

Auto Inteligente Controlado por Bluetooth

Landau - Puhl | Sistemas Embebidos | noviembre 2019

# Resumen

Durante este proyecto se trabajó en la implementación de un auto pequeño que pueda controlarse por medio de una aplicación instalada en un smartphone Android.

Teniendo en cuenta esto, el usuario es capaz de manejarlo de manera remota y poder alternar entre 3 modos distintos, uno en el que podrá manejarlo de forma normal, un modo asistido, en el cual se detectarán colisiones inminentes y frenará automáticamente el auto; y finalmente un modo automático en el cual el auto es capaz de desplazarse de manera autónoma evitando obstáculos.

# Objetivos

La idea del proyecto fue crear un auto pequeño con base en los autos inteligentes actuales, como los autos de Tesla, que tienen piloto automático y sensores de proximidad a lo largo del vehículo.

En esta versión hogareña, se contó con un solo sensor de proximidad. Y para el control, se optó por utilizar una aplicación que funcione en cualquier smartphone Android, para facilitar el uso del vehículo.

Otro objetivo fue establecer tres modos de operación.

# Elementos utilizados

Para la implementación de este proyecto se utilizaron varios componentes:

* Arduino UNO.
* Motor Shield L293D.
* Sensor Ultrasónico HC-SR04.
* Modulo Bluetooth HC-05.
* PowerBank.
* Protoboard.
* Motores DC 100 RPM 6V

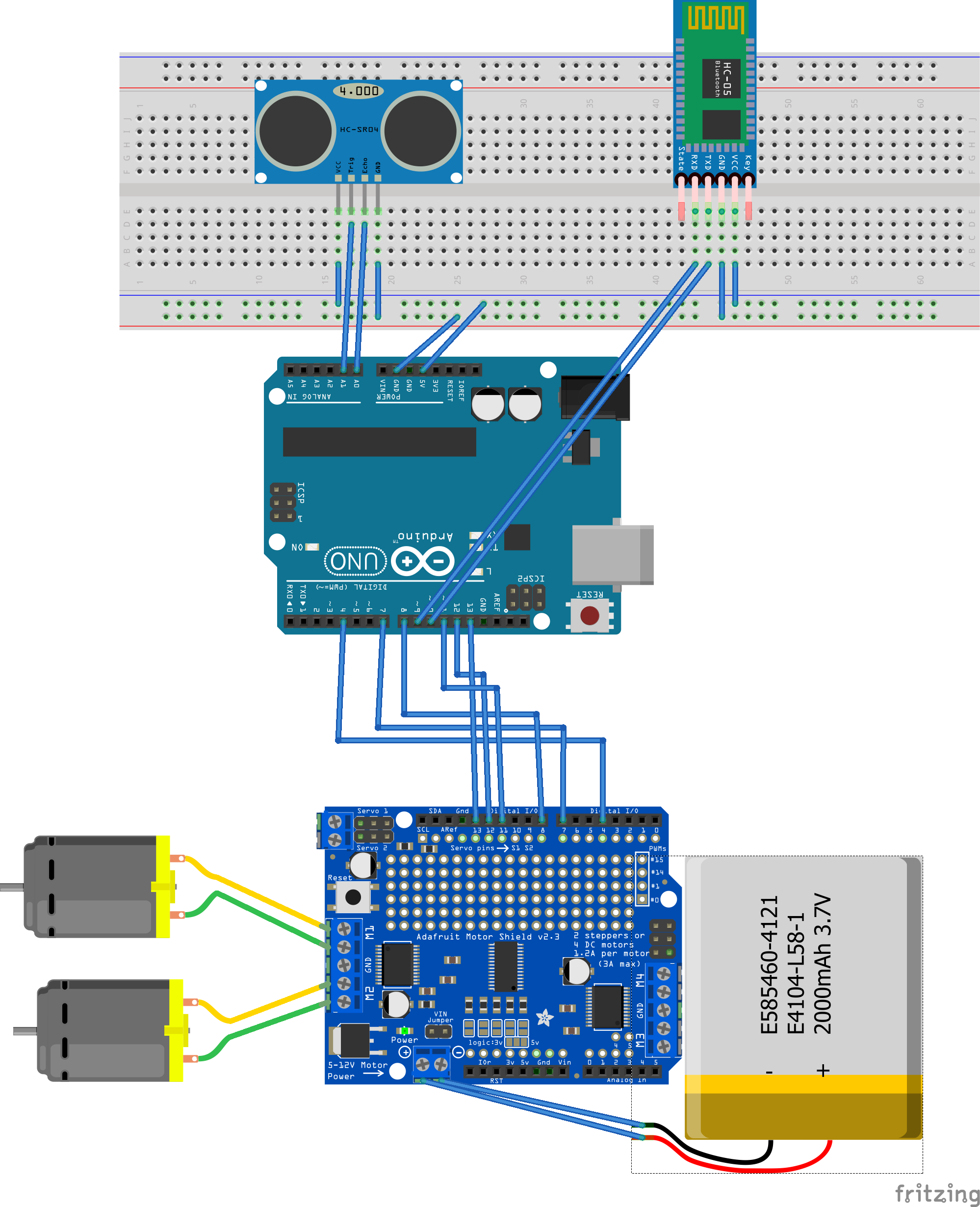


Figura 1: Conexionado.

La placa Arduino debe ser conectada a una batería distinta de la del shield, en nuestro caso, utilizamos el cable USB provisto por Arduino, dejándonos con el problema de que el auto no es completamente inalámbrico.

# Software

Para la programación del código para Arduino utilizamos el IDE oficial de Arduino.

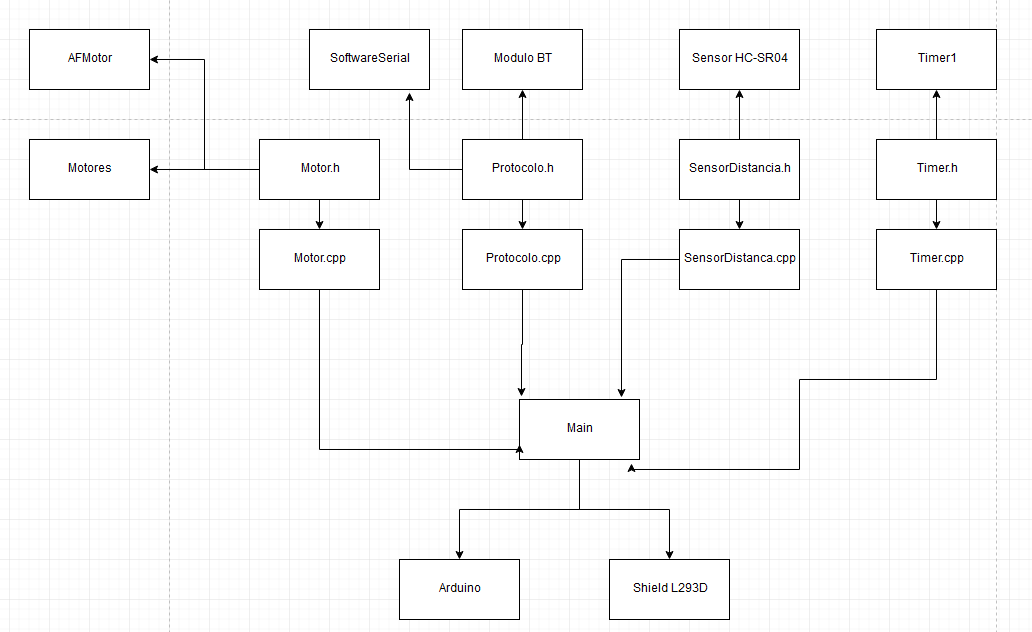


Figura 2: Diagrama de módulos.

Para la programación de la aplicación de Android, utilizamos la herramienta web MIT App Inventor, con la cual pudo llevarse a cabo rápidamente. Consiste de un editor visual donde se añaden los elementos de la interfaz, tales como botones, sliders, etc.; y un editor de lógica, en el cual disponemos de distintos bloques de código que iremos encastrando como si fueran ladrillitos de juguete.

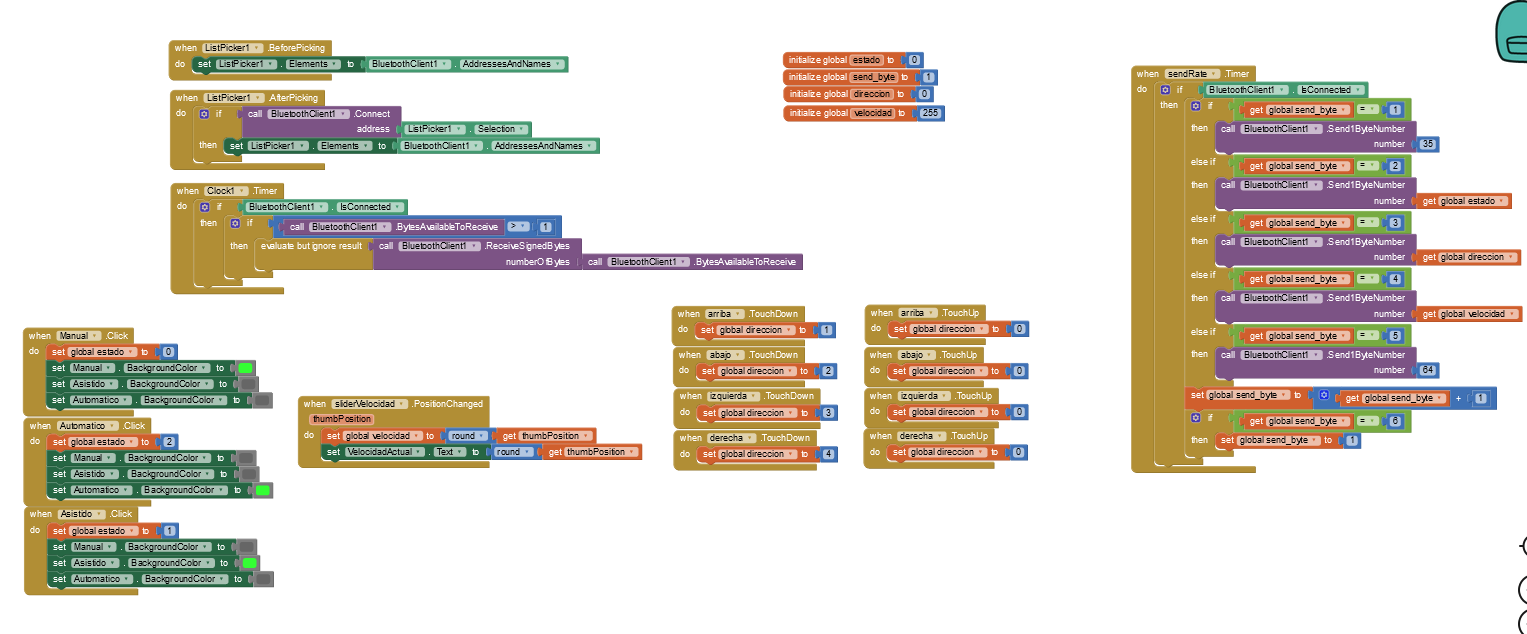


Figura 3: Vista del editor de bloques de MIT App Inventor.

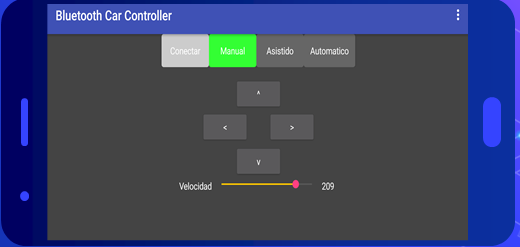


Figura 4: Interfaz Aplicación Android.

Paquete enviado desde la aplicación:

* Carácter de Inicio = ‘#’
* Modo:
  + 0 = Modo Manual.
  + 1 = Modo Asistido.
  + 2 = Modo Automático.
* Dirección:
  + 0 = No Mover.
  + 1 = Adelante.
  + 2 = Atrás.
  + 3 = Izquierda.
  + 4 = Derecha.
* Velocidad: 0 – 255.
* Carácter de Finalización = ‘@’
* Carácter de Escape = ‘\’

El smartphone envía paquetes, pero no recibe confirmación por parte del módulo bluetooth del auto.

# Funcionamiento:

Mediante la aplicación Android, puede controlarse el movimiento del automóvil. Lo primero es conectarse al módulo, con el botón “Conectar”. Una vez enlazado, el usuario puede manejar libremente, en el modo que el elija.

En el modo manual, la conducción es normal y libre. El auto no utilizará el sensor ultrasónico para detectar colisiones.

En el modo asistido, el auto utiliza el sensor para evitar colisiones. En caso de detectar un objeto delante, simplemente frenara y no dejara al usuario avanzar por mas que lo intente.

En el modo automático, el auto avanza libremente sin intervención del usuario. En cuanto detecte un obstáculo, comenzara a girar a la derecha buscando sortearlo.

El auto tiene 2 leds a modo de indicadores, uno rojo y otro amarillo. El led rojo indica la presencia de un obstáculo delante, mientras que el led amarillo indica el estado actual de operación: En modo manual, permanecerá apagado, mientras que en el modo asistido se encenderá de manera fija. En el modo automático, el led amarillo parpadeara cada 1 segundo.

# Limitaciones

El sensor está fijo en la parte posterior del automóvil, permitiendo la detección de obstáculos, sin embargo, solo detectará objetos que sean perpendiculares al auto. Esto se debe al funcionamiento del sensor, el cual envía una onda sonora y luego detecta cuanto tiempo tarda en rebotar y volver. Si el ángulo con el obstáculo no es 90°, la onda puede irse en otra dirección y el sensor no detecta la colisión, avanzando hasta chocar.

En el modo automático, como consecuencia de lo explicado anteriormente, detecta la colisión inicial, pero cuando gira el ángulo con el objeto ya no es de 90°, con lo cual ya no detecta la colisión.

# Conclusiones y pasos a seguir

Tenemos muy en claro que puede mejorarse mucho más. Podríamos arreglar el problema de la detección, o con más sensores, o con un servo y un comportamiento más pausado en el modo automático, por ejemplo: Detectar la colisión, retroceder un segundo, y girar a la derecha; o detectar la colisión y girar 90° para luego volver a avanzar.

También podríamos enviar datos desde el auto y mostrarlos en el teléfono, en vez de solo enviar desde el celular.

Sin embargo, creemos que logramos cumplir los objetivos que nos propusimos, y estamos muy satisfechos teniendo en cuenta que para este proyecto utilizamos sensores, módulos y shields que no conocíamos y tuvimos que investigar cómo usarlos, además del poco tiempo que tuvimos y los compromisos con otras materias.