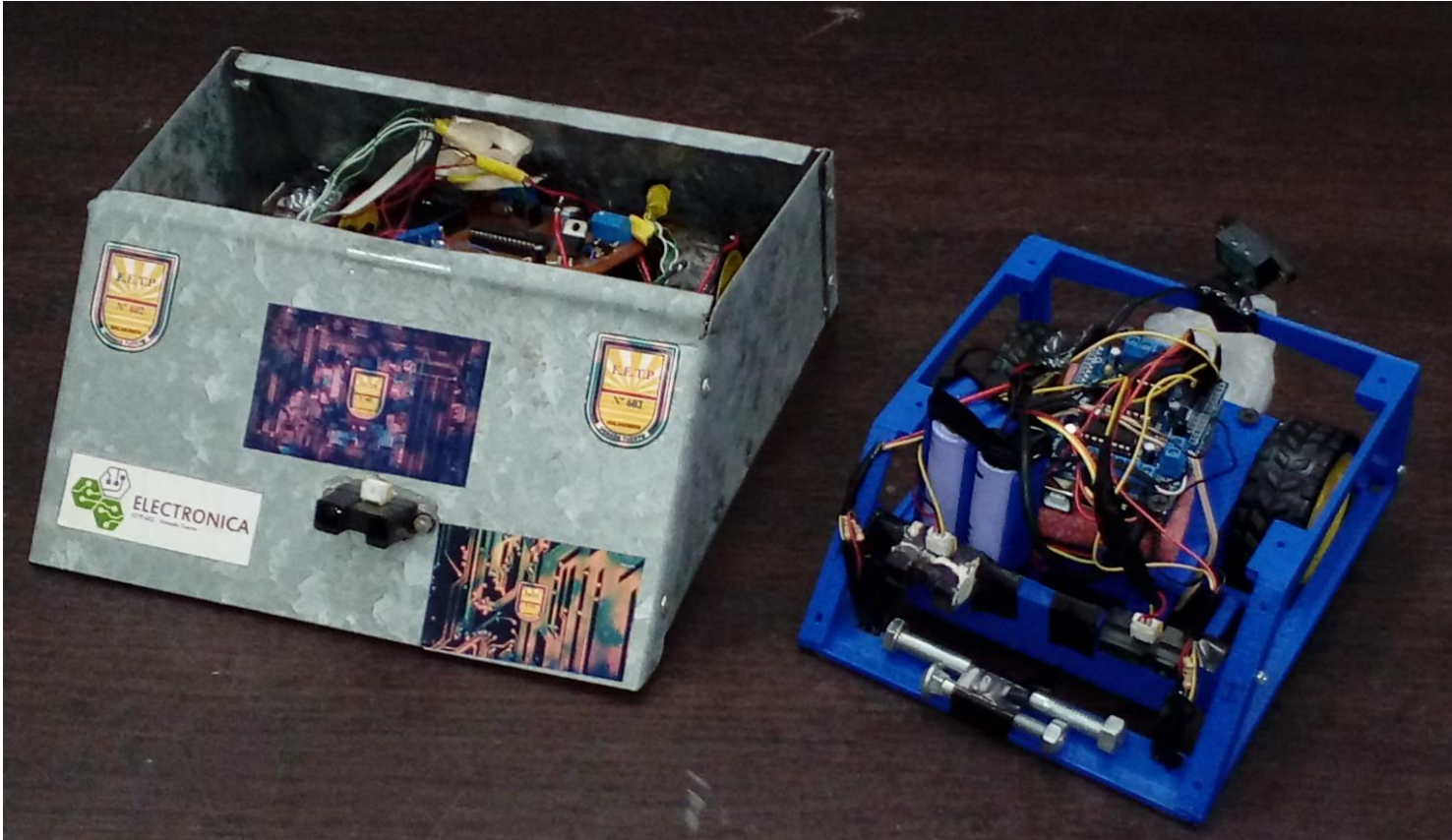

Manual de robot sumo con arduino



Autor: Profesor Palacios José Emanuel

Año: 2018

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

www.eetp602.edu.ar



Manual de Robot Sumo con Arduino

Contenido

Introducción: Robot Sumo.....	3
Materiales.....	3
Interruptor on/off y alimentación.....	4
Detección de la “línea blanca”.....	6
Detección de objetos.....	11
Control de Motores	16
Conectando los motores.....	17
Detalle de comandos que utiliza la librería del shield	19
Detalle del uso de pines	21
Lógica de movimiento	24
Montaje de los componentes.....	26
Programación de la aplicación final.....	32
Bibliografía:.....	36



Manual de Robot Sumo con Arduino



Introducción: Robot Sumo.

En este manual desarrollaremos un robot Sumo funcional básico, que se espera sufra modificaciones por quien lee este manual. Los robots Sumo son robots que tienen como objetivo sacar o quitar a su contrincante de un área de combate específica (dojo), para lo cual deben ser autónomos por lo que necesitan sensores que detecten a su enemigo y así poder envestirlo hasta lograr sacarlo del dojo, en el siguiente enlace puedes ver una competencia robots sumo.

<https://youtu.be/jFRErAX1tY0?t=3m30s>

Los robots sumos deben realizar 3 actividades:

- **Activarse y esperar 5 segundos antes de empezar a moverse.**
- **Detectar al enemigo y atacar.** Esto lo logra ubicando la posición del enemigo por medio de unos sensores de distancia, cuando se detecta un objeto cercano se aproxima a atacarlo.
- **Evitar salir del área de combate.** El área de combate es de color negro con una línea blanca alrededor, entonces se usan unos sensores que son capaces de detectar dicha línea y al encontrarla evitarán pasar sobre ella.

Material.

- 1x Arduino uno
- 1x Chasis que cumpla con las normas de la categoría (se proporcionaran planos de ejemplo).
- 2x Micromotor Metálico Polulu o en su defecto 2 Motor Dc 3v A 12v Tt Engranaje Robot Arduino Smart Car Auto.
- 2x Sensores de distancia Sharp de 150cm.
- 2x Conectores para sensor de distancia Sharp.
- 1x protoboard.
- 1x Driver Pololu A4988 Arduino Cnc Impresora 3d Motor o motor shield V1.0
- 2x Sensor de reflectancia infrarrojo TCRT5000
- 1x Soldadura
- Jumpers hembra-hembra y Jumpers macho-macho varios
- Interruptor 1 polo 2 tiros
- 1pulsador de panel N/A.
- Pistola de silicona.
- Baterías recargables tipo 18650 3.7v 5000 Mah
- Conector para fuente DC de arduino.
- Tornillos cabeza tanque de 1/8 de varias longitudes.

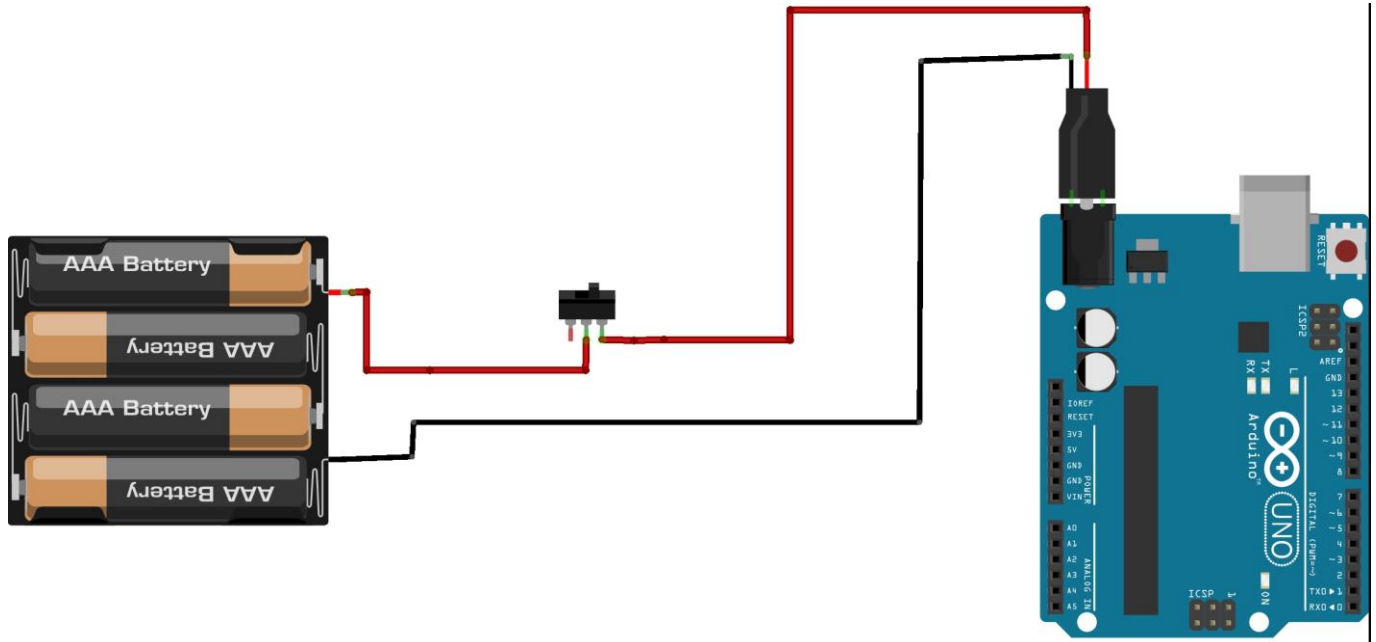


Manual de Robot Sumo con Arduino

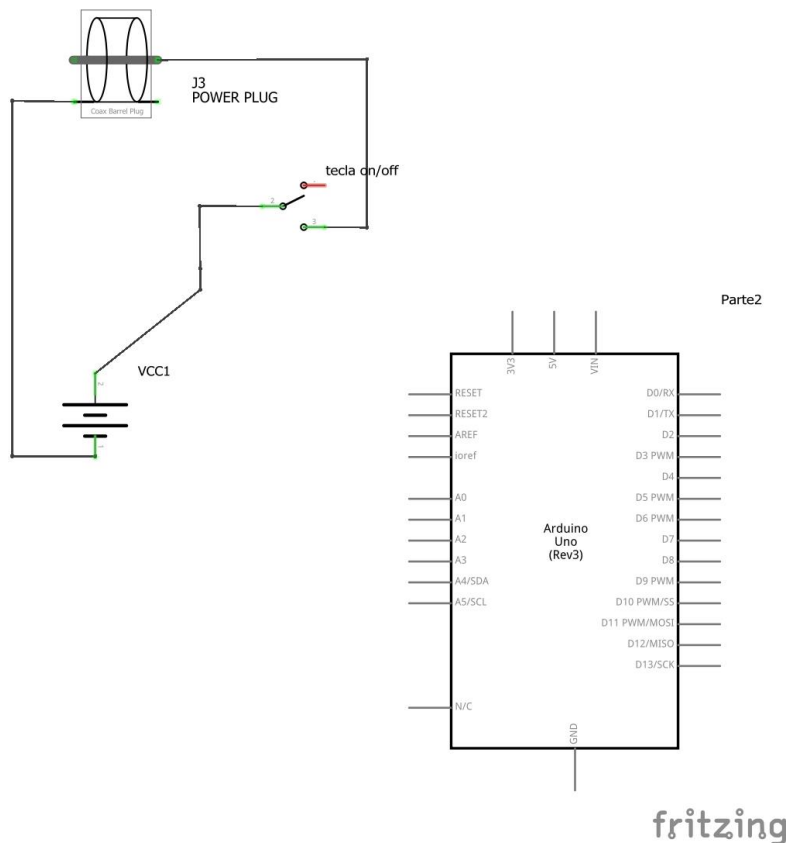


Interruptor on/off y alimentación

Es conveniente agregar un interruptor para la alimentación del circuito, el interruptor debe ser de 2 posiciones, estos interruptores tienen 3 pines, debes soldar el del centro y uno en los extremos:



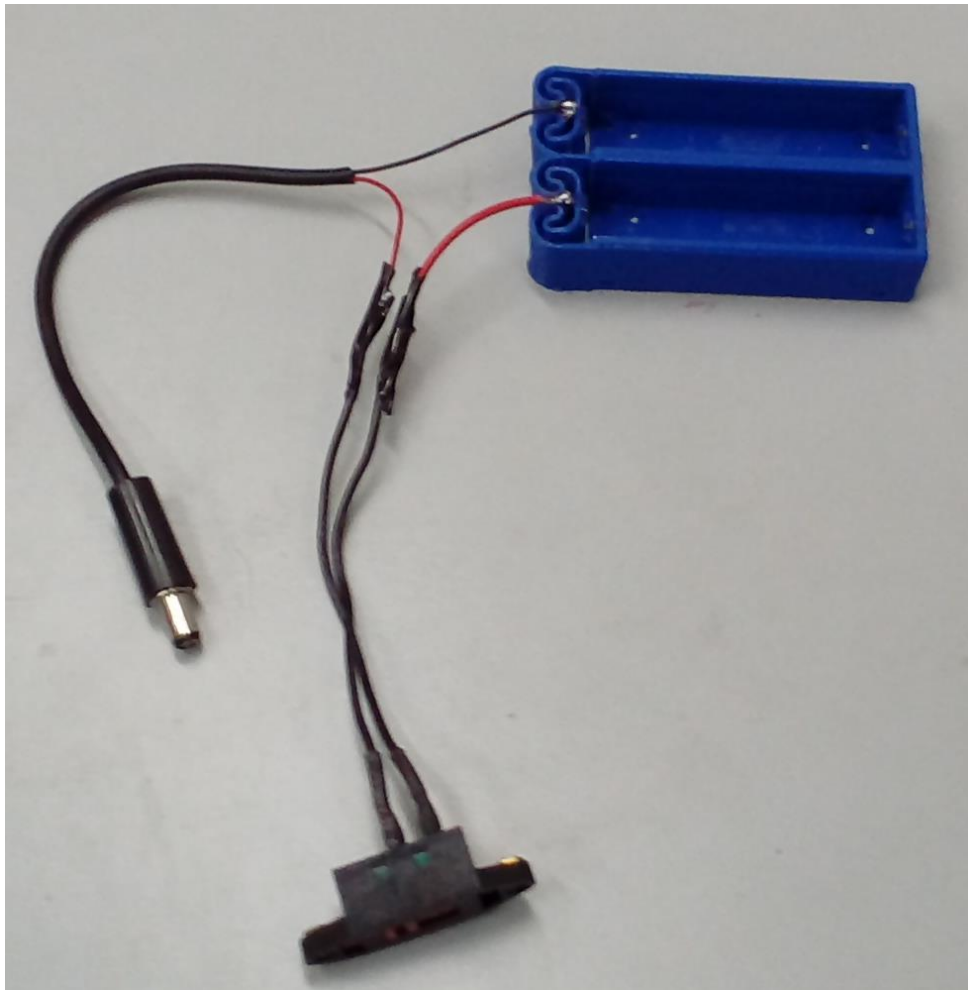
Esquema





Manual de Robot Sumo con Arduino

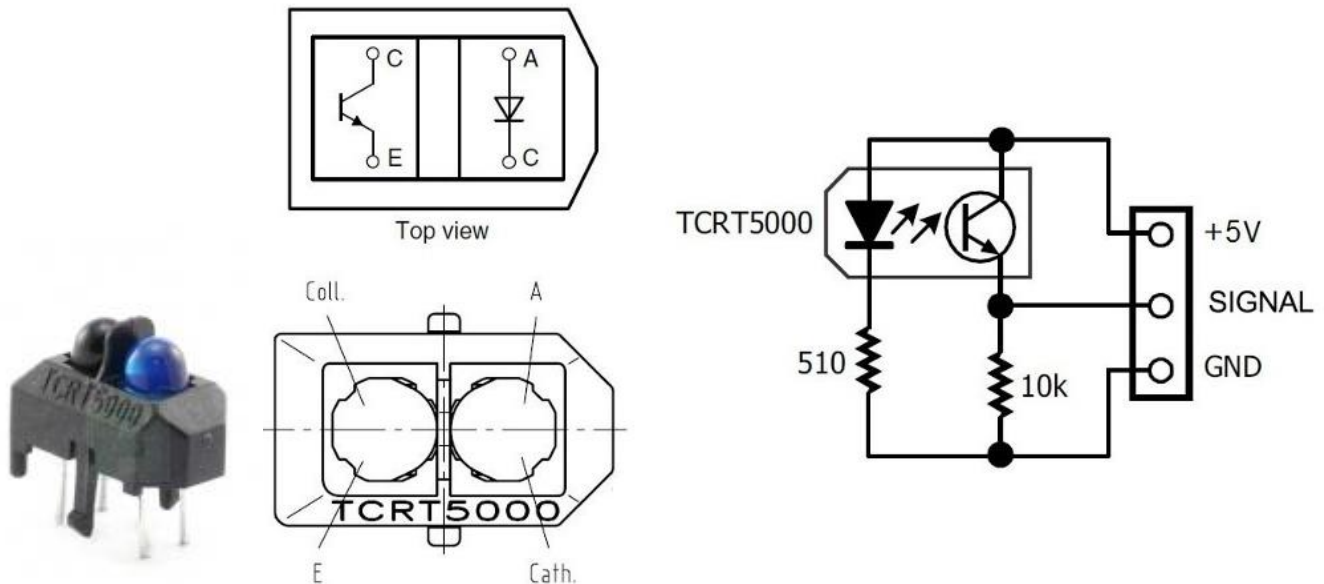
Uno de los cables del interruptor va conectado al voltaje de la batería y el otro al punto en el que queremos alimentar el circuito, como se observa en la siguiente imagen:



Manual de Robot Sumo con Arduino

Detección de la "línea blanca".

Parte importante de cómo armar un Robot Sumo son los sensores de línea, los cuáles evitarán que él mismo salga del área de combate, utilizaremos los sensores TCRT5000 en este manual. A continuación vemos sus terminales y como se debería conectar:



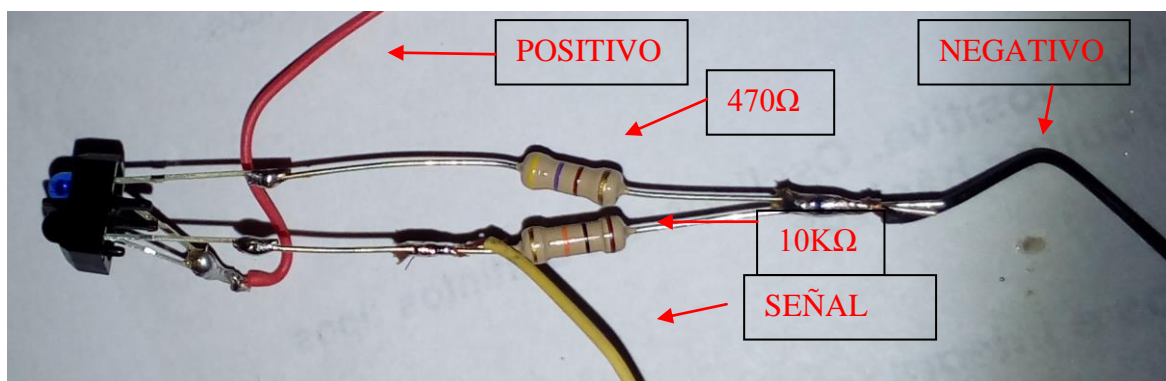
Con la configuración anterior obtendremos a la salida de la señal valores entre 0 y 5 voltios.

Debemos soldar las resistencias al sensor de forma que se puedan doblar los 3 cables que irán al arduino como en la siguiente imagen:

+5v: al positivo

GND: al negativo

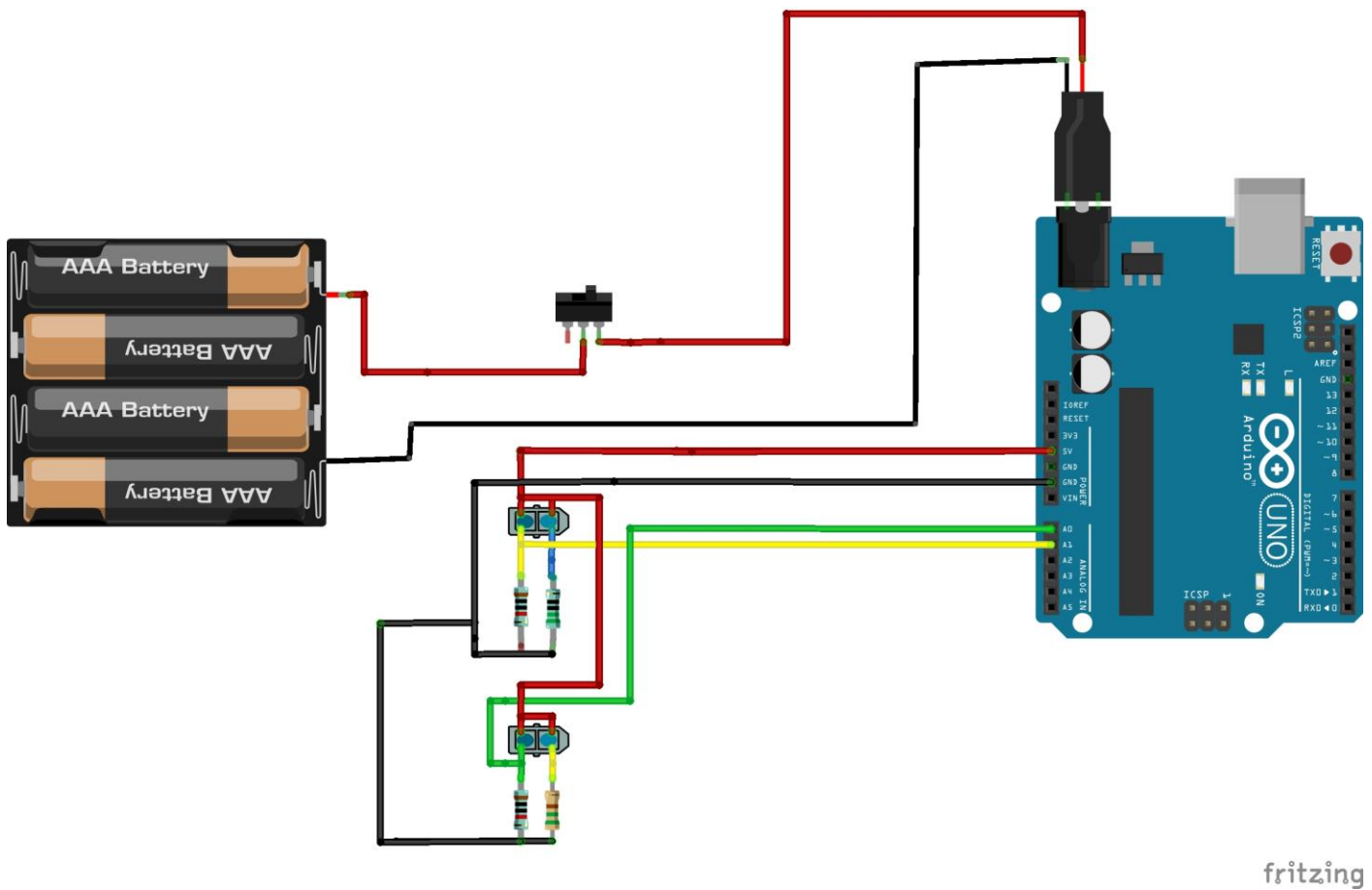
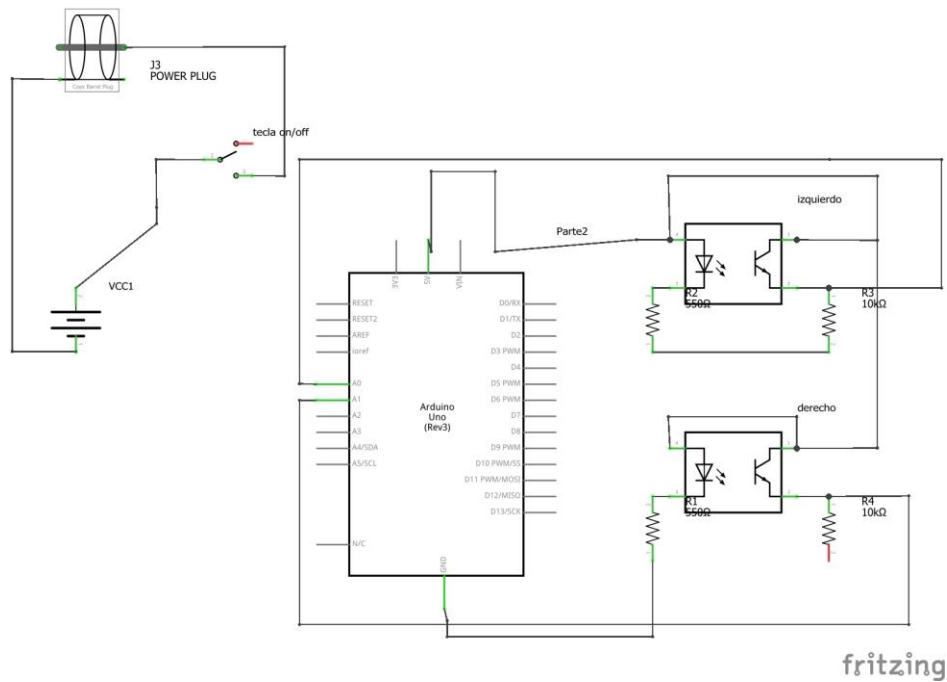
SIGNAL: a la entrada analógica del arduino A0 o A1





Manual de Robot Sumo con Arduino

Luego se conectará cada sensor al arduino como se muestra en el siguiente esquema:





Manual de Robot Sumo con Arduino

Se hará un programa muy corto con el que se probará su funcionamiento. Si el sensor detecta línea blanca se encenderá el LED, hay que destacar que **el sensor envía un voltaje alto cuando detecta línea blanca**.

```

/*****
/* Autor: Palacios José Emanuel docente de la EETPN°602
/* Programa: testeo de sensores de línea blanca o negra
/* Año:2018
*****/

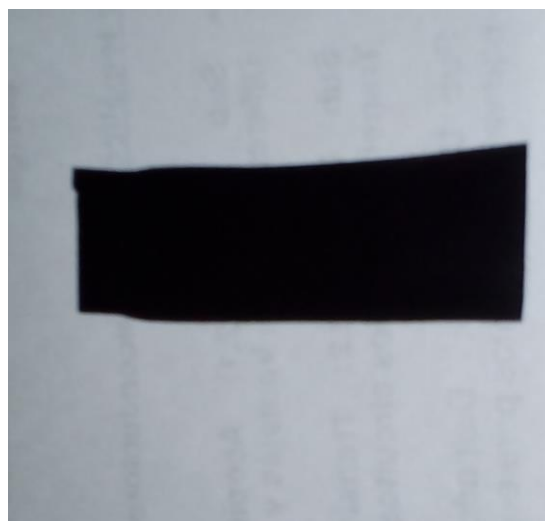
int LED = 13; // definimos la variable LED EN EL PIN 13
int lectura_linea_1; // definimos la variable donde colocaremos el valor del sensor 1
int lectura_linea_2; // definimos la variable donde colocaremos el valor del sensor 2

void setup() {
  pinMode(LED,OUTPUT); //definimos la variable LED que está en el pin 13 como salida
  Serial.begin(9600); //habilitamos el monitor serial
  Serial.println("Iniciando lectura de sensor");
}

void loop() {
  delay(100);
  lectura_linea_1 = analogRead(A0); //leemos el sensor 1
  lectura_linea_2 = analogRead(A1); //leemos el sensor 2
  Serial.println(lectura_linea_1); //mostramos el valor por el monitor serial
  Serial.println(lectura_linea_2); //mostramos el valor por el monitor serial
  if(lectura_linea_1 > 30 || lectura_linea_2 > 30) //si detecto línea blanca activo LED
    digitalWrite(LED,HIGH); //activo el LED
  else
    digitalWrite(LED,LOW); //desactivo el LED
}

```

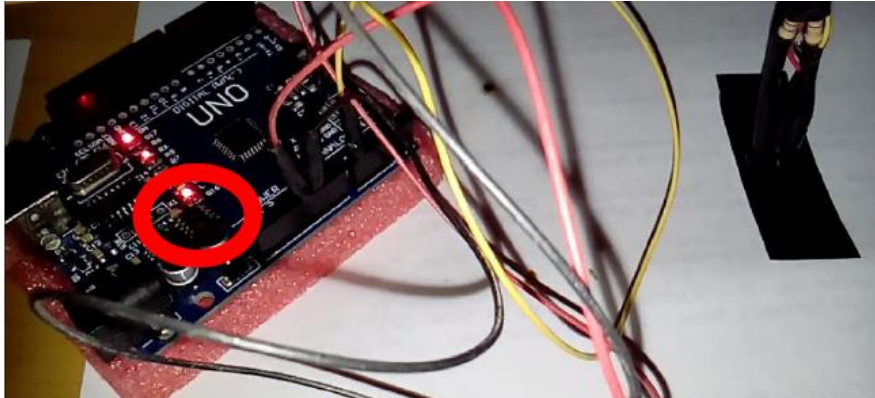
Para probar el sensor utilizaremos algún objeto que sea de color blanco y negro.



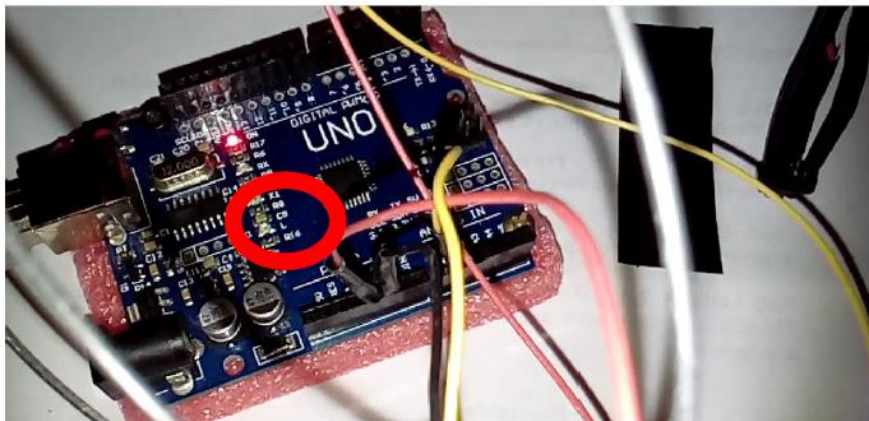
Y finalmente, al probar el sensor detectando el objeto en una superficie blanca o negra, se encenderá y apagará el LED respectivamente:

Manual de Robot Sumo con Arduino

Detectando la superficie negra

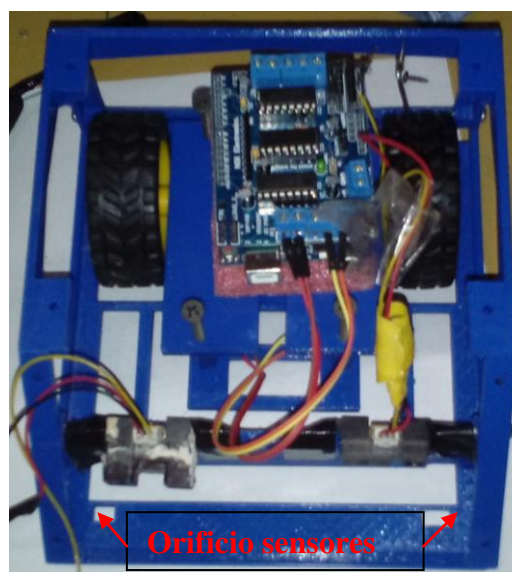


Detectando la superficie blanca



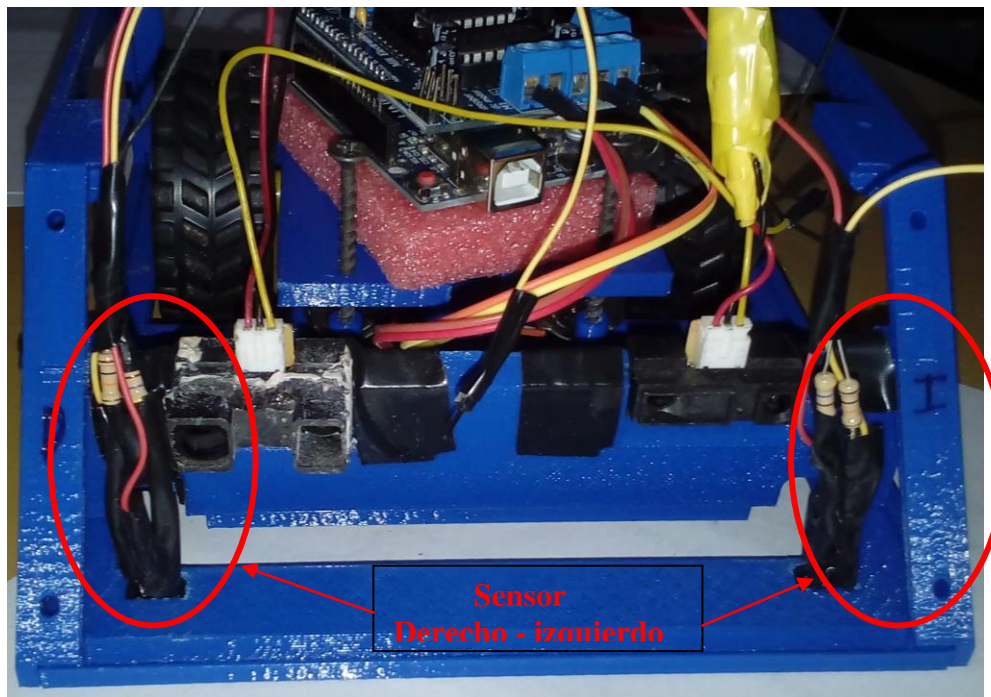
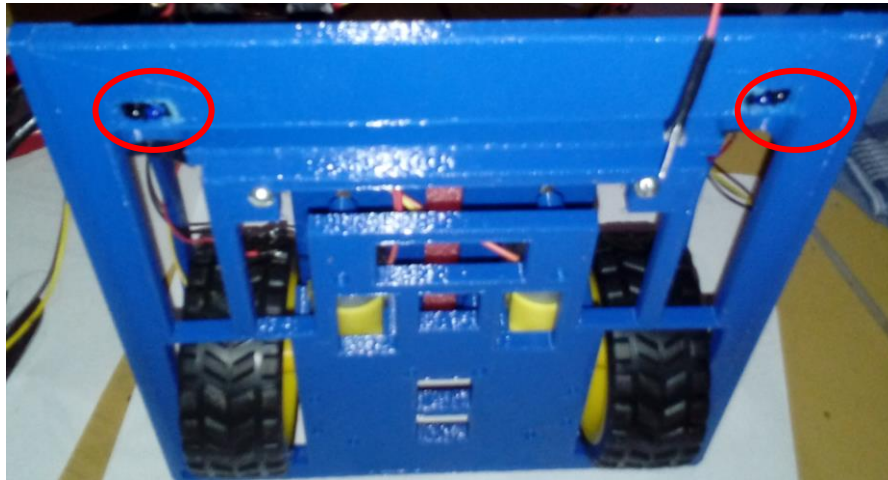
Una vez probados ambos sensores procedemos a colocarlos en la estructura, recomendamos usar silicona para adherirlos a la estructura y para que quede aislado el sensor para evitar hacer algún corto circuito.

El sensor debe ser colocado en el orificio de la parte inferior del chasis de Robot Sumo.



Manual de Robot Sumo con Arduino

Sensores ya colocados



Para saber si funciona el sensor se puede observar con la cámara de un celular y se podrá observar si el led infrarrojo está funcionando:





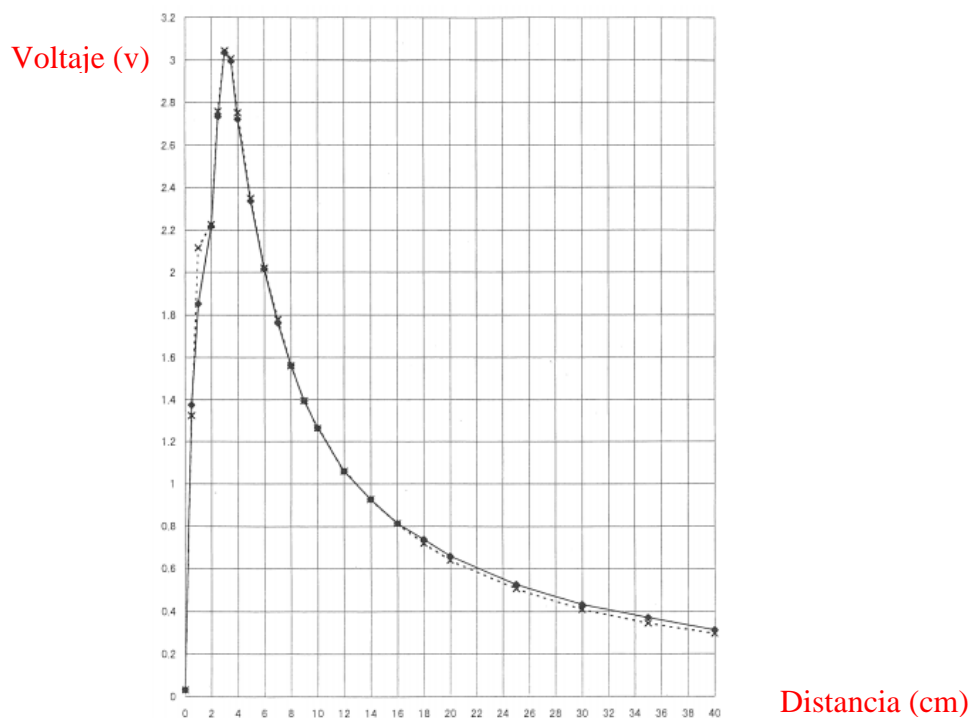
Manual de Robot Sumo con Arduino



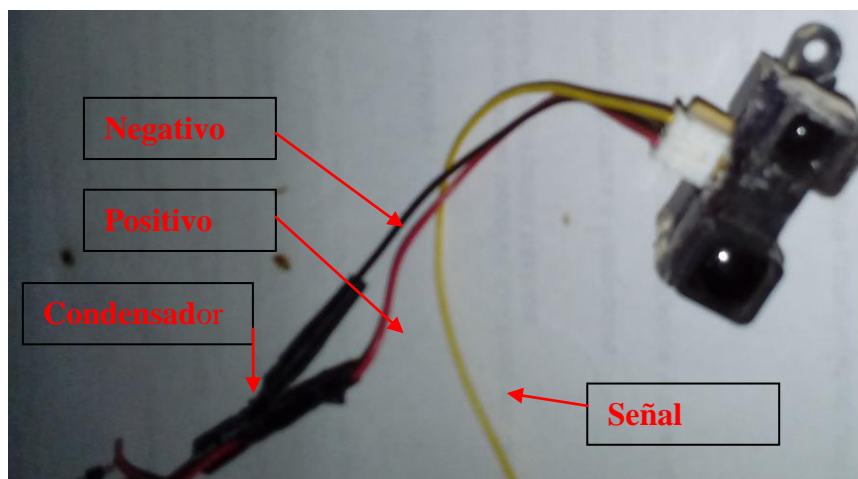
Detección de objetos.

Para detectar el enemigo se debe medir la distancia de objetos que se encuentren al frente o a los lados del Robot Sumo, para esto usaremos los sensores de distancia SHARP, estos nos entregan un voltaje analógico que depende de la distancia que se esté midiendo.

El sensor entrega 3 volts si el objeto se encuentra cerca (no mide distancias menores a 4cm, por lo que se aconseja utilizar otro sistema de medición para distancias menores a esta), y disminuye el voltaje que entrega cuando la distancia va incrementando. La siguiente gráfica explica su funcionamiento:



Los sensores utilizan unos conectores especiales, estos necesitan soldarse a otros conectores para después poder conectarlos al arduino. Puedes usar unos jumpers hembra o macho, si usas hembra puedes conectar directamente a tu arduino, si utilizas macho puedes conectar a un protoboard.

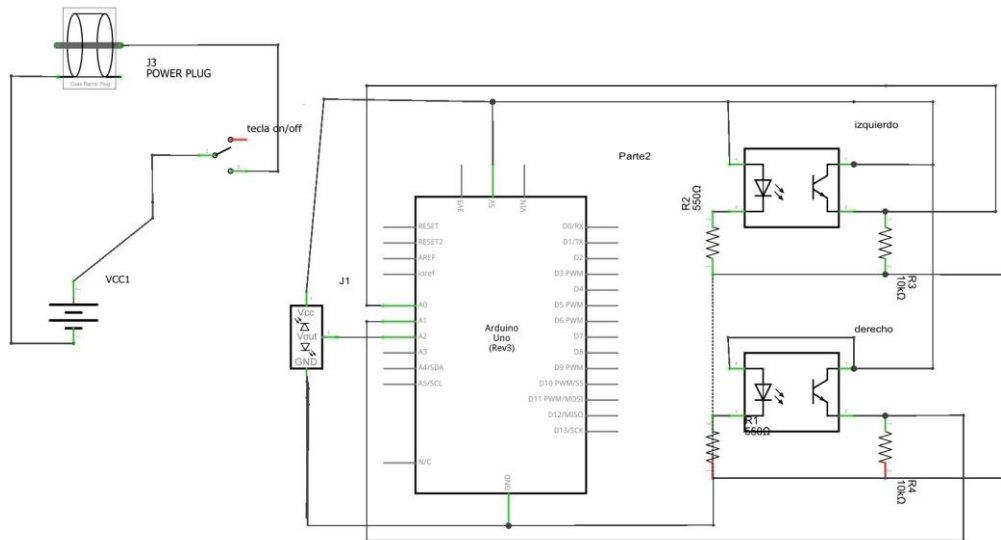




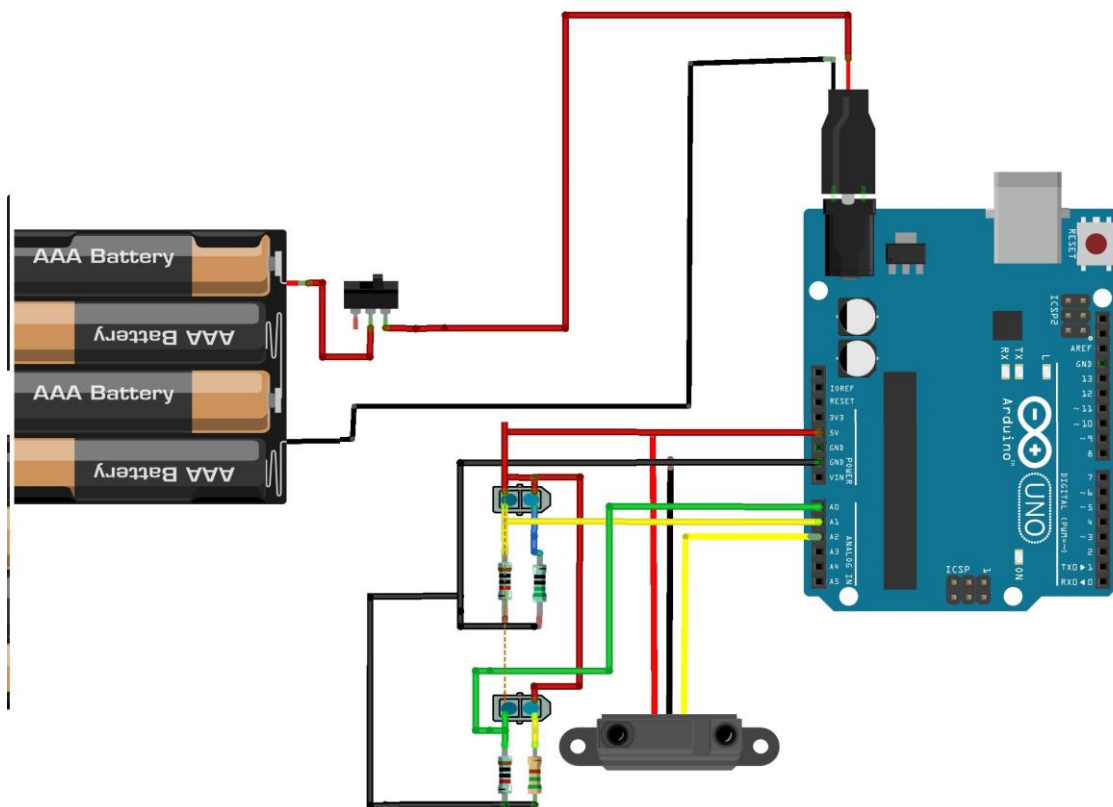
Manual de Robot Sumo con Arduino

Al soldar los conectores recomendamos el uso de termo contraíble o cinta aisladora, tendrá un acabado más ajustado y resistente.

Una vez soldados los conectores se conectarán al arduino y se probará su funcionamiento. Primero conecta el sensor al arduino como en el siguiente diagrama (esquemático y dibujo en fritzing):



fritzing





Manual de Robot Sumo con Arduino

Este código que nos permitirá medir la distancia y enviarla al puerto serie.

```

/*****
/* Autor: Palacios José Emanuel docente de la EETPN°602
/* Programa: testeo de sensores SHARP
/* Año:2018
*****/

// Conexiones
int sensor_distancia_1 = A2;//defino que voy a conectar el sensor en A0

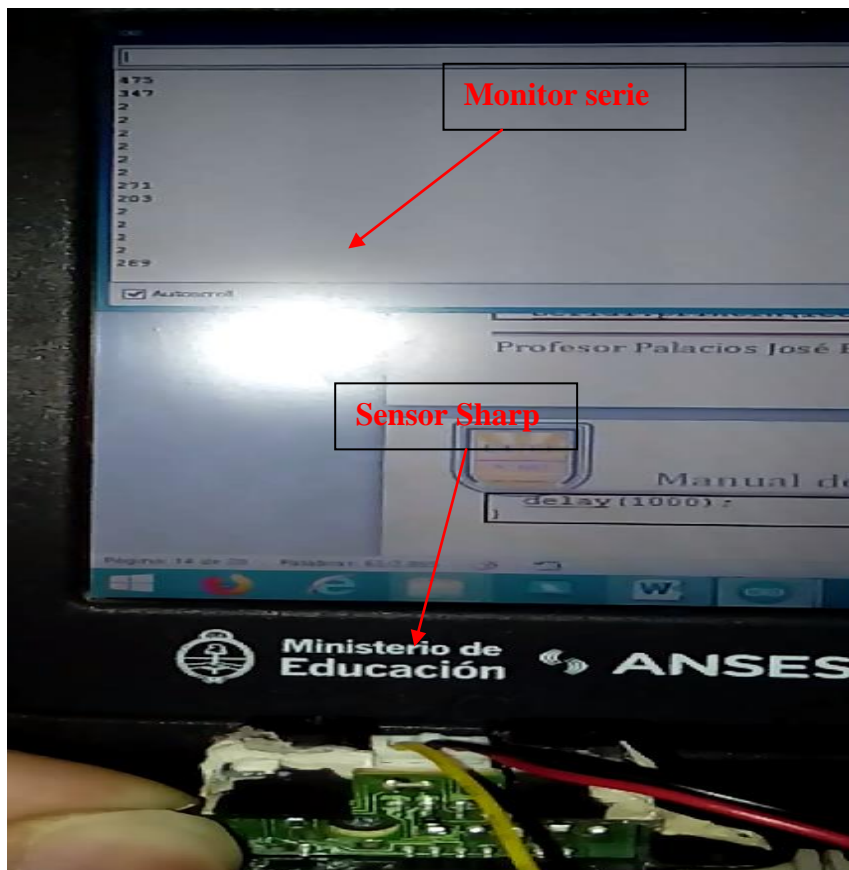
// Inicialización
int lectura = 0;

void setup() {
  Serial.begin(9600);
  Serial.println("Iniciando lectura de sensor");
}

// Programa
void loop() {
  lectura = analogRead(sensor_distancia_1);//leo el sensor conectado en A0
  Serial.println(lectura);//muestro por el monitor serial la lectura realizada
  delay(1000);
}

```

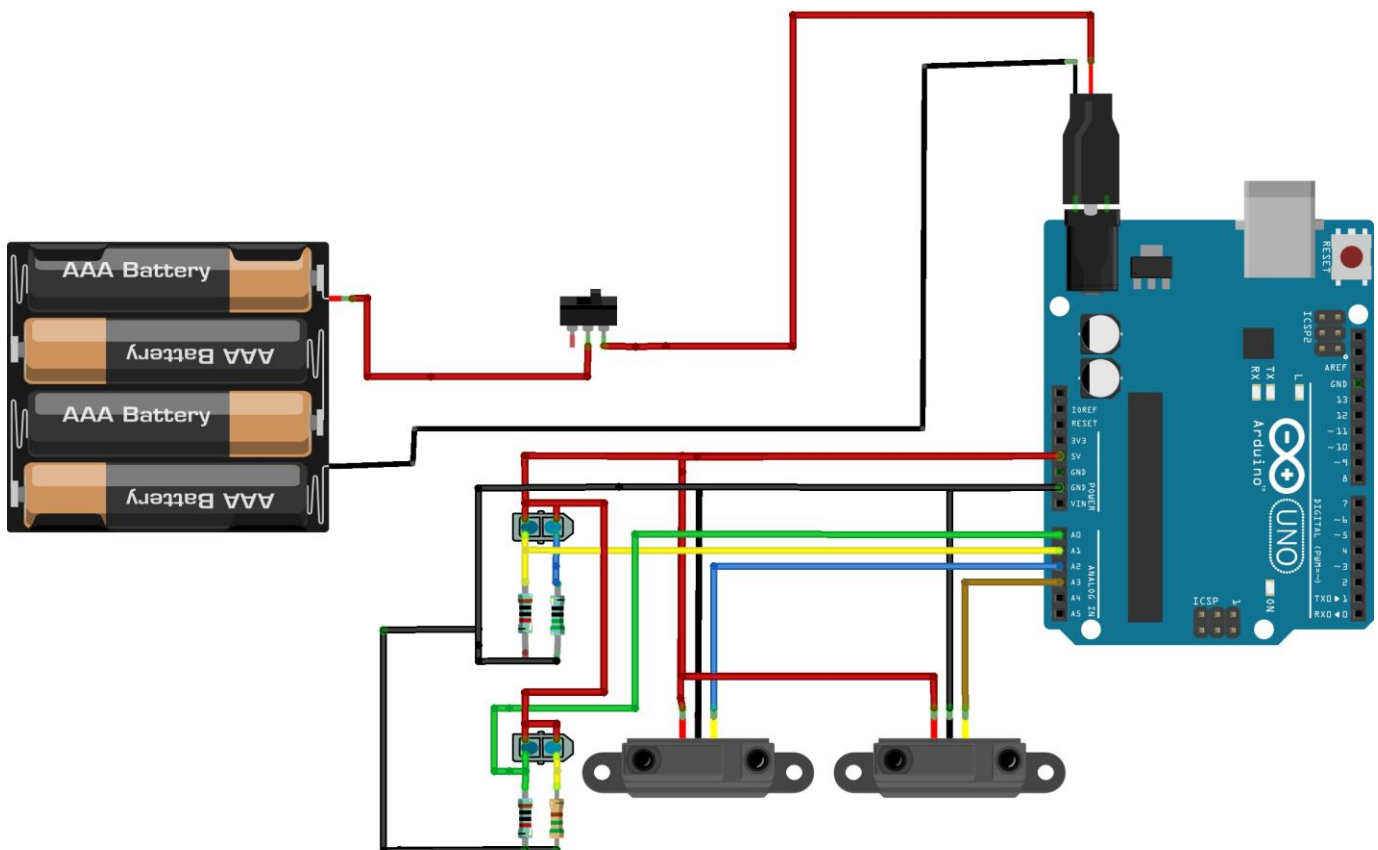
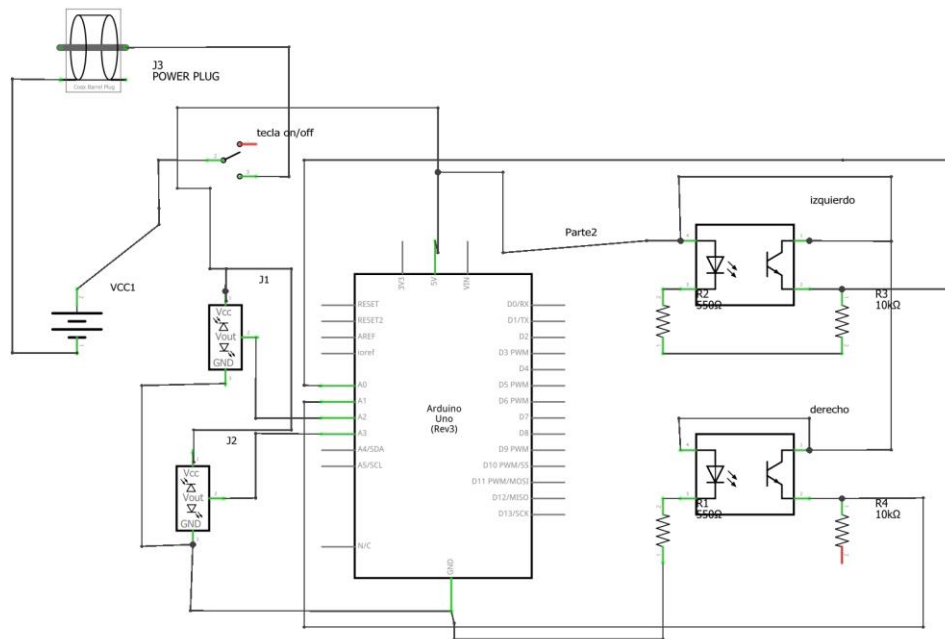
Una vez programado el código abre el monitor serie y te mostrará la lectura del sensor, prueba colocar un objeto frente a él a diferentes distancias y visualiza el dato que envía el monitor serie.





Manual de Robot Sumo con Arduino

Acá el circuito con los dos sensores SHARP conectados.



fritzing



Manual de Robot Sumo con Arduino

Nota: se recomienda conectar en paralelo a los cables de alimentación del sensor Sharp un condensador electrolítico de 10uf x 10v, para eliminar posibles ruidos que genere el sensor. Así como analizar la hoja de datos del sensor que esté utilizando

Para saber si funciona el sensor se puede observar con la cámara de un celular y se podrá observar si el led infrarrojo está funcionando:



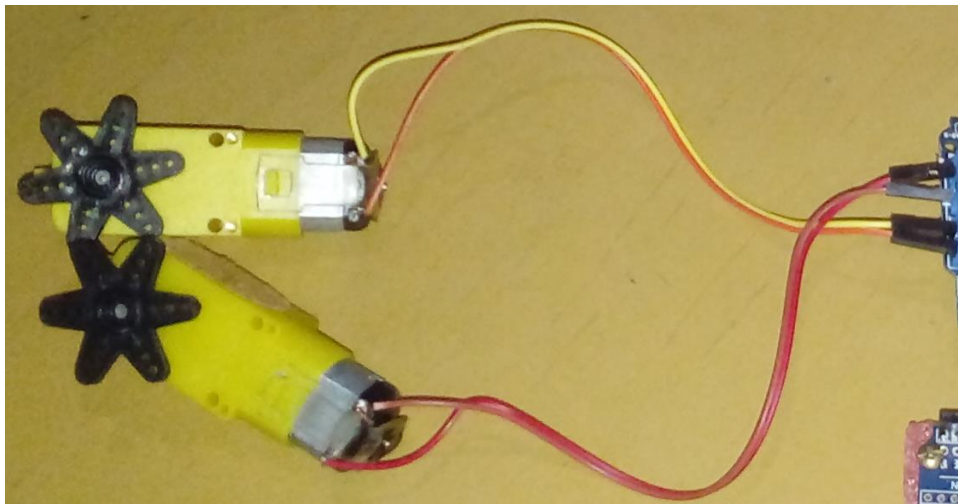
Manual de Robot Sumo con Arduino

Control de Motores

Terminada la parte de los sensores, 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, igual que con los sensores de distancia usaremos jumpers para que sea fácil conectar a los motores.

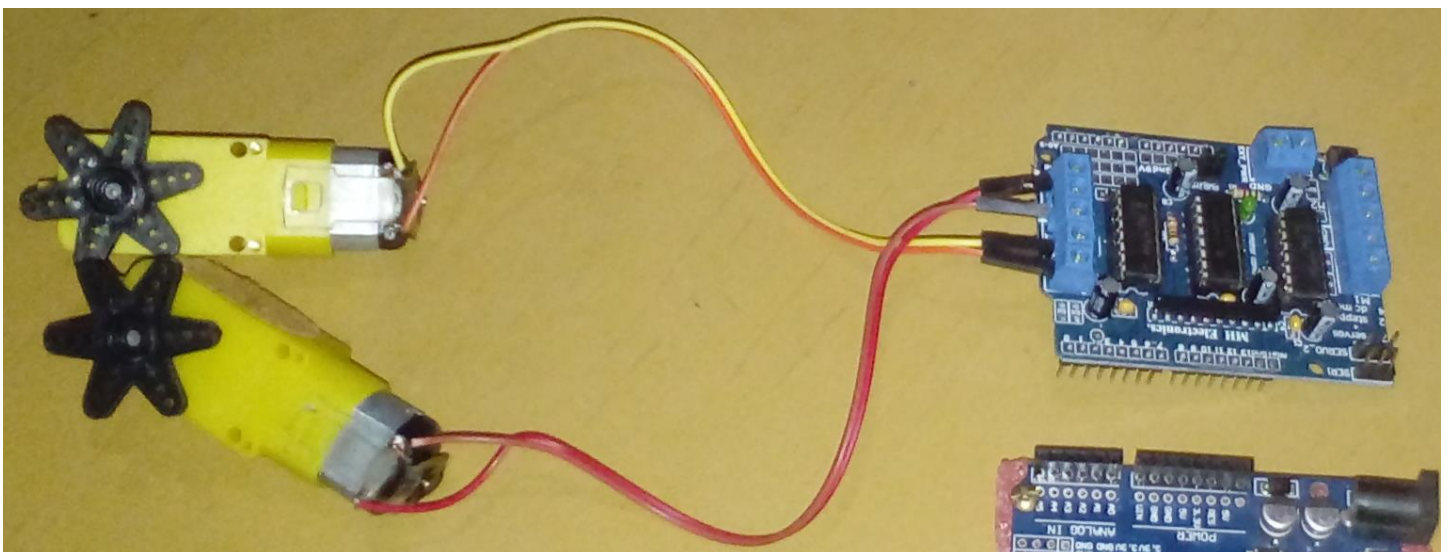
Se deben soldar los cables de manera perpendicular al motor de modo que la longitud del motor sea lo más corta posible (ahora que se ha incluido el cable).



Una vez soldados los motores debemos soldar unos conectores machos para el driver, soldaremos los conectores macho para conectar al protoboard.

Shield motor V1.0

Aquí tienen una foto del Motor Shield V1.0, que pueden encontrar fácilmente en cualquier proveedor:





Manual de Robot Sumo con Arduino



Las características principales son:

- **4 H Bridges** incluidos con dos **L293D** chips.
- Hasta **4 motores CC** con control bidireccional y selección de velocidad de 8 bits.
- Máxima corriente de 0,6 Amperios (Aunque acepta picos de hasta 1,2) con una protección de sobre temperatura.
- Acepta motores cuya alimentación va desde 4,5 a 25V.
- Podemos añadir otros dos motores Servos o paso a paso.
- Dispone de alimentación de motores, separada de la del Shield para evitar ruido e interferencias.
- Bajo precio.
- Compatible con **UNOs** y **Megas** por lo menos.
- Dispone de una librería cómoda para manejar los motores.

En resumen, es un **Shield** barato y práctico para manejar pequeños motores de 5V, pero que se quedará corto si necesitas una cierta potencia en los motores, ya que la intensidad que estos demandaran superará con facilidad los 0,6A que tiene como límite este Shield.

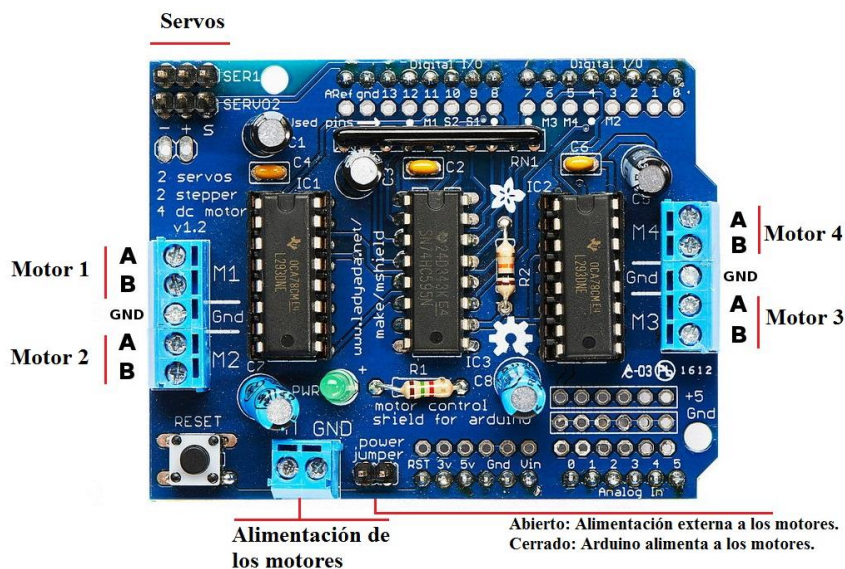
Es ideal para construir robots autónomos con pequeños motores y sin peso y sobre todo como elemento de aprendizaje antes de invertir en otras opciones más sofisticadas.

Conectando los motores

El shield conecta directamente los **H Bridges L293D** a los pines PWM de **Arduino** y además incorpora un viejo conocido nuestro el **74HC595**, un **Shift Register**, para ahorrar pines en la conexión.

El Shield se reserva los pines 3,4,5,6,7,8,9,10,11,12, los pines 9 y 10 se usan para los servos en caso de que los incluyamos y quedan libres los pines 2 y 13, así como el 0 y 1 que se usan para la comunicación USB con nuestro PC.

Los pines A0 a A5 están disponibles para nuestro uso, podemos usarlos como pines digitales también.



Hay conexiones para los bornes de 4 motores, marcados como M1, M2, M3 y M4 y conviene que al conectar sigas el mismo criterio en los bornes y en el Shield porque de lo contrario conseguirás que alguno de los motores gire al revés de lo que pretendes. Lo que no es muy grave porque se arregla con facilidad.



Manual de Robot Sumo con Arduino

Una cosa interesante de este shield es que nos permite separar la alimentación de los motores de la alimentación de Arduino y es de agradecer porque los motores generan mucha interferencia electromagnética que puede hacer que nuestro Arduino se comporte de forma errática.

Por eso siempre que sea posible conviene que separes la alimentación de uno y otro (*Aunque en un robot autónomo va a ser difícil*). Para ello basta con que quites el "power jumper" y la alimentación estará separada. Esto será imprescindible si tus motores son de digamos 9 0 12V porque tu Arduino solo funciona con 5V.

Vamos a conectar un primer motor al Shield para probar. Lo primero montad el Shield a nuestro **Arduino**. Conectar los bornes del motor a los tornillos A y B del **Motor Shield**, dejando libre el pin central marcado como **GND**.

- *No es importante en que borne coloco cada que pin, porque lo único que ocurrirá si lo invertís es que el motor girará al revés. Lo que sí es recomendable es que mantengáis el mismo criterio para todos los motores.*
- *Como en este primer ejemplo vamos a alimentar el motor desde Arduino y el USB, no olvidés conectar el power jumper, para que alimentemos los motores directamente desde Arduino. Posteriormente se recomienda alimentar el módulo de motores directamente desde la batería.*

Ahora vamos a descargar una **librería para manejar el Shield** directamente, que nos permitirá abstraernos del detalle de pines.

La librería que necesitamos es esta [adafruit-Adafruit-Motor-Shield-library-8119eec](https://www.prometec.net/wp-content/uploads/2015/03/adafruit-Adafruit-Motor-Shield-library-8119eec):

<https://www.prometec.net/wp-content/uploads/2015/03/adafruit-Adafruit-Motor-Shield-library-8119eec.zip>

Y para instalarla seguimos el [procedimiento habitual](#):

<https://www.prometec.net/librerias/>



Manual de Robot Sumo con Arduino



Detalle de comandos que utiliza la librería del shield

DEFINICIÓN

AF_DCMotor *nombre_motor*(*num*, MOTOR12_*freq*KHZ);

- *nombre_motor*: el nombre que queramos darle al motor.
- *num*: número del 1 al 4 que indica el conector que utilizaremos en la placa.
- *freq*: puede ser 64, 12, 8 o 1. A más frecuencia, más exactitud y menos ruido pero más consumo.
- Ejemplo: AF_DCMotor motor2(2, MOTOR12_64KHZ); crea un motor llamado motor2 que se deberemos poner en el conector número 2 y que cuya frecuencia de PWM será de 64KHz.
- **NOTA**: para los motores 1 y 2 podemos utilizar cualquier frecuencia, pero para los motores 3 y 4 sólo se usa la frecuencia de 1kHz, cualquier otra frecuencia indicada en la función se desechará.

VELOCIDAD

***nombre_motor*.setSpeed(*num*);**

- *nombre_motor*: el nombre con el que hemos definido del motor.
- *num*: número del 0 al 255 que indicará el nivel de voltaje de salida aplicado al motor: 0 mínimo, 255 máximo.
- Ejemplo: motor2.setSpeed(100); hace que al motor que hemos definido como motor2 se le aplique un nivel de voltaje de 100, siendo el máximo 255.

SENTIDO

***nombre_motor*.run(SENTIDO);**

- *nombre_motor*: el nombre con el que hemos definido del motor.
- *SENTIDO*: la palabra FORWARD o BACKWARD según queramos que el motor gire en un sentido o en otro.
- Ejemplo: motor2.run(FORWARD); hace que al motor que hemos definido como motor2 gire hacia adelante.

• REPOSO



Manual de Robot Sumo con Arduino



nombre_motor.run(RELEASE);

- *nombre_motor*: el nombre con el que hemos definido del motor.
- Ejemplo: motor2.run(RELEASE); hace que al motor que hemos definido como motor2 se pare.

Aca tenemos el programa para probar un motor:

```

/*****
/* Autor: Palacios José Emanuel docente de la EETPN°602          */
/* Programa: testeo de Modulo Motores V1                        */
/* Año:2018                                                      */
*****/

#include <AFMotor.h> //incluye la librería de AdaFruit en nuestro programa
AF_DCMotor Motor1(1); // crea una instancia de un motor conectado a la puerta M1, y le
iniciamos con el parámetro que le pasamos que puede ir del 1, M1 al 4, M4.

void setup()
{
    Motor1.run(RELEASE); // detengo motor
    Motor1.setSpeed(200); // Definimos la velocidad de Motor1 (puede ir desde 0 a 255)
}

void loop()
{
    Motor1.run(FORWARD); // Hacemos que el Motor1 avance
    delay (2000); // Demora de 2000 milisegundos

    Motor1.setSpeed(180); // definimos la velocidad en 180
    Motor1.run(BACKWARD); // Hacemos que el Motor1 retroceda
    delay (2000); // Demora de 2000 milisegundos
}

```

Ahora para usar dos motores.

```

/*****
/* Autor: Palacios José Emanuel docente de la EETPN°602          */
/* Programa: testeo de Modulo Motores V1                        */
/* Año:2018                                                      */
*****/

#include <AFMotor.h> //incluye la librería de AdaFruit en nuestro programa
AF_DCMotor Motor1(1); // crea una instancia de un motor conectado a la puerta M1
izquierdo, y le iniciamos con el parámetro que le pasamos que puede ir del 1, M1 al 4,
M4.
AF_DCMotor Motor2(2); // crea una instancia de un motor conectado a la puerta M2 derecho

void setup()
{
    Serial.begin(9600); // set up Serial library at 9600 bps
    Motor1.setSpeed(255); //definimos la velocidad
    Motor2.setSpeed(255); //definimos la velocidad
}

```




Manual de Robot Sumo con Arduino



```

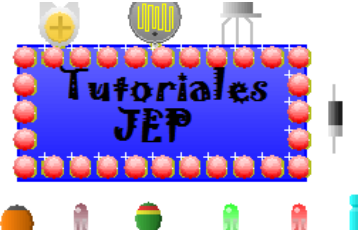
}
void loop()
{
// avanza
  Motor1.run(FORWARD); // avanza
  Motor2.run(FORWARD); // avanza
  delay (2000) ;
//detengo
  Motor1.run(RELEASE); // detengo motor
  Motor2.run(RELEASE); // detengo motor
  delay (2000);
// retrocedo
  Motor1.run(BACKWARD); // retrocedo
  Motor2.run(BACKWARD); // retrocedo
  delay (2000);
//detengo
  Motor1.run(RELEASE); // detengo motor
  Motor2.run(RELEASE); // detengo motor
  delay (2000);
//derecha()
  Motor1.run(FORWARD); // avanza
  Motor2.run(BACKWARD); // retrocedo
  delay(1000);
//detengo
  Motor1.run(RELEASE); // detengo motor
  Motor2.run(RELEASE); // detengo motor
  delay (2000);
//izquierda
  Motor1.run(BACKWARD); // retrocedo
  Motor2.run(FORWARD); // avanza
  delay(1000);
//detengo
  Motor1.run(RELEASE); // detengo motor
  Motor2.run(RELEASE); // detengo motor
  delay (2000);
}

```

Detalle del uso de pines

Este Motor Shield V1 se conecta directamente sobre el arduino y para saber que pines usa y cuales deja libre hemos añadido este pequeño anexo especificando los pines que usa cada motor y cuales están disponibles en caso contrario.

Arduino	Función
Pin 1	Disponible
Pin 2	Disponible
Pin D3	DC Motor #2 / Stepper #1
Pin D4	Reservado para gobernar el Shift Register 74HC595
Pin D5	DC Motor #3 / Stepper #2
Pin D6	DC Motor #4 / Stepper #2
Pin D7	Reservado para gobernar el Shift Register 74HC595
Pin D8	Reservado para gobernar el Shift Register 74HC595
Pin D9	Servo #1, Disponible si no se usa el servo

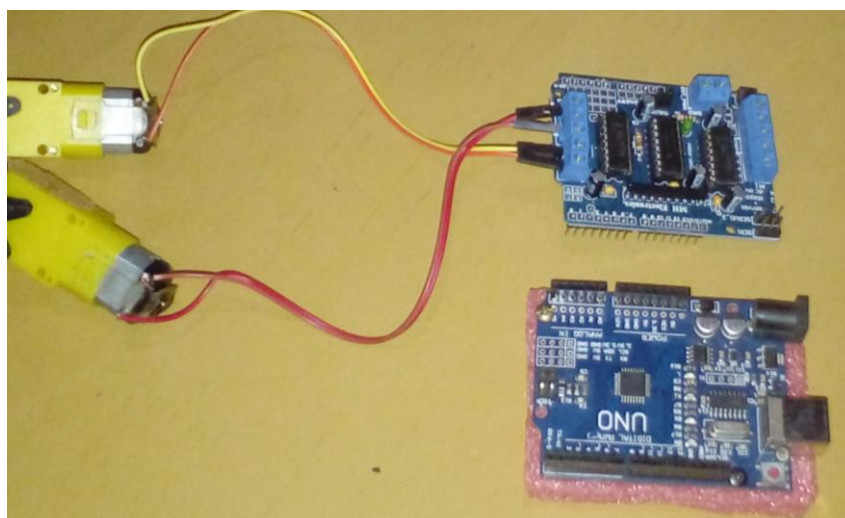


Manual de Robot Sumo con Arduino

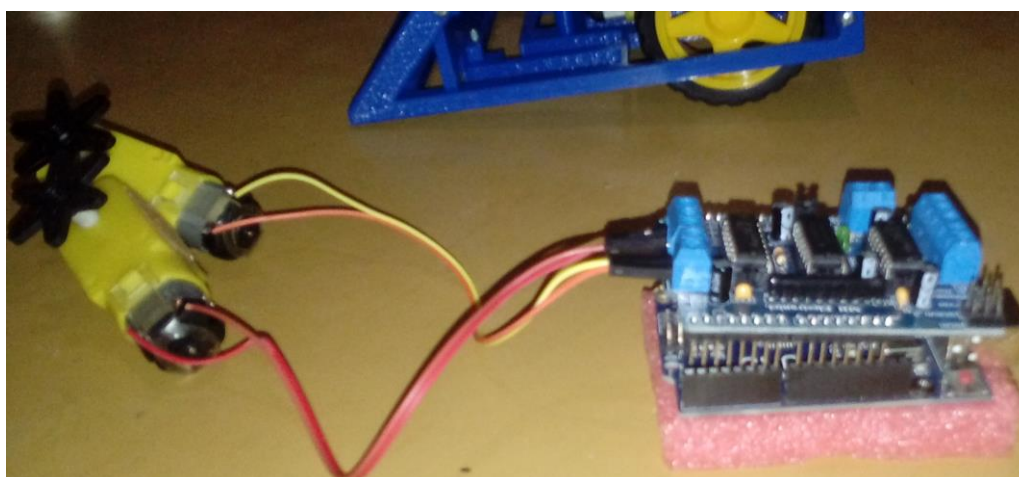
Arduino	Función
Pin D10	Servo #2, Disponible si no se usa el servo
Pin D11	DC Motor #1 / Stepper #1
Pin D12	Reservado para gobernar el Shift Register 74HC595
Pin D13	Disponible
A0,A1,A2,A3,A4,A5	Disponibles. Recuerda que pueden usarse como digitales

Tener presente que no hay ningún pin de **Arduino** conectado a los motores directamente. La gestión de los motores se realiza a través del **shift register** para ahorra pines y por eso tenéis necesariamente que usar la librería para manejarlos.

En la siguiente imagen se observa el arduino uno y el shiel motor V1 separados.



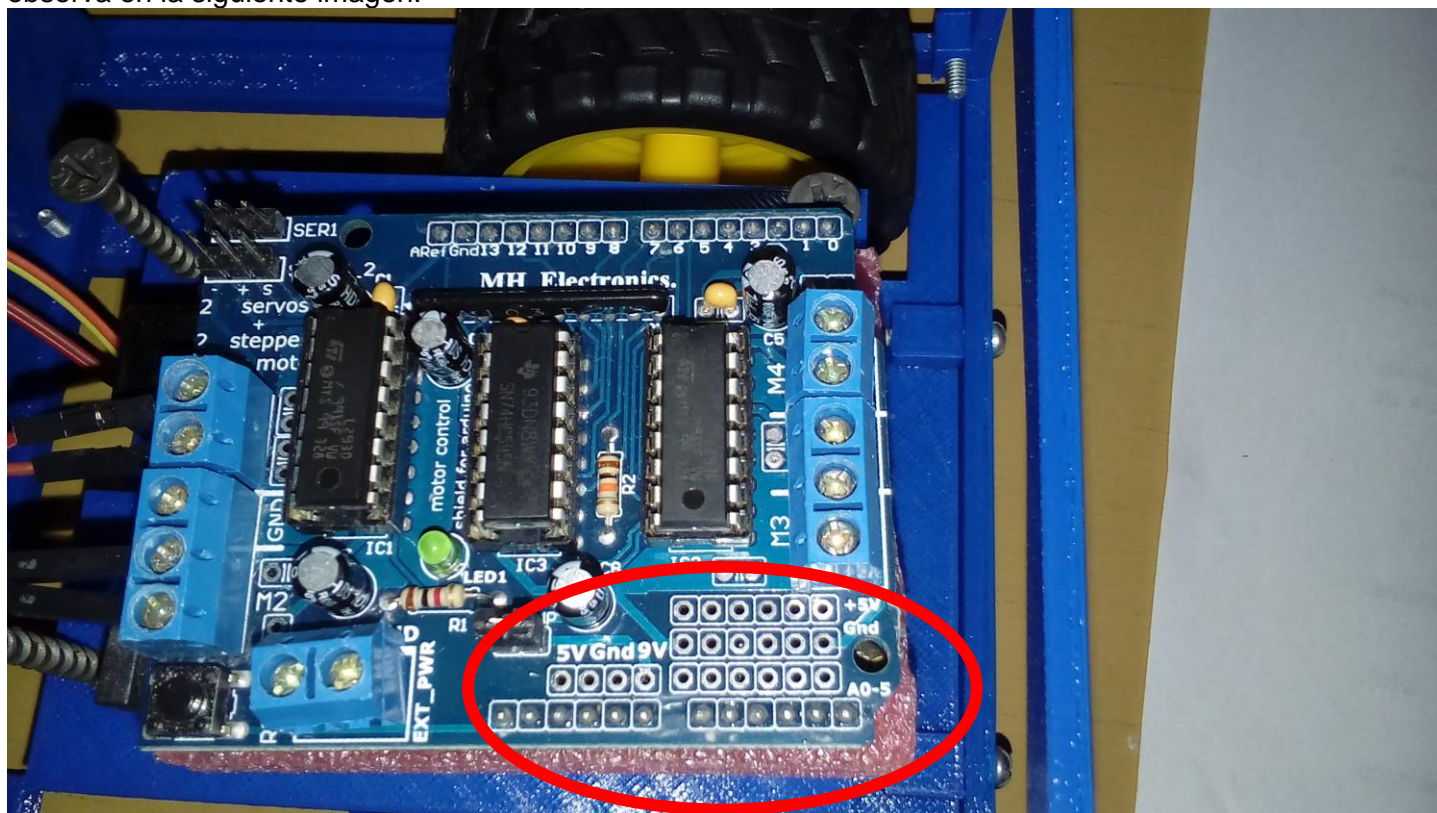
Aquí montamos el shield en el arduino.



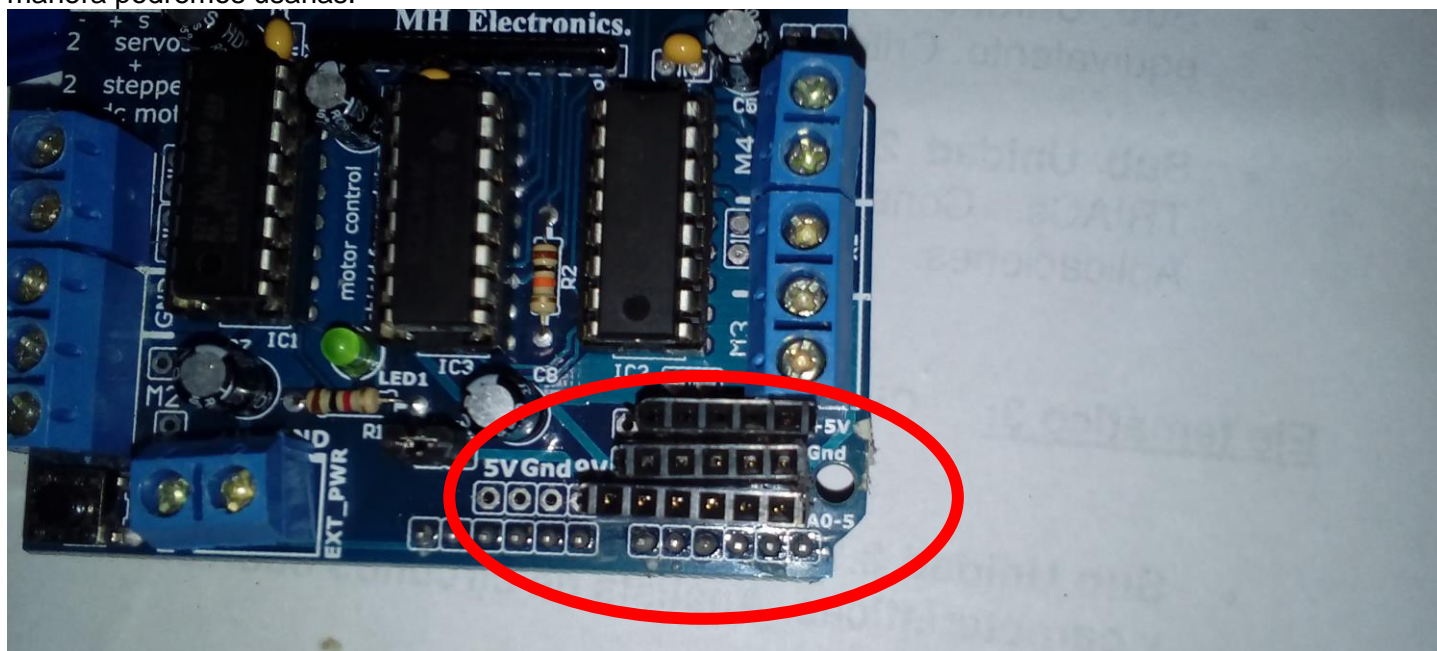


Manual de Robot Sumo con Arduino

Ahora como vemos si usamos el shield tal cual sale de fábrica, no podremos usar los pines de alimentación y las entradas analógicas ya que el shield no las usa pero las tiene conectadas como se observa en la siguiente imagen:



Pero nos provee orificios para poder usar esas salidas, soldaremos en ellas pines hembra y de esta manera podremos usarlas.



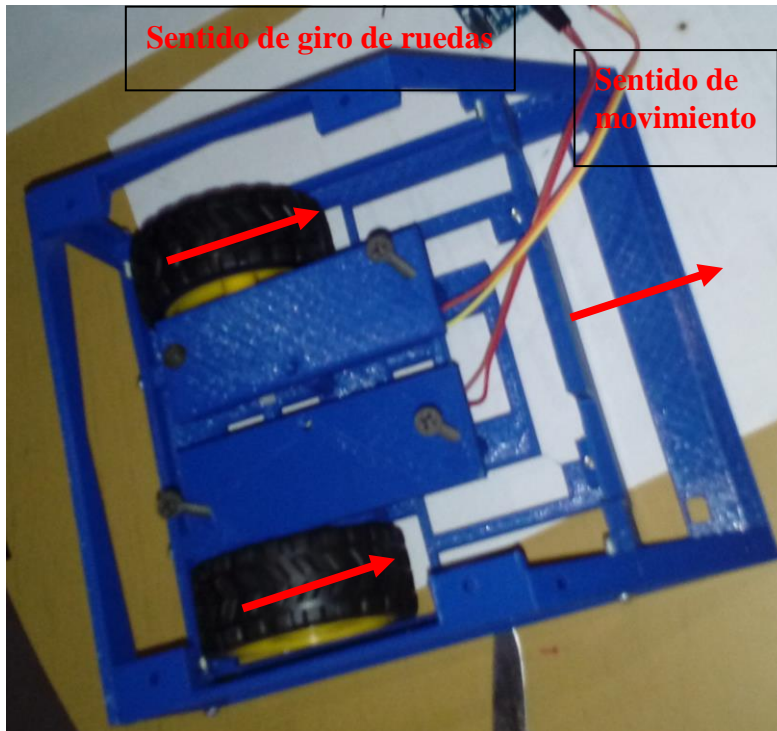


Manual de Robot Sumo con Arduino

Lógica de movimiento

La estructura que hemos elegido para nuestro robot-sumo 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 (avanzando), esto provoca un movimiento recto hacia delante de la estructura.



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





Manual de Robot Sumo con Arduino

Movimiento hacia la derecha: Se hace girar el motor izquierdo hacia delante y el motor de la Derecha hacia atrás. Esto provoca un movimiento de giro ala Derecha de la estructura.



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

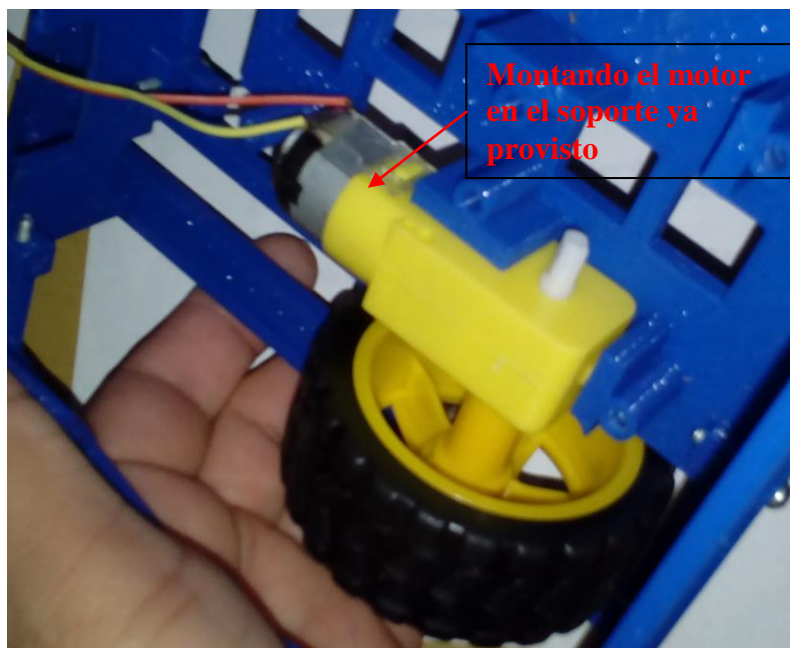
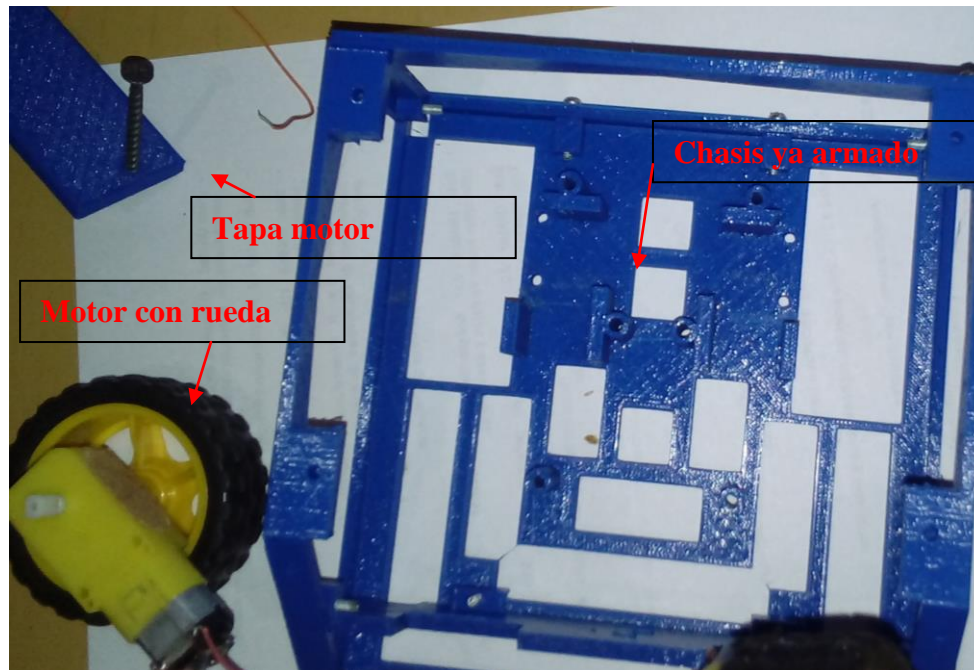


Manual de Robot Sumo con Arduino

Montaje de los componentes.

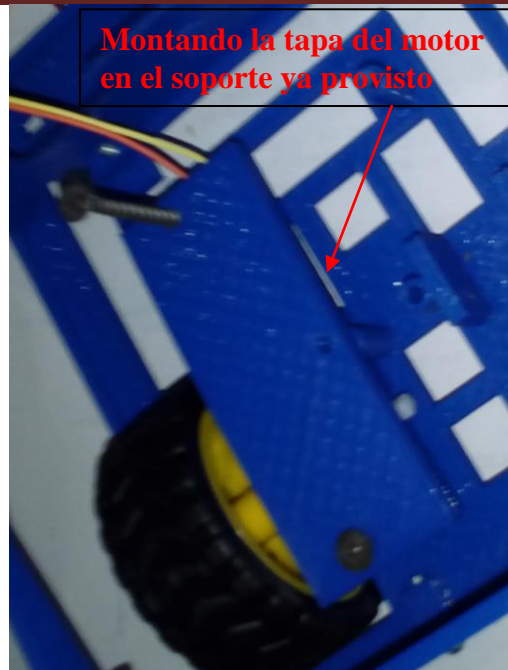
Cuando ya se han probado todos los componentes por separado lo siguiente es montar todo el circuito electrónico, sensores, motores y control en el chasis. Recomendamos el uso de silicona y tornillos, la silicona además de ayudar a fijar los componentes también les permitirá **aislar los circuitos y evitar cortos circuitos**.

Primero se deben montar los motores como se observa a continuación:





Manual de Robot Sumo con Arduino



Montando la tapa del motor en el soporte ya provisto



Ambos motores ya montados

Manual de Robot Sumo con Arduino

Si colocáramos un motor pololu, utilizaremos los Brackets o Bases para micromotor Pololu, coloca la base como se muestra en la imagen:

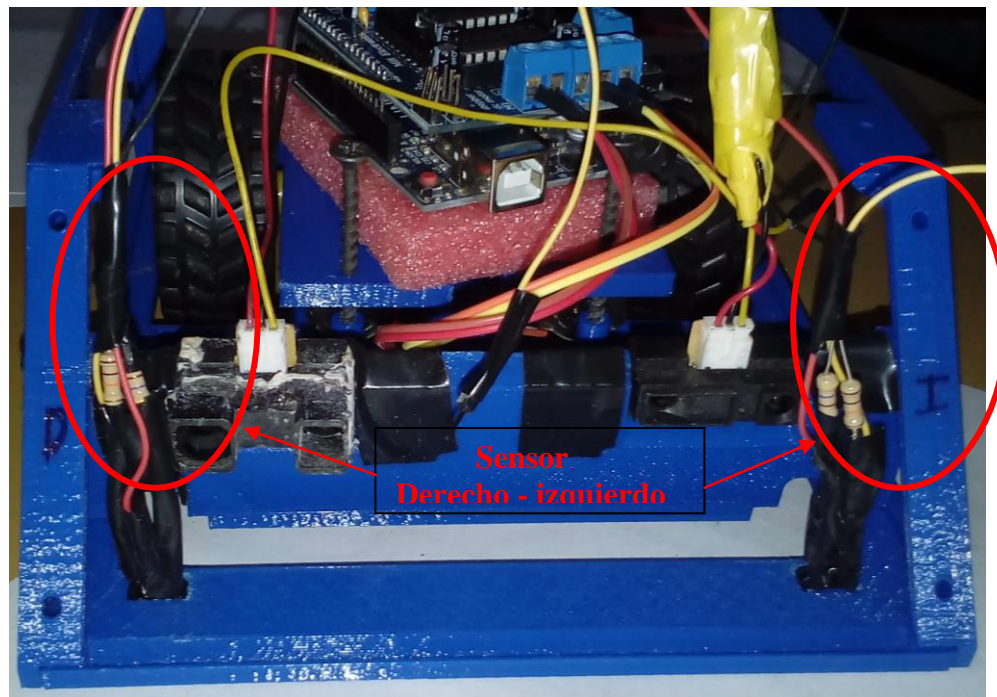


Y atornilla ambos motores con su base en el chasis:

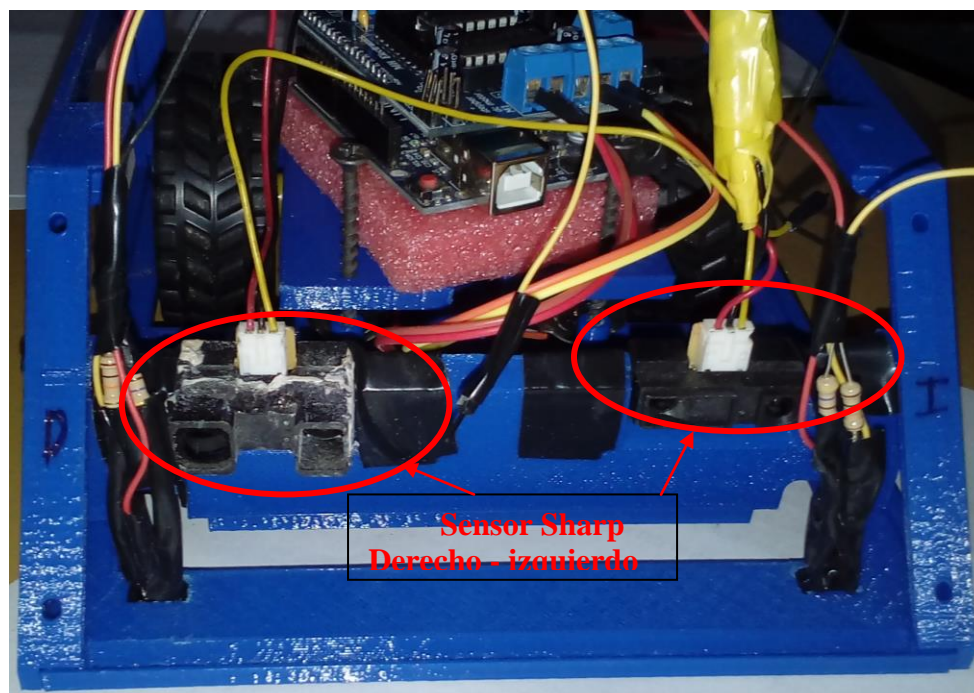


Manual de Robot Sumo con Arduino

Luego los sensores para detectar la línea, esos deben ir colocados por dentro del chasis de modo que el sensor encaje con el orificio asignado, en la siguiente imagen podrán ver cómo debe quedar colocado el sensor. Para mayor comodidad recomendamos desarmar el chasis.



Lo siguiente es colocar los sensores de distancia en la parte frontal del Robot Sumo, en la siguiente imagen se muestra como debe quedar colocado el sensor:

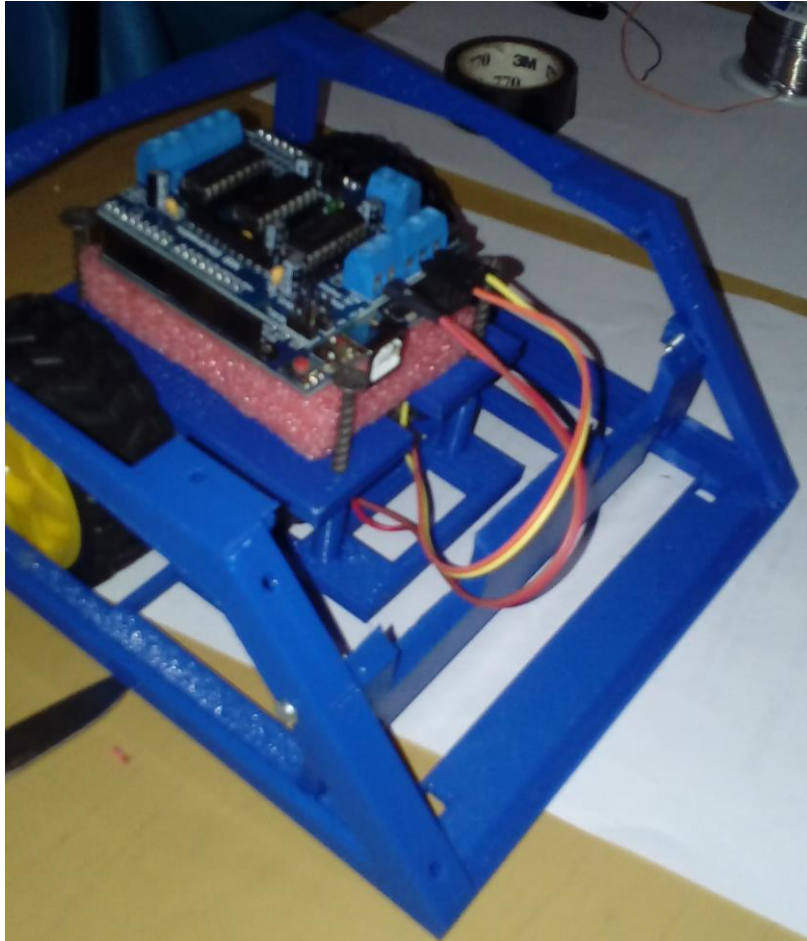




Manual de Robot Sumo con Arduino



Colocar el Arduino en la base destinada para este:

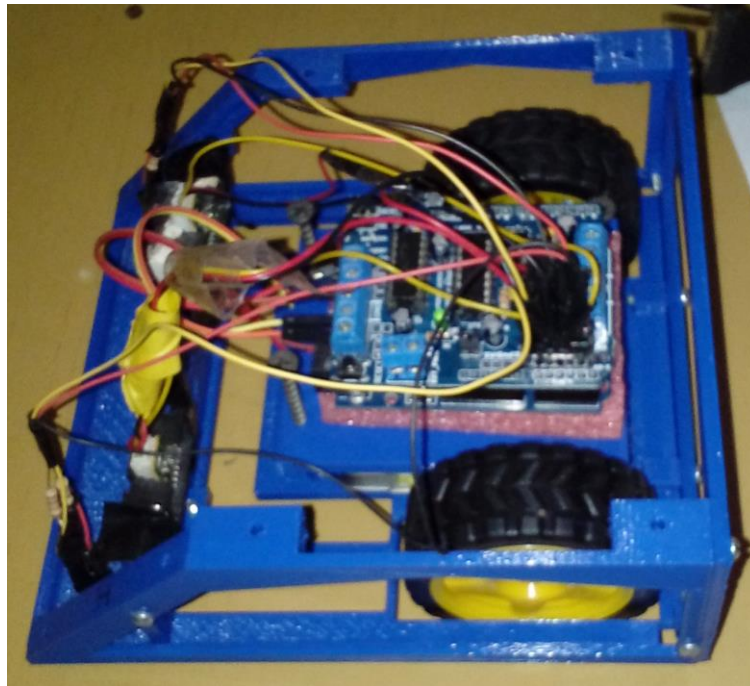


Colocar la batería dentro del Robot Sumo y agregarle peso para que califique en la categoría en la que desee participar.

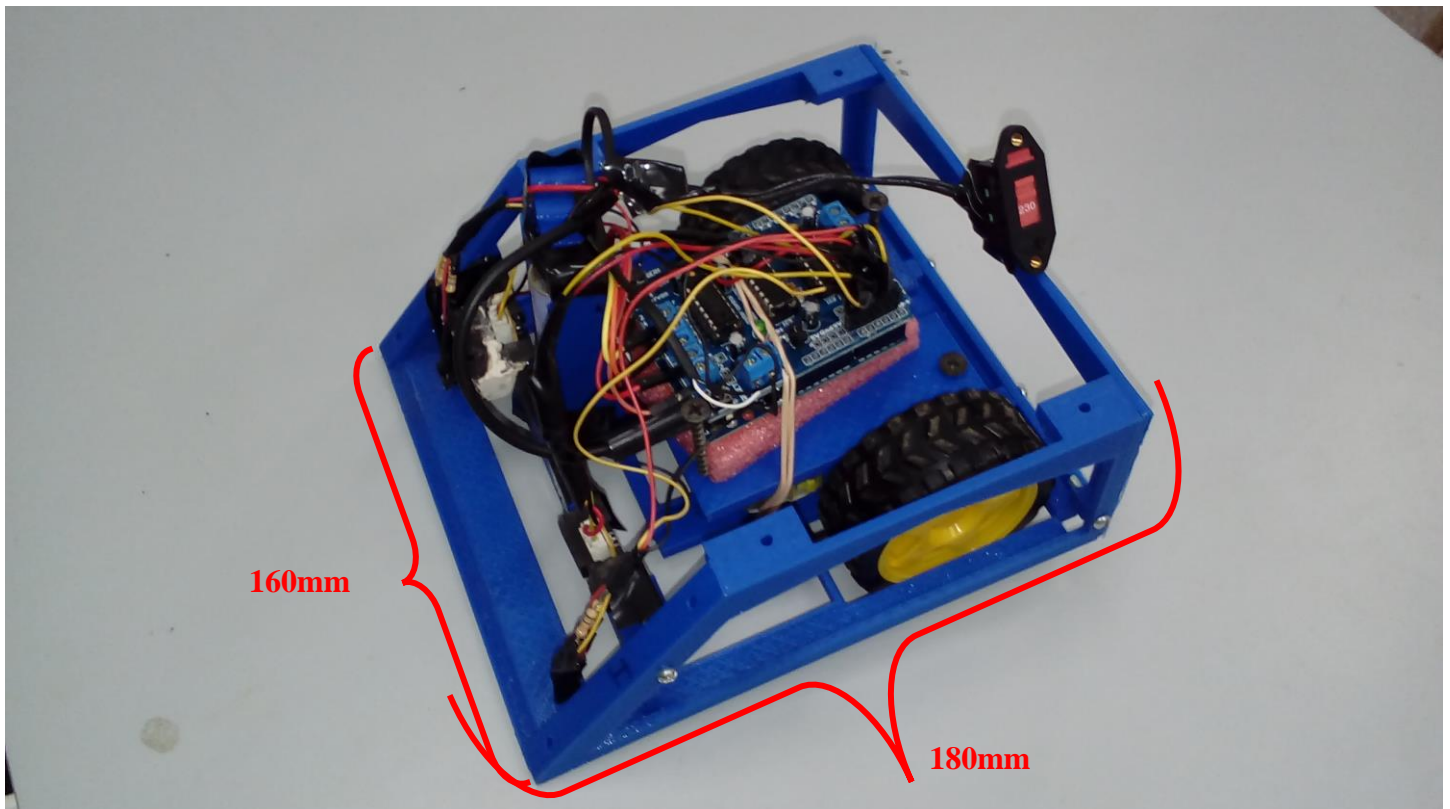


Manual de Robot Sumo con Arduino

Finalmente conectas todo de acuerdo a los circuitos anteriormente vistos y está todo listo para programarlo con la aplicación final:



Recomendamos revisar cuidadosamente todas las conexiones, debido a que el sistema ya incluye demasiados módulos es muy fácil cometer errores. A continuación vemos el Robot sumo ya armado con los visto hasta ahora.





Manual de Robot Sumo con Arduino



Programación de la aplicación final

Para unir todas las funciones se hace un programa que revisa los sensores de línea y los sensores de distancia para tomar una decisión de hacia dónde debe moverse:

```

/*****
/* Autor: Palacios José Emanuel docente de la EETPN°602 */
/* Programa: Sumo Arduino Básico */
/* Año:2018 */
*****/

// Conexiones y librerías
#include <AFMotor.h> //incluye la librería de del shield en nuestro programa
AF_DCMotor Motor1(1); // crea una instancia de un motor conectado a la puerta M1 izquierdo
AF_DCMotor Motor2(2); // crea una instancia de un motor conectado a la puerta M2 derecho

// sensor de línea izquierdo conectado en A0
// sensor de línea derecho conectado en A1
// sensor de distancia izquierdo conectado en A2
// sensor de distancia derecho conectado en A3

// Variables
int li_izq_dela =0; // variable del sensor de linea izquierdo delantero
int li_der_dela =0; // variable del sensor de linea derecho delantero
int dis_izq = 0; // variable de distancia izquierdo
int dis_der = 0; // variable de distancia derecho

// ***** Inicialización *****
void setup() {
    Serial.begin(9600);
    Serial.println("Iniciando lectura de sensor");
    delay(5000); // cuando se active el robot esperamos los 5 segundos antes de iniciar.
    //este tiempo se debe cronometrar para asegurarnos de que sea el correcto
}

//***** programa principal *****
void loop() {
    sensores(); //llamo a la función sensores
    //***** sensores de proximidad Sharp *****
    // Estos valores de 250 pueden cambiar dependiendo de cada sensor, cuando es menor a 250 se refiere a que no detecta
    //nada y mayor a 250 es que el sensor está detectando algo sin embargo eso lo debes comprobar con tus sensores.
    //***** sensores de línea blanca o negra *****
    // Para los sensores de línea blanca o negra los valores mayores a 200 indican que detecto la línea blanca,
    // por lo que si cualquiera de los 2 sensores detecta la línea blanca el robot debe regresar
    // sin embargo eso lo debes comprobar con tus sensores.
    // Tú decides si quieres que solo regrese o también quieres que gire a la derecha o izquierda
    //***** TOMA DE DECISIONES *****
    if((li_izq_dela < 200) && (li_der_dela < 200)&&(dis_izq >= 250)&&(dis_der >= 250)){
        // si ambos sensores leen zona negra y ambos sensores de distancia encuentren al enemigo
        adelante_f(); //llamo a la función adelante fuerte
    }
    if((li_izq_dela < 200) && (li_der_dela < 200)&&(dis_izq < 250)&&(dis_der >= 250)){
        // si ambos sensores leen zona negra y encuentre al enemigo con el sensor derecho
        derecha_f(); //llamo a la función derecha fuerte
    }
    if((li_izq_dela < 200) && (li_der_dela < 200)&&(dis_izq >= 250)&&(dis_der < 250)){
        // si ambos sensores leen zona negra y encuentre al enemigo con el sensor izquierdo
    }

```




Manual de Robot Sumo con Arduino



```

izquierda_f();//llamo a la función izquierda fuerte
}
if((li_izq_dela < 200) && (li_der_dela < 200)&&(dis_izq < 250)&&(dis_der < 250)){
  // si ambos sensores leen zona negra y no encuentro al enemigo con los sensores
  izquierda_d();//llamo a la función izquierda despacio
}
if((li_izq_dela >= 200) || (li_der_dela >= 200)){
  // si leo línea blanca con sensor derecho 0 izquierdo
  reversa_f();//llamo a la función reversa fuerte
  delay(700); //la demora dependera de la velocidad del robot
}
//***** FIN TOMA DE DECISIONES *****
// ¿y si me atacan por atras, o por algun lateral?

}
//***** fin programa principal *****

// funciones creadas por mi para simplificar el programa principal

//***** definir velocidad de motor *****
// Estos valores de 255 o 125 pueden cambiar dependiendo de cada motor ,
// cuando 255 se refiere a que se le envia toda la potencia, y 125 la mitad de la potencia
// sin embargo eso lo debes comprobar con tus motores y elegir la velocidad adecuada.
//*****
void sensores() // lectura de sensores
{
  li_izq_dela = analogRead(A0); // Lectura del sensor de línea delantero izquierdo
  li_der_dela = analogRead(A1); // Lectura del sensor de línea delantero derecho
  dis_izq= analogRead(A2) ; // Lectura del sensor SHARP izquierdo
  dis_der= analogRead(A3) ; // Lectura del sensor SHARP derecho
}
void adelante_d() // muevo el robot hacia adelante despacio
{
  Motor1.setSpeed(125); //definimos la velocidad
  Motor2.setSpeed(125); //definimos la velocidad
  Motor1.run(FORWARD);// avanza
  Motor2.run(FORWARD);// avanza
}
void adelante_f() // muevo el robot hacia adelante fuerte
{
  Motor1.setSpeed(255); //definimos la velocidad
  Motor2.setSpeed(255); //definimos la velocidad
  Motor1.run(FORWARD);// avanza
  Motor2.run(FORWARD);// avanza
}
void reversa_f() // muevo el robot en reversa fuerte
{
  Motor1.setSpeed(255); //definimos la velocidad
  Motor2.setSpeed(255); //definimos la velocidad
  Motor1.run(BACKWARD);// retrocedo
  Motor2.run(BACKWARD);// retrocedo
}
void derecha_d() // muevo el robot hacia la derecha despacio
{
  Motor1.setSpeed(125); //definimos la velocidad
  Motor2.setSpeed(125); //definimos la velocidad
  Motor1.run(FORWARD);// avanza
  Motor2.run(BACKWARD);// retrocedo
}

```



Manual de Robot Sumo con Arduino



```
void derecha_f() // muevo el robot hacia la derecha fuerte
{
  Motor1.setSpeed(255); //definimos la velocidad
  Motor2.setSpeed(255); //definimos la velocidad
  Motor1.run(FORWARD); // avanza
  Motor2.run(BACKWARD); // retrocedo
}
void izquierda_d() // muevo el robot hacia la izquierda despacio
{
  Motor1.setSpeed(125); //definimos la velocidad
  Motor2.setSpeed(125); //definimos la velocidad
  Motor1.run(BACKWARD); // retrocedo
  Motor2.run(FORWARD); // avanza
}
void izquierda_f() // muevo el robot hacia la izquierda fuerte
{
  Motor1.setSpeed(255); //definimos la velocidad
  Motor2.setSpeed(255); //definimos la velocidad
  Motor1.run(BACKWARD); // retrocedo
  Motor2.run(FORWARD); // avanza
}
void detengo() // detengo el robot
{
  Motor1.run(RELEASE); // detengo motor
  Motor2.run(RELEASE); // detengo motor
}
```

```

/***** TABLA DE VERDAD DE TOMA DE DECISIONES *****/
* LINEA IZQUIERDA* LINEA DERECHA * SHARP DERECHO * SHARP IZQUIERDO* ACCIÓN *
* 0 * 0 * 1 * 1 * ADELANTE FUERTE *
* 0 * 0 * 1 * 0 * DERECHA FUERTE *
* 0 * 0 * 0 * 1 * IZQUIERDA FUERTE *
* 0 * 0 * 0 * 0 * IZQUIERDA DESPACIO *
* 1 * 1 * - * - * ATRAS FUERTE *
*****/

```

Ahora solo queda:

- Calibrar los sensores con el programa en un dojo (de dimensiones según el reglamento) y con un contrincante.
- 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 de Robot Sumo en el canal de youtube TUTORIALES JEP. También se encontraran los archivos STL para imprimir con la impresora 3d.

<https://youtu.be/DdqFkJAV6l4>

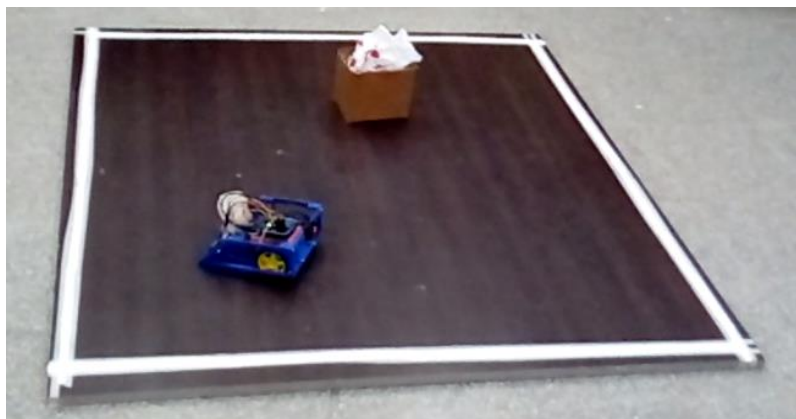


Manual de Robot Sumo con Arduino

Foto ilustrativa del dojo:



Foto ilustrativa de entrenamiento precompetitivo:



En la imagen anterior se observa la prueba de un robot sumo contra un contrincante inmóvil (caja de cartón) en un dojo fuera de reglamento (cuadrado y sin las medidas correspondientes, etc), sin embargo sirve en principio para ir testeando el funcionamiento del robot



Manual de Robot Sumo 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>