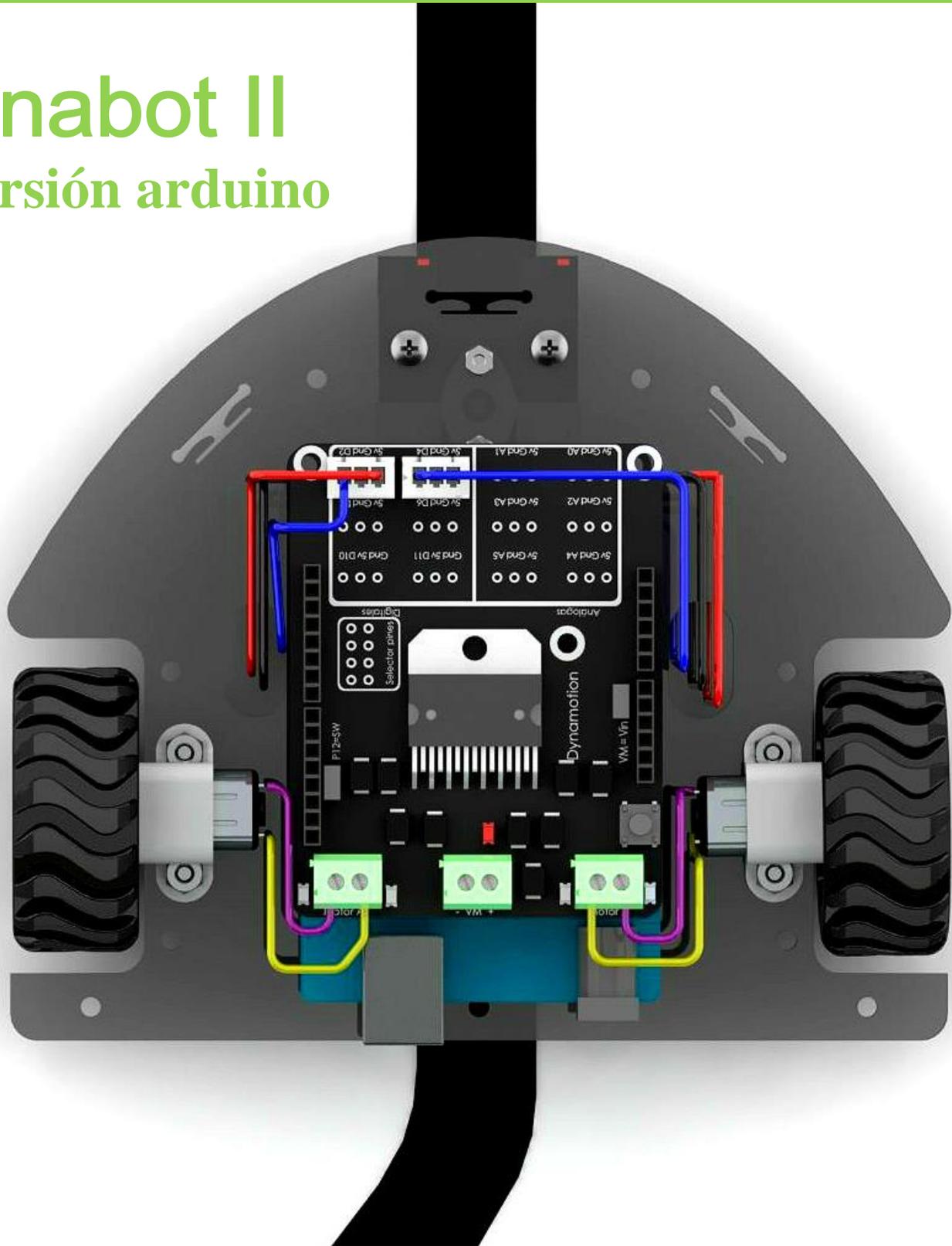


Dynabot II

Versión arduino



Descripción

Dynabot II es un kit diseñado para el aprendizaje de robótica móvil basado en arduino .

Para qué sirve?

Sirve para programar aplicaciones básicas y avanzadas en robótica móvil, entre las practicas que se pueden desarrollar están

- Robot que realice movimientos con figuras geométricas
- Robot seguidor de línea ON /OFF
- Robot seguidor de línea PID
- Robot Resolver laberintos
- Robot Explorador

Ventajas

Modularidad: es un sistema modular que permite integrar otros elementos como sensores de detección de objetos, sensores de posición, sensores inerciales, sensores de ultrasonido etc.

Sistema de control: Al tener arduino como sistema de control, los usuarios pueden aprender a implementar programaciones desde **lenguajes gráficos como Scraths**, hasta lenguaje escrito como Arduino

Expansibilidad: si se requiere más espacio o un sistema más robusto esta plataforma puede expandirse a dos niveles o también a formar un vehículo de tracción 4x4 usando dos plataformas

Partes:



Plataforma
(1)



Dynamotion
(1)



Sensores de línea
(2)



Arduino UNO
(1)



Cable arduino
(1)

Ensamblaje

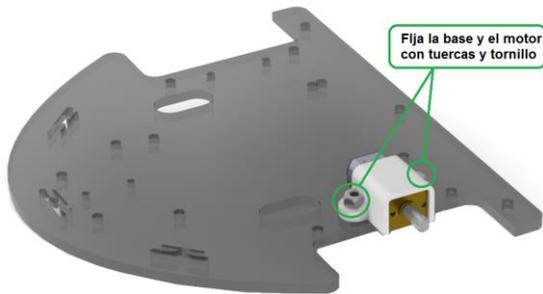


Figura 1

Ubicar el motor con las bases blancas para sujetarlo a la plataforma usando tornillos, los tornillos se ubican desde abajo y las tuercas quedan en la parte superior (figura 1)

Hacer lo mismo con el otro motor, cuando se ubiquen ambos debe quedar como se muestra en la figura 2

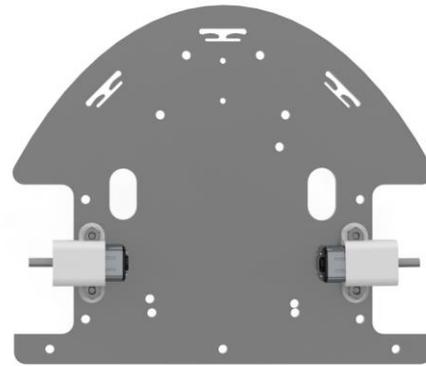


Figura 2

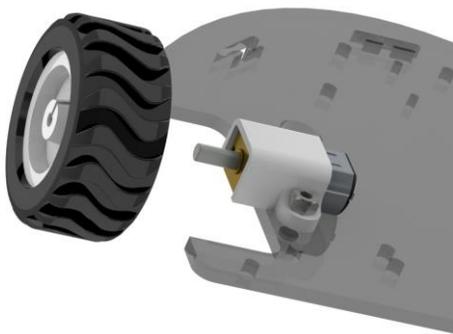


Figura 3

Poner la rueda en el motor, estas entran a presión, tenga en cuenta que la parte plana del eje del motor y de la rueda coincidan (figura 3)



Al hacer presión para ubicar la rueda hágalo con cuidado, no es necesario en todos los casos que entre hasta el fondo, sin embargo verifique que no quede suelta



Se ubica la rueda de apoyo usando los dos accesorios que incluye para darle más altura, se atornilla a la plataforma poniendo los tornillos desde abajo (figura 4)

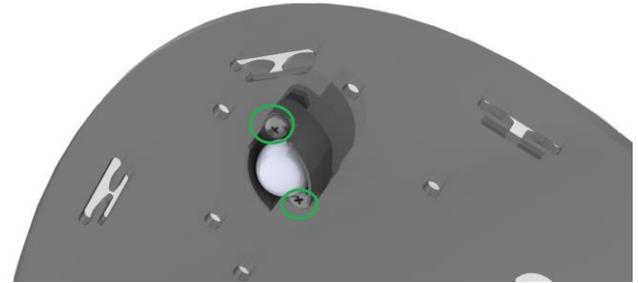


Figura 4



Figura 5

Una vez instalada la rueda de apoyo la plataforma esta lista, para iniciar a integrar la parte electrónica (figura 5)

Se colocan las bases donde se va a poner la board arduino UNO, se usan tres bases hexagonales (Figura 6)

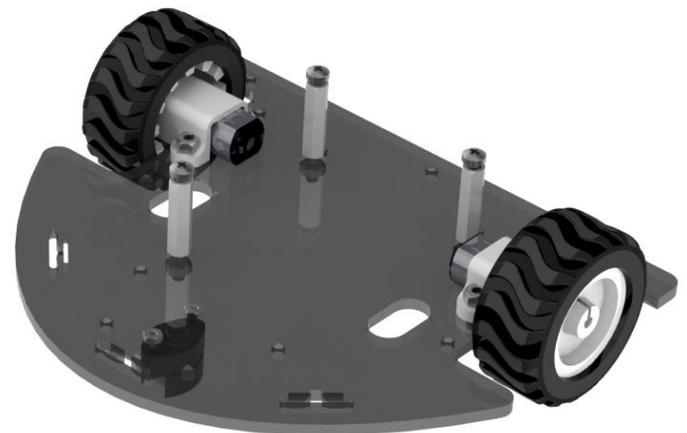


Figura 6

Se ubica la board arduino uno como muestra la figura 7, y se usan tornillos para fijarla a las bases hexagonales

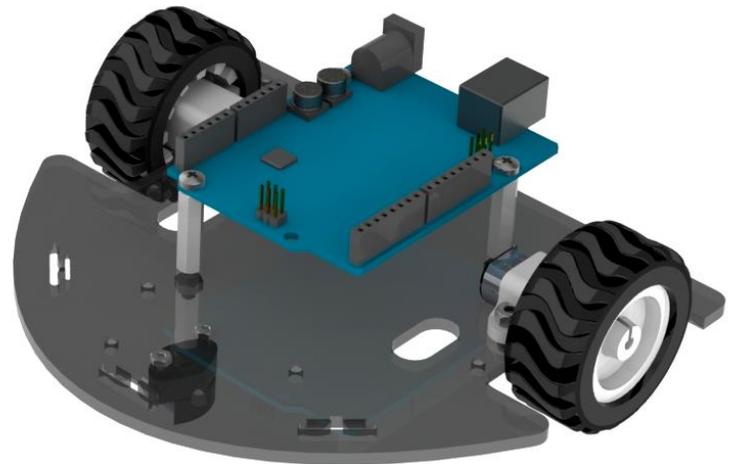


Figura 7

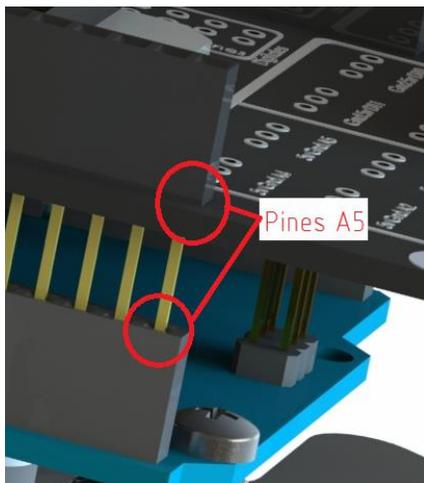


Figura 8

Se ubica la board dynamotion como una shield encima de la board arduino uno, la forma correcta de hacerlo es que las regletas coincidan en el pin A5

Una vez puesta la shield debe verse como se muestra en la figura 9

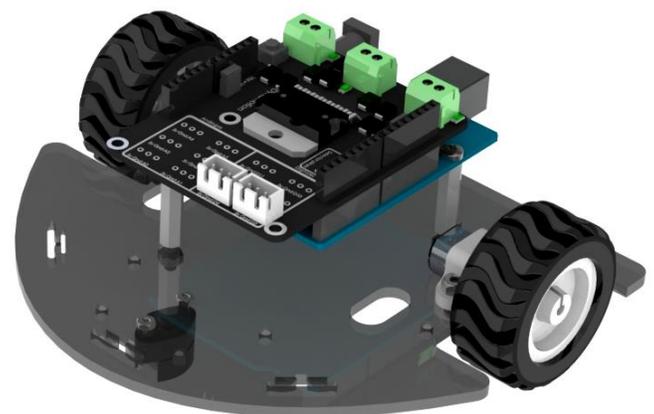


Figura 9

Se colocan los sensores usando los espaciadores y los tornillos como muestra la figura 10

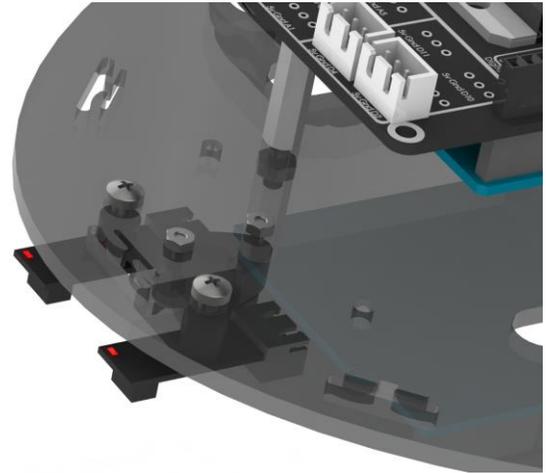


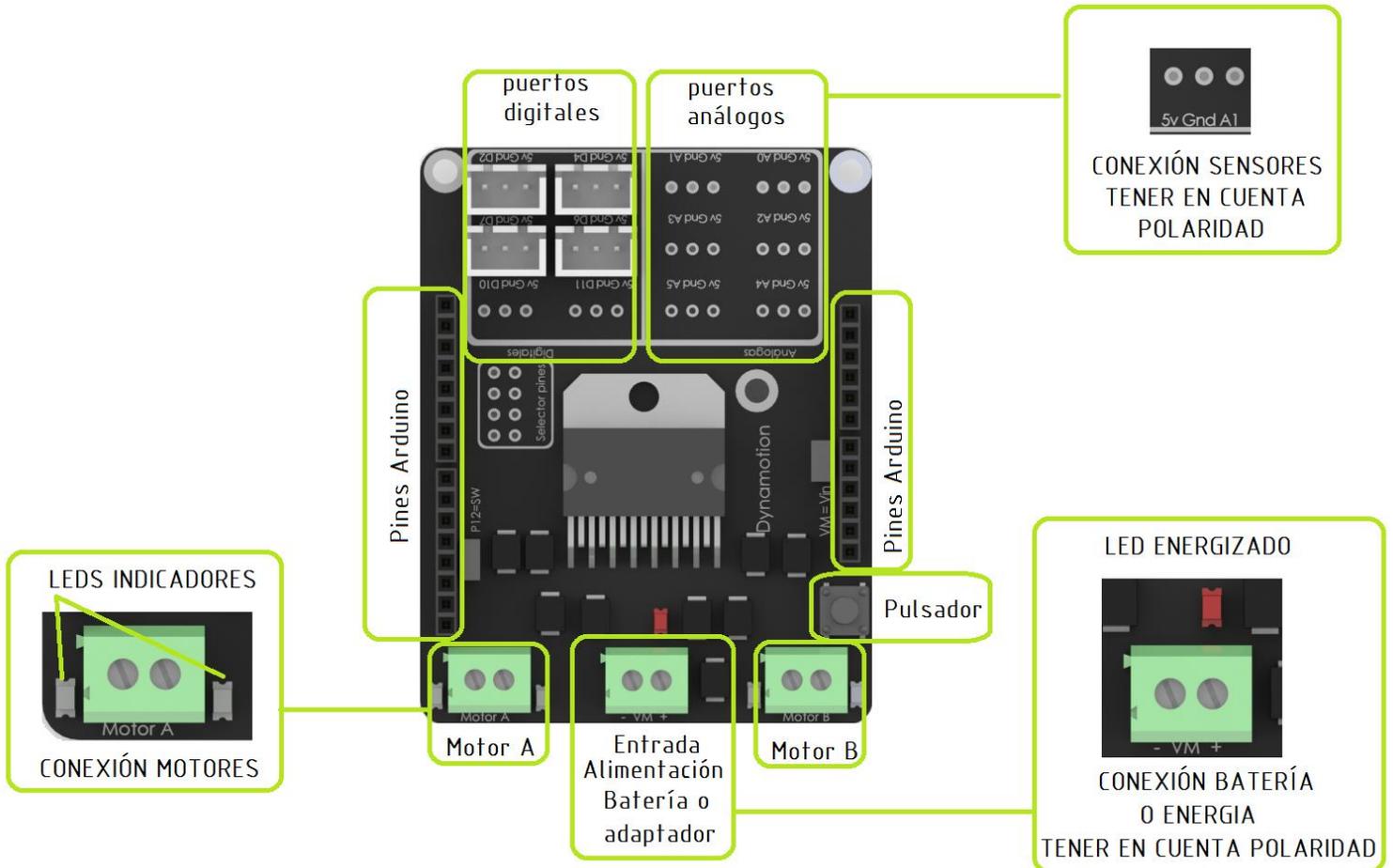
Figura 10



La plataforma esta está lista para usar!!

Figura 11

Conexiones

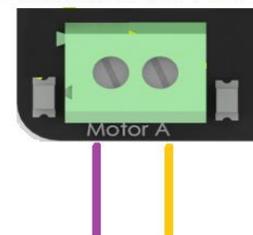


Motores: Los motores tienen dos cables de conexión de color morado y amarillo, estos cables se deben conectar a las borneras respectivas, etiquetadas como motor A y motor B, la conexión se debe hacer como se muestra en la figura siguiente

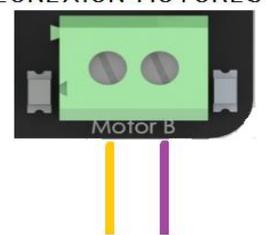


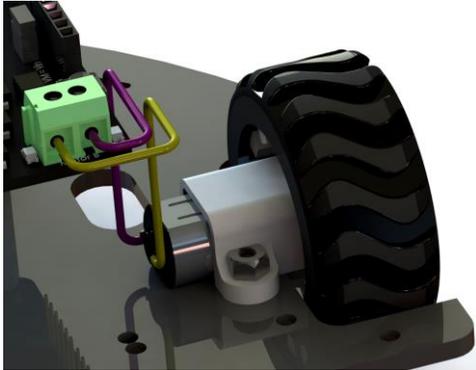
Tenga en cuenta que si los motores no se mueven en el sentido adecuado la conexión puede estar al revés, si lo conecto como muestra la imagen y sigue moviéndose en el sentido contrario, los cables del motor pueden estar invertidos

CONEXIÓN MOTORES

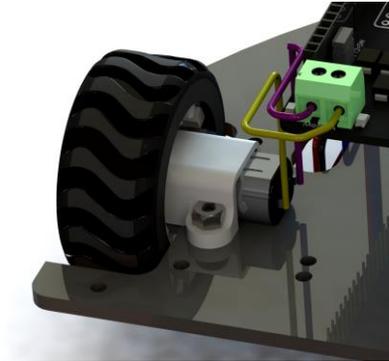


CONEXIÓN MOTORES





Conexión motor derecho



Conexión motor izquierdo

Conexión de Batería: la batería se conecta a la bornera con etiqueta VM tenga en cuenta los signos para la conexión el “-” o tierra se debe conectar el cable negro, en el “+” debe ir el cable rojo, Se recomienda usar una batería de 7.4 a 1000mAh, De Polímero, que pueda dar un voltaje estable



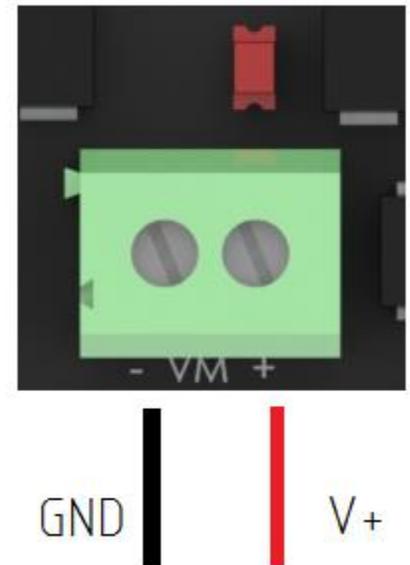
Tenga en cuenta que si conecta al revés los cables de alimentación y tierra la board no encenderá



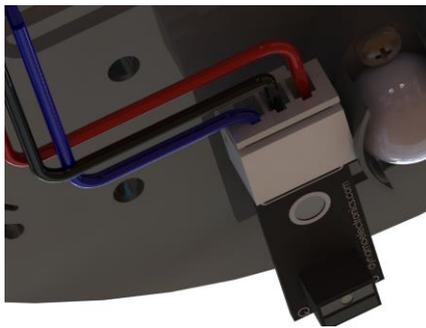
Tenga en cuenta que el voltaje máximo de conexión es de 12V, sin embargo para esta plataforma se recomienda un voltaje máximo de 8.5V



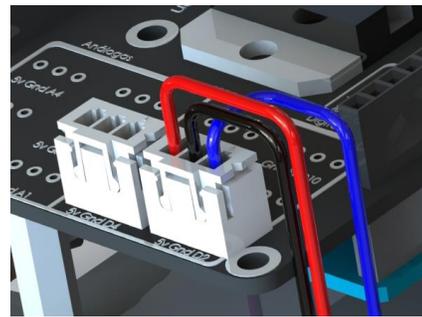
Si usa baterías de polímero - ion recuerde no dejar que el voltaje baje a menos de 6.6V, de ser así la batería podría sufrir daños irreversibles.



Conexiones de Sensores: los sensores se conectan usando los cables para sensores estos cables solo tienen una manera de ser conectados y son seguros, el cable se compone de un conector y de 3 cables de color Rojo, Negro y Azul, el rojo y negro indican alimentación y tierra y el cable azul es de señal, es decir este cable llevará la señal a la board arduino, y la etiqueta que se encuentra en cada conector de la board será el puerto usado en arduino.



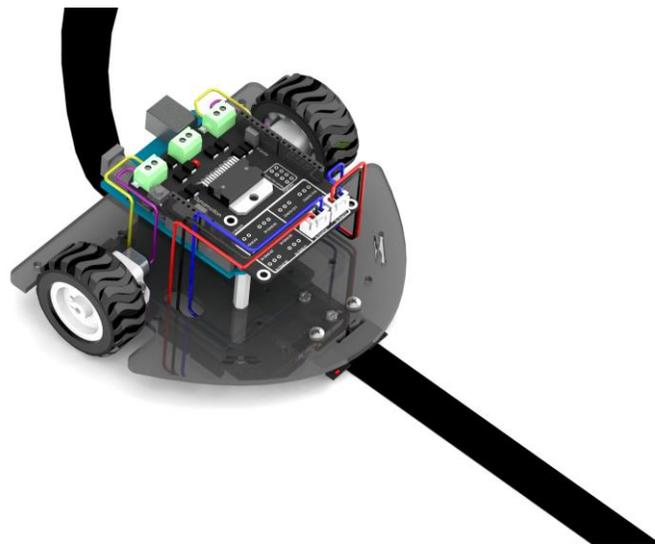
Conexión sensor



Conexión a board



Cuando no use conectores seguros tenga en cuenta conectar bien las conexiones que van a 5V y tierra para evitar hacer un corto, o sobre corrientes en el sistema



Programación básica:

Esta programación fue diseñada usando una batería de 7.4V a 1000mA

Test de conexiones Este programa se usa para verificar que las conexiones estén bien realizadas, copie este programa en arduino y descárgalo al robot, una vez descargado oprima el pulsador, el robot va a realizar los siguientes movimientos, adelante, atrás, giro izquierda, giro derecha

```
/* Ejemplo seguidor de linea basico  
para mayor informacion visita www.dynamoelectronics.com
```

los pines a usar son

Pulsador -- Arduino D12

Motor A atras -- Arduino D3

Motor A adelante -- Arduino D5

Motor B adelante -- Arduino D6

Motor B atras -- Arduino D9

Sensor derecho --- Arduino D2

Sensor izquierdo --- Arduino D4

En este ejemplo se puede ver una programacion basica de un robot seguidor de linea usando dos sensores y un control ON/OFF para los motores, los sensores de linea marcan un alto cuando estan en color blanco y un bajo en color negro, en este caso los sensores van por fuera de la linea negra

```
*/
```

```
void setup() {  
  // Declaracion de pines de entrada y salida  
  
  pinMode(12, INPUT);    // Pulsador entrada  
  pinMode(2, INPUT);    // Sensor Derecho  
  pinMode(4, INPUT);    // Sensor izquierdo  
  pinMode(3, OUTPUT);   // Motor A Atras  
  pinMode(5, OUTPUT);   // Motor A adelante  
  pinMode(6, OUTPUT);   // Motor B adelante
```

```

pinMode(9, OUTPUT);    // Motor B Atras
digitalWrite(12, INPUT_PULLUP); // Pullup para pulsador
}

void loop() {

inicio:           // etiqueta de inicio del programa por uso de saltos

movimientos("parar");
while (digitalRead(12)==1);           // Pulsador para inicio de ejecucion, el programa solo
iniciarÅ; cuando se pulse
while (digitalRead(12)==0);           // una vez sea pulsado se espera a que se deje de
presionar para iniciar
delay(100);                             // retardo de 100ms se evitan rebotes

for(;;)
{
movimientos("adelante");
delay(2000);
movimientos("atras");
delay(2000);
movimientos("giroizq");
delay(2000);
movimientos("giroder");
delay(2000);

if (digitalRead(12)==0)                // Lectura del pulsador de arranque y parada
{ while (digitalRead(12)==0);           // Si fue pulsado espera a que se deje de precionar y
para
delay(100);                             // retardo de 100ms se evitan
rebotes
goto inicio;}                           // Vuelve a inicio a esperar pulso para
arranque
}

}

void movimientos(String dir)
{
if (dir=="adelante")
{ analogWrite(5, 100);
analogWrite(9, 100);
analogWrite(3, 0);
analogWrite(6, 0);
}
}

```

```
}  
else if (dir=="atras")  
{ analogWrite(5, 0);  
  analogWrite(9, 0);  
  analogWrite(3, 100);  
  analogWrite(6, 100);  
}  
else if (dir=="giroizq") {  
  analogWrite(5, 100);  
  analogWrite(9, 0);  
  analogWrite(3, 0);  
  analogWrite(6, 100);  
}  
else if (dir=="giroder")  
{ analogWrite(5, 0);  
  analogWrite(9, 100);  
  analogWrite(3, 100);  
  analogWrite(6, 0);  
}  
else  
{ analogWrite(5, 0);  
  analogWrite(9, 0);  
  analogWrite(3, 0);  
  analogWrite(6, 0);  
}  
}
```