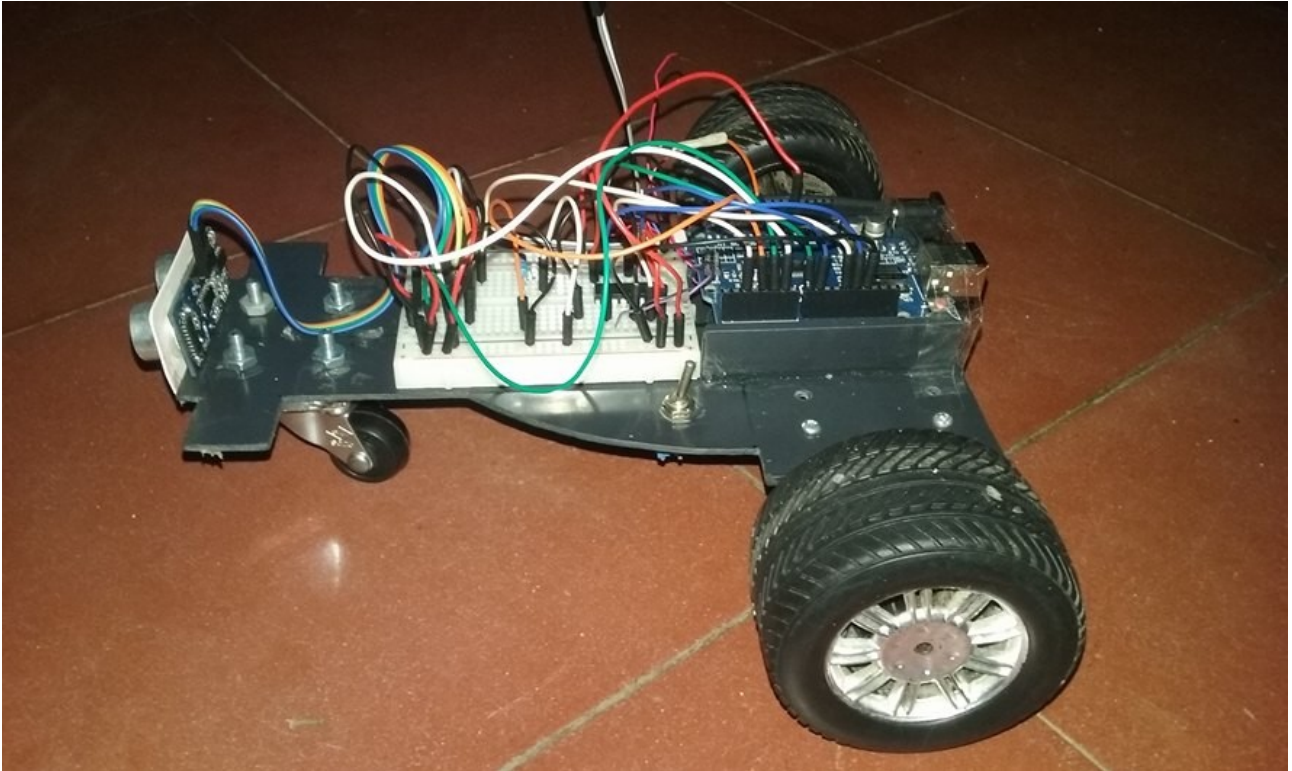


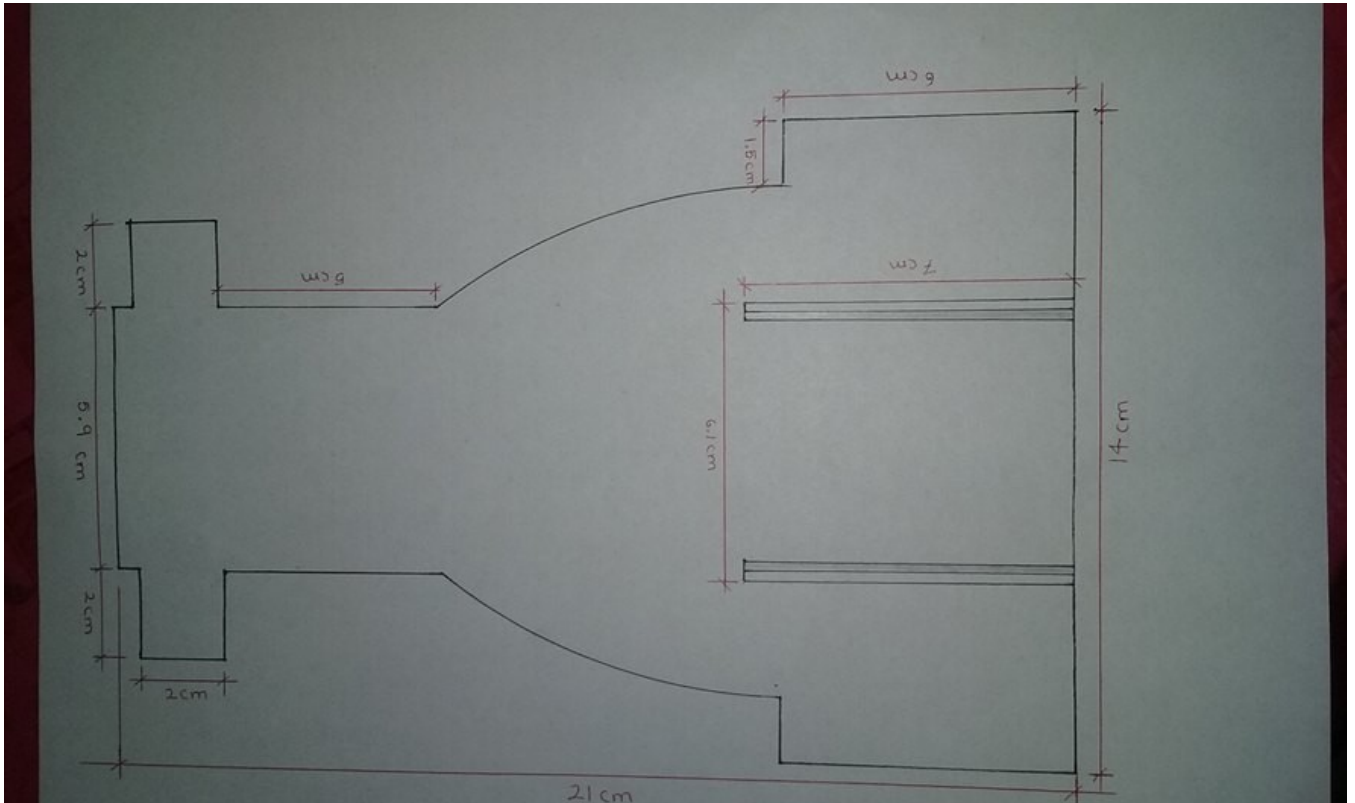
Manual de como armar el carro a control remoto



Así es como quedara cuando este totalmente armado, nosotros utilizaremos los siguientes materiales para llevar acabo este proyecto:

- 2 baterias de 3.7V
- 880mAh
- 1 keyboard dañado
- 2 llantas de algun carrito
- 4 pernos de 1pulg con 3 turcas c/u(para la llanta loca)
- 1 breadboard pequeña
- 1 placa arduino UNO
- 1 sensor IR resector
- cables para breadboard
- 1 resistencia de 220 Ohm
- 8 cables macho/hembra para breadboard
- 1 led cualquier color
- 4 pernos de 1pulg con tuerca(para sujetar los motores)
- 1 llanta loca
- 2 motores DC de 12V
- 1 IC L293B o L293D
- 1 switch o interruptor de palanquita
- pistola de silicona caliente
- 1 taladro
- 1 sensor Ultrasonido HC
- SR04
- 1 cautin
- tenasas, desarmadores

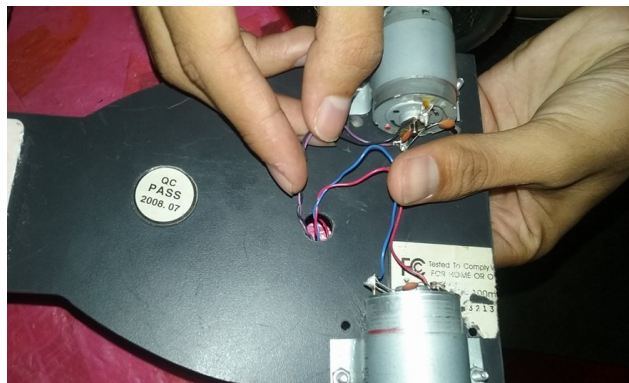
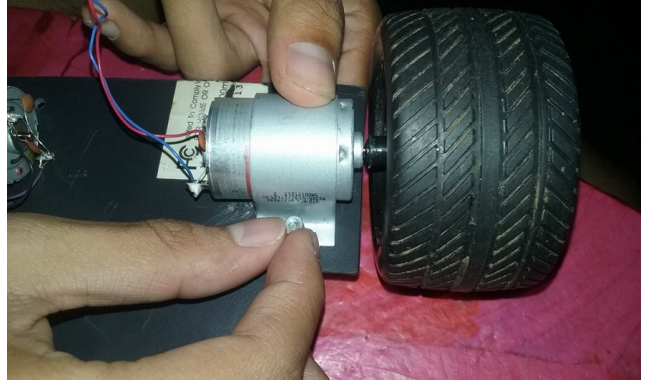
El diseño del chasis es el siguiente:



El primer paso sería crear el chasis el cual llevara montado todo lo que hará que este funcione en nuestro caso utilizamos como base un plástico cuadrado extraído de un teclado y se le dio la siguiente forma

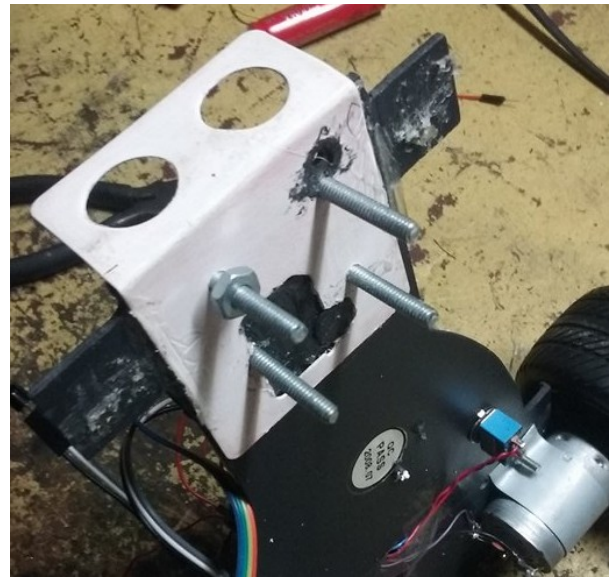


como se puede observar se le hicieron orificios en los cuales irán las abrazaderas para sujetar los motores de las llantas traseras como se muestra a continuación

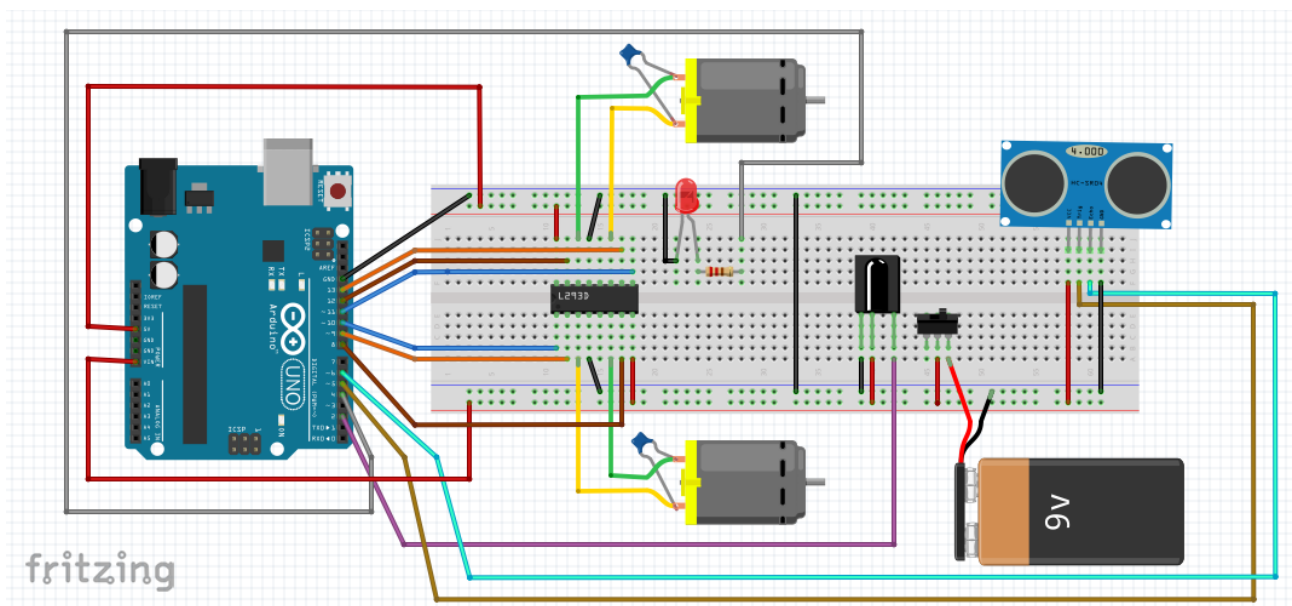


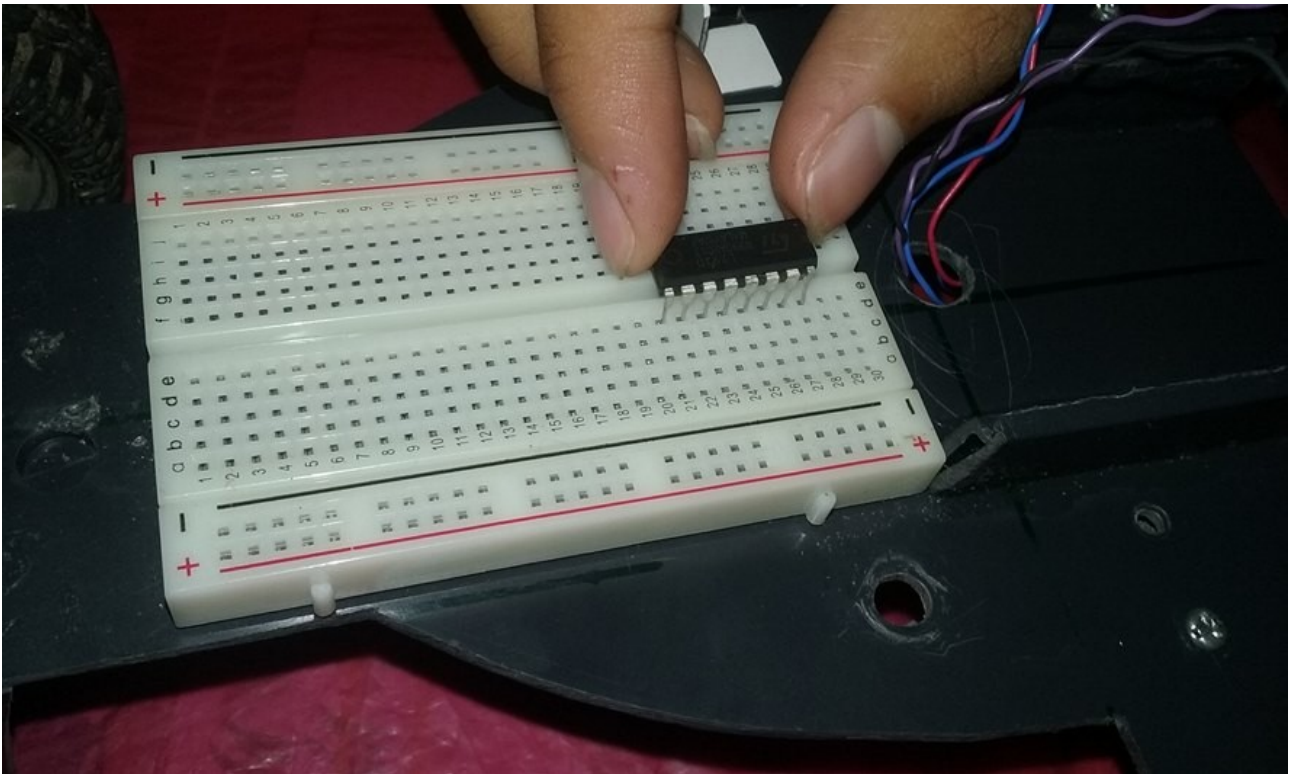
Para la parte de adelante colocamos la llanta giratoria haciéndolo de la siguiente manera lo cual nos permitirá que esta esté bien sujetadas como se muestra a continuación



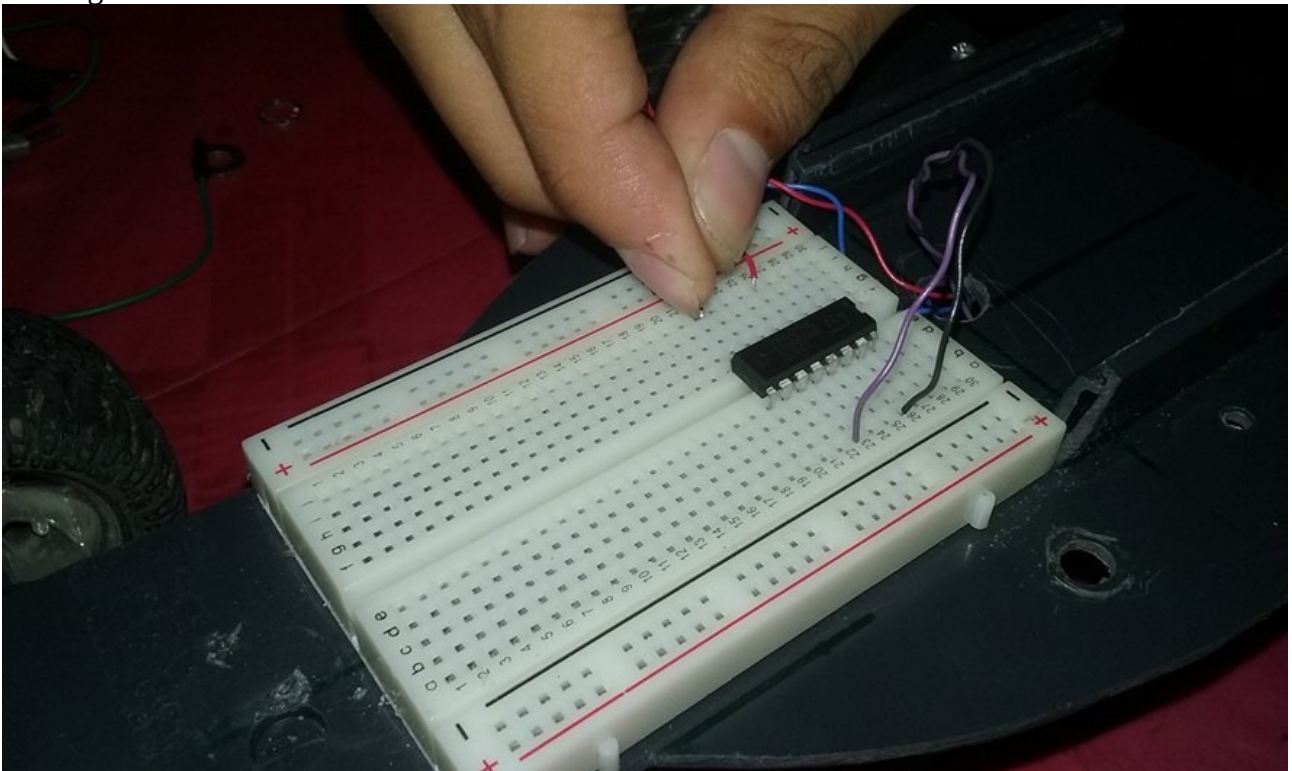


Después de haber adaptado de una forma muy segura las llantas al chasis podemos proseguir con el armado del carrito para ello proseguiremos con la agregación de la breadboard pequeña y IC L293D también conocido como puente H, quedándonos de la siguiente forma

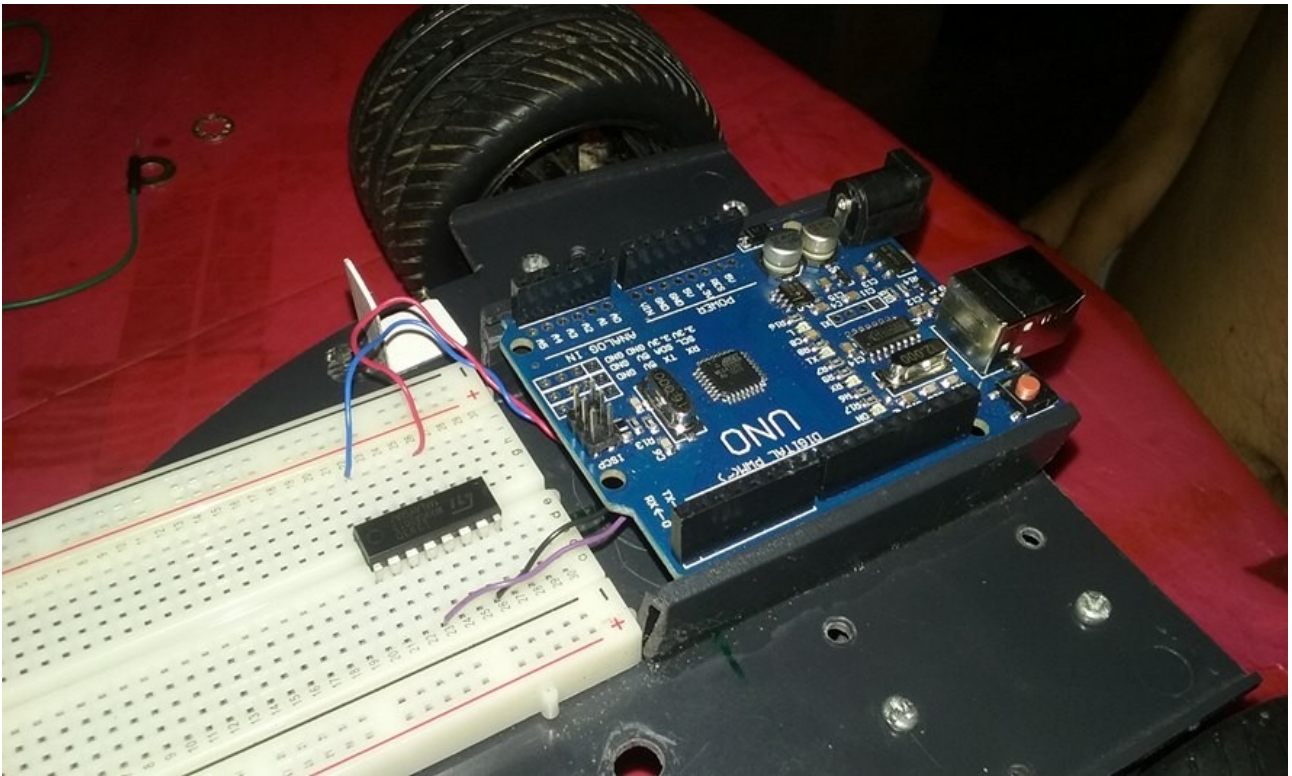
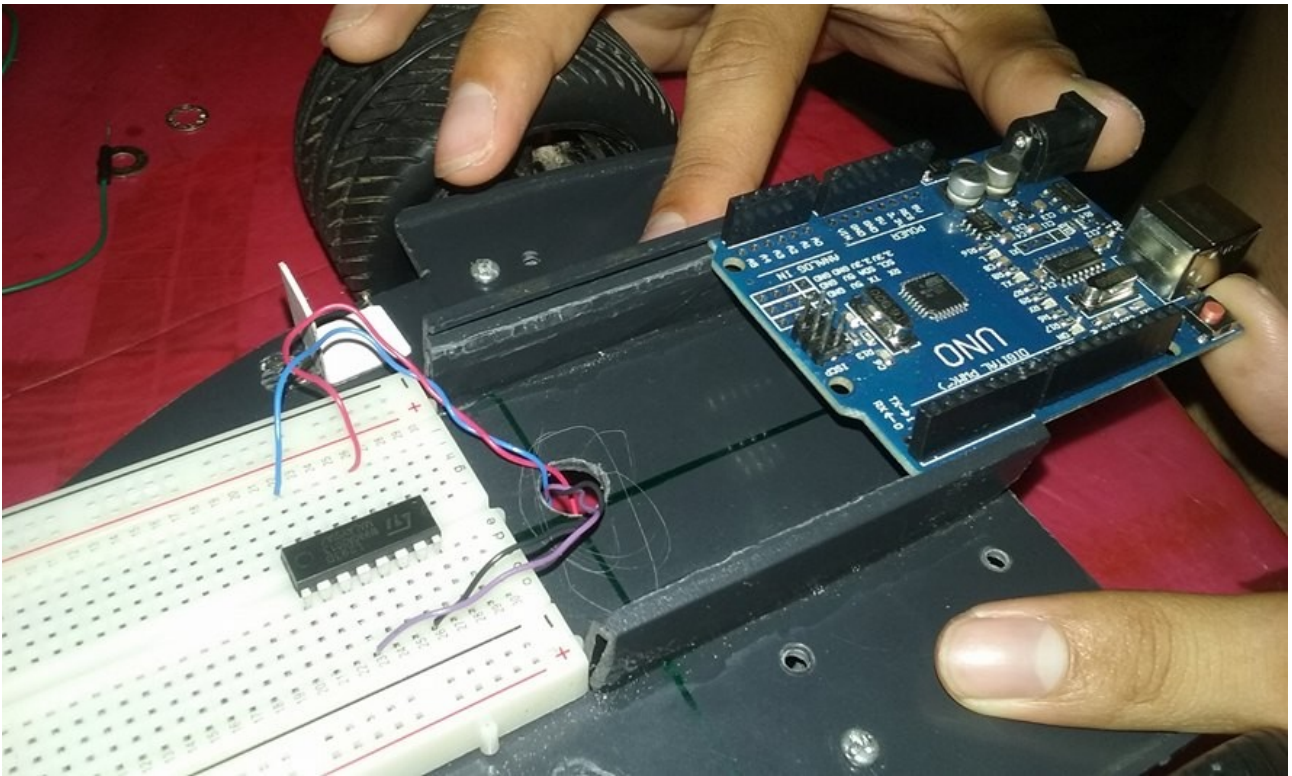




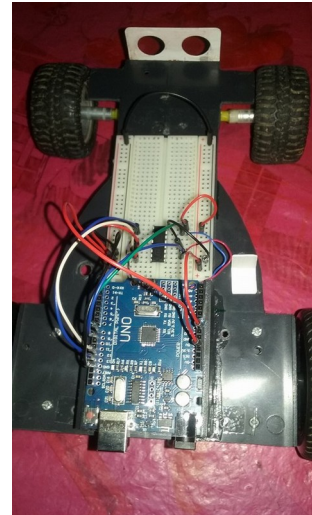
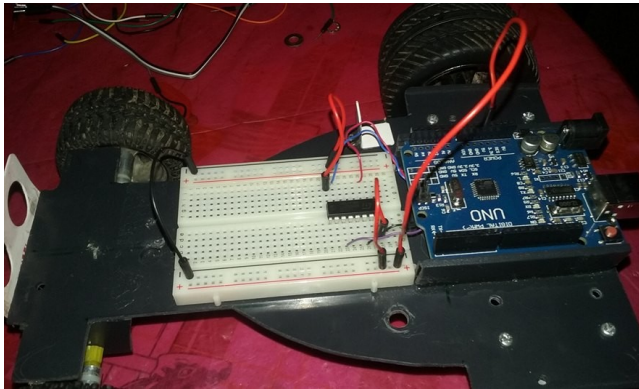
Conectamos los cables de los motores en la breadboard a modo que puedan ser controlados por los pin 3,6,11 y 14 del IC L293D tal como se ve en la imagen siguiente y como se indica en el diseño Fritzing



Agregamos el Arduino UNO en los soportes que le generamos al chasis exclusivamente para mantener la placa fija y evitar que esta se pueda caer con facilidad



Luego proseguimos a conectar los cables como esta indicando en el diseño Fritzing, esperamos te guíes de el para hacer este pasa de la forma mas correcta posible a continuacion te dejamos una imagen de dicho diseño para que la utilices:



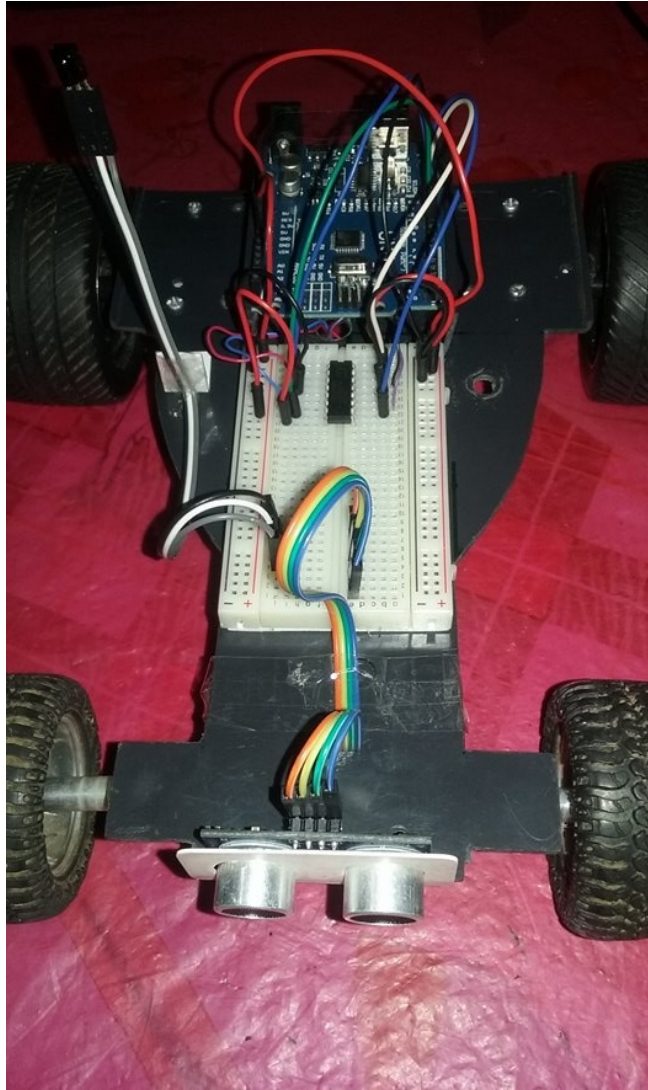
Se hizo una manualidad frente al carro para colocar el ultrasonido el cual se usara para poder detener el carrito en caso que se dirija hacia un objeto con el cual pueda estrellarse, dicha parte es la que se muestra a continuación:



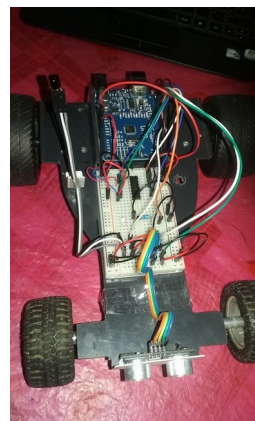
El ultrasonido que conectaremos es el que se muestra en la siguiente imagen



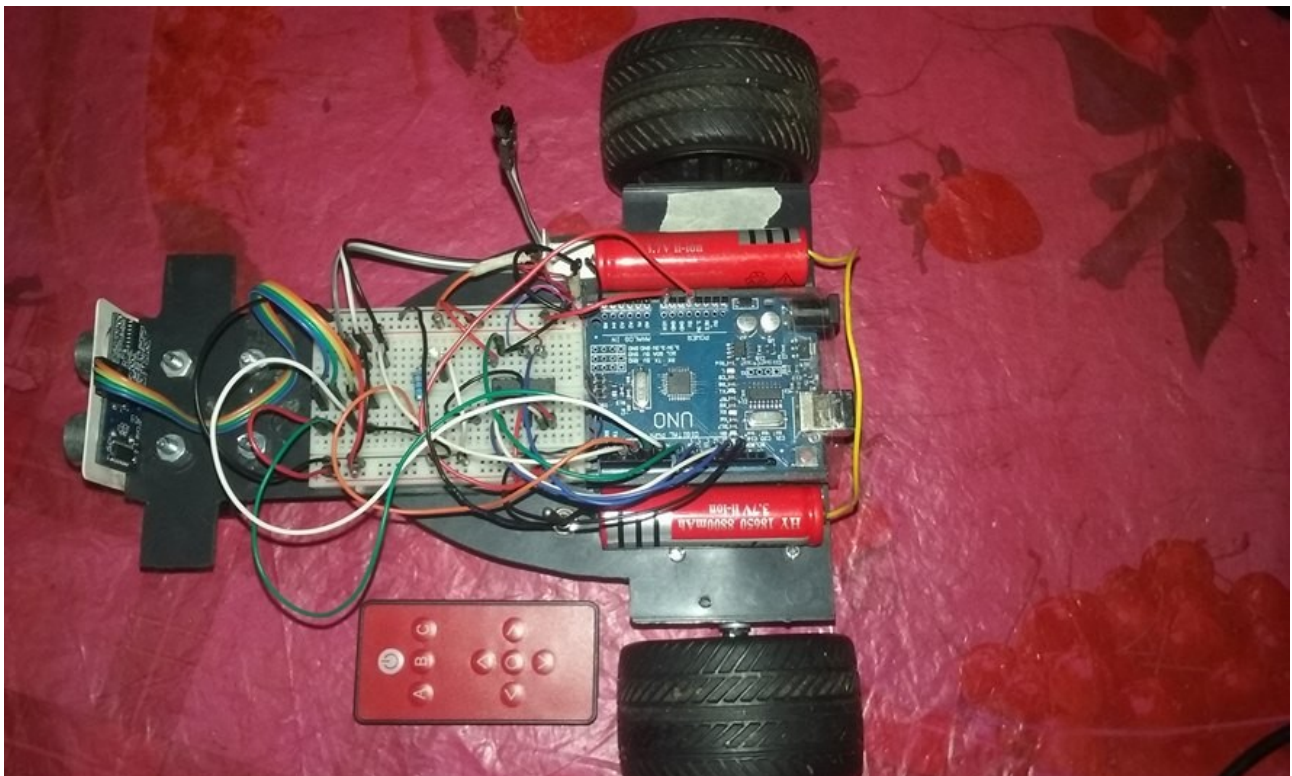
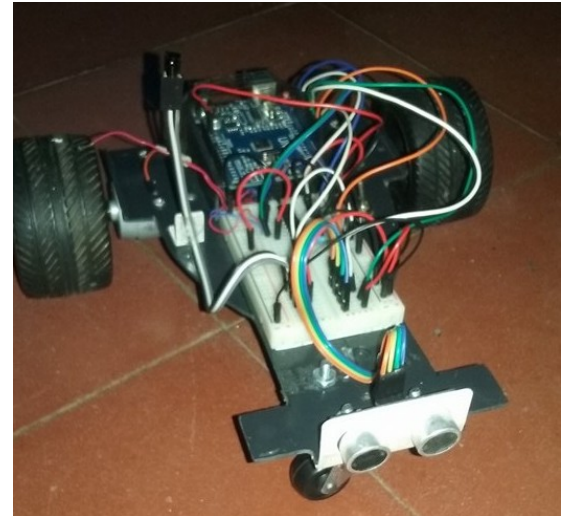
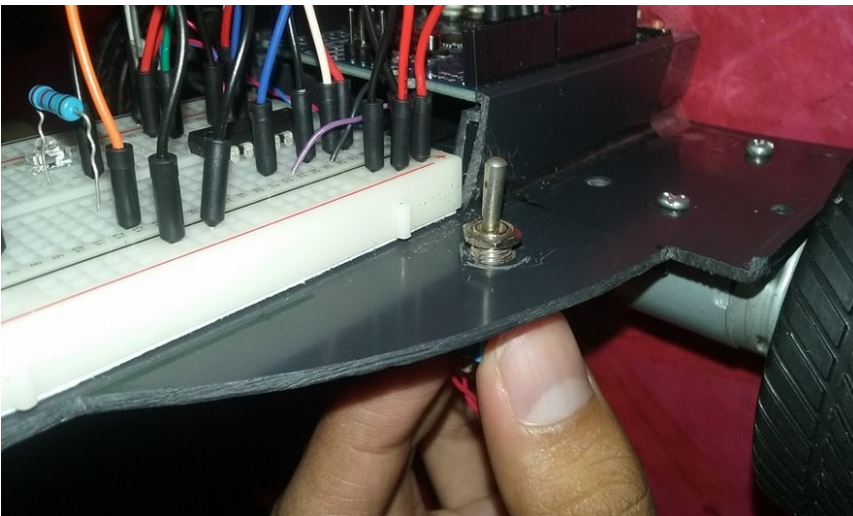
quedándonos de la siguiente forma



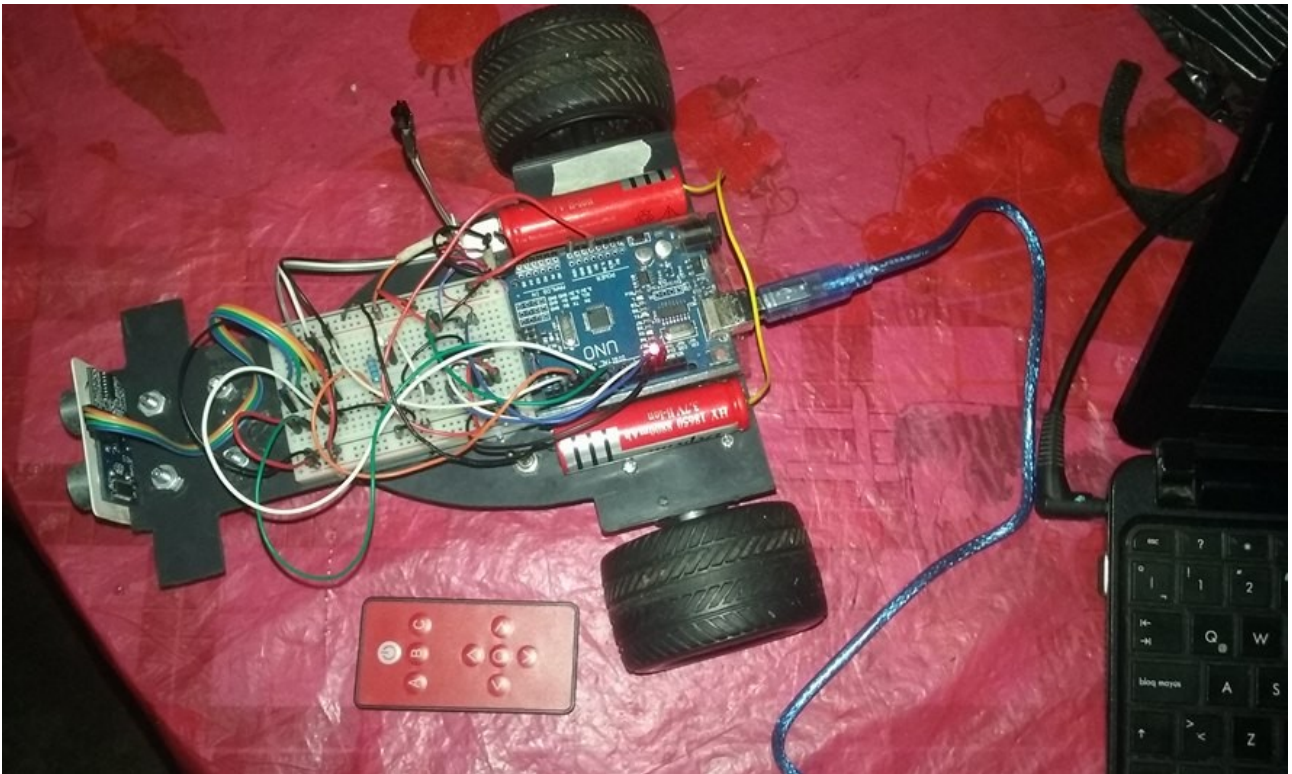
Agregamos el IR para poder darle ordenes a través de un control remoto a nuestro carrito, aunque este debe programarse para el control remoto que utilices, debes abrir el monitor del puerto serial en el IDE de Arduino el cual debes tener instalado, en caso que no lo tengas dirígete a un terminal y con permisos de súper usuario debes digitar o pegar el siguiente comando :~# **apt-get install arduino**, luego habré este IDE y abre el Monitor Serial para poder ver cuales son los códigos que recibe el IR de tu control remoto de esta forma en el código del carrito le asignas dichos códigos a la acción que desees que haga con los botones de tu gusto, pero antes de poder hacer esto conecta dicho IR que es el siguientes



Después de hacer esto procede a conectar un interruptor de corriente para evitar la pérdida de energía de las baterías que estarán alimentando la breadboard



Como puedes observar ya esta todo conectado ahora solo debes copiar y pegar el código en el IDE de Arduino y transferirlo compilado a la placa para ello conecta el Arduino UNO a través del cable USB luego ve a herramientas luego a la opción puerto serial y elige el que corresponde en tu caso, luego da clic en el botón verificar y solo por seguridad para rectificar que pegas el código de una forma correcta y no se borro nada, si no te marca ningún error y te muestra en la parte inferior el tamaño en bytes del compilado procede a cargarlo dando clic en el botón Cargar que es una flecha apuntando hacia el lado derecho, verifica en la parte superior que no te muestre ningún error, ya sea por haber elegido un puerto serial incorrecto, del caso contrario que todo salio a la perfección solo resta probar el carrito, pero recuerda que en un paso anterior a este se te menciono que debías asignar los códigos respectivos al control remoto que estas utilizando, de no haber hecho esto debes hacerlo para que te funcione todo correctamente.



Actividades mar jun 7, 23:09:32 es

Carrito_completo_IR_Ultra_interruccion | Arduino 2:1.0.5+dfsg2-4

Archivo Editar Sketch Herramientas Ayuda

```

Carrito_completo_IR_Ultra_interruccion
#include <IRremote.h> //libreria para usar sensores IR que permite recibir o transmitir códigos de control remoto por infrarrojos.

IRrecv irrecv(2); //variable al que se le pasa el numero del pin donde esta el IRreceptor, Crear el objeto receptor, utilizando un nombre de su el
decode_results results;

long distancia; //varia que guarda la distancia
long tiempo;
int OUT = 5; //Pin TRIGGER
int IN = 6; //Pin ECHO
int revoluciones= 150; //valor de las revoluciones que recibirán los motores

void setup()
{
  pinMode(13, OUTPUT); //Salida del pin activada - pin del motor izquierdo
  pinMode(12, OUTPUT); //Salida del pin activada - pin del motor izquierdo
  pinMode(9, OUTPUT); //Salida del pin activada - pin del motor derecho
  pinMode(8, OUTPUT); //Salida del pin activada - pin del motor derecho
  pinMode(4, OUTPUT); //Salida del pin activada - Pin del Led que detecta las señales del IR emisor,
  pinMode(IN, INPUT); //Salida del pin activada - pin del ECHO del sensor ultrasonido
  pinMode(OUT, OUTPUT); //Salida del pin activada - pin del TRIGGER del sensor ultrasonido
}

Compilación terminada

Tamaño binario del Sketch: 9,334 bytes (de un máximo de 32,256 bytes)

1 Arduino Uno on /dev/ttyUSB2
```