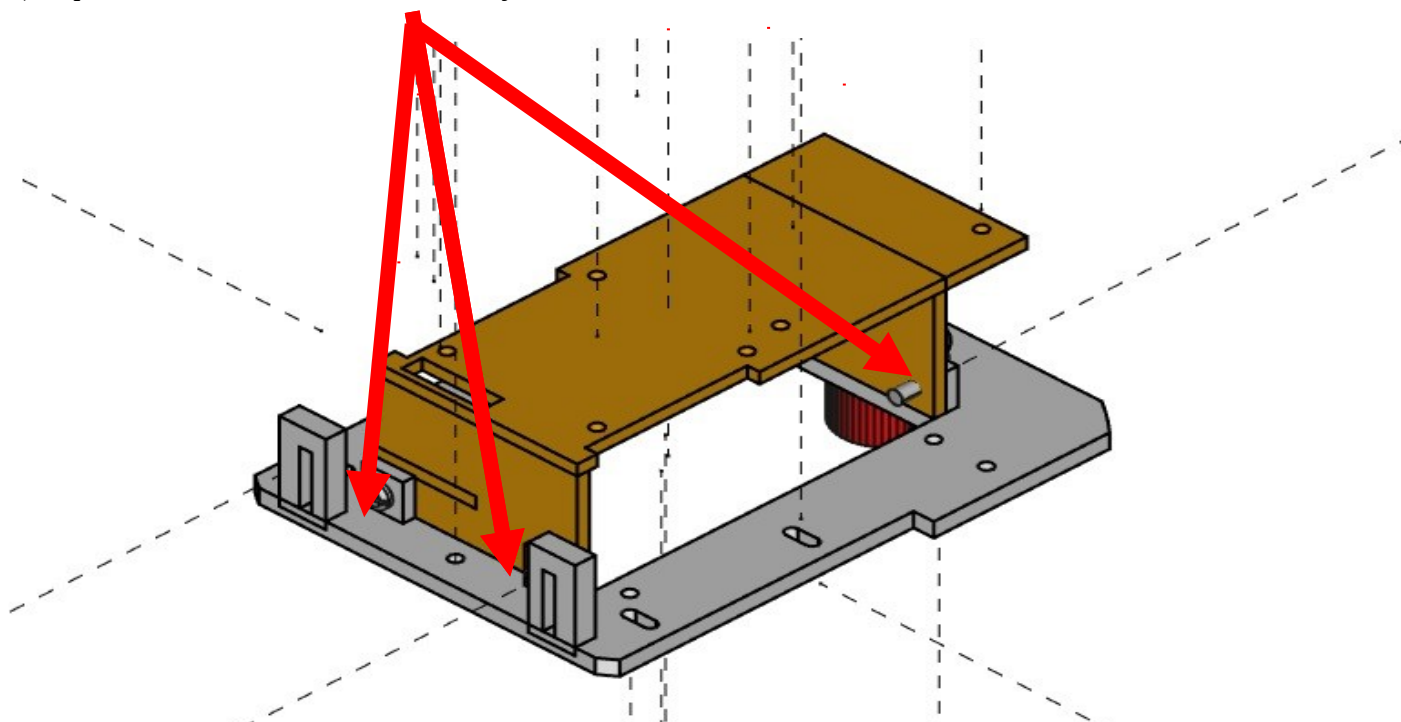
2) Montamos la rueda loca



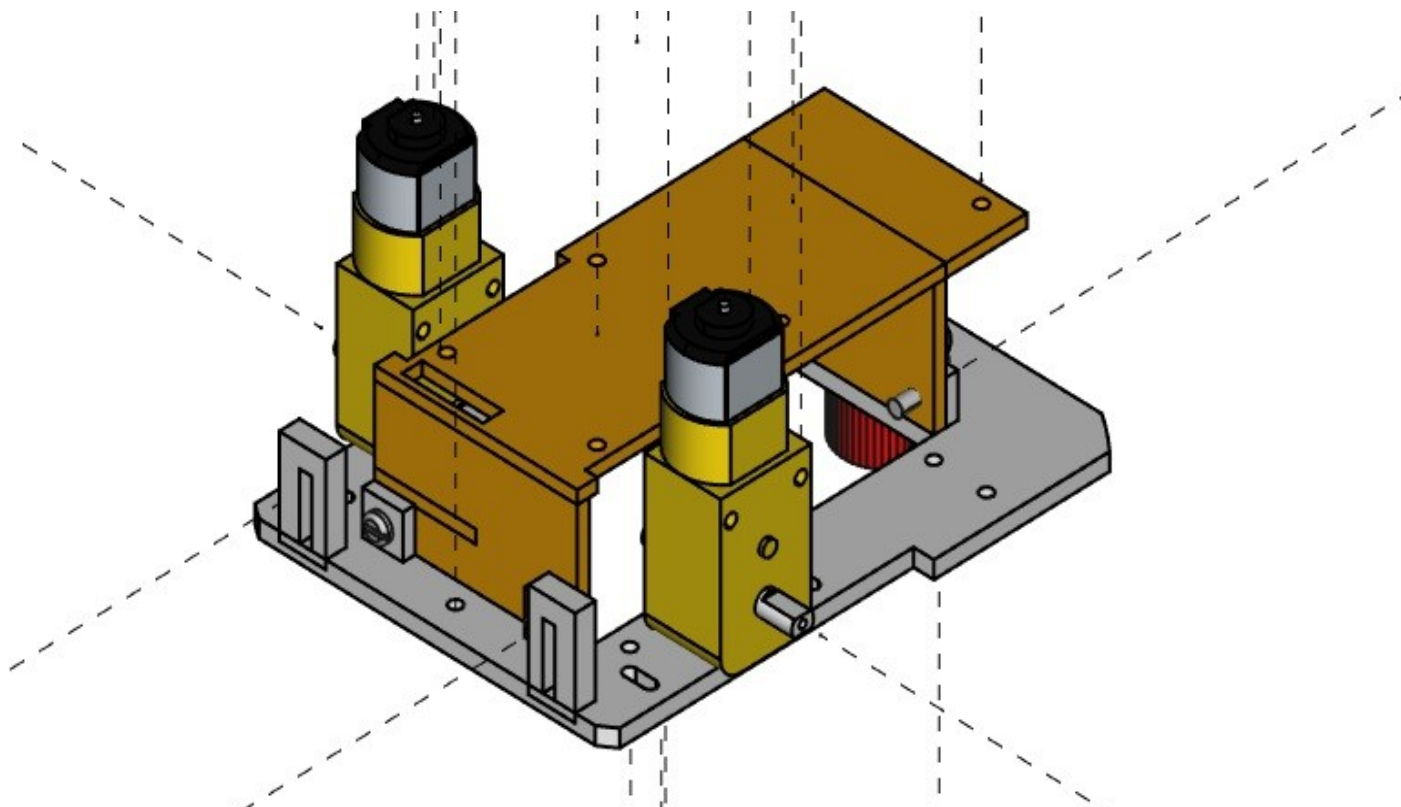
3) Montamos el soporte para los porta baterías.



4) Aquí el detalle de los tornillos de sujeción.

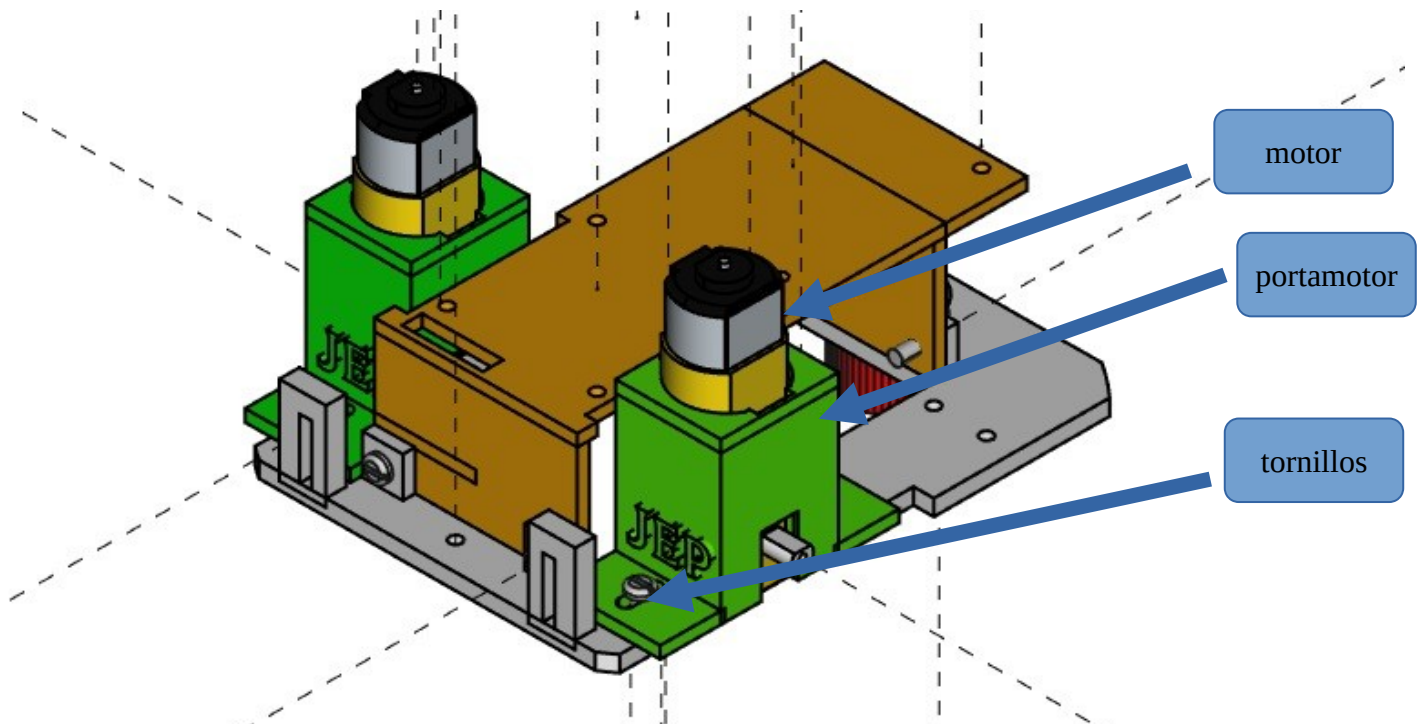


5) detalle de los motores sin los porta motores.

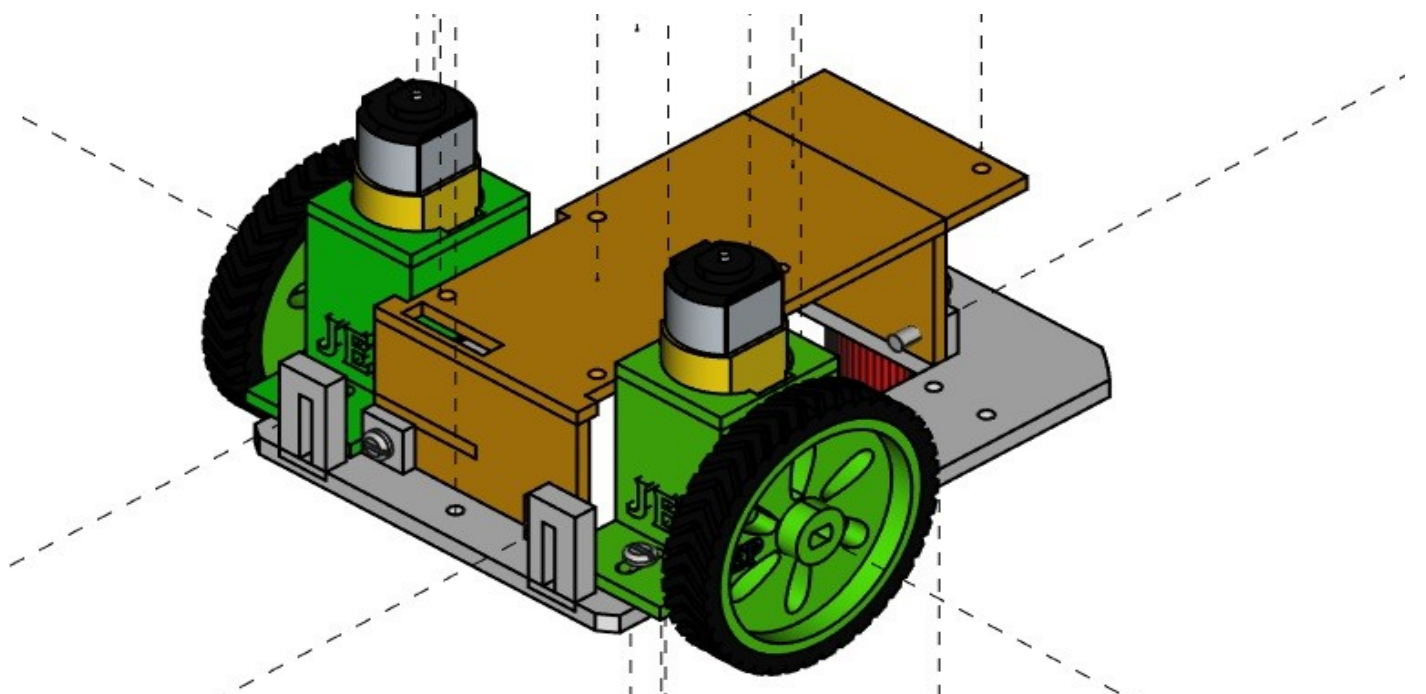


Manual de Robot Laberinto con Arduino

6) Detalle de los motores con los porta motores.

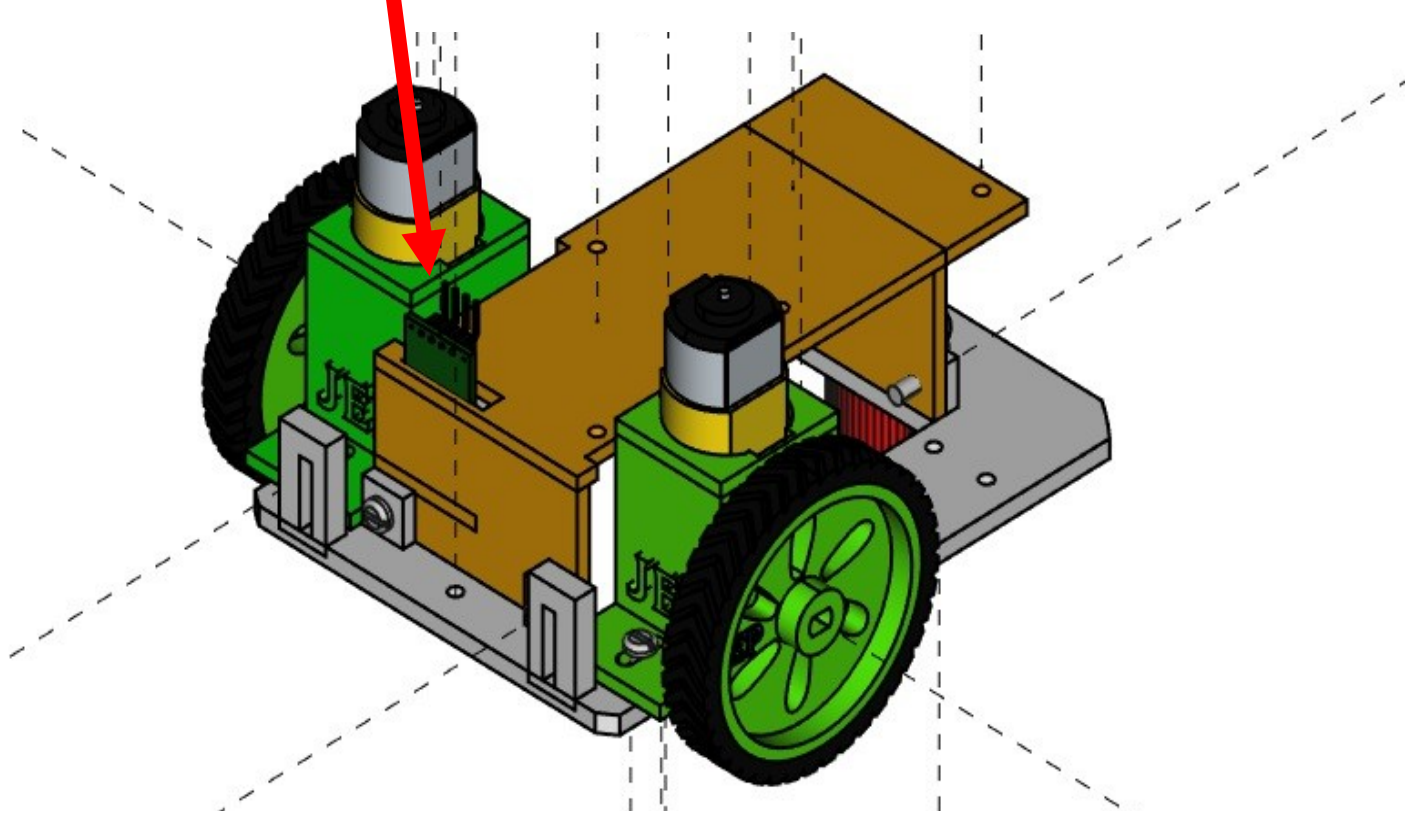


7) Colocamos las ruedas (llanta y cubierta).

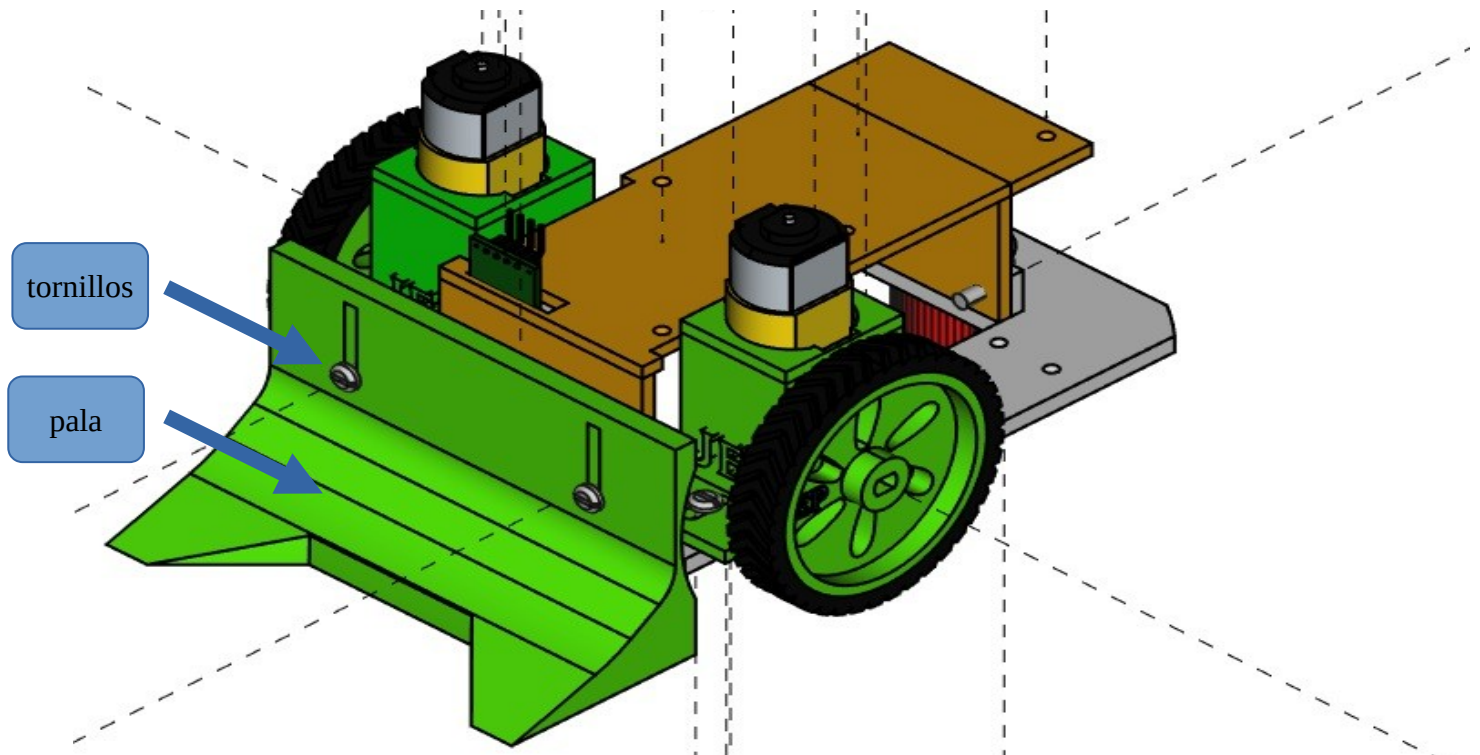


Manual de Robot Laberinto con Arduino

8) Posicionamos el modulo Bluetooth

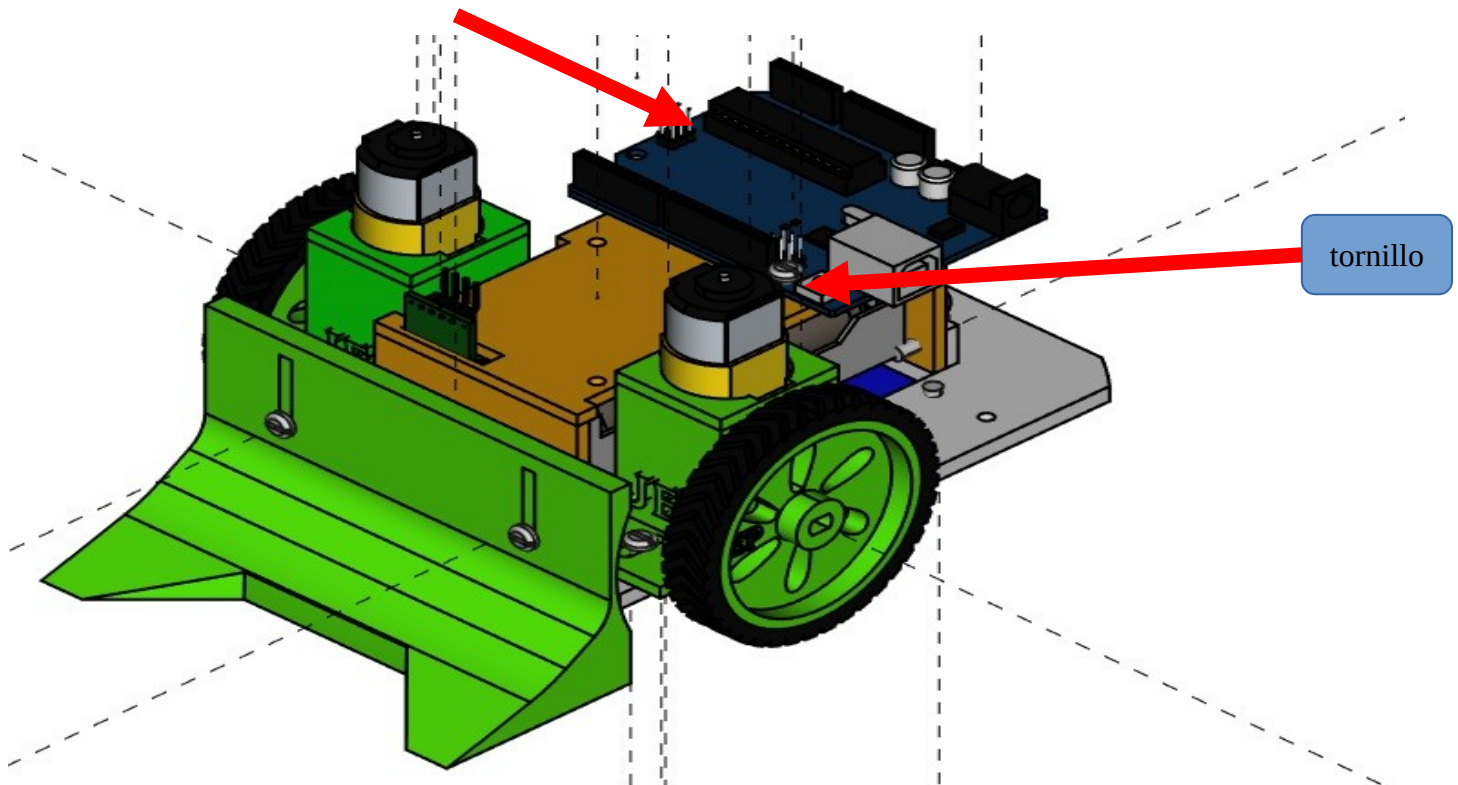


9) Detalle del montaje de la pala.

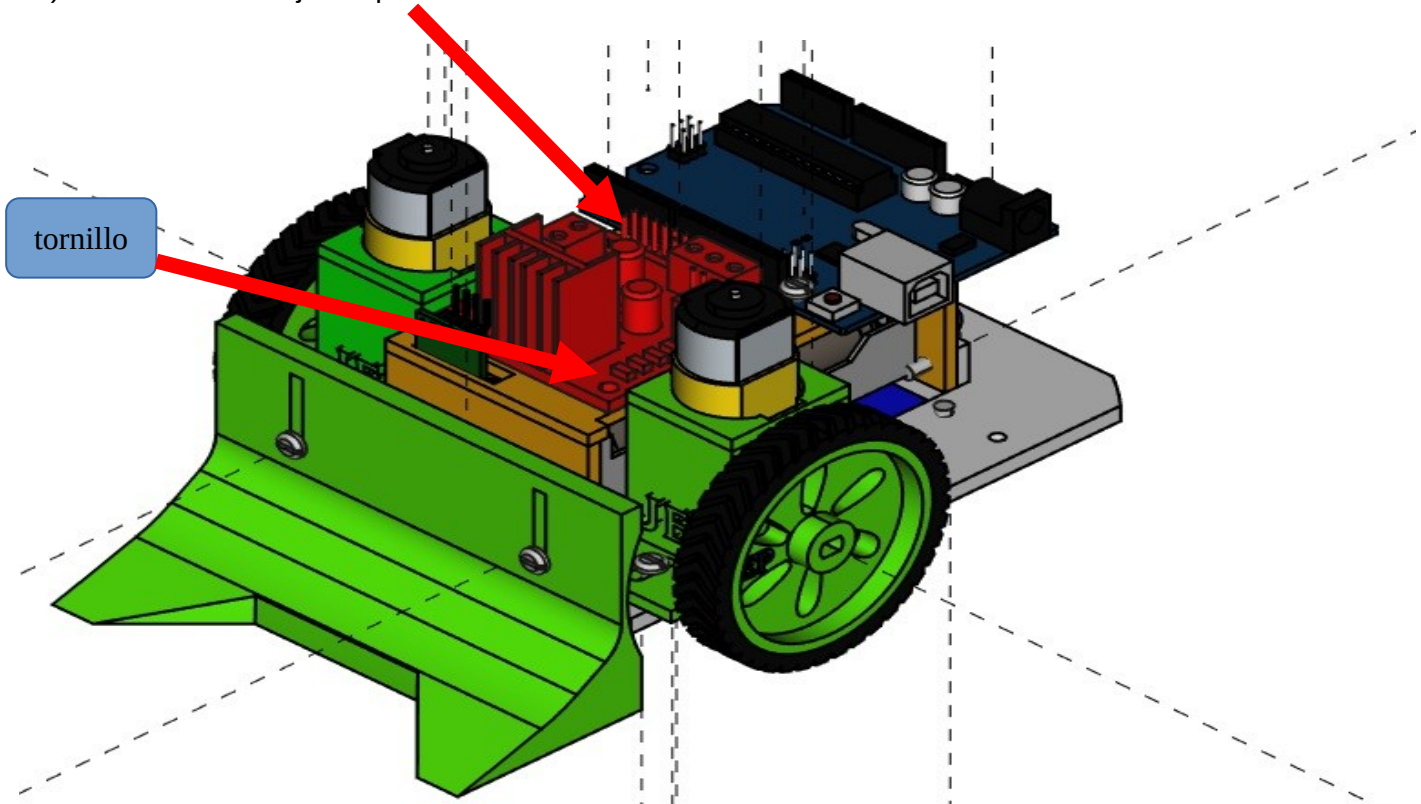


Manual de Robot Laberinto con Arduino

10) Detalle del montaje del Arduino uno.

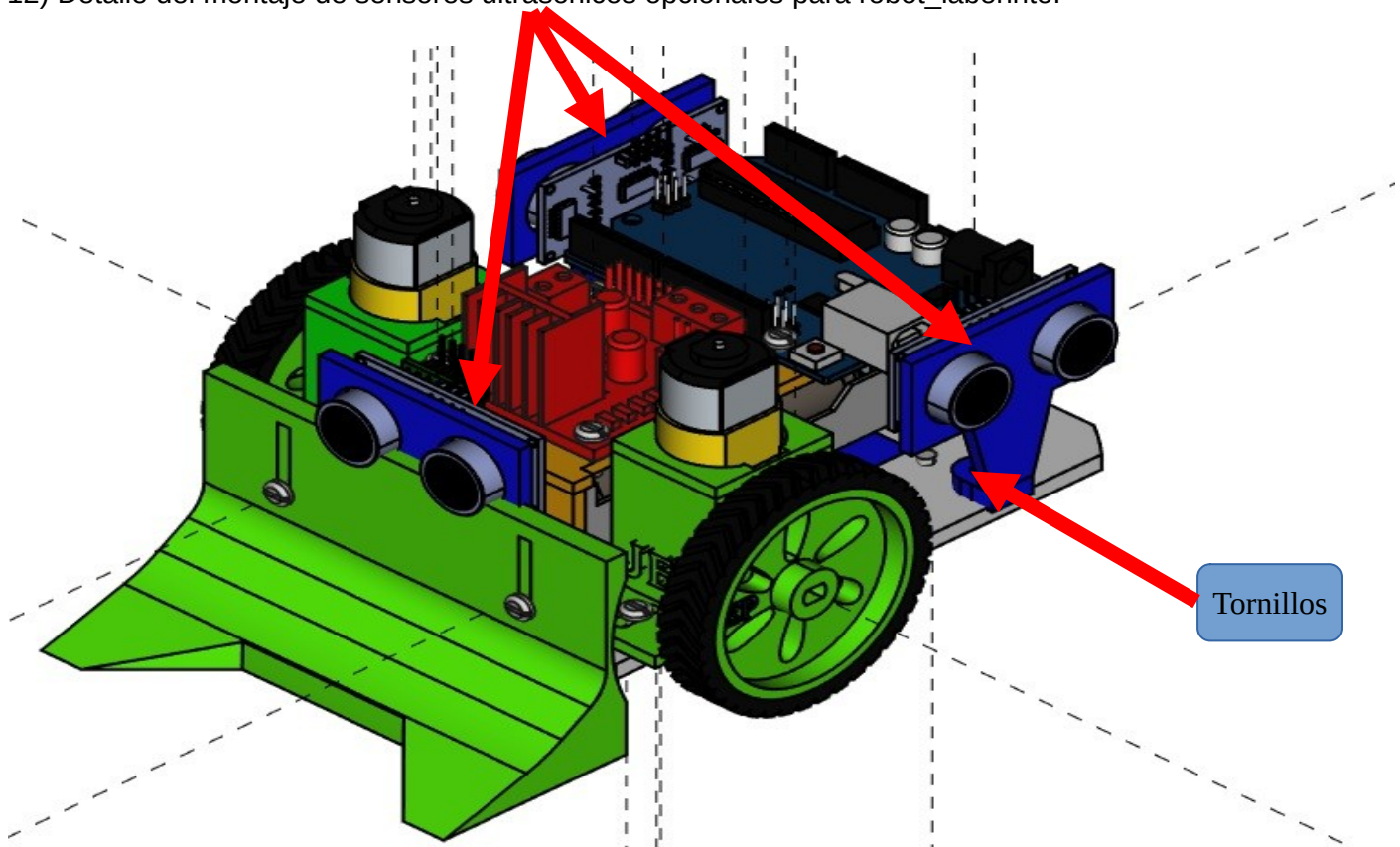


11) Detalle del montaje del puenteh.

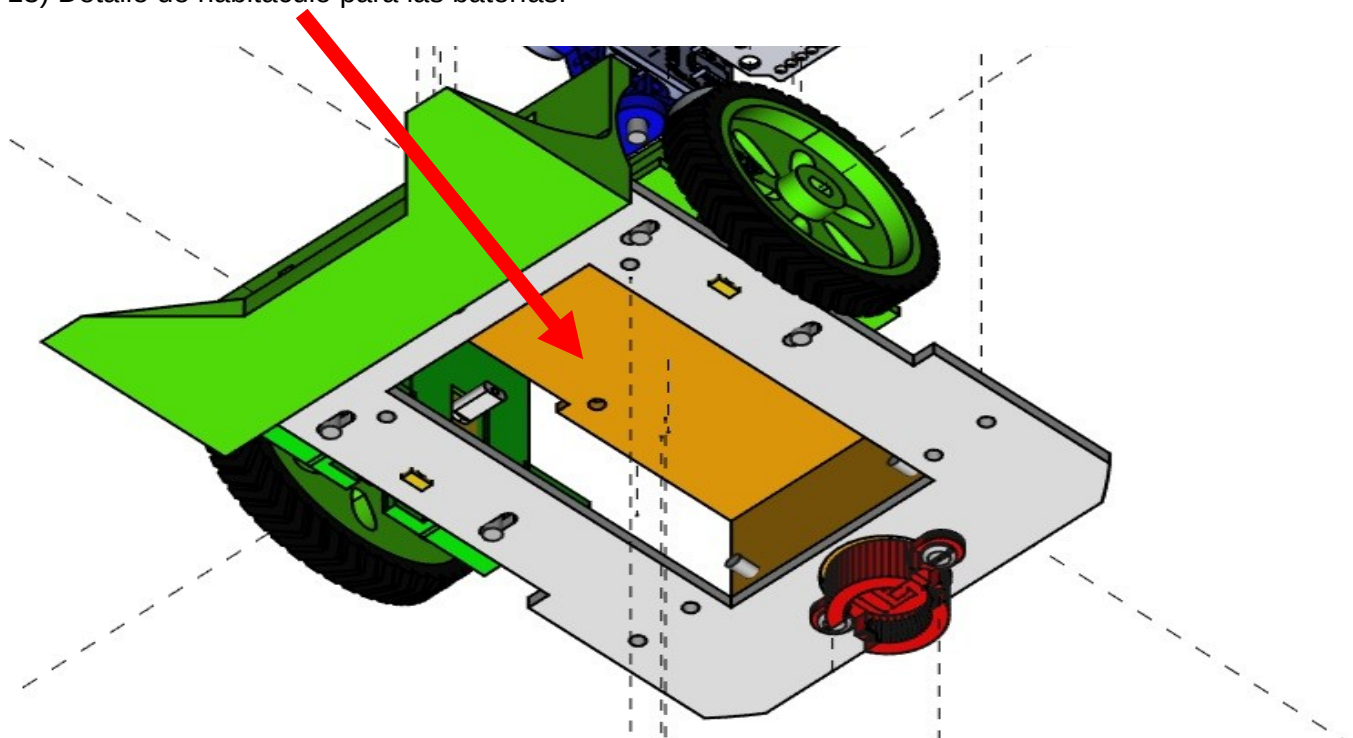


Manual de Robot Laberinto con Arduino

12) Detalle del montaje de sensores ultrasonicos opcionales para robot_laberinto.

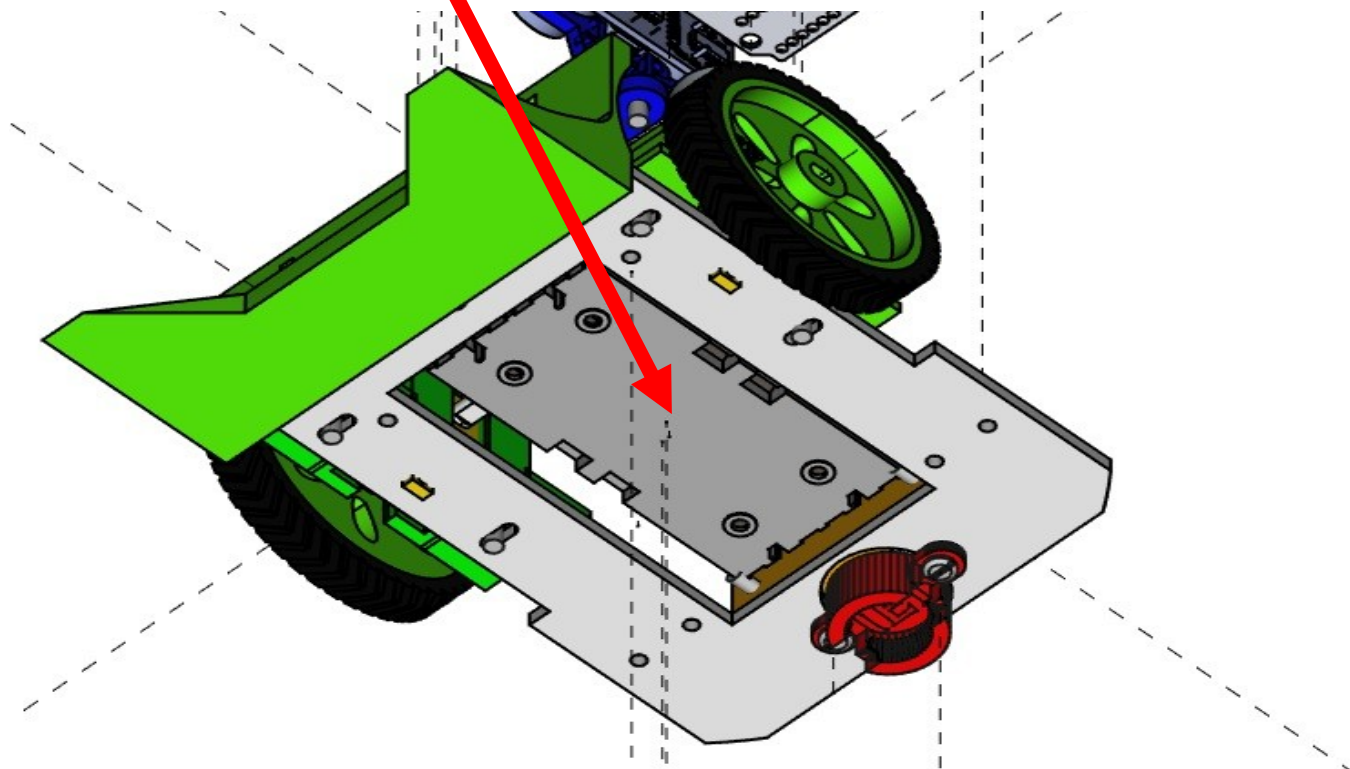


13) Detalle de habitáculo para las baterías.

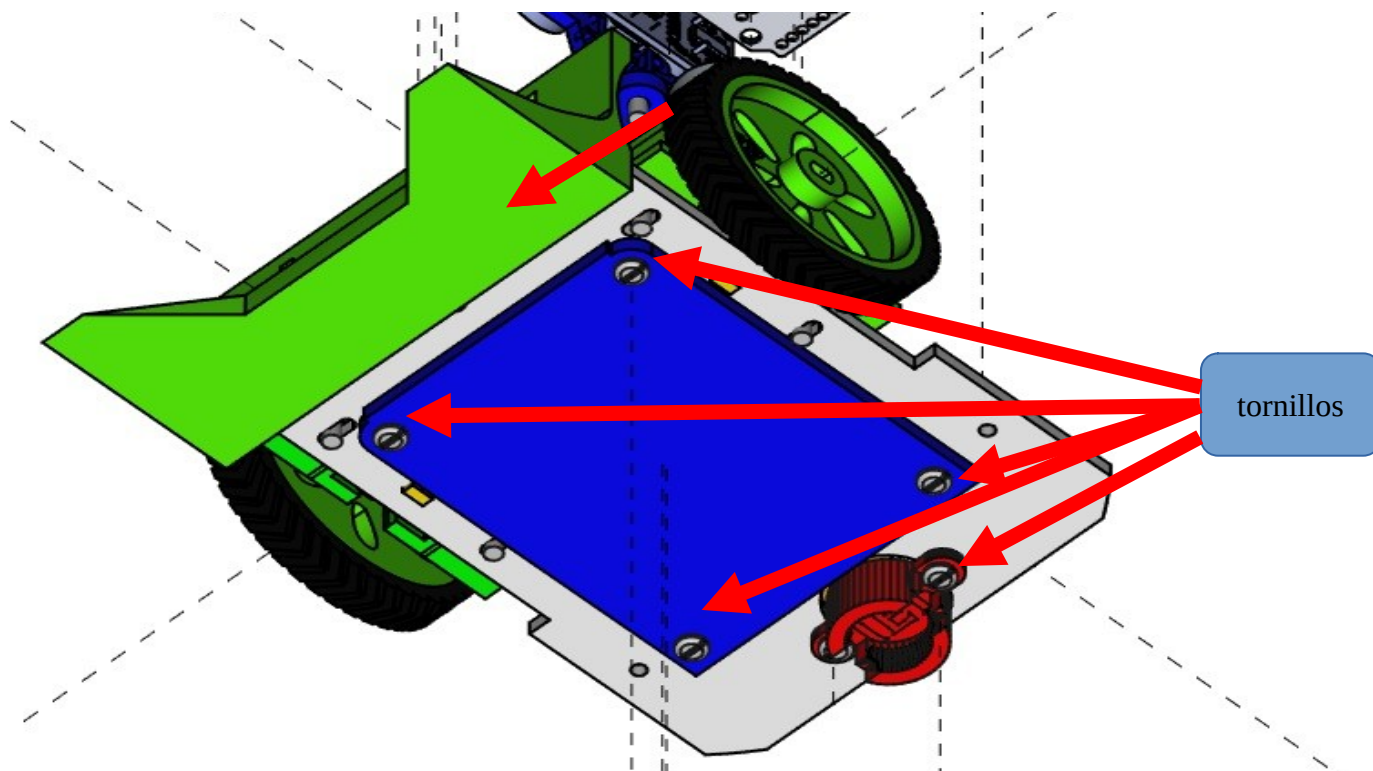


Manual de Robot Laberinto con Arduino

14) Detalle del montaje del portabaterías.

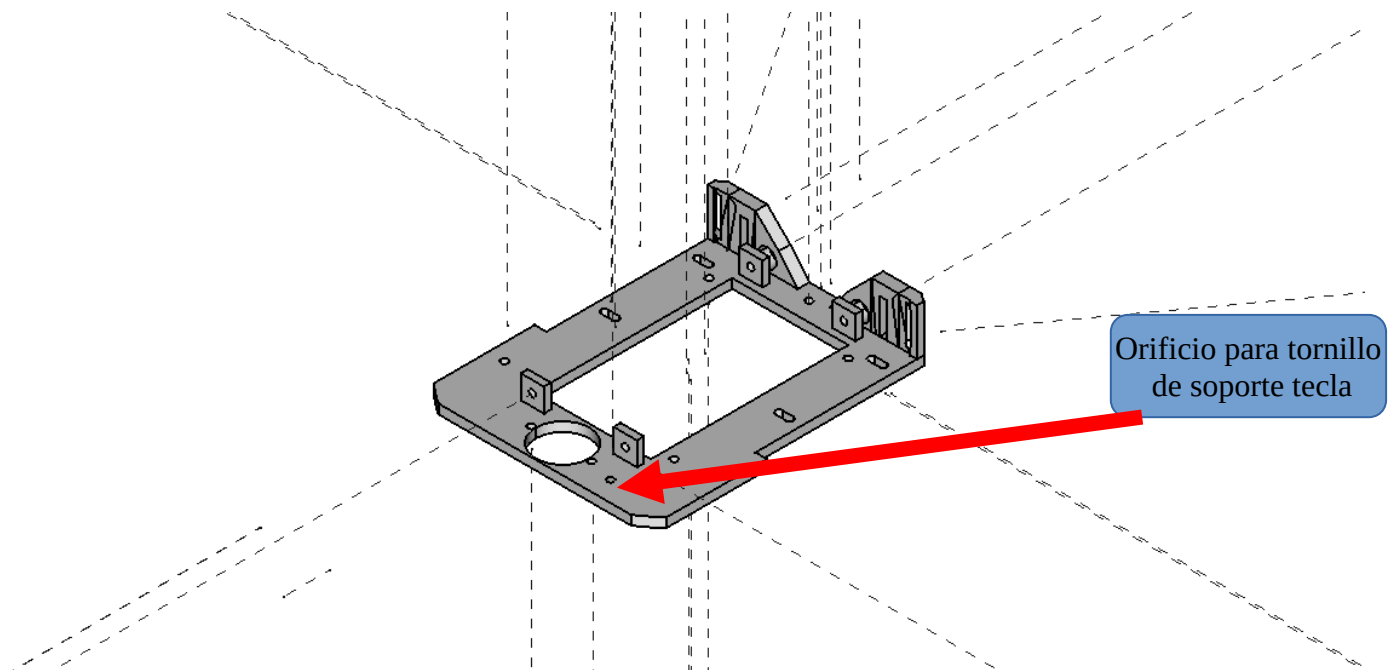


15) Detalle del montaje de la tapa del porta baterías .

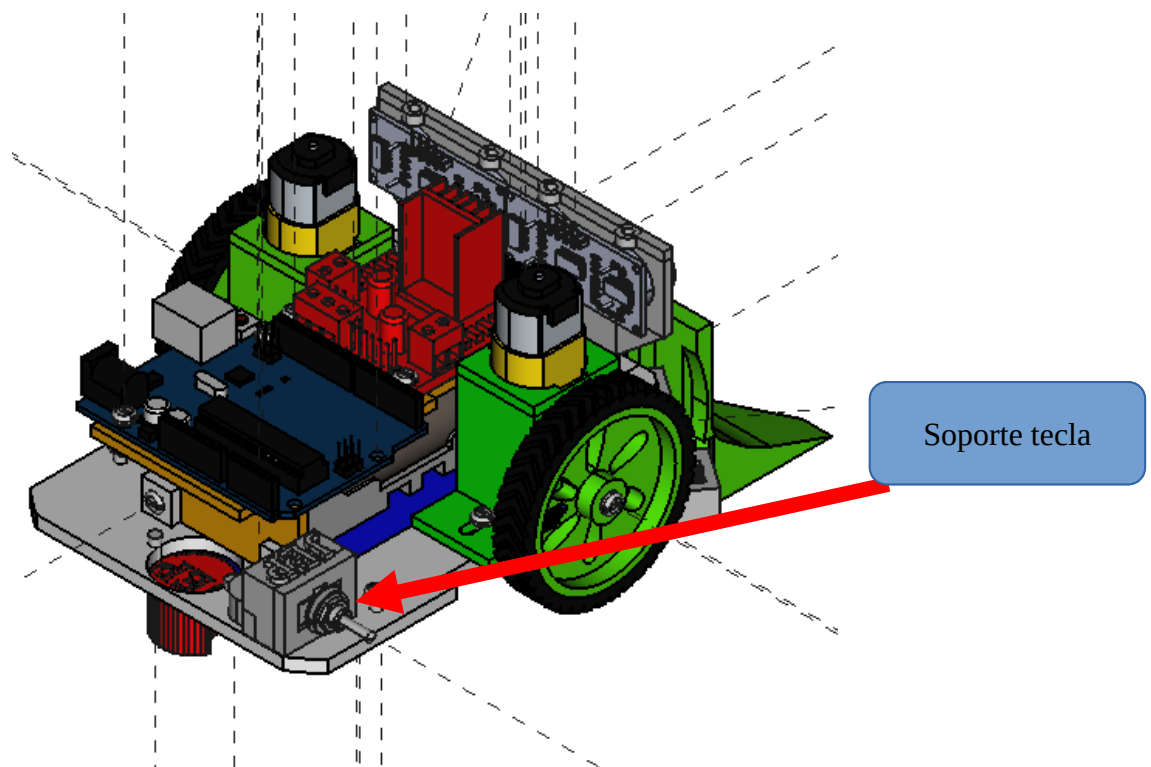


Manual de Robot Laberinto con Arduino

16)Detalle conexión de porta tecla on/off.

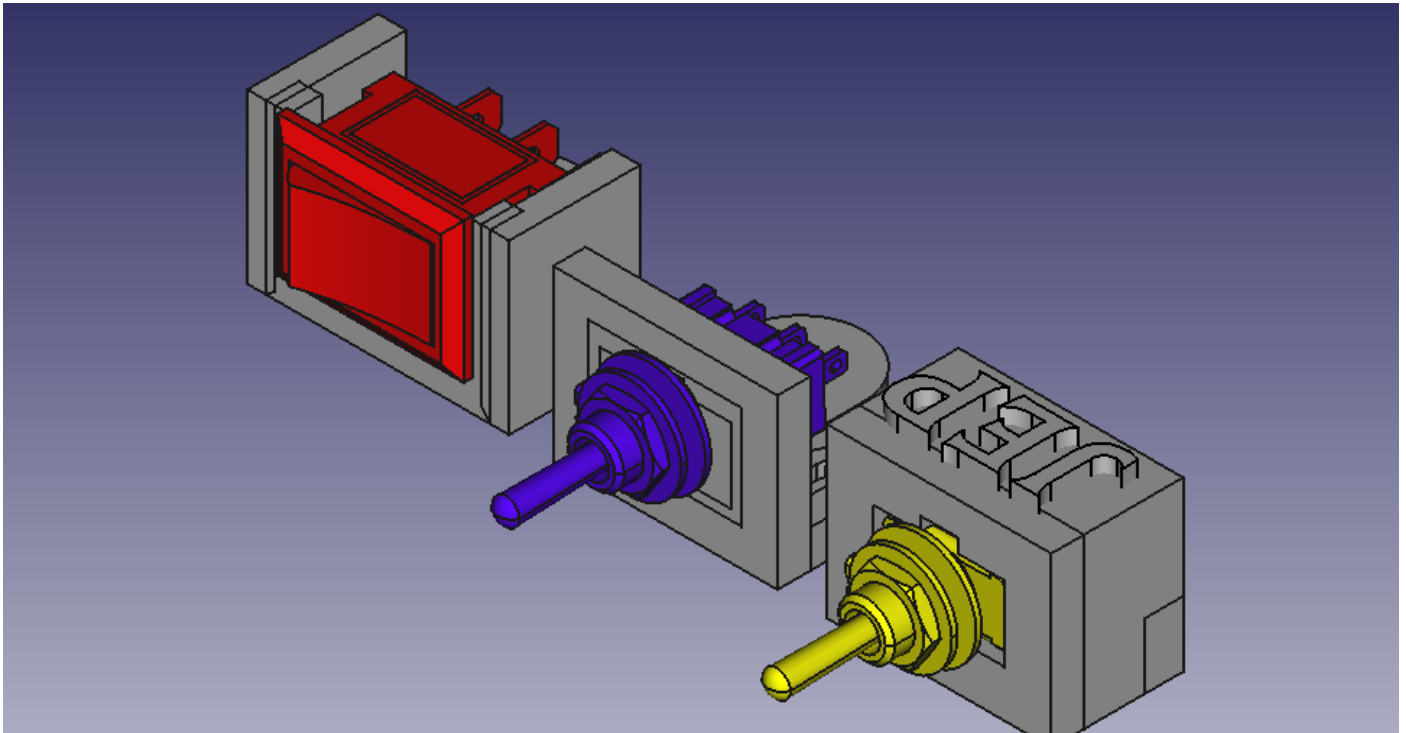


17) detalle con el porta tecla atornillado.

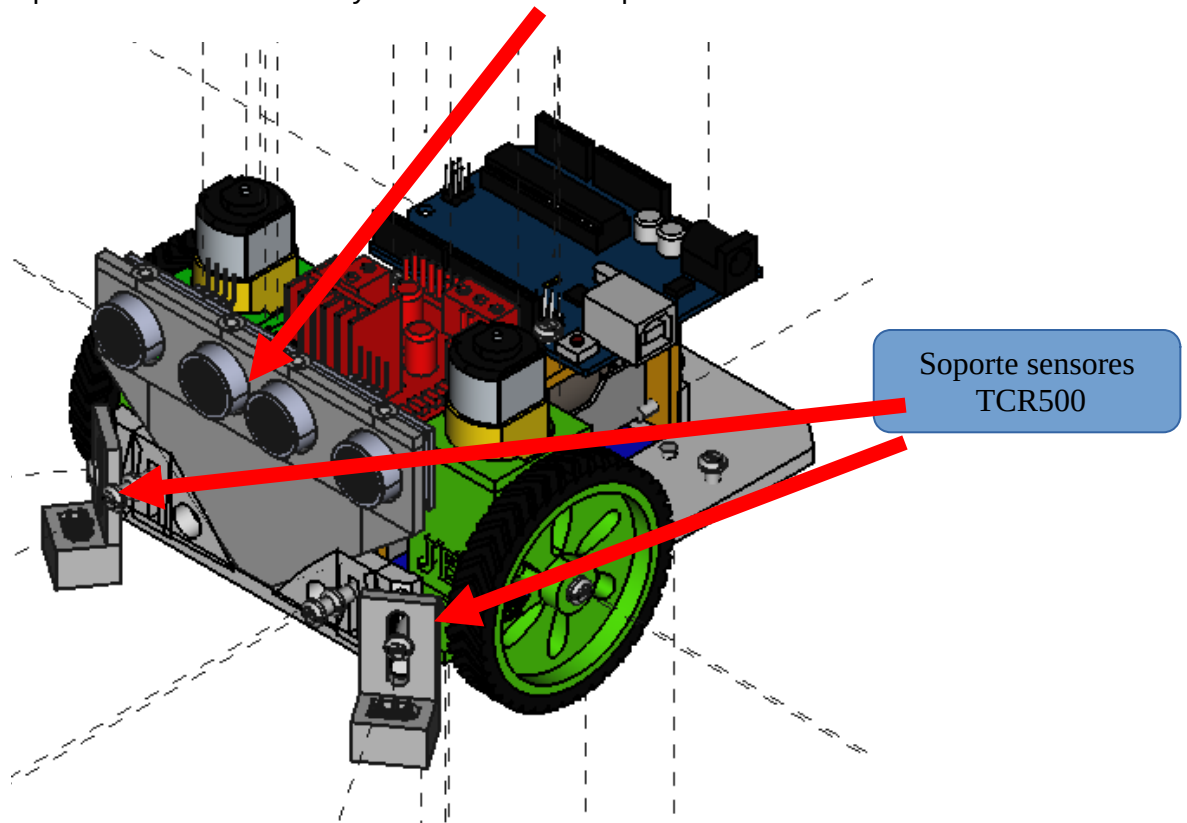


Manual de Robot Laberinto con Arduino

18)Detalle distintos soportes para interruptores de distintos tipos:

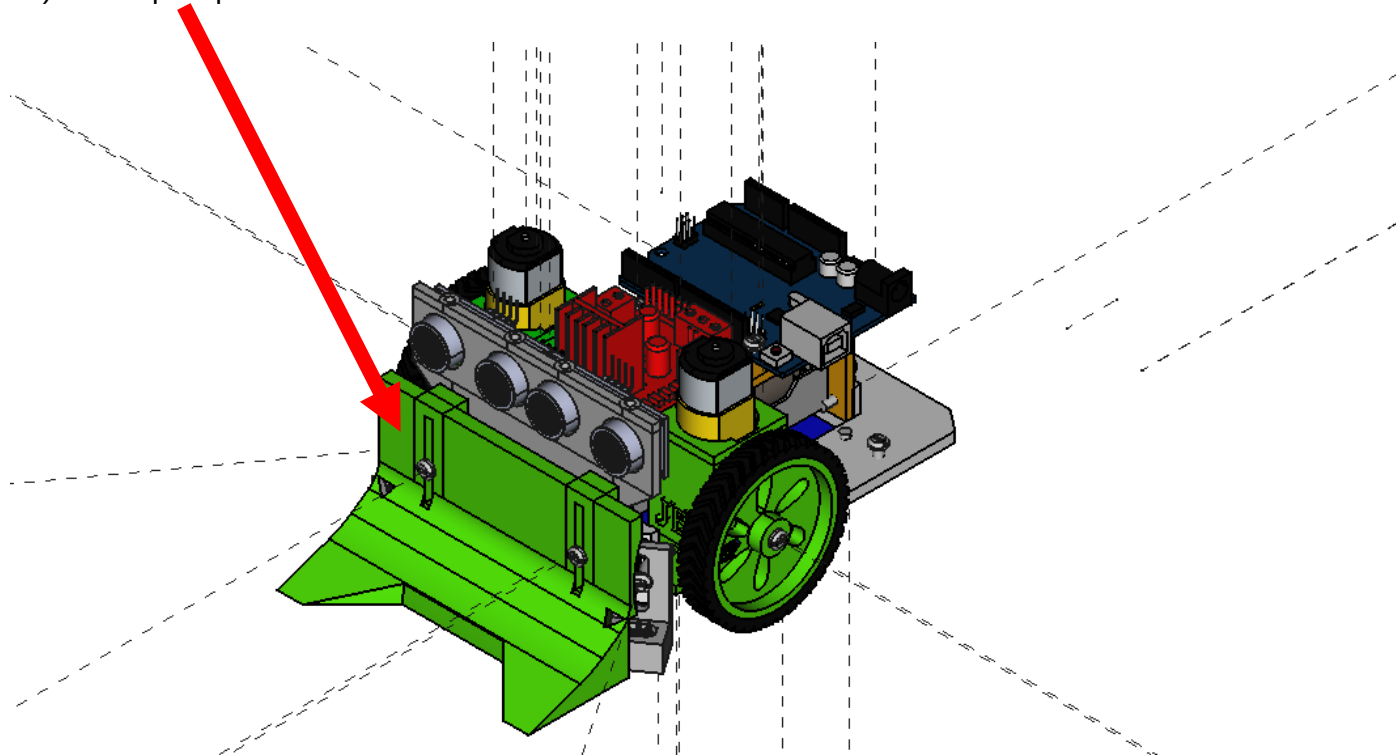


19)Detalle de soporte para sensores TCR500 y sensor ultrasonico para robot sumo.

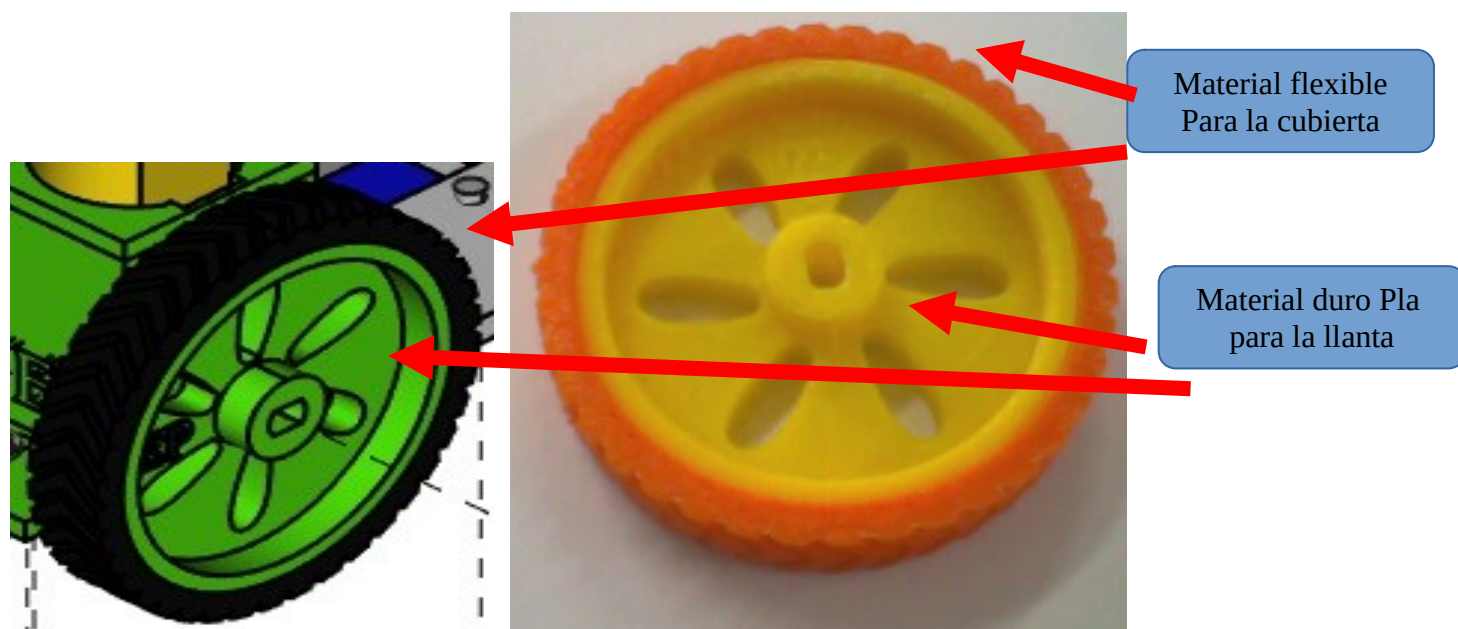


Manual de Robot Laberinto con Arduino

20)Detalle pala para sumo.

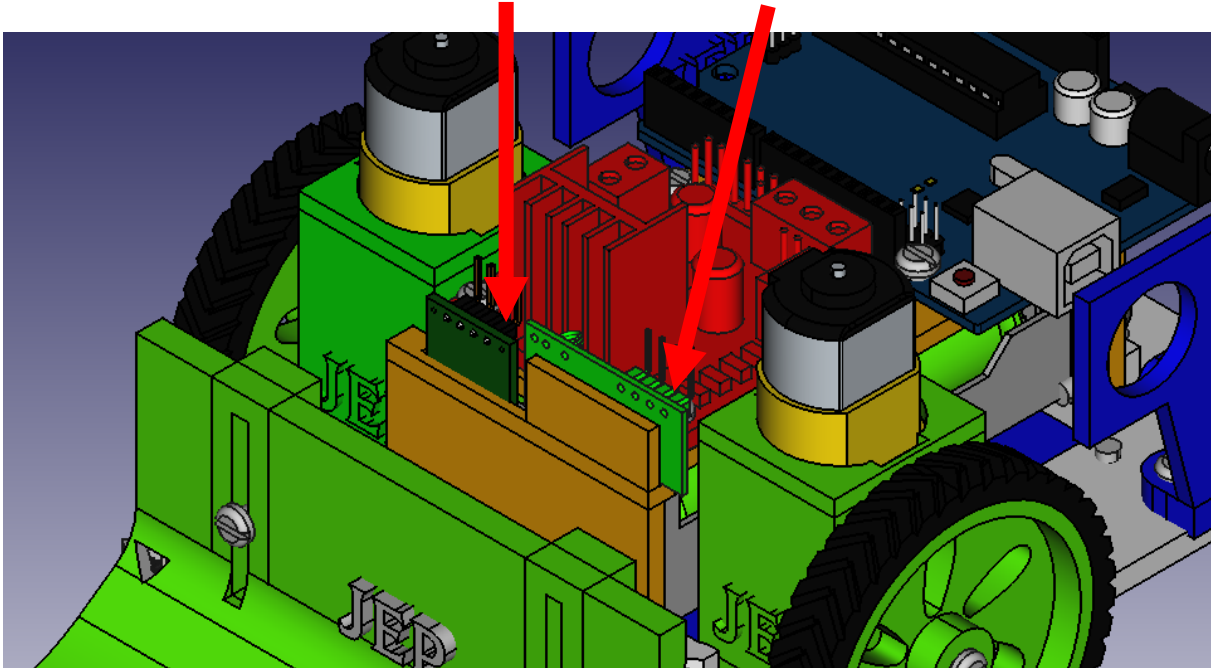


21)Detalle de rueda la cual consta de dos partes la llanta a imprimirse en PLA y la cubierta a imprimirse en material flexible.

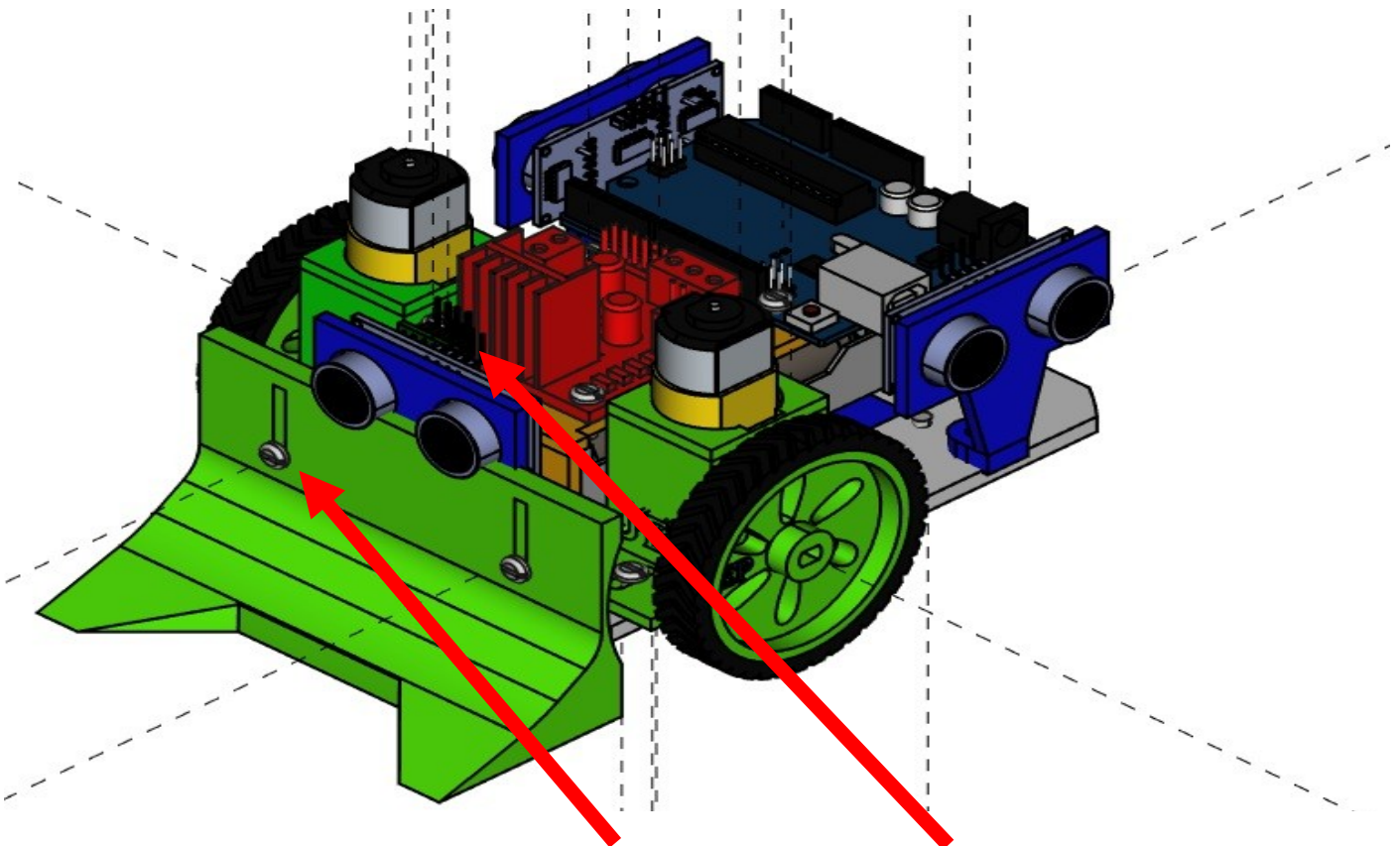


Manual de Robot Laberinto con Arduino

22) Detalle soporte para modulo Bluetooth y modulo RF receptor.



23) Detalle del robot armado.

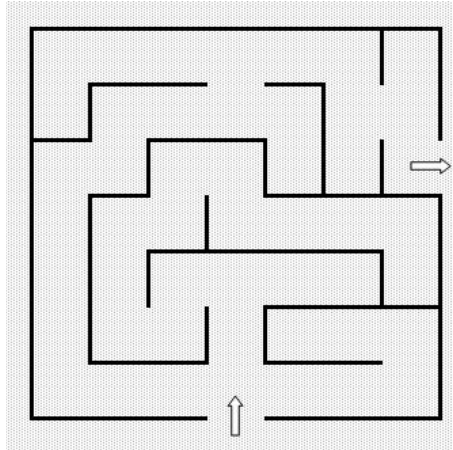


Nota: para el robot laberinto no es necesario instalar la pala y el modulo Bluetooth o el modulo FR.

Lógica para la programación de la aplicación final

Para unir todas las funciones se hace un programa que resuelve un laberinto debemos primero entender como solucionarlo.

El siguiente ejemplo de laberintos podría servir. **Nuestro robot debe ser capaz de resolverlo.**



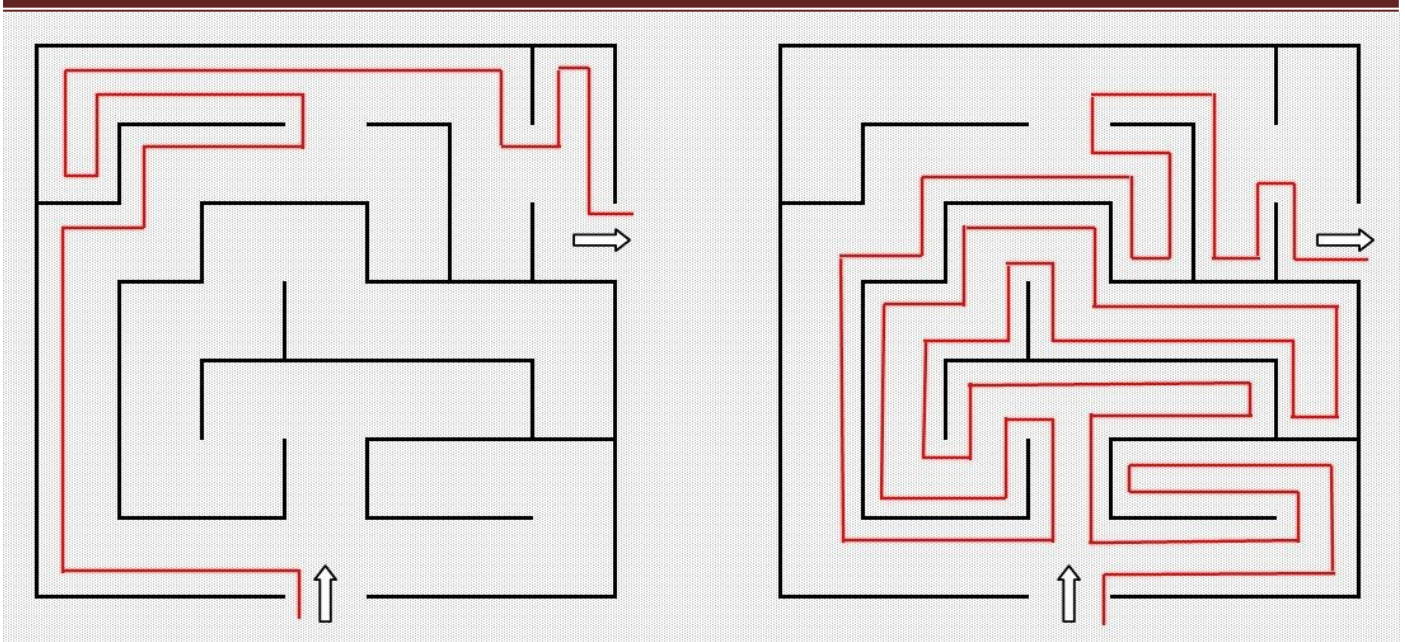
Para resolver este tipo de laberintos será suficiente con implementar un **algoritmo para robot seguidor de paredes**. Este algoritmo consiste en mantener el seguimiento de una pared priorizando un lado, derecha o izquierda, y mantener el seguimiento hasta encontrar la salida del laberinto. Una calle sin salida hay que tomarla como una pared que no acaba.

La mayoría de laberintos recreativos que hay en grandes jardines y parques son laberintos de este tipo precisamente por seguridad, para que puedan ser resueltos sencillamente siguiendo una pared.



Para poder entender mejor cómo funciona un seguidor de pared aquí puedes ver cómo debería resolver el robot uno de los laberintos que mostraba arriba dando prioridad a la derecha o a la izquierda. El algoritmo no es lo complicado, viendo el diagrama se puede entender el funcionamiento, **lo interesante es averiguar la manera de implementarlo en nuestro robot**, en este caso el "robotlaberinto".

Manual de Robot Laberinto con Arduino



En la imagen anterior se observa el laberinto que usamos de ejemplo resuelto con la lógica de ir pegado a la pared izquierda (imagen izquierda) y con la lógica de ir pegado a la pared derecha (imagen derecha)

¿A que ahora se entiende mejor? Te sorprenderías de la cantidad de laberintos que se pueden resolver así.

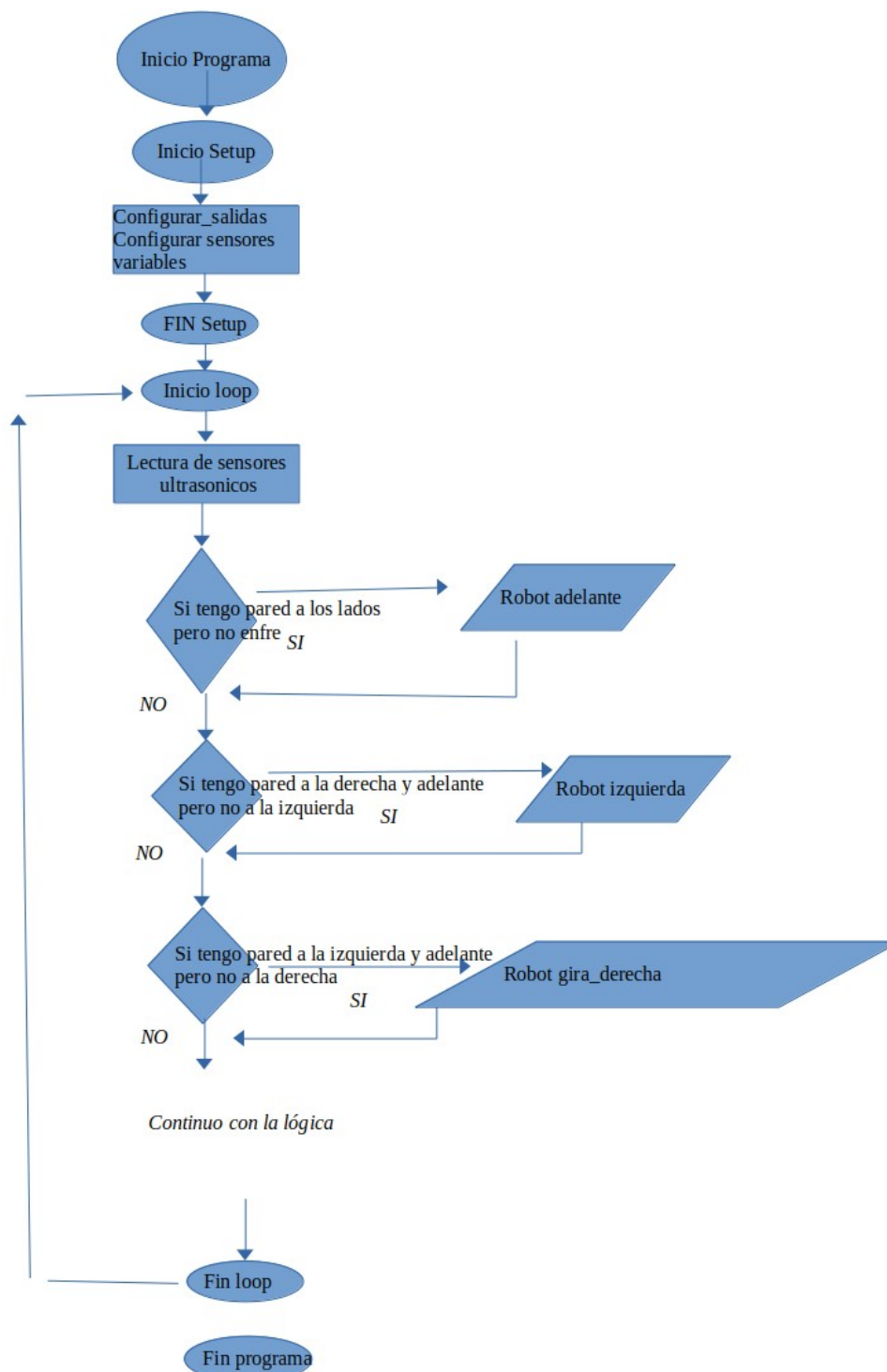
Cuando hagas las pruebas no hace falta que utilices un laberinto de pruebas tan grande como el del ejemplo. Si conseguís que el robot resuelva un laberinto pequeño, podrá resolver uno grande. Realmente no importa el tamaño del laberinto mientras sea de este tipo.

Una vez conocida la solución que le vamos a dar al problema, llega el momento de diseñar la implementación, tanto a nivel de electrónica, mecánica, utilización de sensores como de programación.



Manual de Robot Laberinto con Arduino

El diagrama de flujo sería el siguiente:



Se espera que el lector complete la lógica.



Manual de Robot Laberinto con Arduino

El código de Arduino quedaría de la siguiente manera:

```

/*****
/* Autor: Palacios José Emanuel docente de la EETPN°602          */
/* Programa:Programa final                                     */
/* Año:2022                                                    */
*****/
// Conexiones puente h
//motor 1_izquierdo
int IN1=10; // enmascaro al pin 10 con el nombre de IN1
int IN2=9; // enmascaro al pin 9 con el nombre de IN2

//motor 2_derecho
int IN3=8; // enmascaro al pin 8 con el nombre de IN3
int IN4=7; // enmascaro al pin 7 con el nombre de IN4

// Conexiones sensores ultrasonicos
// sensor 1 derecha
const int Trigger = 2;    // enmascaro al pin 2 con el nombre de Trigger del sensor 1
const int Echo = 3;      // enmascaro al pin 3 con el nombre de Echo del sensor 1

// sensor 1 izquierda
const int Trigger2 = 4;   // enmascaro al pin 4 con el nombre de Trigger2 del sensor 2
const int Echo2 = 5;     // enmascaro al pin 5 con el nombre de Echo2 del sensor 2

// sensor 3 frente
const int Trigger3 = 12;  // enmascaro al pin 4 con el nombre de Trigger2 del sensor 2
const int Echo3 = 13;    // enmascaro al pin 5 con el nombre de Echo2 del sensor 2

void setup()
{
  Serial.begin(9600); // inicio el puerto serial para la comunicación con el puerto serial de ser necesario
  // habilito pines puente h
  pinMode(IN1, OUTPUT); // configuro el pin donde conecte IN1 como salida
  pinMode(IN2, OUTPUT); // configuro el pin donde conecte IN2 como salida
  pinMode(IN3, OUTPUT); // configuro el pin donde conecte IN3 como salida
  pinMode(IN4, OUTPUT); // configuro el pin donde conecte IN4 como salida
  // habilito pines sensores ultrasonicos
  pinMode(Trigger, OUTPUT); //pin como salida
  pinMode(Echo, INPUT);    //pin como entrada
  digitalWrite(Trigger, LOW); //Inicializamos el pin con 0
  pinMode(Trigger2, OUTPUT); //pin como salida
  pinMode(Echo2, INPUT);    //pin como entrada
  digitalWrite(Trigger2, LOW); //Inicializamos el pin con 0
  pinMode(Trigger3, OUTPUT); //pin como salida
  pinMode(Echo3, INPUT);    //pin como entrada
  digitalWrite(Trigger3, LOW); //Inicializamos el pin con 0

  // demora de 5 segundos para darnos tiempo a soltar el robot

  delay(5000); // corroborar el tiempo según reglamento
}

void loop()
{

```




Manual de Robot Laberinto con Arduino

```

lectura de sensores ultrasonicos
// VARIABLES PARA USAR CON EL SENSOR ULTRASONICO_1*****
long t; //tiempo que demora en llegar el eco
long d_derecha; //distancia en centimetros

// ENVIO PULSO PARA EL SENSOR ULTRASONICO _ 1
digitalWrite(Triiger, HIGH); // ponemos el pin trigger en 5v
delayMicroseconds(10); //Esperamos 10us
digitalWrite(Triiger, LOW); // ponemos el pin trigger en 0v
//con esto Enviamos un pulso de 10us PARA EL SENSOR ULTRASONICO_1_derecha

t=pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
d_derecha=t/59; //calculamos la distancia en cm

Serial.print("Distancia derecha: ");
Serial.print(d_derecha); //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
delay(100); //Hacemos una pausa de 100ms

// VARIABLES PARA USAR CON EL SENSOR ULTRASONICO_2_izquierda*****
long t2; //tiempo que demora en llegar el eco
long d2_izquierda; //distancia en centimetros

// ENVIO PULSO PARA EL SENSOR ULTRASONICO _ 2
digitalWrite(Triiger2, HIGH); // ponemos el pin trigger en 5v
delayMicroseconds(10); //Esperamos 10us
digitalWrite(Triiger2, LOW); // ponemos el pin trigger en 0v
//con esto Enviamos un pulso de 10us PARA EL SENSOR ULTRASONICO_2

t2=pulseIn(Echo2, HIGH); //obtenemos el ancho del pulso
d2_izquierda=t2/59; //calculamos la distancia en cm

Serial.print("Distancia2 izquierda: ");
Serial.print(d2_izquierda); //Enviamos serialmente el valor de la distancia 2
Serial.print("cm");
Serial.println();
delay(100); //Hacemos una pausa de 100ms

// VARIABLES PARA USAR CON EL SENSOR ULTRASONICO_3_frente*****
long t3; //tiempo que demora en llegar el eco
long d3_frente; //distancia en centimetros

// ENVIO PULSO PARA EL SENSOR ULTRASONICO _ 2
digitalWrite(Triiger3, HIGH); // ponemos el pin trigger en 5v
delayMicroseconds(10); //Esperamos 10us
digitalWrite(Triiger3, LOW); // ponemos el pin trigger en 0v
//con esto Enviamos un pulso de 10us PARA EL SENSOR ULTRASONICO_3

t3=pulseIn(Echo3, HIGH); //obtenemos el ancho del pulso
d3_frente=t3/59; //calculamos la distancia en cm

Serial.print("Distancia frente: ");
Serial.print(d3_frente); //Enviamos serialmente el valor de la distancia 3
Serial.print("cm");
Serial.println();
delay(100); //Hacemos una pausa de 100ms

// con esto ya leimos las distancias frontal y a los costados

```



Manual de Robot Laberinto con Arduino

```
// ahora queda tomar las decisiones en función de esos datos
// d_derecha
// d2_izquierda
// d3_frente
//***** TOMA DE DECISIONES
//*****
// si la distancia derecha es menor a 10cm(hay pared) y la distancia izquierda es menor a 10cm ( hay pared) y la
distancia en frente es mayor a 10cm( no hay pared cerca)
if((d_derecha<= 10)&&(d2_izquierda<= 10)&&(d3_frente>= 10)){
    // entonces voy hacia adelante

    // hacia adelante
    //giro motor 1 sentido anti horario
    digitalWrite(IN1,LOW); //envió 0V al pin donde conecte IN1
    digitalWrite(IN2,HIGH); //envió 5V al pin donde conecte IN2
    //giro motor 2 sentido horario
    digitalWrite(IN3,HIGH); //envió 5V al pin donde conecte IN3
    digitalWrite(IN4,LOW); //envió 0V al pin donde conecte IN4
    delay(100);// demora de 10milisegundos
}

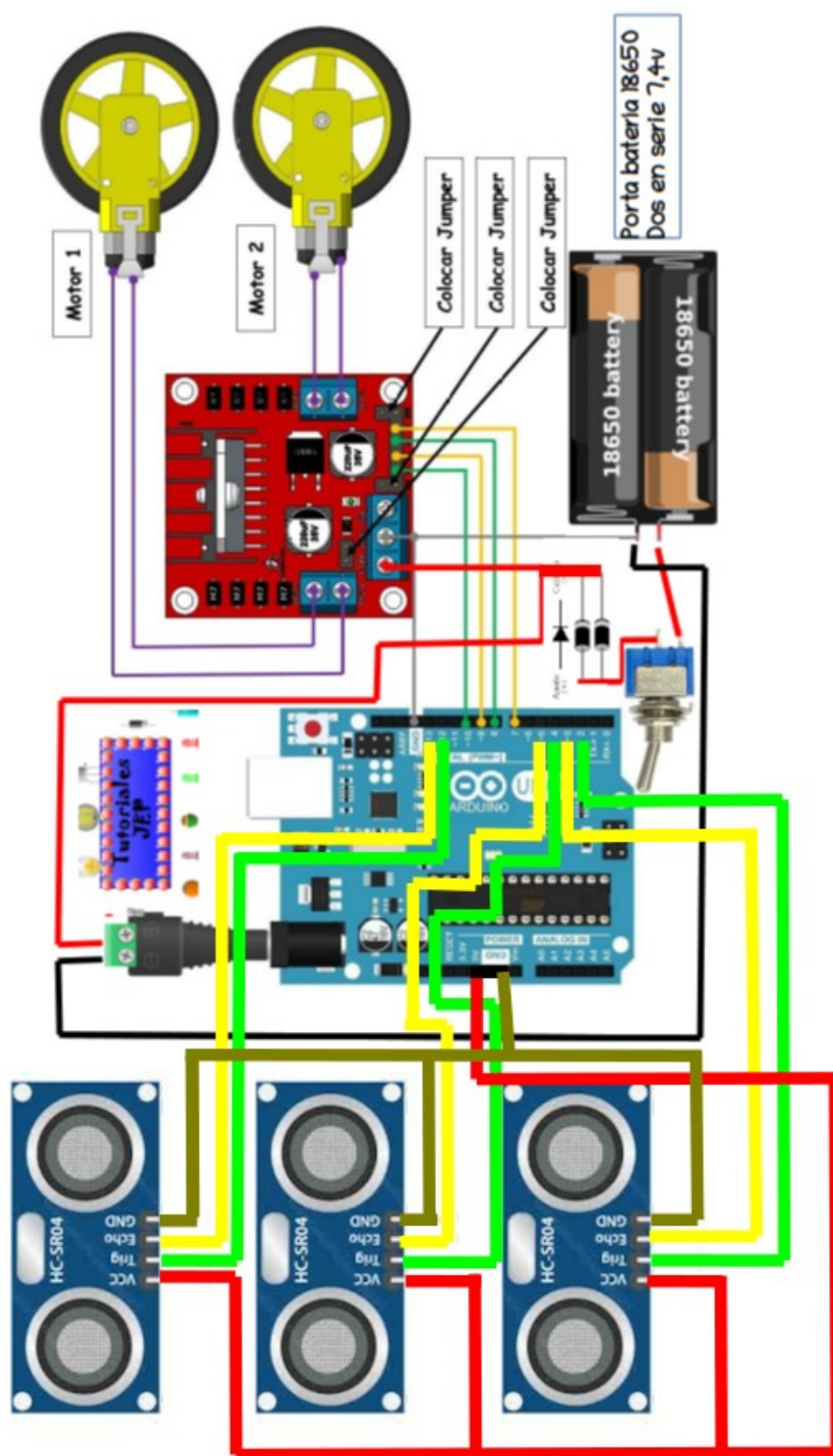
// si la distancia derecha es mayor a 10cm(no hay pared cerca) y la distancia izquierda es menor a 10cm ( hay pa-
red) y la distancia en frente es menor a 10cm( hay pared )
if((d_derecha>= 10)&&(d2_izquierda<= 10)&&(d3_frente<= 10)){

    //voy a la derecha con una rueda

}
// continuar la lógica
}
```

El deja al lector completar el programa en base a la lógica planteada.

Diagrama de conexión final





Manual de Robot Laberinto con Arduino

Ahora solo queda:

- Calibrar el programa en un cancha (de dimensiones según el reglamento).
- Distribución del peso en el chasis para obtener la mayor tracción.
- Adecuarse a las normas de la categoría en la que participe.

Podrán descargar los archivos referentes al chasis del Robot en el canal de youtube TUTORIALES JEP. También se encontraran los archivos STL para imprimir con la impresora 3d.

Robot_futbol_01: <https://youtu.be/j2guSepUVq8>



Manual de Robot Laberinto con Arduino

Bibliografía:

<https://www.arduino.cc/en/main/software>

<https://www.pololu.com/category/22/motors-and-gearboxes>

<http://www.superrobotica.com/S320103.htm>

<https://youtu.be/DdqFkJAV6l4>

Robot_futbol_01: <https://youtu.be/j2guSepUVq8>

<https://juegosrobotica.es/retos/reto-mbot-robot-laberinto/>