|  |
| --- |
| Proyecto Final  INTL |
| G:\ARDUINO\videos parqueo\compañeros.JPG |
|  |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | G:\ARDUINO\ArduinoUno_R3_Pinouts.png |  | G:\ARDUINO\Pines-Sensor-HC-SR04.png |  | G:\ARDUINO\Dual-H-Bridge-DC-Stepper-Motor-Drive-Controller-Board-Module-Arduino-L298N-HE2.jpg | | | Arduino UNO  Inteligencia Artificial-Auto Parque Solo. |  | Sensor de ultrasonido  Inteligencia Artificial-Auto Parque Solo. |  | Puente doble H  Inteligencia Artificial-Auto Parque Solo. | |

**OBJETIVO.-**Nuestro objetivo es lograr en nuestro auto se estacione de forma automática

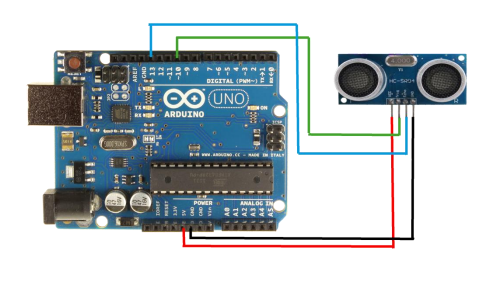
Hacer que nuestros sensores obedezcan las indicaciones grabadas en el arduino.

**MATERIALES UTILIZADOS.-**

* Arduino (hay muchas versiones, aquí uso el UNO)
* Sensor de ultrasonido HC-SR04 (hay parecidos como HY-SRF05)
* Cables USB tipo A y tipo B
* Chasis del coche con motores y ruedas
* Puente doble H L298N
* Cables tanto macho-hembra como hembra-macho
* Pilas recargables
* Cargador portátil

**DESARROLLO.-**una vez definido el proyectoreunimos el material necesario.

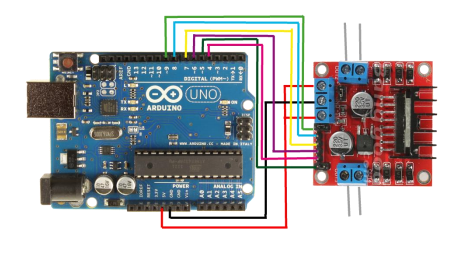
1.-El sensor de ultrasonido: Vcc a 5v, GND a GND como antes, luego Trig a 10 y Echo a 13.



2.- Segundo sensor de ultrasonido: Igual que el anterior, pero Trig a 11 y Echo a 12

3- Puente H:

Este es el componente más complicado de instalar. Primero, Vcc y %v del puente a 5v del Arduino, GND a GND como siempre, luego EN1 a 9, In1 a 8, In2 a 7, In3 a 6, In 4 a 4 y En2 a 5 en las salidas digitales. Debemos tener en cuenta unas cuantas cosas, de serie probablemente las entradas “En” vengan con unos jumpers, estos están para habilitar una velocidad constante sin control de PWM, si deseas modificarla tendrás que quitarlos y conectarlo a salidas PWM del Arduino, como lo tenemos aquí montado. Luego las dos salidas de arriba irán a las dos de un motor y las de abajo a su motor correspondiente



El Arduino

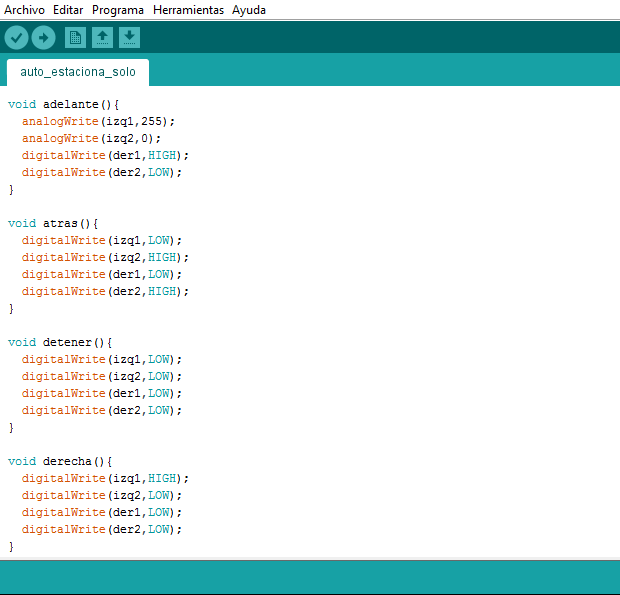
En cuanto al programa es programación en C, con una diferencia, la programación en C es lineal, claro está tienes bucles “for” y demás pero sigue siendo lineal, empiezas de un punto y acabas cuando se ha ejecutado todo desde ese punto hacia abajo. La diferencia es que el Arduino lleva todo en un “void loop”, ósea un bucle constante, lo cual también es lógico al tomar datos del exterior por las entradas. Pero esto tal vez sea lo que más me ha dificultado una tarea como aparcar que es una secuencia he tenido muchos problemas con “ifs”, “fors” (como salirse y empezar a leer desde el principio), etc. Lo cual he tenido que solucionar a base de tiempo y muchas variables auxiliares para vincular un paso al siguiente paso y así. También para delimitar lo que donde debe empezar a leer en su próximo ciclo de lectura, ya que cuantos más “ifs” o “cases” y más variables auxiliares, ira más al grano (por decirlo coloquialmente), e ira más concretamente a lo que le interesa, esto es muy importante a la hora de usar tiempos o sensores, porque cuanto más directo seas más rápido irán los sensores, mayor frecuencia entre muestras y menor errores en acciones temporizadas. Para mi programa he usado la estrategia de dividirlo en otros programas aislando cada funcionalidad para luego juntarlas sabiendo que funciona correctamente cada una por separado.

Partes el programa completo

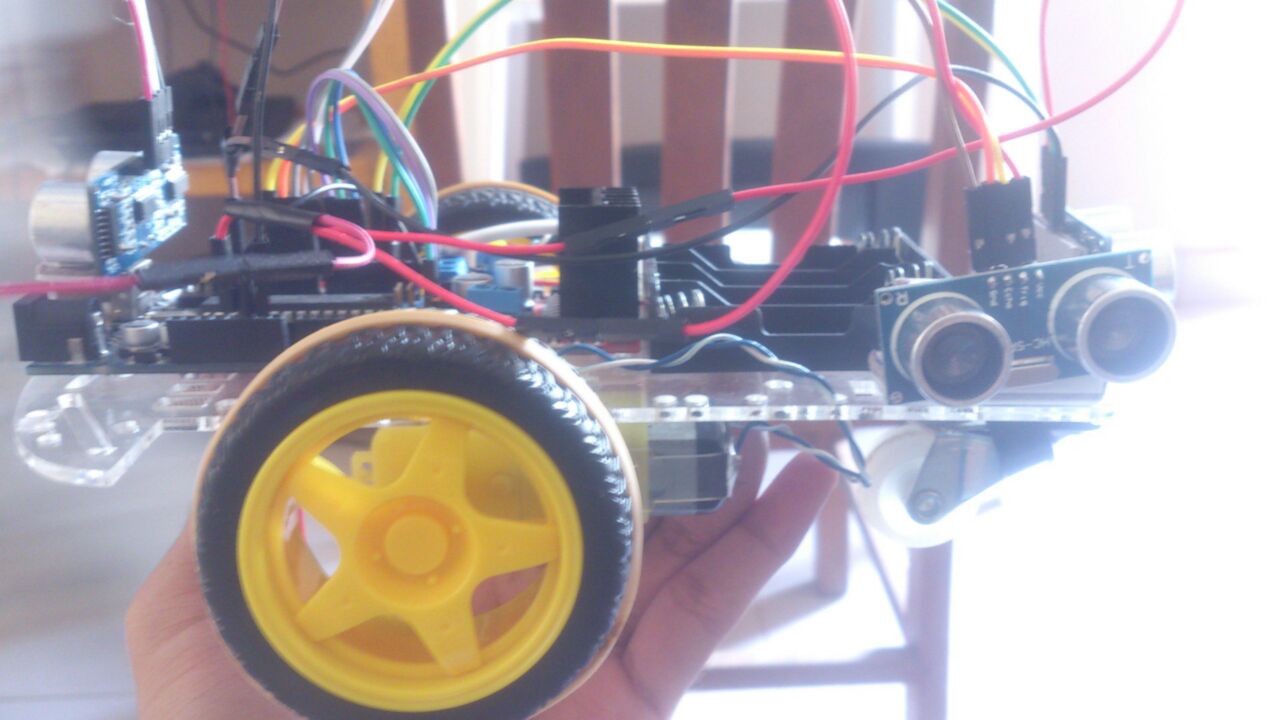
Sensores de ultrasonido: Estos sensores disponen de un rango de medidas, dependiendo de las librerías que usemos, no mayor a 4.5m (dependiendo de dónde te den la hoja de características), con las librearías que he usado yo de 51cm, con menos de 15º de apertura hacia cada lado de su vertical, además las ondas deben rebotar sobre algo casi totalmente paralelo si no quieres tener fallos de lectura. Aun así su margen de lecturas decentes no es a medidas pequeñas (como las que he usado en este proyecto) ni muy grandes, donde mejor ira será en el medio. Posiblemente si usted lo llega a usar verá que de vez en cuando le da algún pico, este pico si le da mientras intenta aparcar le puede fastidiar el programa, mi solución fue hacer que las medidas hagan una media de cada diez medidas, lo puedes ajustar para el numero de medidas que crea, cuanto más grande menos picos tendrá, pero más tardara en medir, aun así haciendo media de diez valores si suprimimos los delays al mínimo ira muy rápido.

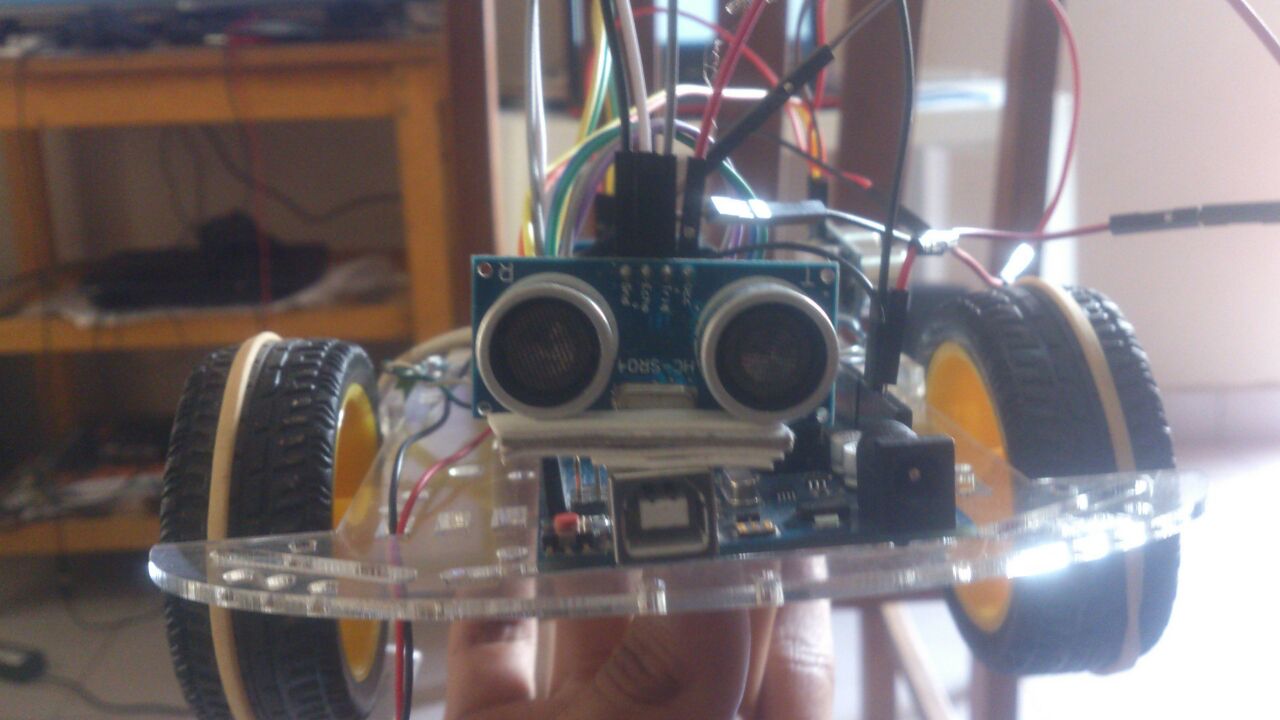
CODIGO DE ARDUINO:

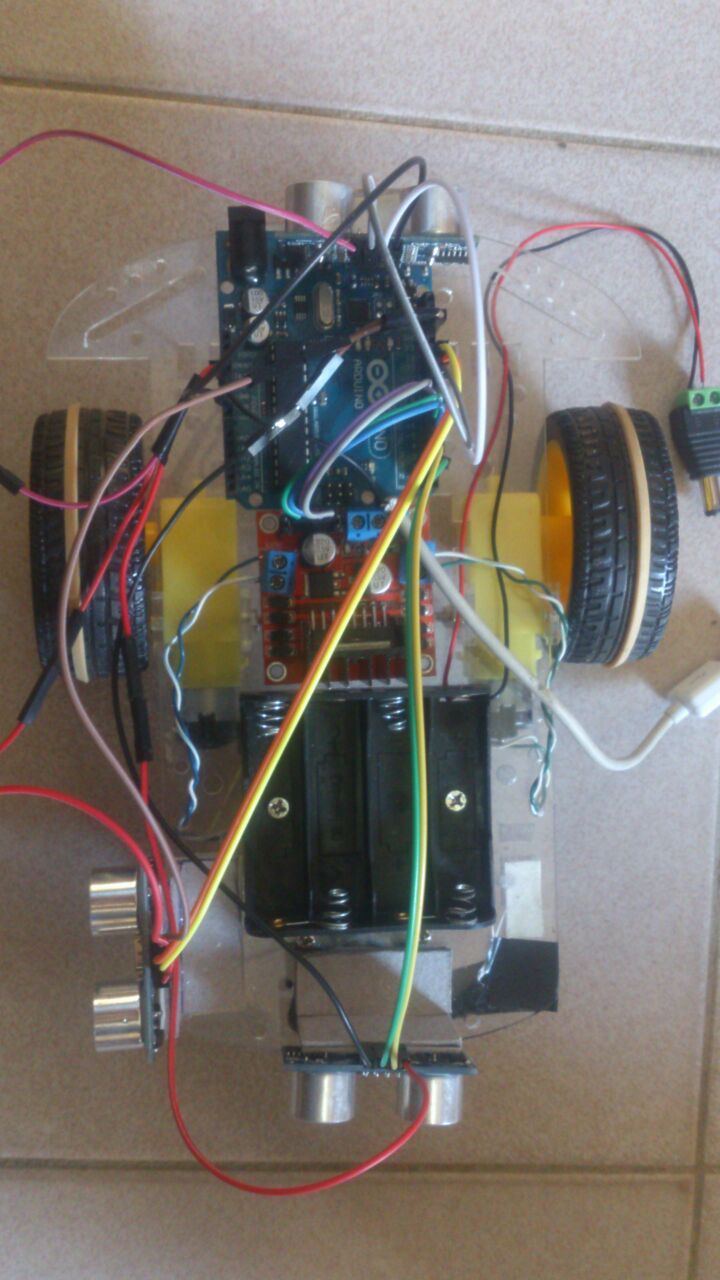




RESULTADO







Conclusión

¿Por qué no desarrollarlo como una asistencia al aparcamiento válida para la mayoría de automóviles? Se podría. Puede hacerse perfectamente, yo he demostrado que se puede hacer un proyecto que nos ayude con nuestro automóvil, pero podemos ir más allá y perfeccionarlo. Claro está, el “Arduino” es muy útil pero no es el mejor open source de su tipo, tenemos “Raspberry Pi” con un mejor procesador y 10 veces más capacidad para el programa, además de muchos más puertos, etc. ¿Cómo mejorarlo? Bueno, este proyecto solo tiene un modo valido en todos los coches, el modo distancia, pero podemos mejorarlo con mejores sensores de ultrasonido que tengan mayor rango de medida y precisión, además podemos ampliar este modo, como un pitido cuando estemos demasiado cerca del coche posterior, cuando pase algo de repente como un niño y porque no, tener más de un sensor detrás para comparar. Pero podemos avanzar más, este modo con leves modificaciones valdrá para un sensor delantero o en los laterales. Aun podemos hacer más, ¿Por qué no usar estos sensores en los laterales para hacer una imagen 2D del aparcamiento? Podemos, sin mucha complejidad, estos sensores pueden recibir datos a gran velocidad, será tan fácil como, teniendo en cuenta la velocidad del coche y la diferencia entre cada medida hacer un escaneo de la zona lateral, podremos verlo en nuestro móvil o pantalla del coche y superponer encima una imagen de nuestro coche para que veamos que tal entraríamos en el hueco. Todas estas cosas son posibles de tener en nuestro coche y Smartphone y cada vez será más habitual. No hay que ver este proyecto como un proyecto acabado, muerto, sino como un proyecto vivo y abierto. El cual podrá derivar en algo mucho mejor con la participación de la gente.

Bibliografía

Las dos páginas principales para el desarrollo:

http://www.arduino.cc/es/

Página para descargar el programa:

<http://arduino.cc/en/Main/Software>

Y sus tutoriales:

<http://arduino.cc/en/Reference/HomePage>

Además de tener una gran comunidad y soporte.

http://appinventor.mit.edu/explore/

También dispone de tutoriales y pagina para crearlo online como comente anteriormente. Otras páginas que he usado como fuente de información:

<http://panamahitek.com/el-puente-h-invirtiendo-el-sentido-de-giro-de-unmotor-con-arduino/>

<http://www.proyectosarduino.com.ar/creando-una-aplicacion-con-appinventor/>

http://spainlabs.com/foro/viewtopic.php?f=9&t=1192

<http://soloelectronicos.com/2014/04/27/control-desde-su-telefono-usandomit-app-inventor/>