



UNIVERSIDAD AUTONOMA DE GUADALAJARA

CopsKiano Móvil

Arquitectura de Microcomputadoras

Maestría en Ciencias Computacionales

Daniel Coppel Vizcarra

Registro: 2861216

Enrique Martínez Salas

Registro: 1922510

13/08/2019

Introducción

Este documento describe el diseño e implementación de un vehículo controlado a distancia a través de un dispositivo móvil. Estará formado por dos grandes bloques: hardware y software. El prototipo del vehículo, construido a escala utilizando una estructura que contiene un microcontrolador, se puede mover con los comandos que envía el usuario, gracias a la creación de un algoritmo que evita colisiones con objetos frontales. En este documento, se detalla todo el proceso.

Alcance del trabajo

Actualmente la tecnología móvil está en constante evolución; Las utilidades que puede ofrecer un dispositivo móvil están aumentando, desde utilidades básicas como hacer llamadas a utilidades más complejas.

De esto surge la motivación por la cual se llevará a cabo este proyecto, que consiste en aplicar la tecnología de control inalámbrico de la tecnología móvil. Los autos a control remoto fueron una gran revolución tecnológica; Poder conducir un automóvil con un control inalámbrico despertó un gran interés en el mundo del ocio. En la actualidad, existen estructuras más complejas de control remoto, como barcos, aviones, helicópteros o los innovadores "drones".

Justificación

El impacto que genera el aumento de la tecnología móvil en la sociedad puede reflejarse en el número de terminales vendidos por año entre personas de todas las edades; esto despierta un interés tanto en los usuarios como en los desarrolladores de aplicaciones, ya que cada vez hay un mayor número de aplicaciones destinadas al ocio o para facilitar ciertas tareas al usuario.

Al unir el concepto de control inalámbrico con tecnología móvil, es el objetivo principal de este proyecto, que consistirá en un prototipo de automóvil que se puede manejar a través de un teléfono inteligente.

Marco teórico

Si queremos desarrollar un buen proyecto, es necesario tener conocimiento de los módulos y la arquitectura de los microcontroladores, ya que pueden hacer muchas aplicaciones. Por lo tanto, se requiere una mente abierta y una buena lógica para pensar en la solución del proyecto y cómo lograr los objetivos propuestos.

Descripción funcional

Para definir la descripción funcional, se clasificará en dos categorías:

- Requisitos del usuario: describe la funcionalidad de un proyecto.
- Diagramas de casos de uso: describa las actividades que debe realizar alguien o algo para llevar a cabo un proceso.

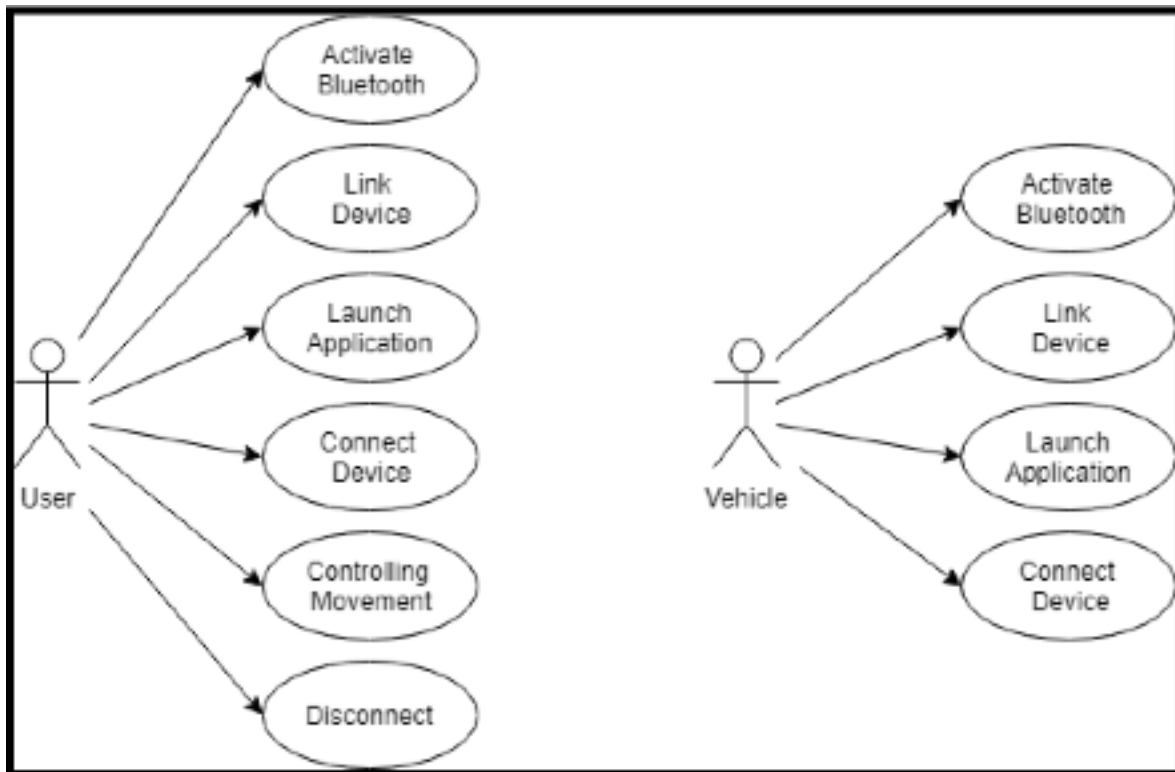
Requisitos de usuario

- El usuario debe tener una conexión inalámbrica habilitada en el dispositivo móvil.
- Debe poder seleccionar el dispositivo con el que establecer la conexión inalámbrica conexión.
- Debe haber una conexión entre el vehículo y el dispositivo móvil.
- Al inicio de la aplicación, se presentará la siguiente interfaz: 8 botones para operar el vehículo, un botón para conectarse a través de Bluetooth y dos botones para encender y apagar las luces delanteras.
- El usuario debe poder conducir el vehículo (avanzar, retroceder, girar a la izquierda y derecha y pare).
- El usuario debe poder encender y apagar las luces delanteras usando el encendido y apagado botones.
- El usuario debe poder conectar el dispositivo con el botón Bluetooth.
- El diseño de la aplicación de la interfaz de usuario debe ser simple e intuitivo.
- La aplicación debe notificar al usuario si se produce algún tipo de error.
- El vehículo debe tener un dispositivo para realizar conexiones inalámbricas.
- El dispositivo inalámbrico del vehículo debe estar encendido y visible para el usuario.
- El vehículo debe estar equipado con un sistema anticolidión para evitar colisiones frontales
- El vehículo debe tener su propia fuente de energía.
- El vehículo debe tener un mecanismo para generar tracción sobre dos ruedas.
- El software cargado en el microprocesador debe recibir y procesar correctamente los pedidos enviados desde la aplicación móvil.

Diagramas de casos de uso

Una vez que se establecen los requisitos del usuario, se dividirán en dos escenarios principales:

- Casos de uso del usuario: se definirán los casos de uso para el control del vehículo por parte de un usuario desde la aplicación de Android.
- Casos de uso del vehículo: estos serán los casos de uso que el vehículo puede hacer como actor principal.



Requisitos

Para definir los requisitos del proyecto, estos se clasificarán en dos categorías principales:

- Requisitos funcionales: se refiere a un requisito de capacidad del usuario.
- Requisitos no funcionales: se refiere a un requisito de restricción del usuario.

Requisitos funcionales

- El dispositivo del vehículo, una vez conectado correctamente al teléfono, aparecerá en la lista de pares del teléfono.
- Si no se puede establecer una conexión, se notificará al usuario.
- El usuario puede intentar conectarse nuevamente hasta que se logre la conexión.
- El usuario operará el vehículo en modo manual usando los botones que aparecerán en la aplicación (avanzar, retroceder, girar, parar y encender y apagar los faros).
- Todos los botones aparecerán en la pantalla de operación de la aplicación: conecte el dispositivo, encienda / apague las luces delanteras y los botones de cambio del vehículo.
- El vehículo tendrá un módulo Bluetooth para transmitir los datos entre la aplicación y el microcontrolador.
- El módulo Bluetooth del vehículo se puede vincular a uno o más dispositivos móviles.
- El módulo Bluetooth del vehículo solo puede conectarse a un dispositivo móvil.
- El LED del módulo Bluetooth parpadeará rápidamente si no hay conexión con un dispositivo.

- Una vez establecida la conexión con un dispositivo, el LED del módulo Bluetooth parpadeará lentamente.
- El módulo Bluetooth debe estar encendido para aparecer como un dispositivo disponible.
- El vehículo estará equipado con un sensor que calcula la distancia entre el vehículo y el obstáculo al que se enfrenta.
- Si el sensor detecta un obstáculo a cierta distancia, el vehículo se detendrá automáticamente.
- Se usará una batería de 9V como fuente de energía externa para el vehículo.
- El vehículo tendrá dos motores eléctricos para mover las ruedas.
- El vehículo también tendrá una "rueda loca" sin tracción para generar estabilidad en la estructura.
- El microcontrolador recibirá los datos enviados desde el módulo Bluetooth, que actuará como esclavo.
- El software debe compilarse sin errores ni advertencias antes de cargarlo en el microcontrolador.

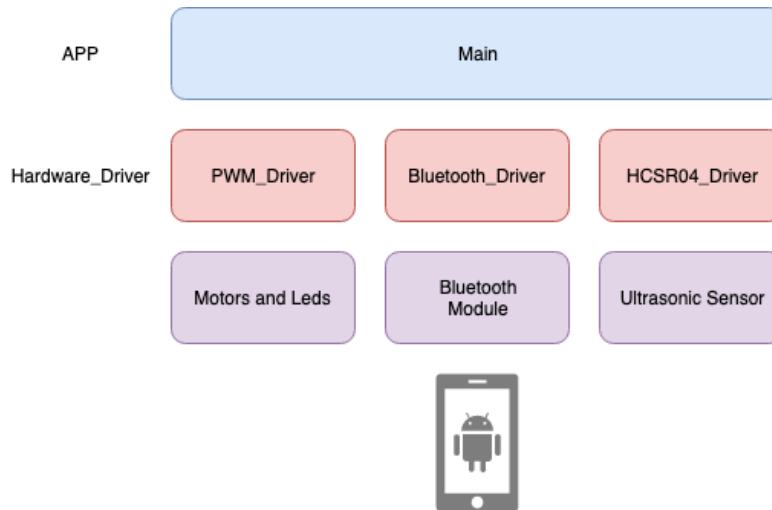
Requisitos No Funcionales

- La aplicación debe funcionar en el sistema operativo Android.
- La aplicación no está diseñada para tabletas o teléfonos inteligentes de más de 5,5 pulgadas.
- Se utilizará el lenguaje C para embebidos
- El sensor detectará obstáculos iguales o mayores al área de un cuaderno tamaño carta y en sentido perpendicular al carrito
- Se utilizará un entorno de desarrollo a nivel de código, compatible con el sistema operativo Windows 10.
- Tanto las conexiones de software como de hardware se realizarán utilizando un solo microcontrolador.
- El modelo de la tarjeta utilizada será "Arduino".

Módulos

- PWM_Driver: módulo para controlar el movimiento de los motores y las luces delanteras.
- Bluetooth_Driver: módulo de comunicación para recibir los comandos.
- HCSR04_Driver: módulo para controlar el sistema anticolidión que tendrá el vehículo.

Arquitectura



Descripción

- **App**

Main: Programa para administrar y controlar los movimientos.

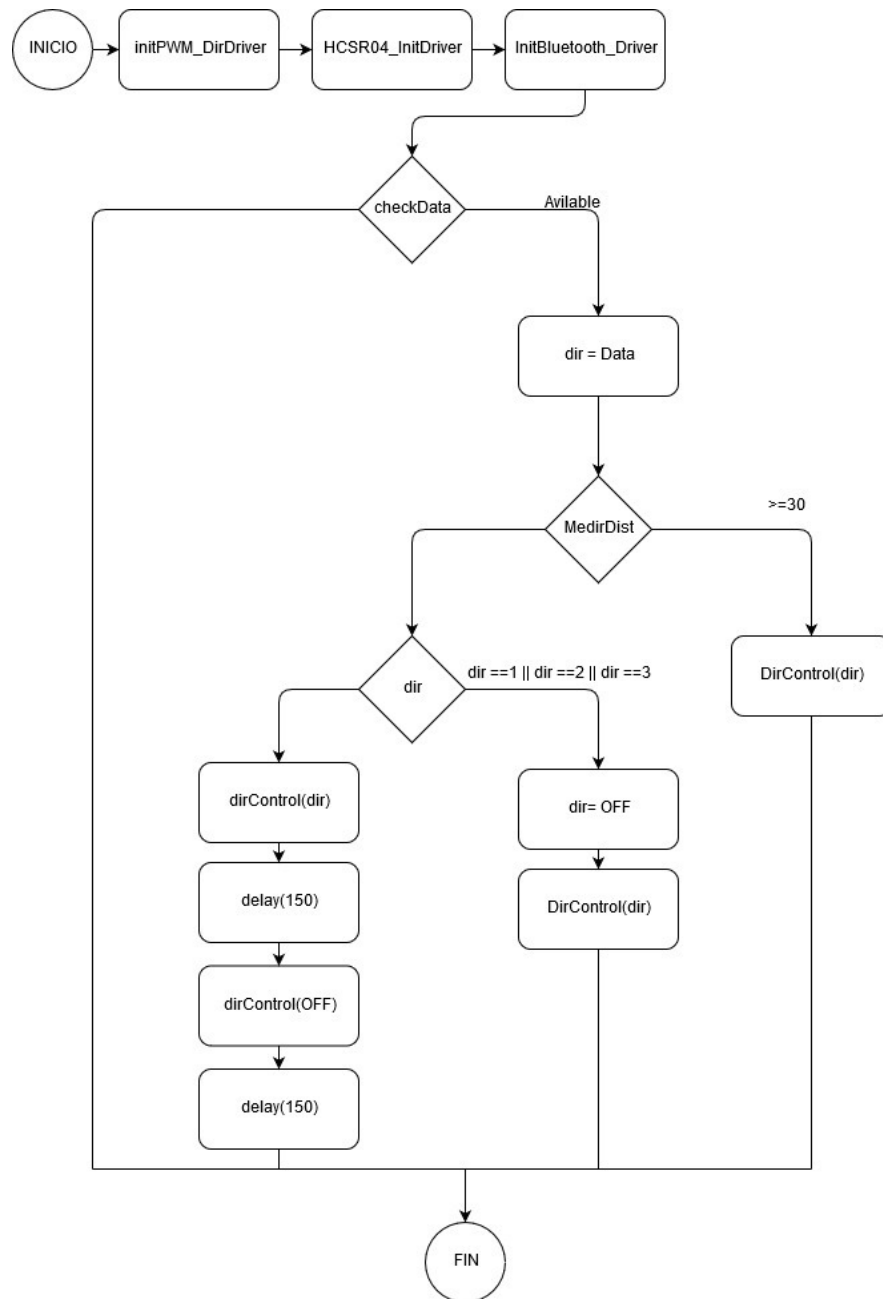
- **Hardware_Driver**

PWM_Driver: Módulo que contiene controladores inicializados para controlar el movimiento de los motores y las luces delanteras.

Bluetooth_Driver: Módulo que contiene controladores inicializados para recibir los comandos.

HCSR04_Driver: Módulo que contiene controladores inicializados para controlar el sistema anticolidión que tendrá el vehículo.

Diagrama de flujo



Materiales

Tarjeta Arduino Mega 2560

Utilizado habitualmente para proyectos de esta magnitud. Su función será recibir y procesar las órdenes que el usuario envíe desde el teléfono móvil para controlar el vehículo.



Módulo Bluetooth HC-05

Este componente es utilizado para realizar la conexión entre el teléfono móvil y el Arduino vía Bluetooth, y así poder transferir los datos que el usuario envía.



Sensor ultrasónico HCSR04

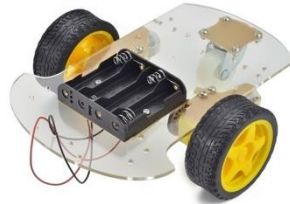
Se trata de un dispositivo cuya finalidad es medir la distancia a la que se encuentra un objeto situado frontalmente, mediante un par de transductores de ultrasonido.



Vehículo

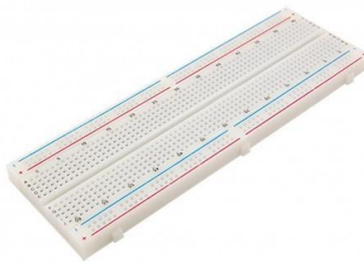
El vehículo que se utilizará para este proyecto está formado por diversas componentes:

- Tabla
- Ruedas
- Motores



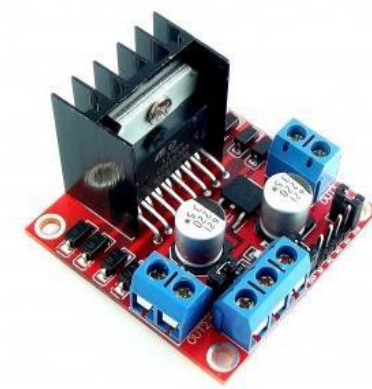
Protoboard

Es una placa en la que puede insertar elementos electrónicos y cables con los cuales se ensamblan los circuitos sin necesidad de soldar ninguno de los componentes.



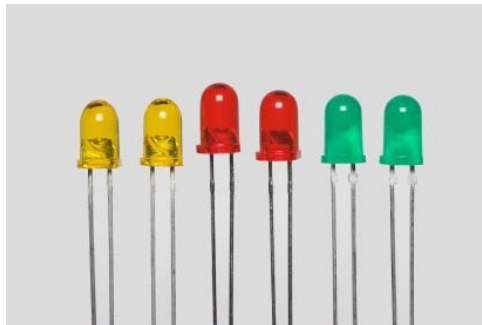
Modulo L298N

Servirá para generar movimiento en los dos motores de corriente continua que el vehículo poseerá y, por lo tanto, se controlarán los posibles giros del motor: girar hacia adelante y hacia atrás.



Luces Led

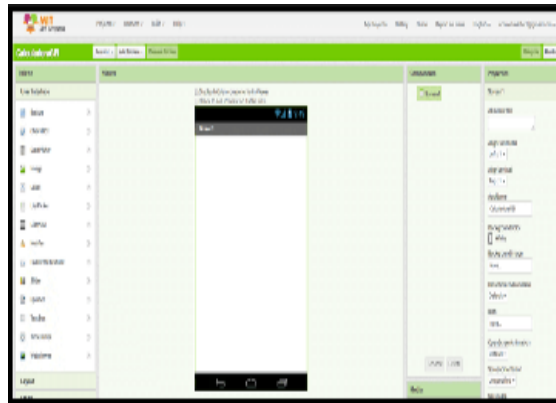
Las luces led simularán los faros del vehículo.



MIT App Inventor

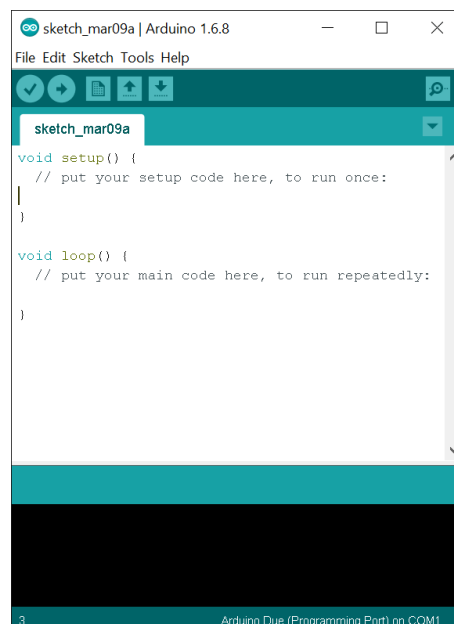
Es una plataforma de Google que le permite diseñar y crear aplicaciones para el sistema operativo Android de una manera simple y visual.

La principal ventaja de usar esta herramienta es la simplicidad de uso, ya que ofrece al usuario la capacidad de diseñar la interfaz de la aplicación e implementar la lógica de una manera muy simple e intuitiva.



Arduino IDE

Es el entorno de programación usado para el desarrollo del código que ejecutara el Arduino.



Problemas encontrados

A lo largo de la realización del proyecto, han surgido dificultades tanto en el software como en el hardware, que han causado la modificación de la planificación de las tareas establecidas desde el principio; En el caso del software, los problemas han sido errores en el código, tanto en el microcontrolador como en la aplicación de Android. Esto es algo muy común en la programación, pero gracias a la documentación proporcionada por Internet y la realización de numerosas pruebas, se resolvieron los errores.

Los problemas en el hardware han sido los más comunes en la implementación del sistema, como fallas en algunos componentes debido a una mala conexión o voltaje incorrecto, así como el desgaste de la batería al realizar numerosas pruebas, piezas sueltas en la estructura del vehículo las cuales tuvieron que ser soldadas (los polos de los motores).

Los problemas, tanto en software como en hardware, originados durante la realización del proyecto, se pudieron resolver de manera eficiente y el tiempo final de entrega del proyecto no se alteró, a pesar de sufrir modificaciones en la planificación inicial de las tareas del proyecto.

Lecciones aprendidas

La realización del proyecto, desde el principio, ha requerido un gran esfuerzo personal debido a la poca experiencia en el entorno del sistema.

La parte más fácil, pero al mismo tiempo ha traído más problemas, ha sido la construcción del vehículo; Todas las conexiones de los componentes de hardware que constituyen la estructura podrían realizarse correctamente gracias a la extensa documentación existente en Internet. Los problemas fueron principalmente para acoplar los circuitos a la estructura del vehículo y la construcción de dicha estructura.

La parte más complicada, en la que se ha empleado el mayor esfuerzo, ha sido en la programación de software, ya que tenía poca experiencia en la programación de embebidos y ninguna con el microcontrolador Arduino.

De todos modos, aprendí mucho en este proyecto, como retomar los fundamentos básicos de la programación en c, su lógica y que los sistemas embebidos son muy útiles, ya que puedes hacer productos que pueden facilitar tu vida diaria.

Como ingeniero de software, he aprendido a organizarme mejor en la realización de proyectos como developer porque muchos de los problemas de un proyecto que no se termina a tiempo se deben a la falta de organización.

LINK de Github del Código del Proyecto

<https://github.com/kikiano/CopsKianoMovil>