**UNIVERSIDAD TECNOLOGICA METROPOLITANA DE AGUASCALIENTES**

****

INFORME DE PROYECTO INTEGRADOR [ARDUINO & APP]

**TEMA:**

DESARROLLO DE CÓDIGO PARA LA ELABORACIÓN DE UN CARRO ARDUINO CONTROLADO POR MEDIO DE UNA APP MÓVIL

**AUTORES:**

RICARDO MARTINEZ CAMPOS

LUIS BRANDON LUNA CASTORENA

ANGEL RAUL RODRIGUEZ OROPEZA

JOSE ALEJANDRO RODRIGUEZ OROPEZA

ARCADIO GONZALEZ DIAZ

ALEJANDRO MANUEL CANO HERNÁNDEZ

EDGAR BECERRA BARBA

ALBERTO ISRAEL CORZO BAUTISTA

1. **Título de la investigación**

INFORME DE PROYECTO INTEGRADOR [ARDUINO & APP]

1. **Planteamiento del problema**

Para la evaluación del tercer parcial se nos asignó elaborar un proyecto el cual está constituido por 3 partes, por la primera parte tenemos la materia de APLICACIONES DE IOT en la cual se nos pidió elaborar un carro a control remoto elaborado con Arduino, por la segunda parte tenemos la materia de DESARROLLO MOVIL MULTIPLATAFORMA en la cual se nos pidió elaborar una aplicación móvil la cual fuera capaz de controlar el carro Arduino que se elaborara en la materia de APLICACIONES DE IOT y finalmente como última parte tenemos la materia de INTEGRADORA II en la cual se nos pidió elaborar o llevar a cabo toda la documentación necesaria de los procesos que se llevarán a cabo durante la elaboración tanto del carro Arduino así como de la aplicación móvil.

El desarrollo de estos aspectos pretende solventar este problema que se nos han asignado en dichas materias las cuales tratan de llegar a un fin en común el cual es mostrar este proyecto en la ExpoTI que se llevará a cabo en la universidad.

* 1. **Antecedentes**

La universidad politécnica superior Carlos III de Madrid realizó el diseño e implementación de un vehículo a escala controlado remotamente, dicho proyecto fue realizado el mes de septiembre del año 2016 por Marco Torrico López bajo la tutoría de Javier Fernández Muñoz.

Este proyecto describe el diseño y la implementación de un vehículo controlado de manera remota a través de un dispositivo móvil. Está formado por dos grandes bloques: hardware y software. El prototipo de vehículo, construido a escala mediante una estructura que contiene un microcontrolador, podrá desplazarse con las órdenes que el usuario envíe o de manera automática, gracias a la creación de un algoritmo inteligente que evita las colisiones con objetos frontales.

Por otro lado, la universidad politécnica de Cartagena (2015) en la ingeniería en electrónica industrial realizó, del mismo modo, el diseño y construcción de un vehículo teledirigido basado en la plataforma Arduino, en donde el proyecto se desarrolló en torno a un coche dirigido por radiocontrol con mando, enfocándose en la comunicación inalámbrica programada la cual gestiono los movimientos del vehículo.

En conclusión, ambos proyectos utilizaron diferentes tecnologías, ya que mientras la universidad superior utilizo tecnologías como Android Studio, para la creación de una aplicación la cual controle dicho vehículo, la universidad politécnica de Cartagena utilizó un control remoto programado y conectado a dicho vehículo mediante bluetooth, dicho esto se llegó a la conclusión que para la implementación de las diferentes materias es necesario la realización de una aplicación móvil la cual controla el vehículo, vehículo que será realizado con diferentes componentes como la tarjeta ESP-32, la cual permitirá la conexión con la aplicación mediante una red WIFI.

* 1. **Justificación**

La solución de crear un carro a control remoto controlado con una aplicación móvil es beneficiosa de realizar por varias razones:

* Se desarrollan grandes capacidades en los alumnos en base a la experiencia que proyectos de esta categoría generan, ya que por la complejidad o la magnitud de estos proyectos el alumno aplica sus conocimientos a un nivel más alto.
* Se promueve el autodidactismo ya que al estar en un proyecto de estas magnitudes esto obliga al alumno a investigar temas que se desconocen o son muy complejos de entender lo cual lleva a generar conocimiento más concretos y claros.
* Se promueve la realización de proyectos de gran nivel para promover la creación de proyectos innovadores dentro del ambiente de la universidad.
* Se promueve el trabajo en equipo de manera eficiente logrando que el alumno forje actitudes de disciplina dentro de un entorno de trabajo en equipo logrando así resultados favorables.

1. **Objetivos**
   1. **Objetivo General**

El objetivo principal de este proyecto es mediante la creación y optimización de software y la construcción de hardware crear un proyecto en conjunto en el cual se construya un carro a control remoto el cual sea controlado o manipulado mediante una aplicación móvil. Dicho carro y aplicación deben de cumplir con funciones básicas para que puedan mantener una comunicación estable y puedan trabajar y funcionar entre estas dos para una solución justa y eficiente.

* 1. **Objetivos Específicos**
* Implementación y diseño de una estructura con forma de automóvil constituida por hardware Arduino y componentes electrónicos.
* Implementación y diseño de una aplicación móvil creada para dispositivos Android usando la tecnología JAVA como lenguaje de programación principal de la aplicación.
* Correcto funcionamiento del carro a control remoto, dicho carro debe de tener la facilidad de moverse en una superficie plana en cualquier dirección sin ninguna dificultad.
* Correcto funcionamiento de la aplicación móvil, dicha aplicación debe de tener la funcionalidad de controlar el carro a control remoto con botones que permiten controlar el carro en todas las direcciones existentes (arriba, abajo, derecha e izquierda).

**4. Marco Teórico**

Describiremos algunas definiciones y conceptos de componentes que utilizamos para

desarrollar este proyecto:

* **ARDUINO ONE**

El Arduino Uno es una placa electrónica basada en el ATmega328. Cuenta con 14 pines digitales de entrada / salida (de los cuales 6 se pueden utilizar como salidas PWM), 6 entradas analógicas, un 16 MHz resonador cerámico, una conexión USB, un conector de alimentación, un header ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador, simplemente conectarlo a un ordenador con un cable USB o al poder con un adaptador de CA o la batería ACC para empezar. (Arduino, 2014).

**¿Qué puede hacer Arduino?**

Arduino puede detectar el medio ambiente mediante la recepción de la entrada de una variedad de sensores y puede afectar a sus alrededores por las luces de control, motores, y otros actuadores. El microcontrolador en la placa se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Proyectos de Arduino pueden ser autónomos o pueden comunicarse con el software que se ejecuta en un ordenador (por ejemplo, Flash, Processing, MaxMSP).

Las tablas pueden ser construidas a mano o compradas preensamblado, el software puede ser descargado de forma gratuita. Los diseños de referencia de hardware (archivos CAD) están disponibles bajo una licencia de código abierto, que son libres de adaptarlas a sus necesidades. Los fundadores de Arduino son: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino y David Mollis.

Un Puente O Puente en H es un circuito electrónico que permite a un motor eléctrico DC girar en ambos ampliamente usados en robótica y como convertidores de potencia. Los sentidos, avance y retroceso. Los puentes H están disponibles como circuitos integrados, pero también pueden construirse a partir de componentes discretos.

* **CONEXIÓN HC-06 y ARDUINO**

La conexión que tendrás que hace entre el bluetooth y el Arduino, será como la que se muestra a continuación y aunque el módulo que aparece en la imagen no es el mismo que el HC-06-solo sirve como referencia, ya que nuestro módulo también cuenta con los mismos 4 pines para su conexión.

* **MODULOS BLUETOOTH**

Se denomina Bluetooth al protocolo de comunicaciones diseñado especialmente para dispositivos de bajo consumo, que requieren corto alcance de emisión y basados en transceptores de bajo costo. Opera mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz. Su uso es adecuado cuando puede haber dos o más dispositivos en un área reducida sin grandes necesidades de ancho de banda.

Bluetooth tiene la ventaja de simplificar el descubrimiento y configuración de los dispositivos, ya que éstos pueden indicar a otros los servicios que ofrecen, lo que redunda en la accesibilidad de estos sin un control explícito de direcciones de red, permisos y otros aspectos típicos de redes tradicionales. El Wifi utiliza el mismo espectro de frecuencia que el Bluetooth con una potencia de salida mayor que lleva a conexiones más sólidas.

Los módulos de bluetooth HC-05 y HC-06 son módulos muy populares para aplicaciones con microcontroladores PIC y Arduino. Se trata de dispositivos relativamente económicos y que habitualmente se venden en un formato que permite insertarlos en un protoboard y cablearlo directamente a cualquier microcontrolador, incluso sin realizar soldaduras. En esta entrada del blog vamos a explicar un poco del funcionamiento de estos módulos y cómo configurarlos. También abordaremos las diferencias entre el HC-05 y el HC-06.

* **SENSOR ULTRASONIDO HC-SR04**

El HC-SR04 es un sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 450 cm. El sensor funciona por ultrasonidos y contiene toda la electrónica encargada de hacer la medición. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno. De muy pequeño tamaño, el HC-SR04 se destaca por su bajo consumo y gran precisión.

* **MOTORES DC**

El motor de corriente continua (también denominado motor de corriente directa, motor CC o motor DC) es una máquina que convierte la energía eléctrica en mecánica, provocando un movimiento rotatorio, gracias a la acción que se genera del campo magnético.

* **SERVO MOTOR**

Básicamente un servomotor es un motor de corriente continua con un potenciómetro que le permite saber la posición en la que se encuentra y así poder controlarla. Para controlar el servomotor se le envía pulsos cada 20 ms es decir 50Hz. La anchura del pulso es lo que codifica el Ángulo de giro, es decir lo que se conoce como PWM, codificación por ancho de pulso. Esta anchura varía según el servomotor, pero normalmente va entre 0.5 y 2.5 m más, aunque pueden variar. Dependiendo del tamaño del servo y su consumo es posible que no puedas alimentarlo desde tu placa Arduino, en ese caso es necesario una fuente de 5V independiente para poder moverlo, en mi caso uso un micro serva por lo que consume poca corriente y se puede alimentar directamente por el Arduino. Sobre el peso que pueden levantar se puede deducir con el par del servo. Normalmente los servos indican el torque que pueden realizar para un servo estándar suele ser 5kg/cm es decir puede mover 5 kg a 1 cm de distancia. En caso de querer mover lo a 5 cm el servo solo podrá mover 1kg.

* **ANDROID STUDIO**

Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014.

* **JAVA**

Java es un lenguaje de programación y una plataforma informática que fue comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán, probablemente, a menos que tengan Java instalado, y cada día se crean más. Java es rápido, seguro y fiable.

* **LENGUAJE C**

La programación de PIC en C actualmente se encuentra en auge en Latinoamérica. Ha surgido una creciente demanda de información en castellano sobre cómo programar PIC en C, para lo cual han salido al mercado una importante cantidad de programas para compilar. El crecimiento ha sido de forma exponencial ya que los costos han presentado una reducción repercutiendo en la variedad de productos al alcance de un conjunto mayor de personas.

* **PROTOBOARD**

Una placa de pruebas está compuesta por varios bloques de plástico perforados y numerosas láminas delgadas, de una aleación de cobre, estaño y fósforo, que unen dichas perforaciones, creando una serie de líneas de conducción paralelas. Las líneas se cortan en la parte central del bloque para garantizar que dispositivos en circuitos integrados de tipo dual in-line package (DIP) puedan ser insertados perpendicularmente y sin ser tocados por el proveedor a las líneas de conductores. En la cara opuesta se coloca un forro con pegamento, que sirve para sellar y mantener en su lugar las tiras metálicas.

**5. Diseño metodológico y técnicas de investigación**

**Código Fuente – Carro Arduino**

#include "BluetoothSerial.h"

#if !defined(CONFIG\_BT\_ENABLED) || !defined(CONFIG\_BLUEDROID\_ENABLED)

#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it

#endif

Definimos la librería y activamos la misma en este caso activamos la función de bluetooth

BluetoothSerial SerialBT;

int received;// received value will be stored in this variable

char orden;// received value will be stored as CHAR in this variable

// Motor A

const int IN1 = 23;

const int IN2 = 22;

// Motor B

const int buzeer = 18;

const int IN3 = 19;

const int IN4 = 4;

void setup() {

Serial.begin(9600);

SerialBT.begin("SKELE"); //Bluetooth device name

Serial.println("The device started, now you can pair it with bluetooth!");

Serial.println("Use your own app to control the car");//print on serial monitor

pinMode (IN1, OUTPUT);

pinMode (IN2, OUTPUT);

pinMode (IN3, OUTPUT);

pinMode (IN4, OUTPUT);

pinMode (buzeer, OUTPUT);

}

Activamos el serial del bluetooth y declaramos las variables a recibir del dispositivo bluetooth, así como establecer las salidas y entradas de los motores del carrito, después se inicia en el void setup activando el serial local y activando el bluetooth con nombre establecido “SKELE” y nuestras salidas de señal al Arduino (ESP32).

void Clacson(){

digitalWrite (buzeer, HIGH);

delay(500);

digitalWrite (buzeer, LOW);

}

void Adelante ()

{

//Direccion motor A

digitalWrite (IN1, LOW);

digitalWrite (IN2, HIGH);

//Direccion motor B

digitalWrite (IN3, LOW);

digitalWrite (IN4, HIGH);

}

void Atras ()

{

//Direccion motor A

digitalWrite (IN1, HIGH);

digitalWrite (IN2, LOW);

//Direccion motor B´´´´

digitalWrite (IN3, HIGH);

digitalWrite (IN4, LOW);

}

void Derecha ()

{

//Direccion motor A

digitalWrite (IN1, LOW);

digitalWrite (IN2, LOW);

//Direccion motor B

digitalWrite (IN3, LOW);

digitalWrite (IN4, HIGH);

}

void Izquierda ()

{

//Direccion motor A

digitalWrite (IN1, LOW);

digitalWrite (IN2, HIGH);

//Direccion motor B

digitalWrite (IN3, LOW);

digitalWrite (IN4, LOW);

}

void Parar ()

{

//Direccion motor A

digitalWrite (IN1, LOW);

digitalWrite (IN2, LOW);

//Direccion motor B

digitalWrite (IN3, LOW);

digitalWrite (IN4, LOW);

}

Después se declaran las funciones con cada respectivo uso, para el manejo del carrito de manera adecuada, en este caso solo usamos 2 llantas, cada llanta activa y desactiva el motor para girar y avanzar.

void loop() {

orden =(char)SerialBT.read();

if (Serial.available()) {

SerialBT.write(Serial.read());

}

if (SerialBT.available()) {

Serial.print ("Received:");//print on serial monitor

Serial.println(orden);//print on serial monitor

SerialBT.write(orden); //print on serial monitor

if(orden=='e')

{

Adelante();

}

else if(orden=='a')

{

Atras();

}

else if(orden=='d')

{

Derecha();

}

else if(orden=='i')

{

Izquierda();

}

else if(orden=='0')

{

Parar();

}

else if(orden=='p')

{

Clacson();

}

}

}

Para finalizar activamos el ciclo que nos estará activando las funciones de nuestro código, y estas se mandan a llamar con char o strings, dentro de la variable orden donde se almacenan los datos enviados de nuestra aplicación, se imprimen las variables recibidas de manera local y manda un mensaje de recibido a serial local.

**Código Fuente – App móvil**

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Captura de pantalla de computadora

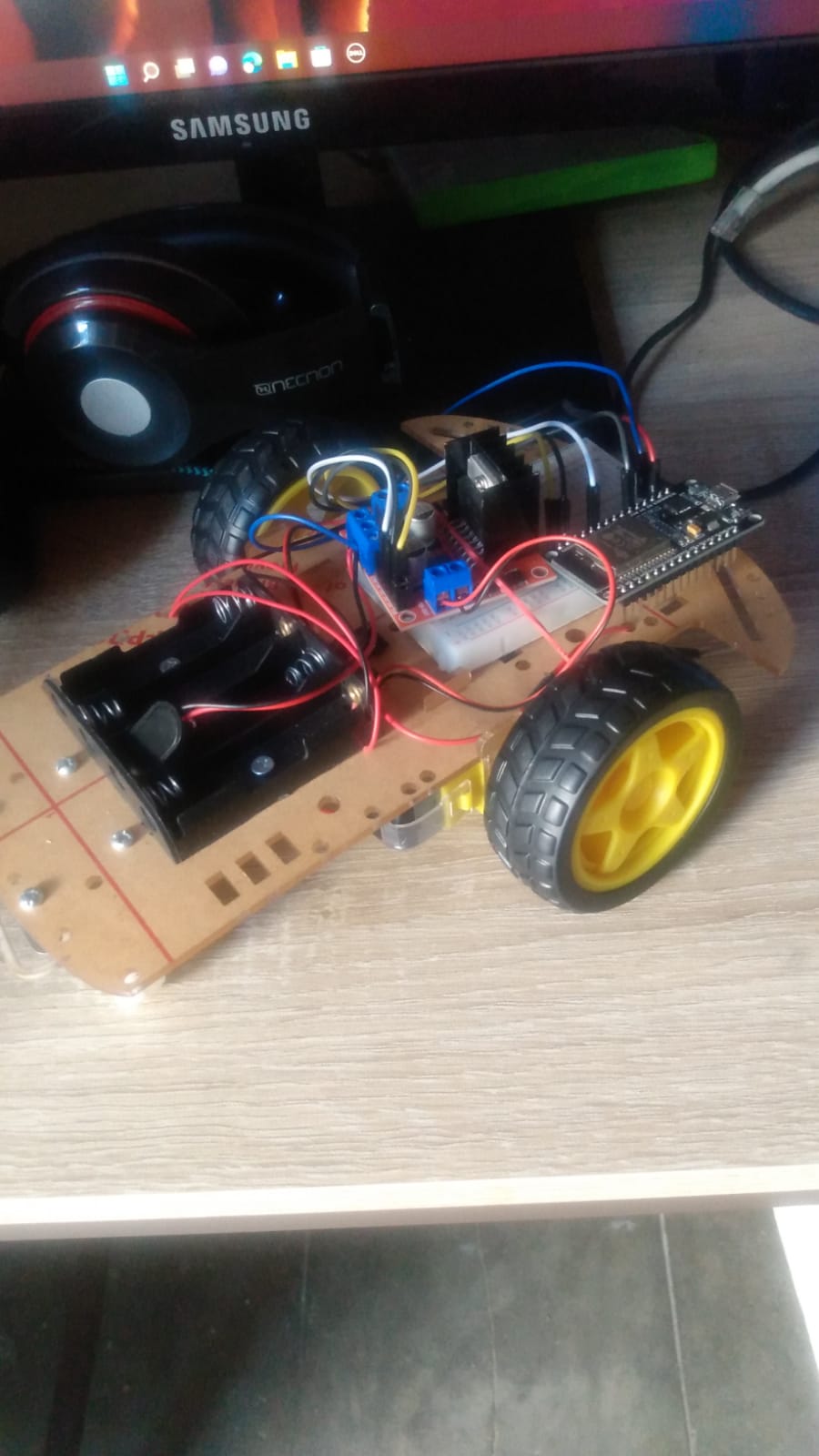
Descripción generada automáticamente

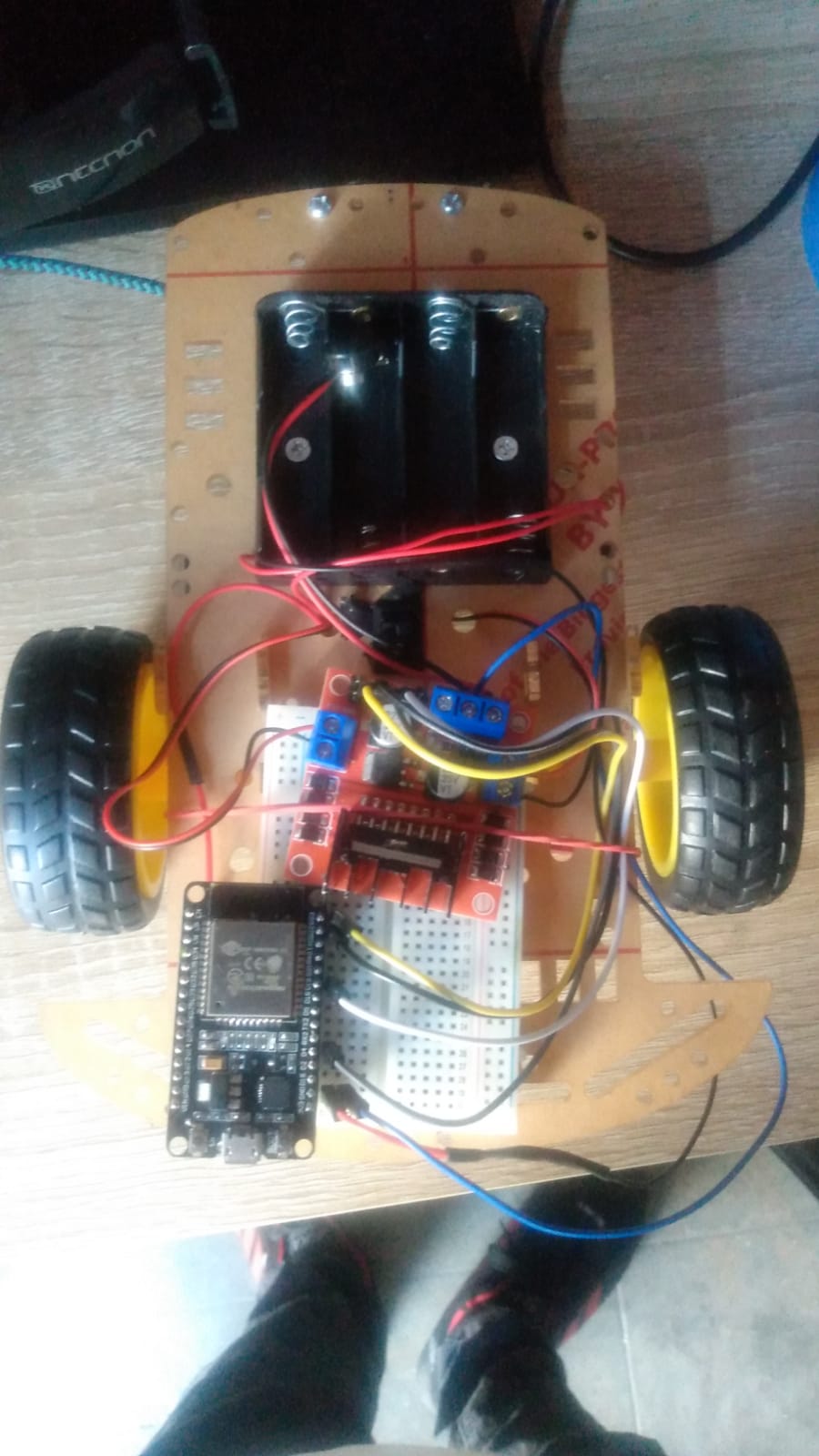
Texto

Descripción generada automáticamente

**6. Resultados**

**Resultado Carro Arduino**





**Resultado Aplicación Móvil**



**7. Conclusiones y recomendaciones del diseño del protocolo**

El presente trabajo se enfocó en el aprovechamiento de las nuevas tecnologías o innovación con el Arduino, con la intención de proporcionar herramientas para contribuir en los procesos de gestión de proyectos de investigación, y considerando los resultados obtenidos de la misma. Durante la elaboración de la propuesta, se describió el desarrollo e implementación de un sistema de información.

Y en recomendaciones del diseño del protocolo este proyecto y del trabajo de investigación se pretenden varios criterios, desde el mantener los recursos necesarios para hacer al servicio perdurable, hasta modificaciones en el diseño técnico del sistema de información. También se incluyen recomendaciones para la gestión, en la que se podrá potenciar el desarrollo de la investigación. En lo referente al presupuesto del servicio, para lograr las metas de este proyecto, tanto actuales como futuras, es importante contar con el recurso mínimo necesario para contar con el servicio de manera ininterrumpida del servidor en el que se almacena el conjunto de aplicaciones. Este recurso se refiere principalmente al pago regular de los servicios de contratación para el servidor dedicado y el ancho de banda de este.

**8. Bibliografía**

* Torrico Lopez, M. (2016, 23 septiembre). *Escuela Politécnica Superior | UC3M*. Universidad politécnica superior Carlos III de Madrid. <https://www.uc3m.es/ss/Satellite/UC3MInstitucional/es/TextoMixta/1371208450589/Escuela_Politecnica_Superior>
* Alejandro Sarmiento. (2017, 11 mayo). *Carro a control remoto - Sistemas Digitales*. Wikidot. <http://sistdig.wikidot.com/wiki:carro-a-control-remoto>
* Arturo Pimentel. (2019, 4 diciembre). *Proyecto Carrito Controlado Por Bluetooth [34m769w9qo46]*. IDOCPUB. <https://idoc.pub/documents/proyecto-carrito-controlado-por-bluetooth-34m769w9qo46>
* Ricardo Daniel Carrada Peña. (2014, 10 diciembre). *Carro RC controlado por Arduino*.Issuu. <https://issuu.com/danielricardocarradapena/docs/proyectofinal>
* SEBASTIÁN BETANCUR, P. (2019, 18 agosto). *Proyecto Carro Eléctrico a Control Remoto*. fdocuments.mx. https://fdocuments.mx/document/proyecto-carro-electrico-a-control-remoto.html
* colaboradores de Wikipedia. (2022, 21 julio). *Java (lenguaje de programación)*. Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)>
* colaboradores de Wikipedia. (2022a, marzo 11). *Android Studio*. Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/Android_Studio>
* colaboradores de Wikipedia. (2022b, abril 30). *Placa de pruebas*. Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/Placa_de_pruebas>