

MANUAL DE ARMADO Y
MANTENIMIENTO

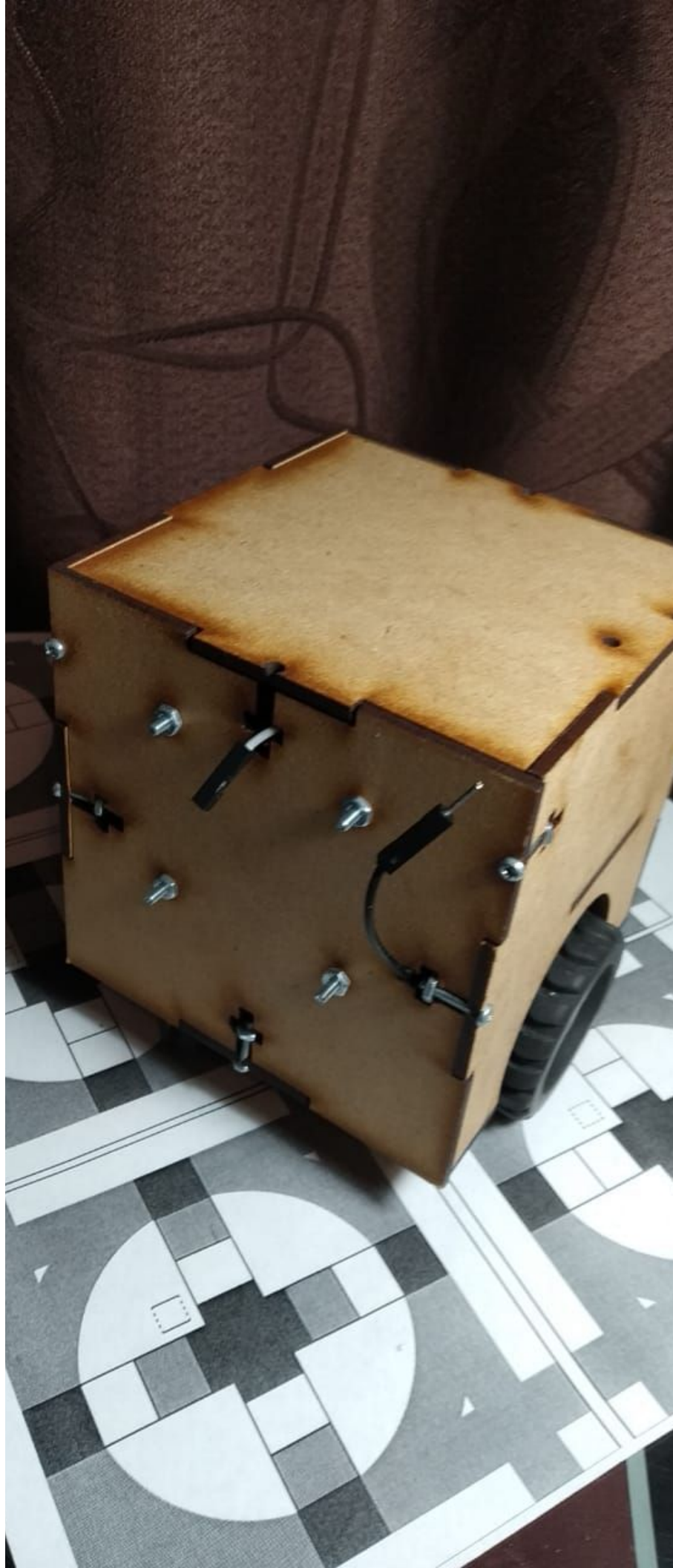
ROBOT 1.0

Nicolas Leonardo Maldonado Garzón

Grupo de Investigación ALIFE

Facultad de Ingeniería

Universidad Nacional de Colombia



Introducción

ESTE MANUAL DESCRIBE CÓMO REPRODUCIR , ARMAR Y REALIZAR EL MANTENIMIENTO DE UN ROBOT, PARTIENDO DE UN LISTADO DE COMPONENTES Y NOCIONES BÁSICAS EN ELECTRÓNICA.

EL MANUAL CUBRE DESDE EL PROCESO DE SOLDADURA DE ALGUNOS COMPONENTES ELECTRÓNICOS, HASTA LA INDICACIÓN DE CÓMO REALIZAR LA CONEXIÓN DE CADA UNO DE LOS MÓDULOS DE FORMA PARTICULAR. FINALMENTE EXPLICA FUNCIONAMIENTO DE UN PEQUEÑO PROGRAMA PARA UTILIZAR EL ROBOT EJERCENDO MOVIMIENTOS BÁSICOS A CONTROL REMOTO POR MEDIO DE COMANDOS BLUETOOTH.

ESTE DISEÑO ES OPCIONAL, ES POSIBLE HACER MODIFICACIONES SEGÚN SE CONSIDERE PERTINENTE.

IMPORTANTE:

LEA TODO EL MANUAL ANTES DE PROCEDER A ARMAR O UTILIZAR EL ROBOT.

Tabla de contenido

1. Lista de partes
2. Montaje
3. Uso y
Mantenimiento
4. Programas de
ejemplo.

1.

LISTA DE PARTES

ELEMENTOS ELECTRÓNICOS

1. **1 Microprocesador Arduino UNO**
<http://tdrobotica.co/uno-r3/705.html>
2. **1 Módulo Bluetooth HC-05**
<http://tdrobotica.co/bluetooth-modem-hc-05/72.html>
3. **1 Módulo Driver L298N**
<http://tdrobotica.co/modulo-driver-l298/544.html>
4. **1 Línea de 8 sensores QTR8RC**
<http://tdrobotica.co/sensor-de-linea-qtr-8rc-digital-arreglo-de-8/1288.html>
5. **2 Micromotores DC**
<http://tdrobotica.co/micromotor-hp-21013-kg-cm150-rpm/1241.html>
6. **1 Regulador de voltaje 7805**
7. **1 PCB universal de minimo 6 x 3**
8. **Regletas de pin macho**
9. **Jumpers macho-hembra y hembra-hembra.**
10. **1m Cable UTP**
11. **Batería recargable con una tensión entre 7 y 12V y una intensidad de corriente de 1A.**

ELEMENTOS ESTRUCTURALES

1. **2 Ruedas Pololu 42 x 19mm**
<http://tdrobotica.co/par-ruedas-negras-4219mm-econ/1172.html>
2. **2 Ruedas Ball Caster 3/8"**
<http://tdrobotica.co/ball-caster-38-plastico-pololu/262.html>
3. **2 Soportes para micromotor**
<http://tdrobotica.co/soporte-para-micromotor-plastico-blanco/1070.html>
4. **30 Tornillos M3 12mm**
5. **30 Tuercas hexagonales M3**
6. **1/8 Lamina de MDF de 3mm de grosor**
7. **1 Tira de Foamy negro de 3mm de grosor de 11 x 192 mm**

2.

MONTAJE

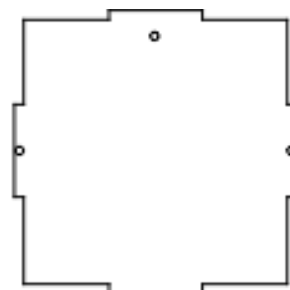
ARMAZÓN O CARCASA

Partes

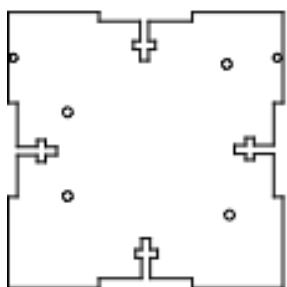
Para el caso de la carcasa, que sirve tanto de estructura del robot, como de soporte de todos los componentes internos, se usa el diseño incluido en este manual (Anexo 1)¹. Este diseño ya cuenta con las perforaciones y medidas necesarias para ajustar cada uno de los componentes de manera precisa.

El diseño de las 6 caras del robot y el soporte interno de la batería se debe cortar sobre 1/8 de MDF del grosor de 3mm utilizando una cortadora láser.

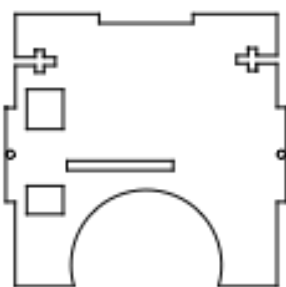
Como materias primas sustitutas para la carcasa pueden usarse láminas de cartón y/o acrílico, pero debido a las propiedades y resistencia del MDF, se recomienda cortar sobre este material.



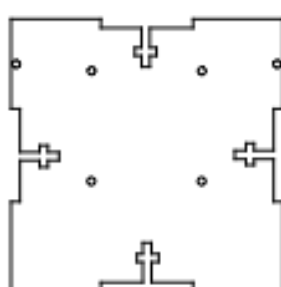
TAPA A
Superior



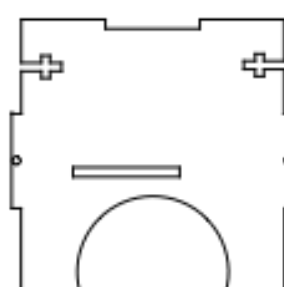
TAPA B
Arduino



TAPA C
Lateral izquierda



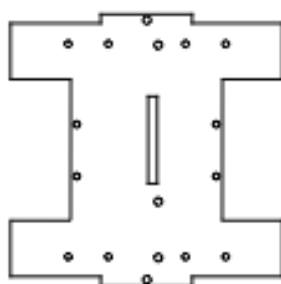
TAPA D
Módulo driver L298N



TAPA E
Lateral derecha



TAPA F
Batería



TAPA G
Sensores y ruedas

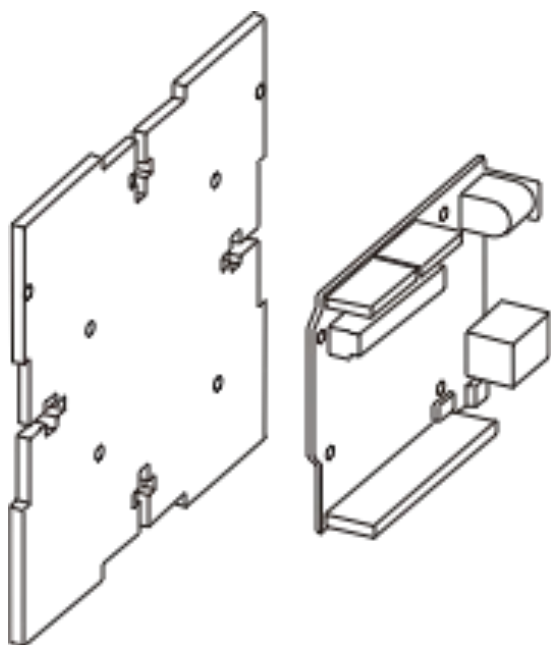
1. Diseño descargable en [OnShape](#)

2.

MONTAJE

ARMAZÓN O CARCASA

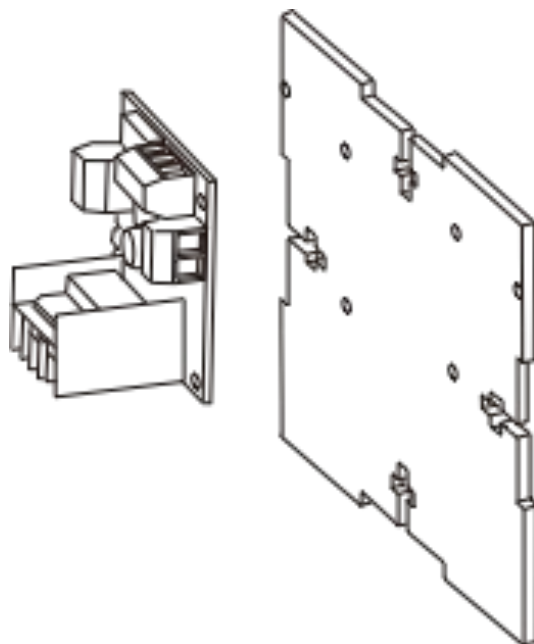
Ensamble sin cableado



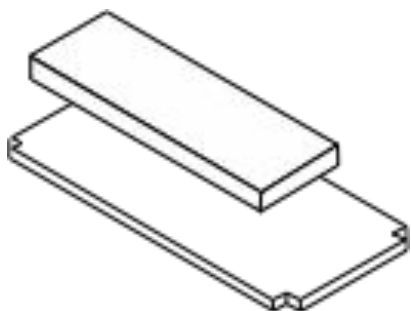
PASO 1: TAPA B + ARDUINO

Sujetar tanto el Arduino como el módulo L298N a las tapas B y C respectivamente, con el uso de 4 tornillos cada uno.

Sujetar la batería al soporte interno con el uso de elásticos o algún material que no afecte negativamente el estado de la misma.



PASO 2: TAPA D + MODULO L298N



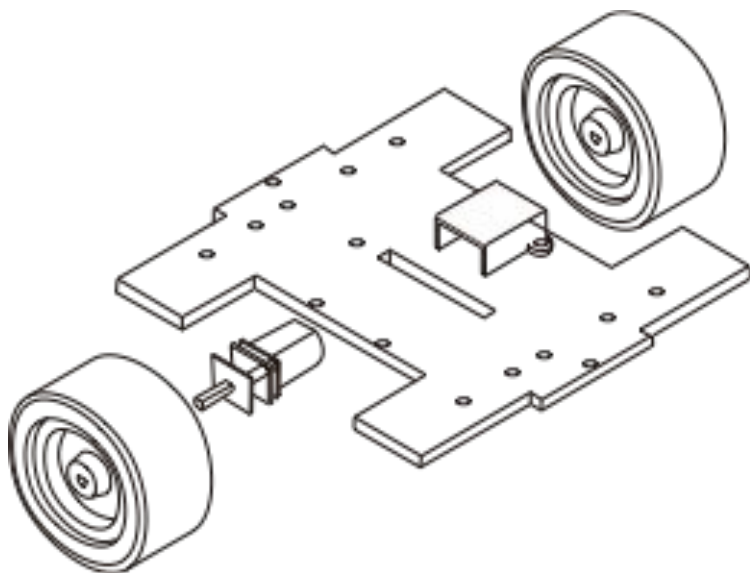
PASO 3: SOPORTE + BATERIA

2.

MONTAJE

ARMAZÓN O CARCASA

Ensamble sin cableado



PASO 4: TAPA G + RUEDAS + MOTOR

Soldar una línea de 11 pines macho en la tarjeta de los sensores de tal forma que el extremo largo quede en sentido contrario de la ubicación de los sensores.

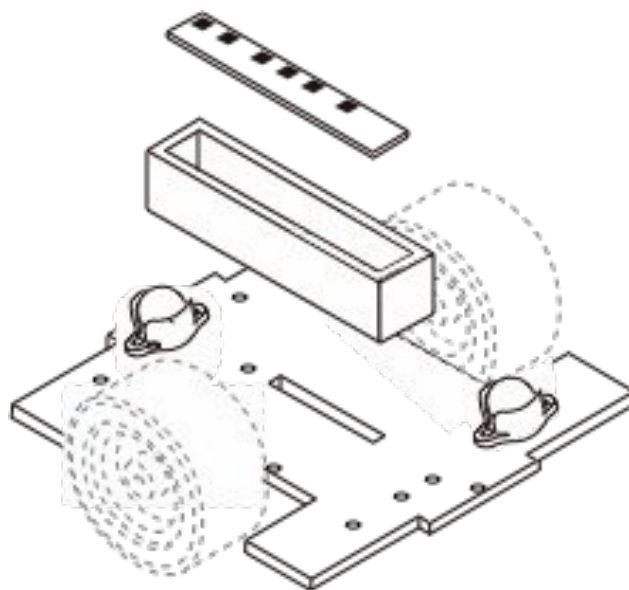
En la cara inferior, sujetar la línea de sensores con tres tornillos, posteriormente sujetar las ruedas locas de forma diagonal a la línea de sensores usando dos tornillos en cada una,

Colocar el protector de los sensores.

Ensamblar cada una de las ruedas con los motores y la cara superior de la tapa G, ajustarlos con los soportes para motor usando dos tornillos en cada uno.

Cortar la tira de foamy en 4 pedazos, 2 de 75mm y dos de 21mm de largo, pegar estos 4 pedazos formando un rectángulo, este posteriormente protegerá a los sensores de la luz ambiente.

La posición de las ruedas y el sensor es opcional.



PASO 5: TAPA G + RUEDAS LOCAS + SENSOR + PROTECTOR SENSOR

2.

MONTAJE

PINOUT

El diagrama del circuito corresponde al cableado necesario para alimentar con electricidad todos los circuitos y partes eléctricas del robot, la forma de conexión de cada pieza se especificará más adelante.

PINOUT Arduino UNO











Pines Digitales

- 2 - TX Bluetooth(Amarillo)
- 3 - RX Bluetooth(Morado)
- 4 - Activador Sensores(Cyan)
- 5 - IN1 Motores(Gris)
- 6 - IN 2 Motores(Gris)
- 7 - Sensor 7(Cafe)
- 9 - Sensor 1(Cafe)
- 10 - IN 3 Motores(Gris)
- 11 - IN 4 Motores(Gris)

Pines Análogos

- A0 - Sensor 1(Ocre)
- A1 - Sensor 3(Ocre)
- A2 - Sensor 4(Ocre)
- A3 - Sensor 5(Ocre)
- A4 - Sensor 6(Ocre)
- A5 - Sensor 8(Ocre)

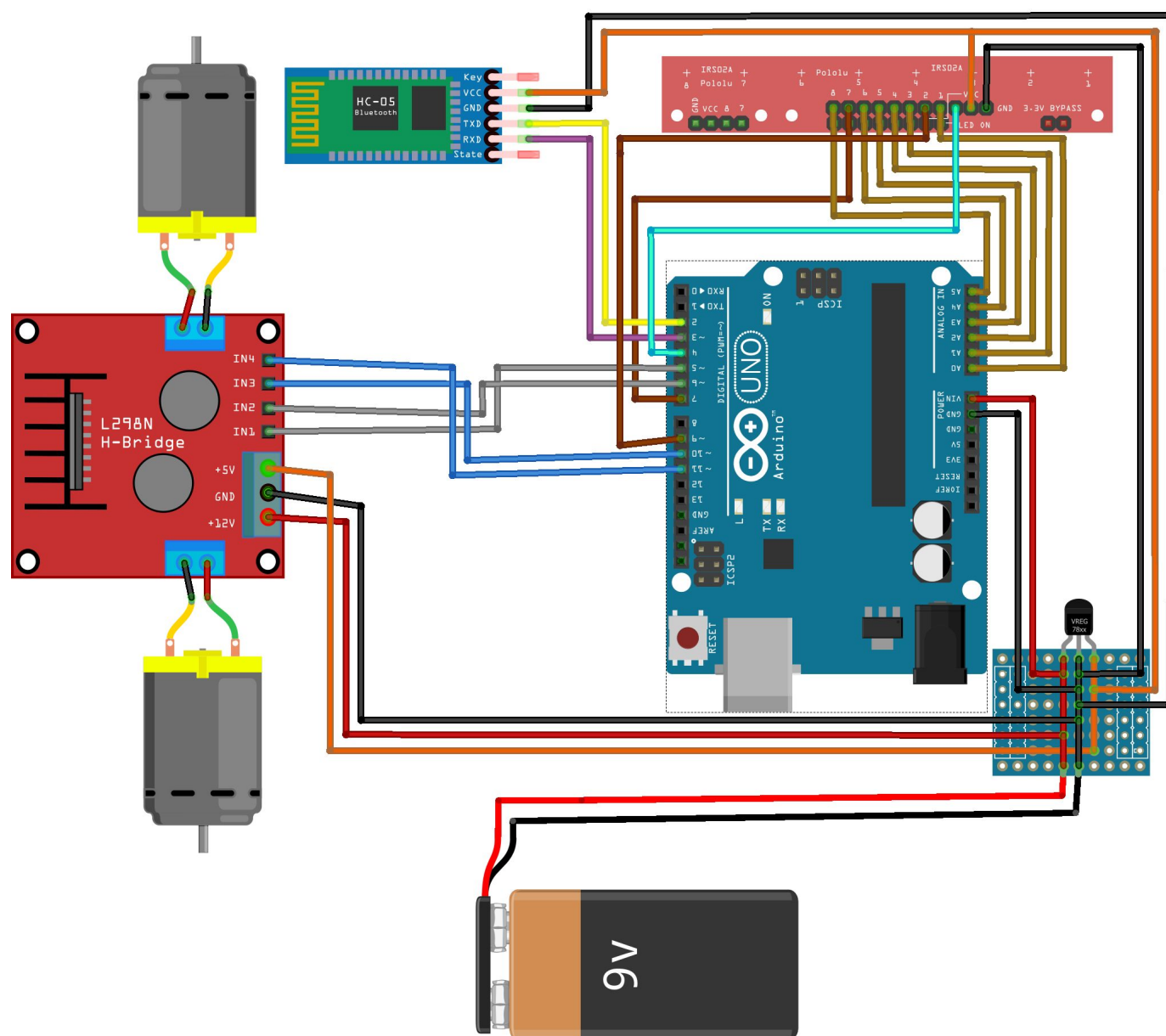
El diseño del PINOUT del conjunto de circuitos del robot cuenta con un sistema de colores para facilitar su comprensión.

	Neutro o tierra		Activador de los sensores
	9V		5V
	Señal de los sensores a los puertos análogos del arduin		Señal de los sensores a los puertos digitales del arduino
	Recepción de arduino a transmisión del módulo bluetooth		Transmisión de arduino a recepción del módulo bluetooth
	Activadores del motor 1(Ambas direcciones)		Activadores del motor 2(Ambas direcciones)

2.

MONTAJE

PINOUT

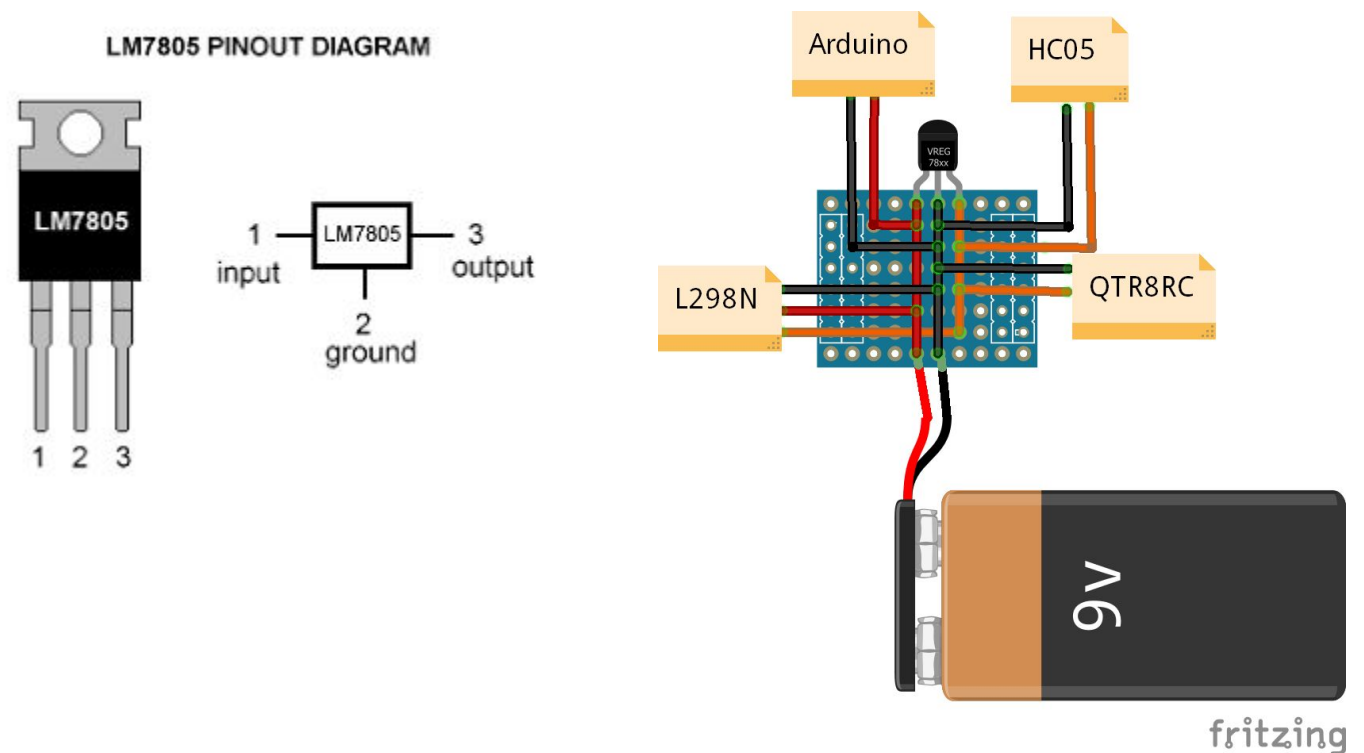


2.

MONTAJE

CIRCUITOS

Para empezar, se suelda el Regulador de Voltaje 7805 a la PCB universal, este regulador nos permitirá manejar fuentes de voltaje que varían entre los 6 y los 12 Voltios, manteniendo un flujo constante de 5V. A la entrada del regulador(primer pata desde la izquierda hacia la derecha) soldamos 3 pines macho, estos serán los utilizados para establecer una conexión con la fuente que proveerá de electricidad todo el circuito, la entrada Vin del Arduino y al Vcc del driver L298N de los motores. A la segunda pata del conversor(la central) se sueldan 5 pines macho, equivalentes a la entrada de GND desde la fuente, el GND del arduino, el GND del driver L298N, el GND de los sensores QTR8RC y el GND del módulo HC-05. Finalmente a la última pata se sueldan 3 pines macho para abastecer con 5V estables a la parte lógica del driver L298N y los Vcc de los sensores y el módulo HC-05.

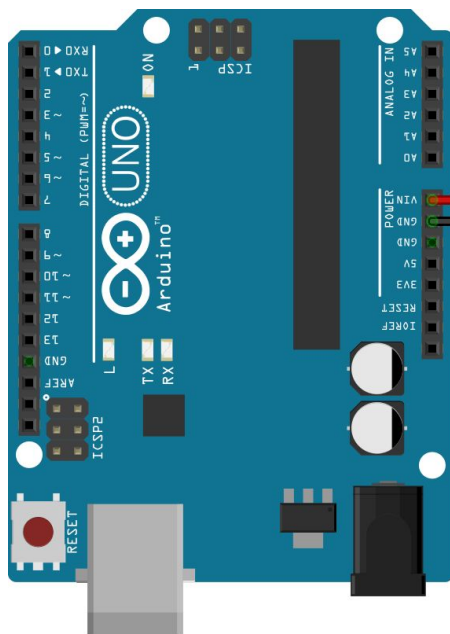
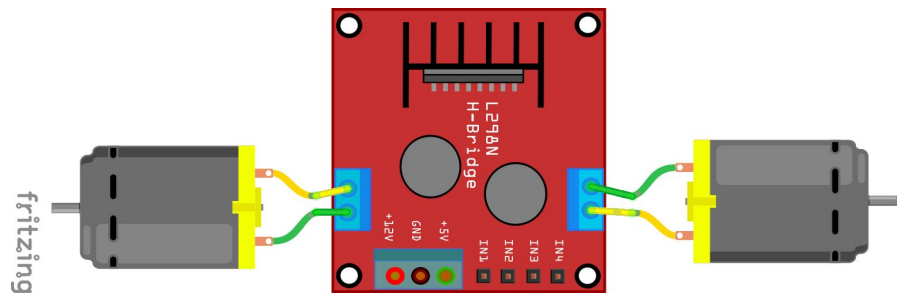


2.

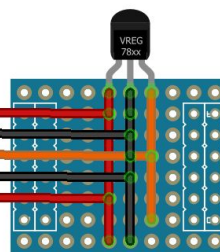
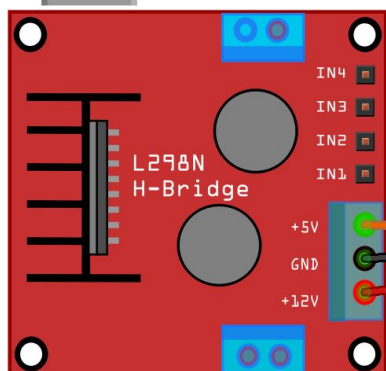
MONTAJE

CIRCUITOS

Se procede soldando dos cables a cada motor DC y ajustando el otro extremo de cada cable a los puertos del Driver L298N previamente sujetado a la cara OUT1 y OUT2 para el motor 1, y OUT3 y OUT4 para el motor 2.



Conectar el Arduino con la PCB utilizando dos jumpers macho hembra, el primero desde uno de los pines macho que conducen la potencia eléctrica entre 6 y 12V, hasta el puerto Vin del arduino y el segundo, desde uno de los pines que conducen tierra hasta uno de los piden GND del arduino.

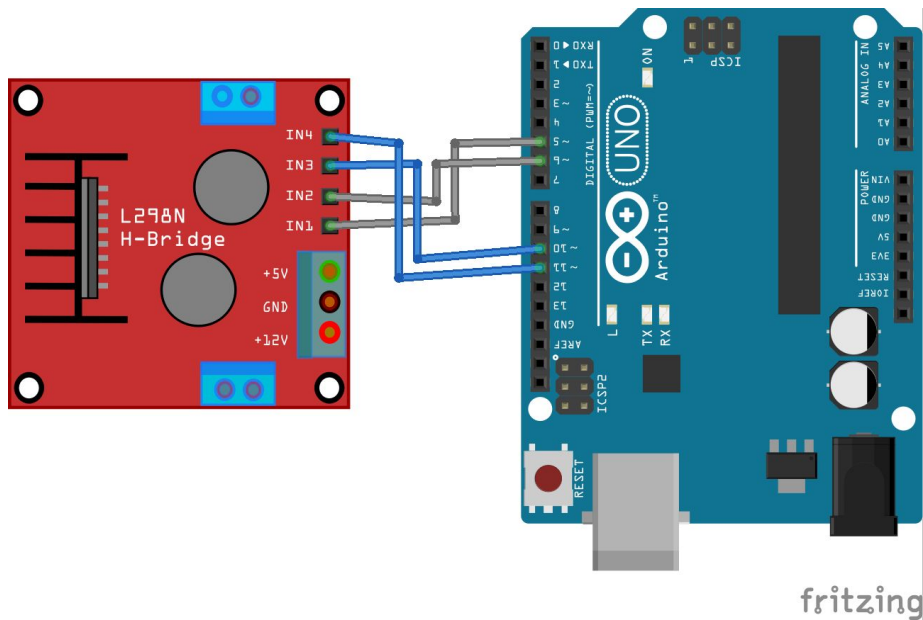


Cortar uno de los extremos hembra a tres jumpers hembra-hembra, con ellos, conectar el Driver L298N con la PCB, el primero desde uno de los pines macho que conducen la potencia eléctrica entre 6 y 12V, hasta el puerto Vcc del driver, el segundo desde uno de los pines que conducen tierra hasta uno de los piden GND del driver y el tercero desde uno de los pines a 5V hasta la entrada de 5V del driver.

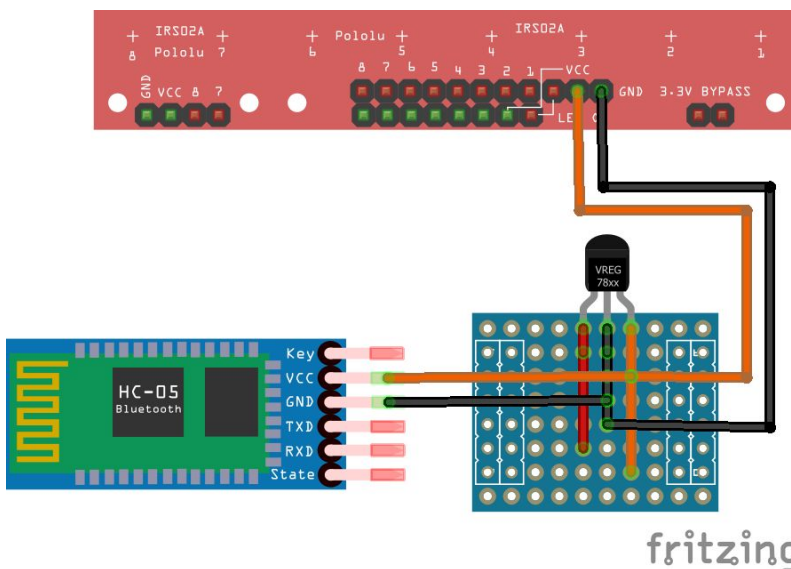
2.

MONTAJE

CIRCUITOS



Conectar el Arduino con el Driver L298N utilizando cuatro jumpers macho-hembra, estos deben ir uniendo los pines digitales 5, 6, 10 y 11 del arduino con los pines IN1, IN2, IN3 e IN4. Se utilizan estos pines debido a que son salidas PWM y de esta forma, graduar la diferencia de potencial que se quiere mandar hacia los motores.



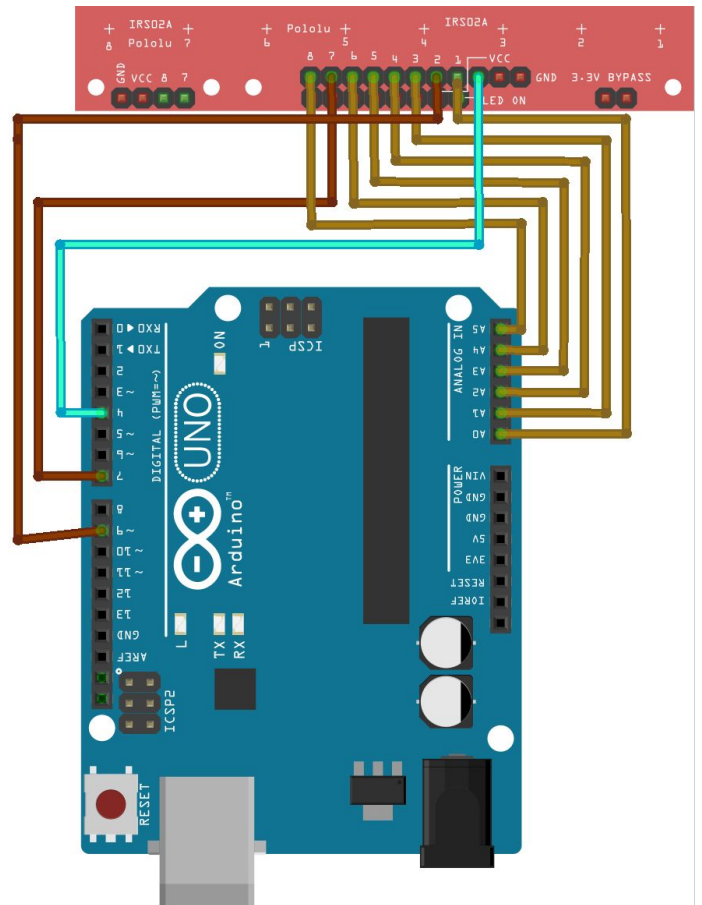
Conectar el Módulo Bluetooth HC-05 y la línea de sensores QTR-8RC utilizando cuatro jumpers hembra-hembra. Dos de estos deben ir desde los dos pines macho libres que conducen tierra, hasta los pines GND de ambos circuitos, y los otros dos cables utilizan los dos pines libres que conducen 5V hasta las entradas Vcc de las tarjetas.

2.

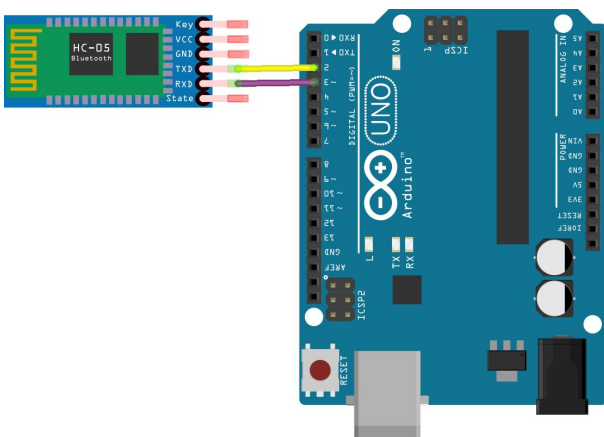
MONTAJE

CIRCUITOS

Conectar el Arduino con la línea de sensores utilizando 9 jumpers macho-hembra de la siguiente manera, 6 de los cables deben ir de las entradas análogas A0 a A5 a los pines de las salidas de los sensores 1, 3, 4, 5, 6 y 8 respectivamente, dos de los otros tres cables cubrirán los sensores 2 y 7, e irán conectados con los pines digitales 9 y 7 respectivamente. Finalmente, el último cable será el activador de los sensores y conectara el pin digital 4 con el único pin macho libre en la tarjeta de los sensores, el pin de entrada LED.



fritzing



fritzing

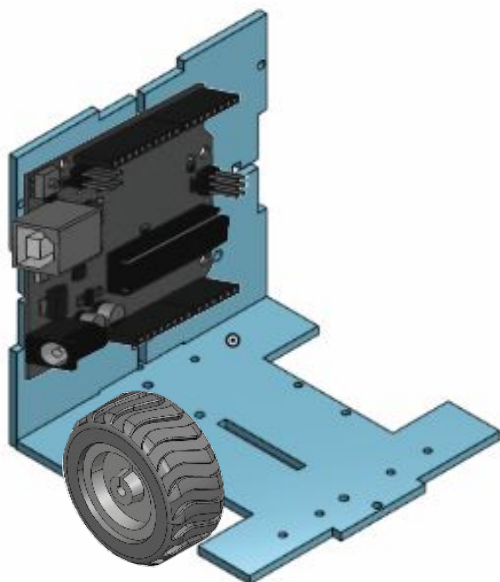
Conectar el Arduino con el Módulo Bluetooth HC-05, utilizando dos jumpers macho-hembra, estos deben ir uniendo los pines digitales 2 y 3 del arduino con los pines de Transmisión(TX) y Recepción(RX) del módulo respectivamente. En estos pines se realiza la configuración del segundo serial del Arduino para poder mantener el serial por defecto libre y poder seguir comunicándonos con el, aun mientras este se comunica con el módulo bluetooth.

2.

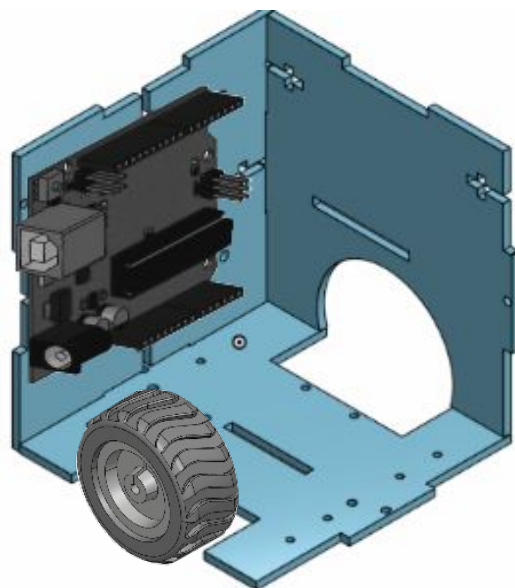
MONTAJE

ARMAZÓN O CARCASA

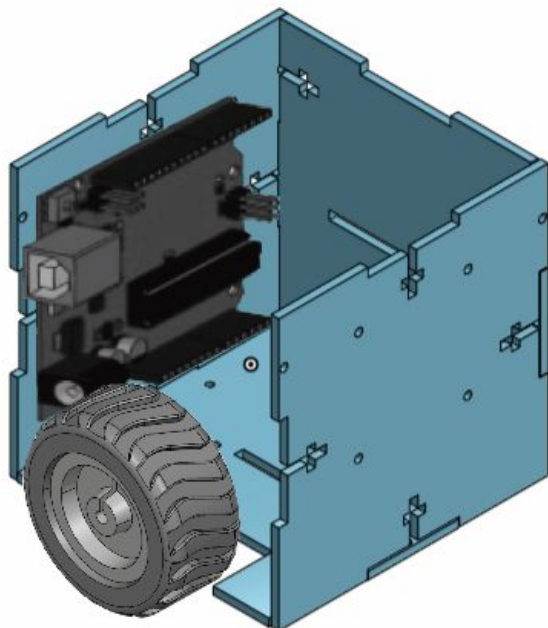
Ensamble final



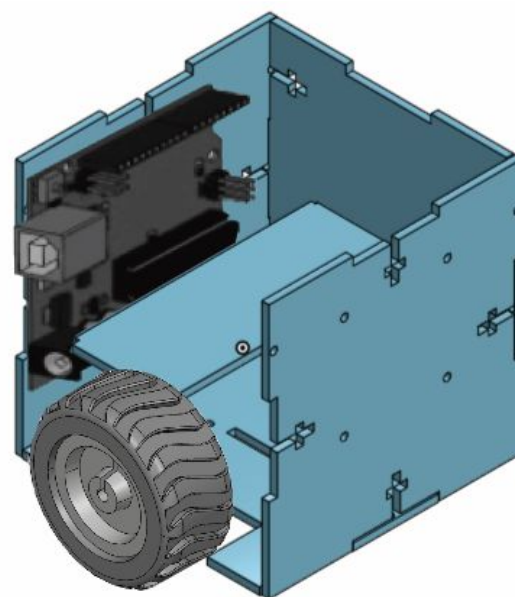
TAPA B + TAPA G



+ TAPA C



+ TAPA D



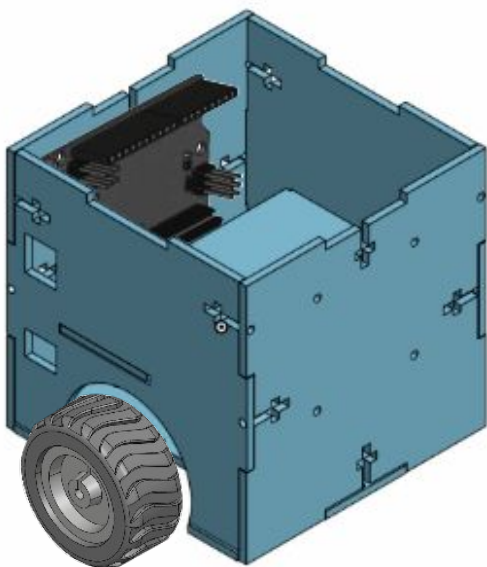
+ TAPA F

2.

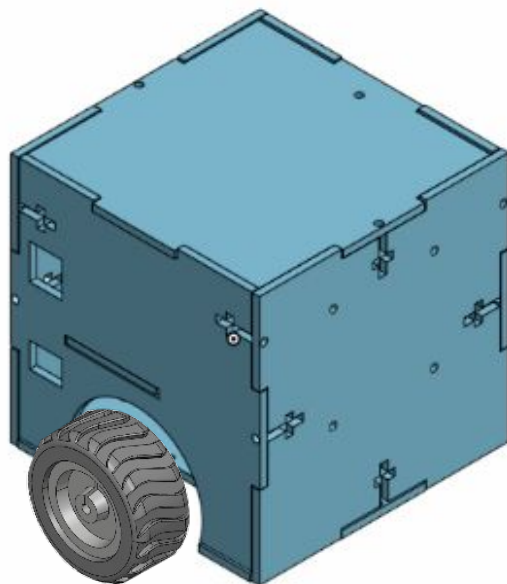
MONTAJE

ARMAZÓN O CARCASA

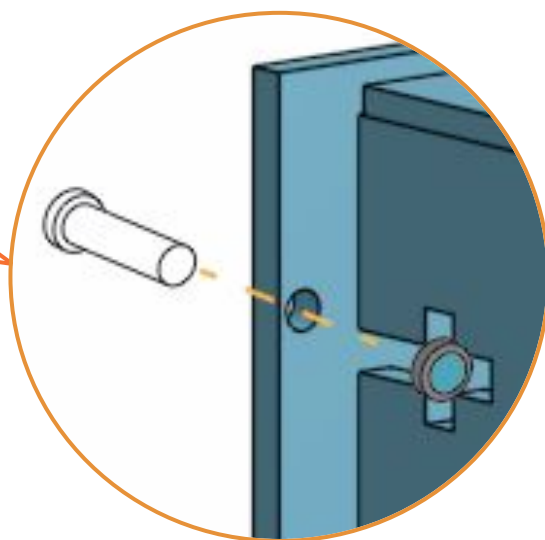
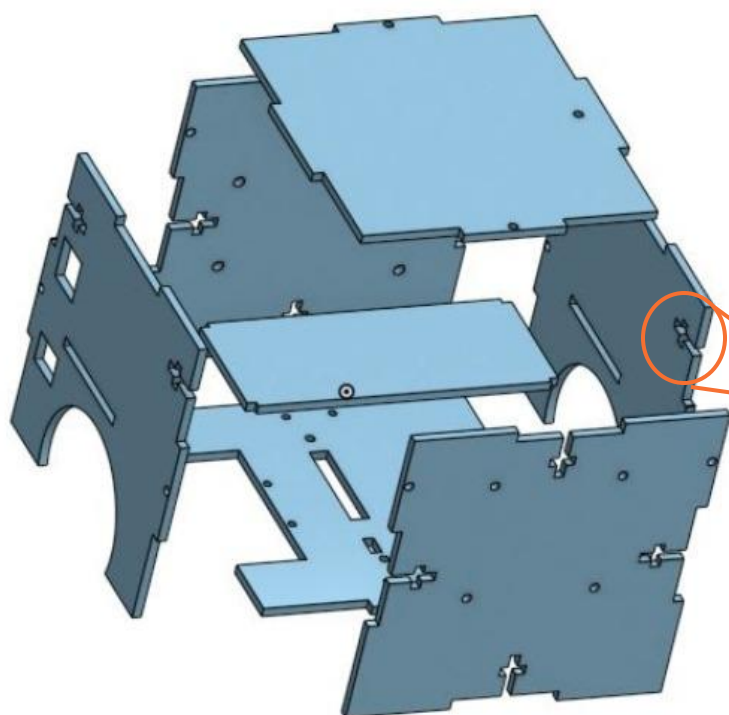
Ensamble final



+ TAPA E



AÑADIR TAPA A



3.

USO Y MANTENIMIENTO

CÓDIGO

Driver L298N y Motores

Ya que el Arduino UNO como microcontrolador debe ser programado, en esta sección se describen los segmentos de código arduino y procedimientos necesarios que se deben tener en cuenta para utilizar el driver de las llantas,² el módulo bluetooth³ y los sensores infrarrojos.⁴

DRIVER L298N

Para hacer uso de este driver es necesario declarar como salida los pines que enviarán los pulsos hacia las entradas de los motores.

```
int IN1=5;           pinMode(IN1,OUTPUT);
int IN2=6;           pinMode(IN2,OUTPUT);
int IN3=10;          pinMode(IN3,OUTPUT);
int IN4=11;          pinMode(IN4,OUTPUT);
```

Las 4 entradas del driver están configuradas de tal manera que si se enciende alguna genera un tipo de movimiento en alguno de los dos motores. Por ejemplo, al encender el pin IN1, el driver encenderá el motor 1 hacia adelante pero al encender el IN2, el driver encenderá el mismo motor 1 hacia atrás. Es muy importante tener esto en cuenta ya que si se encienden los pines IN1 e IN2 al tiempo, el driver hará corto.

El driver funciona de igual manera para encender el motor 2 pero utilizando los pines IN3 e IN4, debido a la estructura que tiene el robot y como los motores se encuentran en contraposición el uno del otro, para que el robot avance en una misma dirección es necesario que uno de los motores gire hacia adelante y el otro hacia atrás, por ende a continuación se describen las 4 configuraciones en código que se utilizarían para mover el robot.

-
2. [Aquí](#) puede encontrar un código para realizar prueba sobre los motores
 3. [Aquí](#) puede encontrar un código para realizar prueba sobre el módulo WIFI
 4. [Aquí](#) puede encontrar un código para realizar prueba sobre los sensores

3.

USO Y MANTENIMIENTO

CÓDIGO

Driver L298N y Motores

Para Avanzar

```
digitalWrite(IN1,LOW);  
digitalWrite(IN2,LOW);  
digitalWrite(IN3,LOW);  
digitalWrite(IN4,LOW);  
analogWrite(IN1, 0);  
analogWrite(IN2, value_of_speed);  
analogWrite(IN3, value_of_speed);  
analogWrite(IN4, 0);
```

Para Retroceder

```
digitalWrite(IN1,LOW);  
digitalWrite(IN2,LOW);  
digitalWrite(IN3,LOW);  
digitalWrite(IN4,LOW);  
analogWrite(IN1, value_of_speed);  
analogWrite(IN2, 0);  
analogWrite(IN3, 0);  
analogWrite(IN4, value_of_speed);
```

Para Girar a la Derecha

```
digitalWrite(IN1,LOW);  
digitalWrite(IN2,LOW);  
digitalWrite(IN3,LOW);  
digitalWrite(IN4,LOW);  
analogWrite(IN1, 0);  
analogWrite(IN2, value_of_speed);  
analogWrite(IN3, 0);  
analogWrite(IN4, value_of_speed);
```

Para Girar a la Izquierda

```
digitalWrite(IN1,LOW);  
digitalWrite(IN2,LOW);  
digitalWrite(IN3,LOW);  
digitalWrite(IN4,LOW);  
analogWrite(IN1, value_of_speed);  
analogWrite(IN2, 0);  
analogWrite(IN3, value_of_speed);  
analogWrite(IN4, 0);
```

La variable `value_of_speed` es la encargada de definir la velocidad a la cual se quiere que giren los motores, este valor debe ser predeterminado por el usuario dependiendo de la funcionalidad que le quiera dar al robot. Esta es la razón por la cual se seleccionaron los pines PWM del arduino, ya que nos pueden otorgar una salida análoga que puede variar entre 0 y 255, donde 0 es tierra y 255 la potencia máxima.

3.

USO Y MANTENIMIENTO

CÓDIGO

Módulo Bluetooth HC-05

MÓDULO BLUETOOTH HC-05

Para usar el módulo HC-05 es necesario establecer comunicación serial entre el arduino y dicho módulo, por ende, para no ocupar el serial por defecto del arduino hacemos uso de la librería `SoftwareSerial`.

```
#include <SoftwareSerial.h>

SoftwareSerial modulo(2,3);

modulo.begin(9600);
```

De esta manera se define que el receptor de este nuevo puerto serial será el pin digital 2 y el transmisor será el pin digital 3. Se inicializa el canal de comunicación con el módulo sobre una velocidad de 9600 baudio(esta es usualmente la frecuencia más estable). Para poder establecer la comunicación con el Arduino basta con hacer uso de estos dos métodos:

Recepción de mensajes provenientes del módulo.

```
String receivedString = modulo.readString();
```

Envío de datos desde el arduino hacia el módulo.

```
modulo.println(stringToSend);
```

Se debe tener en cuenta que para poder establecer conexión con el módulo es necesario un dispositivo que tenga bluetooth y una herramienta que permita la comunicación serial por los puertos COM. Para probar esta conexión se recomienda usar un PC con bluetooth como dispositivo y una herramienta como PuTTY.

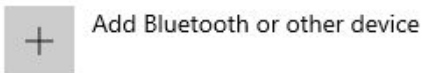
3.

USO Y MANTENIMIENTO

CÓDIGO

Módulo Bluetooth HC-05

Bluetooth & other devices



Bluetooth



Now discoverable as "DESKTOP-BT5GKLH"

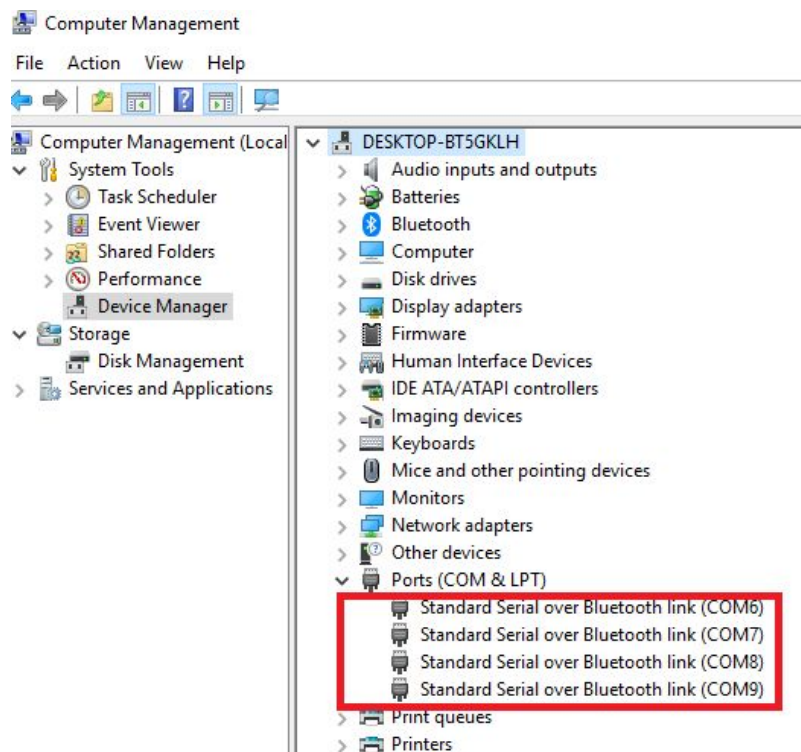
Other devices



El siguiente paso consiste en buscar los números de los puertos COM que están asignados al bluetooth, para este paso (desde el SO Windows 10) es necesario acceder al administrador de dispositivos.

Una vez se tenga la lista con los números de los puertos, se debe probar puerto por puerto a cual está accediendo el módulo bluetooth. Para esta prueba utilizamos la herramienta de comunicación como PuTTY.

Primero se debe establecer conexión con el módulo HC-05, para ello, conecte el montaje a la batería. Encender el bluetooth del computador y establecer conexión con el módulo.



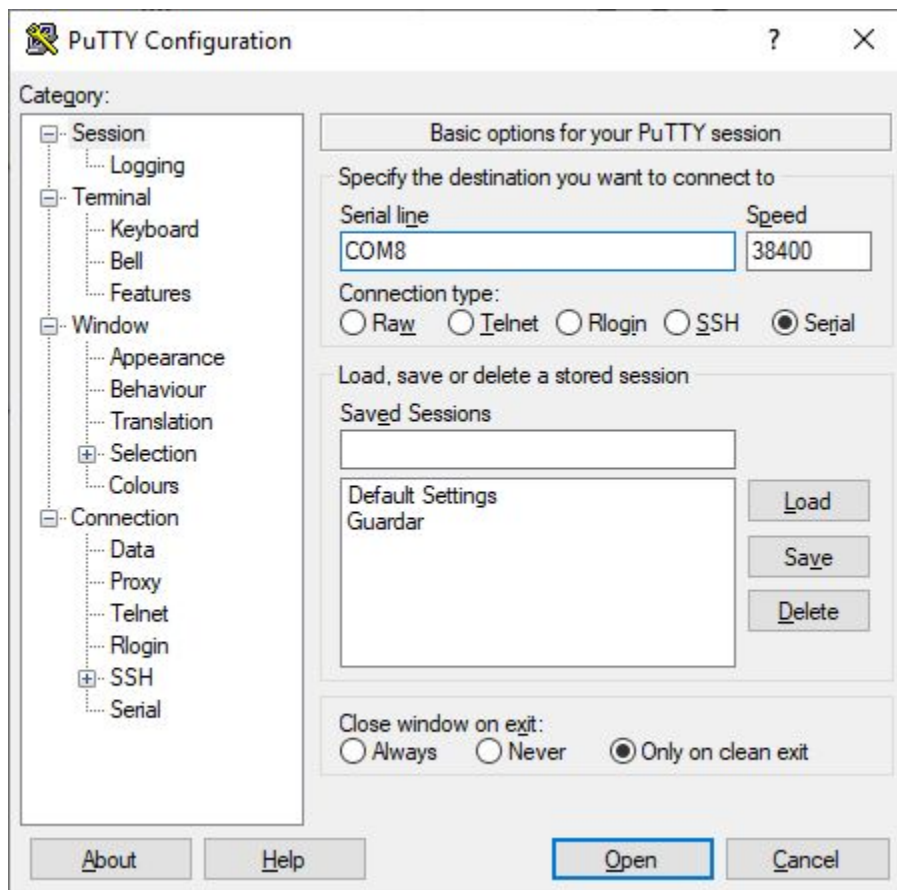
3.

USO Y MANTENIMIENTO

CÓDIGO

Módulo Bluetooth HC-05

Se ajusta la velocidad de la frecuencia en 38400 baudio que es la frecuencia más estable sobre la que trabaja este tipo de módulos y se varían los puertos hasta que establezca la conexión. El bombillo del módulo HC-05 parpadea rápidamente desde el momento en que el módulo se enciende, y lentamente al momento en que establece conexión con otro dispositivo.



3.

USO Y MANTENIMIENTO

CÓDIGO

Arreglo de Sensores
QTR-8RC

LINEA DE SENSORES QTR-8RC

Para usar la línea de sensores QTR-8RC es necesario utilizar la librería `QTRSensors` y definir los pines que servirán como entrada de los datos que tomen los sensores y el activador de los sensores.

```
#include <QTRSensors.h>

#define NUM_SENSORS    8
#define TIMEOUT        2500
#define EMITTER_PIN    4

QTRSensorsRC qtrrc((unsigned char[]) {A0,9,A1,A2,A3,A4,7,A5},
    NUM_SENSORS, TIMEOUT, EMITTER_PIN);
unsigned int sensorValues[NUM_SENSORS];
```

Se definen tres constantes, la primera hace referencia a la cantidad de sensores que se usarán, la segunda describe el tiempo del ciclo de lectura de los sensores en milisegundos y la tercera es el pin que activará la lectura de los sensores. Luego se inicializa el arreglo de sensores definiendo los puertos sobre los cuales estarán las entradas de información en el arduino, para esto utilizamos las 6 entradas análogas(A0-A5) y 2 digitales(7 y 9).

```
delay(500);
for (int i = 0; i < 400; i++)
{
    qtrrc.calibrate();
}
```

Dentro del setup de nuestro programa calibramos la línea de sensores, esto le permite a cada sensor adaptarse a la intensidad lumínica a la que está expuesto.

3.

USO Y MANTENIMIENTO

CÓDIGO

Arreglo de Sensores QTR-8RC

```
Serial.begin(9600);
```

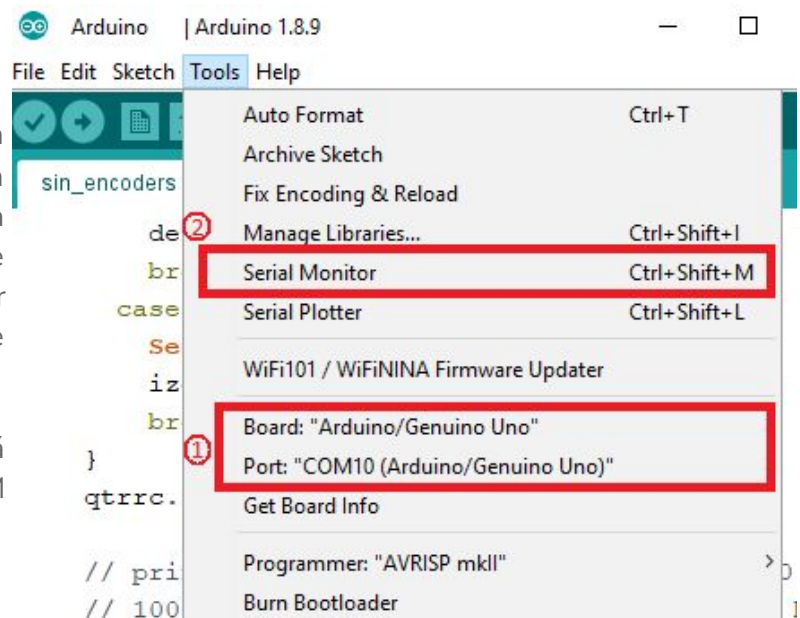
Inicializamos el puerto serial del arduino para poder verificar que sobre que información se inicializaron los sensores. Recorremos el arreglo de sensores y solicitamos los valores de las lecturas mínimas y máximas que obtuvieron los sensores en su proceso de calibración.

```
for (int i = 0; i < NUM_SENSORS; i++)
{
    Serial.print(qtrrc.calibratedMinimumOn[i]); Valor mínimo de calibración
    Serial.print(' ');
}
Serial.println();

for (int i = 0; i < NUM_SENSORS; i++)
{
    Serial.print(qtrrc.calibratedMaximumOn[i]); Valor máximo de calibración
    Serial.print(' ');
}
Serial.println();
```

Estos valores nos otorgan el rango de medición sobre el cual funcionarán los sensores. Para poder verificar esta información, enviamos dicha información desde nuestro arduino por medio de su puerto serial principal, si se quiere visualizar esto, es necesario ingresar al monitor serial de nuestro arduino, para ellos es necesario:

1. Seleccionar el tipo de tarjeta que se está utilizando y verificar a qué puerto COM está asignado nuestro arduino,
2. Abrir el Monitor Serial.

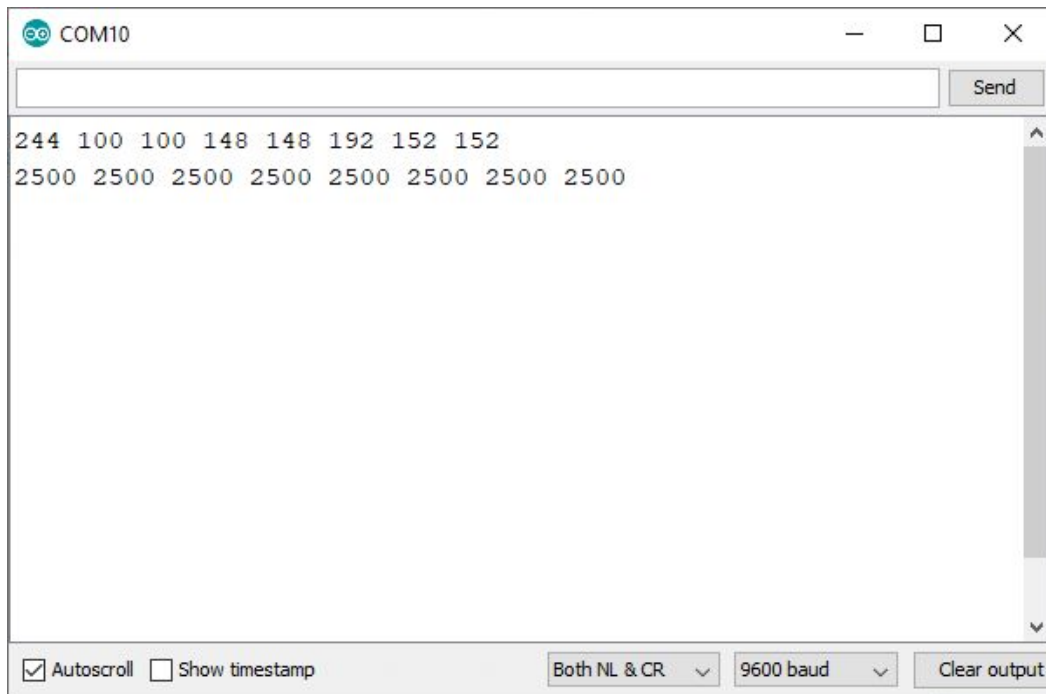


3.

USO Y MANTENIMIENTO

CÓDIGO

Arreglo de Sensores
QTR-8RC



Al encender el arduino, se puede observar desde el monitor serial los valores sobre los que está calibrado cada sensor. Se recomienda hacer uso de una base que tenga una escala de grises para que los sensores puedan identificar un rango muy amplio(Anexo 2).

Una vez calibrados los sensores están listos para usarse, para ellos basta con utilizar la función

```
qtrrc.readLine(sensorValues);
```

Esta función actualiza el valor de cada sensor, así que para visualizarlo podemos realizar un recorrido sobre el arreglo de sensores e imprimir dicho resultado en el monitor serial.

```
cadena = "";  
for ( unsigned char i = 0; i < NUM_SENSORS; i++ ){  
    Serial.print(sensorValues[i]);  
    Serial.print('\t');  
}  
Serial.println();
```

4.

PROGRAMAS DE EJEMPLO

CÓDIGO

ROBOT A CONTROL REMOTO

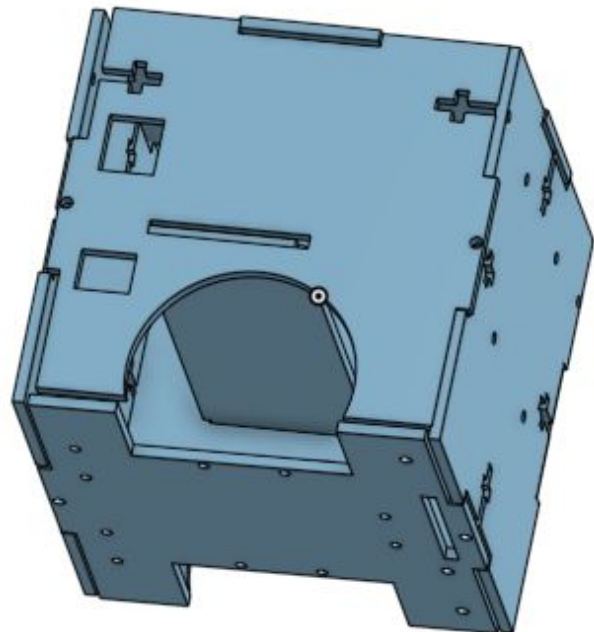
Entre la gran cantidad de posibles usos que podemos encontrar para este tipo de robots, encontramos el de un robot a control remoto, cabe notar que este ejemplo no hace uso de los sensores ubicados en la parte inferior del robot.

El código de este programa lo puede descargar [aquí](#).

ROBOT SEGUIDOR DE LINEA

Para poder implementar dicho programa, es necesario realizar un cambio en la tapa inferior del robot, esto con el fin de colocar el arreglo de sensores en sentido perpendicular al de las ruedas impulsadas por los motores. Este diseño lo podrá encontrar para ser cortado en el anexo 3. Para implementar este programa es innecesario el uso del módulo bluetooth.

El código de este programa lo puede descargar [aquí](#).



SE LE RECUERDA AL LECTOR DE ESTE MANUAL QUE LOS DISEÑOS Y CONEXIONES AQUÍ EXPLICADOS, FUERON SELECCIONADOS ENTRE OTROS MODELOS Y POR SER LOS QUE MÁS SE AJUSTABAN A LAS NECESIDADES DEL PROYECTO. SI EL LECTOR ASÍ LO DESEA, PUEDE DISPONER DE LOS DISEÑOS Y REALIZAR LAS MODIFICACIONES CONSIDERE NECESARIAS Y LE PERMITAN ALCANZAR SUS OBJETIVOS INDIVIDUALES DE UNA FORMA MÁS EFICAZ.

ADICIONALMENTE SE LE EXTIENDE UNA INVITACIÓN AL LECTOR PARA CONTINUAR COMPLEMENTANDO ESTE MANUAL CON POSIBLES APLICACIONES PARA ESTE TIPO DE ROBOTS.

