



**UNIVERSIDAD TECNOLÓGICA ISRAEL**

**TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:**

**INGENIERO/A EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES**

**TEMA: DESARROLLO DE UN VEHÍCULO A CONTROL REMOTO CON VISIÓN  
EN PRIMERA PERSONA.**

**AUTOR: JUAN ANDRÉS SOLÍS POVEDA**

**TUTOR: Mg René Ernesto Cortijo Leyva**

**QUITO- ECUADOR**

**AÑO: 2018**

## DECLARACIÓN

Yo, Juan Andrés Solís Poveda, declaro bajo juramento que el tema planteado para aprobación es de mi autoría, y no ha sido previamente presentado para ningún grado o calificación profesional y he consultado las referencias bibliográficas que se incluyen en este documento.

---

Juan Andrés Solís Poveda  
CC No. 1720235876

# CERTIFICACIÓN DEL TUTOR

UNIVERSIDAD TECNOLÓGICA ISRAEL

## APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación **DESARROLLO DE UN VEHÍCULO A CONTROL REMOTO CON VISIÓN EN PRIMERA PERSONA**, presentado por el **Sr. Juan Andrés Solís Poveda**, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M. 18 de septiembre del 2018

TUTOR

.....

Ing. René Ernesto Cortijo Leyva, Mg.

## **AGRADECIMIENTO**

Agradezco infinitamente a quienes me ayudaron a cumplir mis sueños y metas en mi carrera.

A mi familia quien siempre me acompaño y me cuidó durante toda mi carrera, dándome todo su amor y apoyo, sin importar las circunstancias que estuviera.

Agradezco profundamente a Paulina Pizarro que supo ayudarme e impulsarme a cumplir mis sueños, dándome su paciencia y su ayuda desinteresada.

## **DEDICATORIA**

Dedico mi monografía a mi Madre Dina María Poveda Jara, que me guio durante toda mi vida y supo darme todo su amor y comprensión, siempre permitiste que cumpla mis sueños y metas, siendo la persona que más admiro en mi vida por toda tu bondad. Eres mi luz y mi vida, mi mejor amiga y mi compañera.

A mi padre Juan Antonio Solís Cruz, que, aunque ya está en el cielo, supo darme los mejores consejos que me ayudaron a salir adelante.

## TABLA DE CONTENIDOS

INTRODUCCIÓN .....	1
Antecedentes .....	1
Planteamiento del problema .....	1
Justificación.....	2
Objetivos .....	2
• Objetivo General: .....	2
• Objetivos Específicos: .....	2
Alcance.....	3
Descripción de los capítulos .....	3
CAPÍTULO I.....	5
1.1. Robótica .....	5
1.1.1. Clasificación de robots .....	6
1.1.2. Robots móviles: .....	8
1.2. Navegación de Robots Móviles:.....	8
1.2.1. Robots controlados por sensores: .....	8
1.2.2. Robots controlados por mandos físicos: .....	8
• Bluetooth:.....	9
• WIFI ( <i>Wireless Fidelity</i> ): .....	10
1.3. Arduino .....	11
1.4. Regulador de Voltaje Step – Down .....	12
1.5. Batería.....	12
1.5.1. Baterías LiPo .....	13
CAPÍTULO II.....	15
2.1. Diagrama de bloques del sistema.....	16
2.2. Diagrama de flujos del sistema.....	17
Módulos del sistema.....	19
2.3. Aplicación de mando del sistema .....	19
2.4. Módulo de control .....	19
2.4.1. Placa NodeMCU ESP8266.....	20

2.4.2.	Módulo ESP8266.....	21
2.4.3.	Modos de Operación.....	21
2.4.4.	Puente H L298N .....	22
2.5.	Actuadores.....	24
2.5.1.	Motor Reductor .....	24
2.5.2.	Servomotores .....	24
2.6.	Sistema FPV (Visión en primera persona).....	25
2.6.1.	Cámara FPV .....	26
2.6.2.	Gafas FPV .....	27
2.6.3.	Transmisor FPV.....	28
2.7.	Programas .....	30
2.7.1.	Software SolidWorks.....	30
2.7.2.	App Inventor2.....	30
2.7.3.	Arduino IDE (Entorno de desarrollo integrado).....	31
2.7.4.	Proteus .....	31
2.8.	Análisis de costos para el desarrollo del proyecto .....	32
2.8.1.	Placa de control para el circuito .....	32
2.8.2.	Cámara para el sistema FPV.....	33
2.8.3.	Transmisor para el sistema FPV .....	34
2.8.4.	Gafas para el sistema FPV.....	34
2.8.5.	Presupuesto del proyecto .....	35
2.9.	Análisis de tiempo.....	36
CAPITULO III.....		38
3.1.	Diseño de fuente para los servomotores .....	38
3.2.	Dimensionamiento de la fuente de alimentación del circuito .....	40
3.3.	Diseño del montaje de los elementos del vehículo .....	42
3.4.	Diseño de la cubierta del vehículo.....	44
3.5.	Diseño de la aplicación para el sistema .....	45
3.6.	Desarrollo de la aplicación para el control del vehículo.....	47
3.7.	Desarrollo de la aplicación para el control de los servomotores .....	51
3.8.	Programación del NodeMCu8266 para vehículo a control remoto .....	54
3.8.1.	Programación para el movimiento del vehículo .....	56
3.8.2.	Programación de los servomotores para movimiento de la cámara. ....	58

3.9.	Implementación de la fuente de alimentación para los servomotores .....	60
3.10.	Implementación del vehículo a control remoto .....	62
3.11.	Pruebas de funcionamiento .....	66
3.12.	Análisis de resultados.....	67
CONCLUSIONES.....		69
RECOMENDACIONES .....		70
BIBLIOGRAFÍA .....		71
ANEXOS.....		72



## LISTA DE FIGURAS

<b>Figura 1 1</b>	Esquema general de un robot .....	5
<b>Figura 1 2</b>	Brazo robótico industrial.....	6
<b>Figura 1 3</b>	Prótesis de un brazo robótico .....	7
<b>Figura 1 4</b>	Avioneta no tripulada para fumigar cultivos.....	7
<b>Figura 2. 1</b>	Diagrama de bloques del sistema.....	16
<b>Figura 2. 2</b>	Diagrama de flujo del sistema.....	18
<b>Figura 2. 3</b>	NodeMCU8266.....	21
<b>Figura 2. 4</b>	ESP8266.....	22
<b>Figura 2. 5</b>	Puente H L298N.....	23
<b>Figura 2. 6</b>	Motor reductor .....	24
<b>Figura 2. 7</b>	Mobius action cam .....	26
<b>Figura 2. 8</b>	Gafas FVP .....	28
<b>Figura 2. 9</b>	Transmisor Eachine TS832 .....	28
<b>Figura 2. 10</b>	Bandas de frecuencia del TS832 .....	29
<b>Figura 2. 11</b>	Cronograma de actividades .....	37
<b>Figura 3. 1</b>	Diagrama circuital LM7805 .....	39
<b>Figura 3. 2</b>	Placa del circuito LM7805 .....	40
<b>Figura 3. 3</b>	Placa del circuito LM7805 .....	40
<b>Figura 3. 4</b>	Diseño del montaje de los elementos del vehículo .....	42
<b>Figura 3. 5</b>	Diseño del montaje segunda planta.....	43
<b>Figura 3. 6</b>	Diseño del montaje de los servomotores.....	43
<b>Figura 3. 7</b>	Diseño de cubierta del vehículo a), b), c), d) .....	44
<b>Figura 3. 8</b>	Diseño final de la cubierta del vehículo en 3D a), b), c), d) .....	45
<b>Figura 3. 9</b>	Diseño de la aplicación para el sistema .....	46
<b>Figura 3. 10</b>	Ventada de diseño de App Inventor2.....	47
<b>Figura 3. 11</b>	Opciones de paleta en App Inventor2 .....	48
<b>Figura 3. 12</b>	Control remoto para el movimiento de giro de las llantas .....	49
<b>Figura 3. 13</b>	Programación en bloques para el vehículo .....	49
<b>Figura 3. 14</b>	Pantalla de visualización de la dirección IP .....	51
<b>Figura 3. 15</b>	Diseño completo de la aplicación .....	52
<b>Figura 3. 16</b>	Programación en bloques para los servomotores.....	53
<b>Figura 3. 17</b>	Programa final realiza en APP INVENTOR.....	53
<b>Figura 3. 18</b>	Ingreso de enlace para agregar la librería ESP8266 en Arduino .....	54
<b>Figura 3. 19</b>	Descarga de la librería ESP8266.....	55
<b>Figura 3. 20</b>	Diagrama de flujo para los motores reductores .....	55
<b>Figura 3. 21</b>	Diagrama de flujo para los motores reductores .....	59
<b>Figura 3. 22</b>	Diseño de circuito impreso .....	61
<b>Figura 3. 23</b>	Elementos soldados a la placa.....	62
<b>Figura 3. 24</b>	Implementación del circuito regulador de voltaje .....	62
<b>Figura 3. 25</b>	Armado de la primera planta del vehículo .....	63
<b>Figura 3. 26</b>	Armado de la segunda planta del vehículo .....	64

<b>Figura 3. 27</b> Armado del servomotor en la cubierta del vehículo .....	64
<b>Figura 3. 28</b> Proyecto completado, vista lateral .....	65
<b>Figura 3. 29</b> Proyecto completado, vista frontal .....	65

## LISTA DE TABLAS

<b>Tabla 1. 1</b>	Clases de transmisores bluetooth.....	10
<b>Tabla 1. 2</b>	Estándares de bluetooth.....	10
<b>Tabla 1. 3</b>	Estándar WIFI.....	11
<b>Tabla 1. 4</b>	Baterías LiPo en serie.....	13
<b>Tabla 2. 1</b>	Características del NodeMCU 8266.....	20
<b>Tabla 2. 2</b>	Características del módulo ESP8266.....	22
<b>Tabla 2. 3</b>	Tabla de verdad de los motores.....	23
<b>Tabla 2. 4</b>	Características técnicas de la cámara mobius.....	26
<b>Tabla 2. 5</b>	Características técnicas de las gafas FPV Eachine EV800.....	27
<b>Tabla 2. 6</b>	Características técnicas del transmisor FPV Eachine TS832.....	29
<b>Tabla 2. 7</b>	Especificaciones técnicas del Arduino YUN.....	33
<b>Tabla 2. 8</b>	Comparación de las cámaras mobius y GoPro.....	34
<b>Tabla 2. 9</b>	Comparación de los transmisores TS832 y TS353.....	34
<b>Tabla 2. 10</b>	Comparación entre las gafas EV800 y AOMWAY V1.....	35
<b>Tabla 2. 11</b>	Presupuesto total del proyecto.....	35
<b>Tabla 3. 1</b>	Medición del consumo de corriente máxima del circuito.....	41
<b>Tabla 3. 2</b>	Distribución de pines.....	63
<b>Tabla 3. 3</b>	Distribución de pines de alimentación.....	63
<b>Tabla 3. 4</b>	Análisis de funcionamiento del vehículo con la App.....	66
<b>Tabla 3. 5</b>	Análisis de funcionamiento de los servomotores con la App.....	66
<b>Tabla 3. 6</b>	Análisis de funcionamiento del circuito completo.....	67

## **RESUMEN**

En el presente proyecto se desarrolla un VEHÍCULO A CONTROL REMOTO CON VISIÓN EN PRIMERA PERSONA, permite al usuario tener la sensación de manejar n forma directa, ya que la imagen se proyecta en tiempo real.

El prototipo está enfocado a ser un vehículo no tripulado, comandado mediante una aplicación instalada sobre la plataforma Android con tecnología WIFI, para controlar el movimiento de giro de las llantas del vehículo en todas las direcciones, además de manipular la rotación de la cámara a través de servomotores en sentido horizontal y vertical.

Mediante una metodología experimental, se presenta el vehículo como un desarrollo tecnológico y de innovación de sistemas robóticos, con la utilización de técnicas de control remoto de dispositivos y de transmisión de video actuales.

Palabras claves: FPV, NODEMCU, ESP8266, servomotor, App Inventor2, LiPo

## **ABSTRACT**

In the present project a VEHICLE WITH REMOTE CONTROL WITH VISION IN FIRST PERSON is developed, allows the user to have the sensation of driving directly, since the image is projected in real time.

The prototype is focused on being an unmanned vehicle, commanded by an application installed on the Android platform with WIFI technology, to control the turning movement of the vehicle's tires in all directions, in addition to manipulating the rotation of the camera through of servomotors in horizontal and vertical direction.

Through an experimental methodology, the vehicle is presented as a technological development and innovation of robotic systems, with the use of techniques of remote control of devices and current video transmission.

**KEYWORDS:** FPV, NODEMCU, ESP8266, servomotor, App Inventor2, LiPo

# INTRODUCCIÓN

## **Antecedentes**

Al realizar el proceso de investigación del planteamiento del problema se buscó en los diferentes repositorios digitales de todas las universidades, esto da como resultado que todos los métodos empleados para resolver investigaciones similares se centran con diferentes objetivos, distintos métodos de desarrollo en hardware y software.

En el presente diseño e implementación se utiliza elementos y dispositivos actuales en el mercado, que permiten un óptimo desempeño en el funcionamiento del prototipo. En los anexos, se presentan los proyectos encontrados en los repositorios digitales.

## **Planteamiento del problema**

Este proyecto se enfocará especialmente en vehículos terrestres no tripulados con manejo remoto y con un sistema de vista en primera persona, lo cual se desarrollará un prototipo (hardware y software) de vehículo con aplicación para procesos de búsqueda y exploración en sitios desconocidos o de acceso peligroso.

Con la aplicación del método FPV (First Person View / Vista en Primera Persona), se logrará que el piloto tenga una visión más precisa de la ruta sin que la distancia represente dificultad de manejo y sin afectar la visión del entorno. El vehículo será controlado a través de una aplicación móvil creada en App Inventor.

Para el desarrollo de este proyecto se trabajará en torno a tres puntos esenciales que son los fundamentos teóricos, diseño e implementación del prototipo, los mismos serán expuestos en el mismo orden indicado.

## **Justificación**

En el aspecto tecnológico del proyecto, se busca desarrollar un prototipo que este enfocado con el uso de hardware y software actuales, con ello se utiliza el dispositivo móvil celular para el mando del vehículo y las gafas FPV permitan la visualización de todo su entorno.

El proyecto permitirá enfocarse para varios usos, uno de ellos es la utilización del dispositivo como un elemento de búsqueda en lugares de difícil acceso o de entornos desconocidos, ya que al tener la comunicación inalámbrica de un amplio alcance y con el sistema de visión en primera persona permitirá reconocer o encontrar ambientes que el usuario no pueda llegar.

En la implementación del vehículo, se utilizará los elementos que permitan el correcto funcionamiento del equipo, con el objetivo de siempre buscar el costo beneficio de cada uno de ellos, para que el valor económico represente también una ventaja en el mercado actual.

## **Objetivos**

- **Objetivo General:**

- Desarrollar un vehículo a control remoto con visión en primera persona (FPV – First Person View)

- **Objetivos Específicos:**

- Definir los elementos electrónicos y mecánicos del vehículo a control remoto con FPV y herramientas de software para el prototipo.

- Diseñar el esquema electrónico del hardware del vehículo.
- Crear una aplicación para el control del vehículo.
- Implementar un sistema de comunicación WIFI entre el vehículo y el sistema de control remoto.
- Implementar el sistema de visión en primera persona (FPV), a través una cámara panorámica instalada en el vehículo y gafas de visión en tiempo real en sitios remotos.
- Realizar pruebas de funcionamiento.

### **Alcance**

Se construirá el vehículo a escala con los elementos electrónicos seleccionados, se desarrollará un entorno WIFI entre el sistema de control remoto y el vehículo, que permita dirigir el equipo de manera inalámbrica. Para el mando se realizará una aplicación compatible con el sistema operativo de Android. Finalmente se implementará un sistema FPV (First Person View) dentro del vehículo que pueda visualizar todo el entorno mediante unas gafas.

### **Descripción de los capítulos**

La estructura del documento escrito de la tesis está compuesta por cuatro capítulos, en los cuales se explica a detalle el desarrollo completo del proyecto de titulación. En el capítulo uno se presenta los fundamentos teóricos, en los cuales están basados todos los estudios y análisis del proyecto, es decir, las características, modo de uso, funcionamiento y demás parámetros que se requieren conocer antes de comenzar cualquier etapa de implementación.



En el capítulo dos se describe el marco metodológico, es decir, cuales son las distintas fases del proyecto de investigación, así como los métodos y técnicas empleadas en cada uno de ellos.

En el capítulo tres se desarrolla la propuesta del proyecto de titulación, en ella se encuentran gráficos, esquemas, que representen la idea principal del proyecto, los módulos que lo componen, los aspectos técnicos, análisis de costos y las ventajas del producto.

En el capítulo cuatro esta finalmente la implementación del producto, es decir, el proceso de construcción tanto en hardware como en software de una forma detallada, pruebas de funcionamiento y el análisis de los resultados.

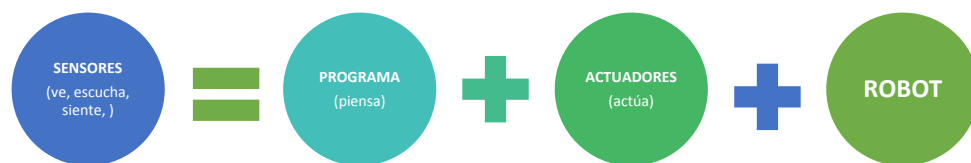
# CAPÍTULO I

## FUNDAMENTACIÓN TEÓRICA

### 1.1. Robótica

La robótica es una actividad que integra la informática, la electrónica y la mecánica, para el desarrollo de robots; actividad que a partir de la creación de la computadora ha tenido una evolución exponencial, los robots van a ser cada vez más compactos, de un coste menor y sobre todo con una capacidad de procesamiento que permite que los robots sean más rápidos y autónomos.

Los robots son máquinas programadas que realizan tareas definidas por el humano conformado por partes mecánicas para el movimiento, sensores que equivalen a los sentidos humanos y procesadores que son el cerebro de los robots (Figura. 1.1); todos estos elementos son seleccionados técnicamente con las funciones que se requiere que cumplan.



**Figura 1 1** Esquema general de un robot

En 1942 Issac Asimov, escritor y profesor de bioquímica en la Universidad de Boston, de origen ruso, cuyas obras giran alrededor de la ciencia ficción, la historia y la divulgación científica, en su libro *Círculo Vicioso* planteó las tres leyes fundamentales de la robótica, que hasta la actualidad son consideradas válidas:

- 1) “Un robot no hará daño a un ser humano o, por inacción, permitirá que un ser humano sufra daño.”
- 2) “Un robot debe obedecer las órdenes dadas por los seres humanos excepto si estas órdenes entrasen en conflicto con la 1ª ley.”
- 3) “Un robot debe proteger su propia existencia en la medida en que esta protección no entre en conflicto con la 1ª o la 2ª Ley.”

### 1.1.1. Clasificación de robots

Los robots han sido clasificados de diferentes maneras, sin embargo, para el presente proyecto resulta conveniente clasificarlos en función del entorno de trabajo, de acuerdo a Reyes Cortés en el libro de robótica indica que se dividen en 3 grupos:

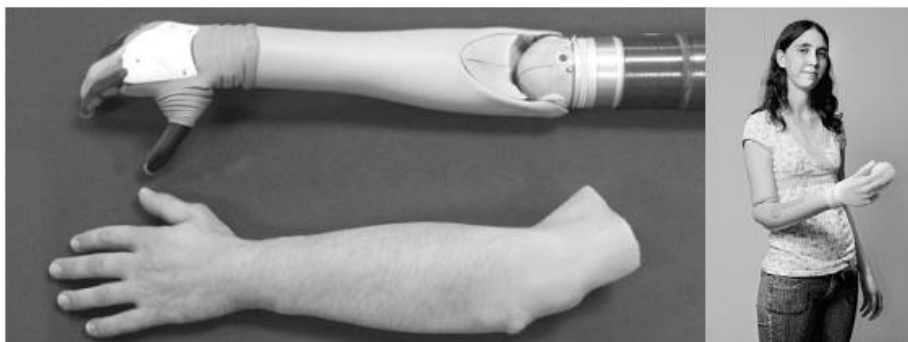
- **Robots industriales:** Están diseñados para realizar trabajos industriales de forma automatizada, tales como levantar cargas pesadas a velocidad altas y con la mayor precisión.



Figura 1 2 Brazo robótico industrial

- **Robots médicos:** Básicamente son prótesis, que además de cumplir las funciones requeridas para sustituir partes del cuerpo, son diseños muy similares a las

extremidades del cuerpo humano. Su respuesta depende de señales nerviosas y musculares. También hay robots que trabajan como asistentes quirúrgicos, los cuales son desarrollados con la más alta precisión posible.



**Figura 1 3** Prótesis de un brazo robótico

- **Robots móviles:** Son robots dotados de un sistema de locomoción con la capacidad de navegar a través de un determinado ambiente de trabajo y sus aplicaciones son enfocadas especialmente a tareas que normalmente son riesgosas o nocivas en los humanos, es por ello que se requiere de vehículos no tripulados. Las avionetas para fumigar cultivos son un ejemplo esta clasificación de robots.



**Figura 1 4** Avioneta no tripulada para fumigar cultivos

El trabajo está enfoca en robots móviles, ya que el vehículo a construir está dentro de esta clasificación de robots.

### **1.1.2. Robots móviles:**

Los robots móviles tienen la capacidad de moverse en cualquier entorno en el que sean diseñados y dependerán de los dispositivos que se utilicen para su movimiento, como ruedas, tracks, piernas, etc. Dentro de los robots móviles hay una sub-clasificación de acuerdo al entorno en el que se movilizan: aéreos, submarinos y terrestres.

Los móviles no tripulados tienen sus orígenes con en el desarrollo de las aeronaves llamadas “torpedos aéreos”, de uso militar, que posteriormente con su evolución se desarrollaron en aeronaves de investigación, de reconocimiento, de combate e inclusive de entretenimiento, sí, la apertura para que estas aeronaves, actualmente llamadas “drones”, sean también de uso civil, al igual que los vehículos submarinos y terrestres.

## **1.2. Navegación de Robots Móviles:**

En la actualidad existe una gran cantidad de dispositivos que permiten el control de los robots móviles no tripulados, de los cuales destacan los utilizados mediante mandos físicos, que le permiten al usuario comandar cada uno de sus movimientos y otros que son autómatas los cuales utilizan sensores para guiar su camino.

### **1.2.1. Robots controlados por sensores:**

Los robots controlados por sensores son sistemas autónomos que residen en la misma máquina, esto les permite operar sin una conexión física, se los llama inteligentes ya que pueden tomar decisiones por sí mismos a partir de lecturas, selección e integración de los datos tomados por los sensores, los más comúnmente utilizados son los sensores de proximidad, sensor ultrasónico, sensores de movimiento, entre otros.

### **1.2.2. Robots controlados por mandos físicos:**

El control remoto más utilizado es el que contiene una serie de botones físicos que permiten mover al equipo en cualquier dirección, una variante de control es el llamado joystick, que está formado por una palanca que puede girar sobre una base que informa el ángulo o dirección que tiene en ese momento al dispositivo que está controlado, cualquiera de los dos mandos son dispositivos de control principal cuyo mercado ha trascendido del campo militar al uso civil.

Otro tipo de mando físico son los programas o aplicaciones que se instalan en un computador, teléfono móvil o Tablet; éstos tienen un alcance muy similar a los anteriores ya que se pueden conectar vía Bluetooth o por medio de WIFI, el alcance dependerá de cual se escoja, al igual que la programación.

- **Bluetooth:**

La tecnología Bluetooth fue desarrollada por ERICSSON en 1994, luego la en 1998 se formó el grupo Bluetooth Special Interest (Bluetooth SIG) dentro de las cuales estan Agere, Ericsson, IBM, Intel, entre otros. (Prabhu, 2004)

Es una especificación tecnológica de las redes inalámbricas, la función principal es de transmitir y recibir datos y voz entre distintos dispositivos con un enlace de radiofrecuencia de 2.4 GHz, es decir no requiere ningún tipo de cable de conexión, esto permite crear redes inalámbricas personales o domesticas para sincronizar y también compartir cualquier información entre usuario.

Para este estándar se ha definido tres clases o categorías de transmisores, en donde el alcanza va a variar en función de la potencia radiada, dependerá de cada fabricante colocar un transmisor Bluetooth con características ya definidas en los dispositivos que presente al mercado.

La tecnología Bluetooth no necesita estar en línea de vista con el otro dispositivo, es decir pueden estar separados por un muro u obstáculo, el estándar bluetooth se divide en múltiples normas que están dentro de la IEEE 802.15X.

**Tabla 1. 1** Clases de transmisores bluetooth

Clase	Potencia Máxima(mW)	Alcance en exterior(metros)
Clase 1	100	100
Clase 2	2.5	10
Clase 3	1	1

Bluetooth Technology 2004

**Tabla 1. 2** Estándares de bluetooth

ESTÁNDAR	CARACTERÍSTICAS
IEEE 802.15.1	Este estándar puede alcanzar velocidades de hasta 1 Mbps
IEEE 802.15.2	Trabaja en frecuencias de 2.4 GHz
IEEE 802.15.3	Estándar de alta tasa de transmisión de hasta 20 Mbps a 2.4 GHz
IEEE 802.15.4	Estándar de baja tasa de transferencia de 200 kHz, es decir de corto alcance.

Bluetooth Technology 2004

- **WIFI (*Wireless Fidelity*):**

WIFI (Wireless Fidelity) es una tecnología de comunicación inalámbrica que tienen como objetivo conectar a Internet múltiples equipos electrónicos como computadoras, teléfonos móviles, tablets y todo tipo de equipo que tenga un chip con esta tecnología. (Huidrobo Moya & Roldán Martínez, 2016)

La comunicación inalámbrica WIFI comprende un conjunto de estándares de redes de uso inalámbrico que están especificadas dentro de la 802.11X, esto permite la compatibilidad e interoperabilidad de los dispositivos que trabajen con esta tecnología; para su funcionamiento se requiere de un enrutador o router que esté conectado a Internet y cuente con una antena la cual va a irradiar la señal de la misma y será captada por el módulo WIFI que posea el equipo portable.

También se puede realizar una red local mediante un dispositivo móvil y que éste trabaje como router y varios equipos se conecten a él; el router debe proporcionar un usuario y contraseña, para que de esta manera se garantice estar dentro del dominio.

La principal ventaja sobre las redes inalámbricas locales que tienen WIFI es su alcance y dependerá mucho del tipo de proyecto para valorar y escoger uno de ellos. Se presenta a continuación los estándares más utilizados por los dispositivos celulares.

**Tabla 1. 3** Estándar WIFI

<b>Estándar</b>	<b>Velocidad de transmisión máxima(Mbps)</b>	<b>Throughput máximo (Mbps)</b>	<b>Banda de Frecuencia</b>	<b>Radio de cobertura</b>
IEEE 802.11a/h	54 Mbps	22 Mbps	5 GHz	185 m
IEEE 802.11b	11 Mbps	6 Mbps	2.4 GHz	140 m
IEEE 802.11g	54 Mbps	22 Mbps	2.4 GHz	150 m
IEEE 802.11n(40MHz)	>300 Mbps	>100 Mbps	5 GHz	300 m
IEEE 802.11n(20MHz)	144 Mbps	74 Mbps	2.4 GHz 5 GHz	300 m

Comunicaciones en redes Wlan, Huidrobo J., Roldan D. 2005

### 1.3. Arduino

El dispositivo comenzó en el 2005 en Ivrea, provincia de Turín, en Italia dentro del Instituto de Interactividad y Diseño, a cargo de los profesores David Cuartielles y Massimo Banzi quienes propusieron crear un dispositivo que permita desarrollar herramientas de hardware de fácil programación para diferentes aplicaciones electrónicas, que puedan ser utilizadas por diversos niveles académicos y de costos accesibles.

Este proyecto fue creado bajo el principio de open source, es decir de código abierto, para que todos los usuarios puedan manipular y modificar sin ningún tipo de encriptación, es un programa de dominio público.

El dispositivo está formado básicamente de un microcontrolador de entrada y salida, que es programada mediante una biblioteca que reduce la escritura de la programación en CC++, un micro controlador llamado MCU es un chip en el cual contiene un procesador, memoria y periféricos de entrada y salida.



No está enfocado para un sector específico de la industria tecnológica, por lo que es muy utilizado en las universidades, ya que forma parte de proyectos de ingeniería de todo tipo, de esta manera los usuarios pueden experimentar físicamente todos los conocimientos y experiencias adquiridas en las aulas.

Es una plataforma de inicio al mundo tecnológico, para incrementar el entusiasmo por ésta rama, desde edades tempranas.

#### **1.4. Regulador de Voltaje Step – Down**

En la alimentación de los servomotores se requiere 5 voltios recomendados, pero la fuente que se va a utilizar es de 7.4 voltios, para ello se requiere un regulador de voltaje que permita disminuir a la cantidad de tensión deseada y de manera constante. Se utilizará un transistor LM7805 el cual permitirá regular el voltaje.

El transistor LM7805 es un transistor que contiene en su interior una serie de transistores bipolares NPN y PNP en cascada, en el exterior el integrado presenta tres pines el primero es de alimentación de entrada, el segundo es tierra y el tercer pin es de voltaje de salida, estos deben estar acompañados por capacitores y un diodo de protección para el circuito, éstos un óptimo rendimiento a los servomotores que están incluidos en el proyecto.

#### **1.5. Batería**

Es un dispositivo el cual puede acumular o almacenar energía eléctrica con diferentes tipos de sales o polímeros en su interior, que permite ser suministrado a equipos para su alimentación y operación. La cantidad de energía que pueda suministrar dependerá de la capacidad de alimentación y almacenaje del dispositivo, así como de sus componentes.

El presente proyecto utilizará una batería para Radio control denominada RC, la cual permitirá suministrar energía a todos los dispositivos.

Se clasifican de la siguiente manera:

- NiCE son baterías de Níquel Cadmio
- NiMH son baterías de Níquel Metal Hidruro
- IonLitio son Baterías de Iones de Litio
- LiPo son Baterías de Polímero de Litio

### 1.5.1. Baterías LiPo

Son baterías recargables que son muy utilizadas en los sistemas eléctricos de radiocomunicación en especial los sistemas aéreos, por lo que se les considera como la principal fuente de alimentación para que los vuelos eléctricos sean viables y reemplacen a los que funcionan con combustible, las baterías LiPo entre sus principales características tiene:

- Son de un peso ligero y pueden fabricarse casi de cualquier forma y tamaño.
- Tiene una gran capacidad de almacenamiento de energía eléctrica.
- Las baterías LiPo tienen una alta tasa de descarga para alimentar sistemas eléctricos más exigentes.

Las baterías en serie denominada por la letra “S”, se presentan a continuación en la tabla 1.4:

**Tabla 1. 4** Baterías LiPo en serie

<b>Baterías LiPo en Serie</b>	
Baterías de 3.7 voltios	1 celda x 3.7 voltios (1S)
Baterías de 7.4 voltios	2 celdas x 3.7 voltios (2S)
Baterías de 11.1 voltios	3 celdas x 3.7 voltios (3S)
Baterías de 14.8 voltios	4 celdas x 3.7 voltios (4S)
Baterías de 18.5 voltios	5 celdas x 3.7 voltios (5S)
Baterías de 22.2 voltios	6 celdas x 3.7 voltios (6S)

Las baterías LiPo tienen celdas de 3.7 voltios y 4.2 voltios cuando están totalmente cargadas, las celdas están conectadas en serie para proporcionar voltajes más altos, se puede encontrar en el mercado baterías de hasta 6 celdas.

Las baterías también pueden ser conectadas en paralelo para aumentar la capacidad, en las especificaciones se denota con la letra "P".

La nomenclatura puede ir junta si se requiere, si tiene solo 1P no se necesita especificar ya que no tiene conectado en paralelo ninguna celda adicional, pero si se tiene por ejemplo 2S2P quiere decir que se tiene 2 celdas conectadas en serie y éstas conectadas a su vez en paralelo con otro conjunto de 2 celdas en serie.

## **CAPÍTULO II**

### **PROPUESTA**

En el presente capítulo se describe el proceso del funcionamiento del vehículo a control remoto con sistema FPV, cuales son los aspectos importantes que se deben tener en cuenta como alimentación del circuito y sistemas de control y visualización.

El proyecto se iniciará con el armado del vehículo a escala, el cual estará formado por un circuito de control para los motores de las llantas mediante el motor driver o puente H L298N, así el vehículo puede moverse en todas las direcciones.

El motor driver está conectado a los pines del NodeMCU8266, el cual está programado mediante el lenguaje IDE de Arduino para que pueda controlar cada uno de los movimientos del vehículo, se debe tomar en cuenta que el NodeMCU8266 va a ser alimentado con una fuente de 5 voltios y el motor Driver soporta fuentes de 7 a 12 voltios por lo que se requiere un regulador de voltaje ya que se utilizará una única alimentación para el circuito, éste se va a utilizar el módulo LM2596, que regule de los 7,4 voltios de salida de la batería a 5 voltios.

Con el vehículo terminado a escala, se procede a realizar el armado de los servomotores que son fundamentales en el sistema FPV del proyecto, ya que son los encargados de suministrar movilidad a la cámara tanto horizontalmente como verticalmente, para ello se utilizó dos servomotores SG90 que pueden girar hasta 180 grados y que están montados sobre una estructura plástica. La programación del SG90 se encuentra en el NodeMCU8266 en lenguaje ID de Arduino.

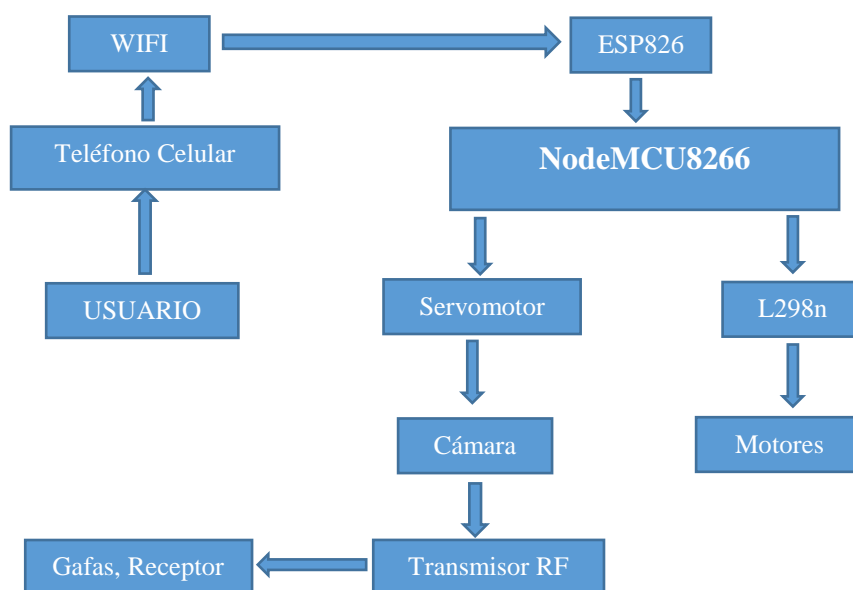
Los servomotores están conectados al transmisor de radio frecuencia TS832, esto permitirá enviar todas las imágenes en tiempo real del vehículo hacia el receptor que se encuentra en las gafas FPV.

Luego se desarrollará la aplicación en APP Inventor 2 para el celular, se va a controlar remotamente a todo el sistema del vehículo de una manera inalámbrica, con lo que permitirá concluir el sistema FPV.

En la alimentación de todo el sistema se utilizará una batería LIPO de 7,4 voltios con 1.5 Amperios de corriente de tipo 2S, finalmente se modelará una carcasa de todo el vehículo en el programa SolidWorks, esto hace que sea estéticamente adecuado en la presentación del proyecto final.

## 2.1. Diagrama de bloques del sistema

Para el presente proyecto se ha diseñado el diagrama de bloques en el cual se detalla los elementos principales del circuito y su comunicación entre ellos en la figura 2.1.



**Figura 2. 1** Diagrama de bloques del sistema

## 2.2. Diagrama de flujos del sistema

El diagrama de flujo que se presenta en la figura 3.2, especifica el modo de comunicación de todo el sistema, comienza con el encendido de la aplicación instalada en el teléfono móvil celular del usuario, éste contiene los botones de mando que permiten controlar al vehículo. La aplicación se conecta con el módulo NodeMCU8266 vía WIFI, es decir una conexión uno a uno.

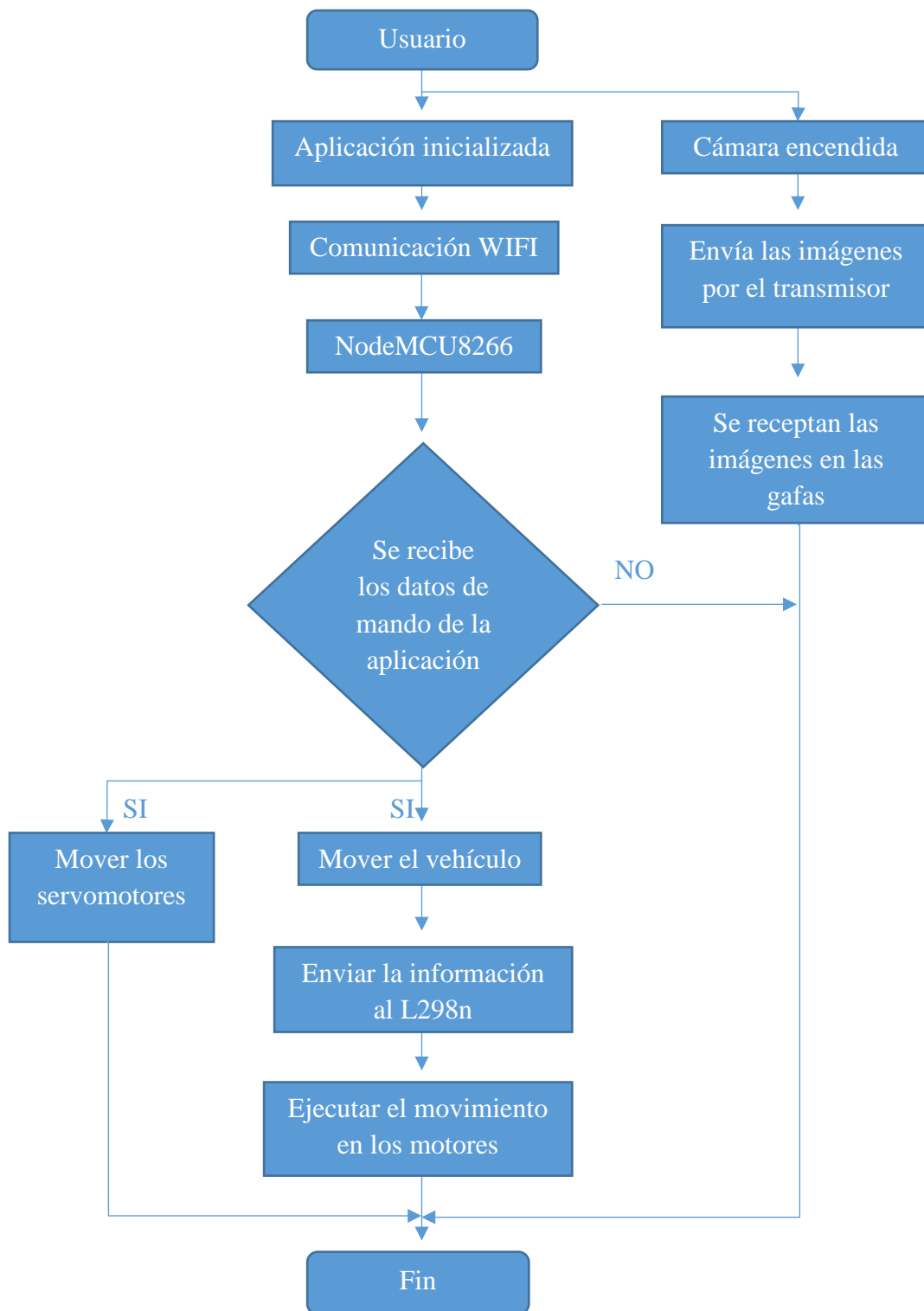
La información que fue recibida por el controlador es administrada hacia los dos sistemas de actuadores que tiene el proyecto, uno, es el movimiento de giro de los motores del vehículo y el otro es la rotación de los servomotores.

La información que contiene el movimiento de los motores es enviada desde el NodeMCU8266 hacia el módulo L298n que permite el giro de las llantas hacia una dirección determinada.

La señal que contiene el control de los servomotores, logra que éstos giren en dirección horizontal y vertical, que en la cámara instalada sobre ellos, pueda tener un ángulo de visión de su entorno más extenso.

El sistema FPV o visión en primera persona comienza con el encendido de la cámara en la base superior del vehículo, está conectada a un transmisor de video que envía la señal en tiempo real hacia las gafas que tiene colocado el usuario, para que pueda tener una imagen real del entorno del vehículo.

El usuario debe verificar el estado de carga de las gafas y colocar la antena de recepción que permite que lleguen las imágenes desde el vehículo, el usuario podrá observar todo el entorno del prototipo que sea manejado por la aplicación. En la figura 2.2 se presenta el diagrama de flujo.



**Figura 2. 2** Diagrama de flujo del sistema

## **Módulos del sistema**

El sistema completo que compone el vehículo a control remoto con visión en primera persona, se subdivide en módulos, en los cuales cada uno de los elementos y dispositivos que lo forman son descritos y expuestos como parte de la propuesta.

Los módulos del sistema son:

- Aplicación de mando del sistema
- Módulo de control
- Actuadores
- Sistema FPV(visión en primera persona)

### **2.3. Aplicación de mando del sistema**

La aplicación de mando del sistema es el control remoto, que permite al usuario dar el movimiento de giro de los motores y los servomotores hacia la rotación de la cámara en el sistema FPV. Los requisitos básicos que necesita la aplicación para ser instalada en un teléfono celular son:

- El teléfono celular debe ser un Smartphone
- El teléfono celular debe utilizar el sistema operativo Android igual o mayor a la 2.3 Gingerbread

Las marcas y modelos no son un requerimiento adicional, por lo que todos los Smartphone pueden ser utilizados para la instalación de la aplicación.

### **2.4. Módulo de control**

El módulo de control principal NodeMCU8266 ejecuta todas las instrucciones que son enviadas desde la aplicación comandada por el usuario, ésta permite que los actuadores sean administrados en cada una de las acciones requeridas. Además, se requiere el módulo L298n que permite la conexión de los motores de corriente directa para su funcionamiento.



### 2.4.1. Placa NodeMCU ESP8266

Es una placa que está basada en el módulo WIFI ESP8266 de la empresa Espressif Systems, el termino NodeMCU se refiere al Firmware (software de control) que usa lenguaje de programación de código abierto. Los códigos de programación que admite la placa son:

- Comandos AT
- Código Arduino
- MicroPython
- Espruino
- ESPbasic

Para el presente proyecto se va a utilizar el NodeMCU 8266, con características similares que las placas Arduino, pero su ventaja es que ya viene incorporado el módulo WIFI ESP8266 y además se los puede programar con código Arduino (Del Valle, 2018). Las características principales de las placas se presentan en la tabla 2.1 a continuación:

**Tabla 2. 1** Características del NodeMCU 8266

<b>NodeMCU8266</b>	
Voltaje de Alimentación (USB):	5 voltios DC
Voltaje de Entradas/Salidas:	3.3 voltios DC
SoC:	ESP8266 (Módulo ESP-12)
CPU:	Tensilica Xtensa LX3 (32 bit)
Frecuencia de Reloj:	80MHz/160MHz
Instruction RAM:	32KB
Data RAM:	96KB
Memoria Flash Externa:	4MB
Pines Digitales GPIO:	17 (pueden configurarse como PWM a 3.3V)
Pin Analógico	ADC: 1 (0-1V)
Chip USB-Serial:	CP2102
Protocolos:	802.11 b/g/n
Stack de Protocolo:	TCP/IP integrado



**Figura 2. 3** NodeMCU8266

### 2.4.2. Módulo ESP8266

El módulo permite la comunicación inalámbrica WIFI compatible con el protocolo TCP/IP, el cual permite dar acceso a cualquier microcontrolador a una red, utiliza una CPU de 32 bits, se debe tomar muy en cuenta que trabaja a voltajes de alimentación de entre 3 a 3,6 voltios por lo que niveles mayores a éstos daña el módulo. Soporta IPv4 y los protocolos TCP/UDP/HTTP/FTP.

### 2.4.3. Modos de Operación

El modulo está diseñado para estar en tres diferentes modos de consumo de energía de acuerdo a su tipo de trabajo como dispositivos IoT o móvil; las arquitecturas de consumo de energía son:

- Active mode o modo activo: es decir que trabaja en pleno rendimiento.
- Sleep mode o modo dormido: en este modo solo el RTC (Real Time Clock) está en actividad mantiene la sincronización. Permite quedarse en modo alerta para los posibles eventos que se susciten y requiera activarse, siempre mantiene en memoria los datos de conexión con la comunicación WIFI, su consumo está entre los 0,6 mA a 1mA.
- Deep sleep o modo de sueño profundo: en este caso el RTC está encendido, pero no operativo, primero debe pasar por el modo dormido antes de activarse, en este modo se debe tener cuidado con los datos ya que funciona como si estuviese

apagado, por lo que todos los datos que no están almacenados se perderán, consume 20 micro amperios. En la tabla 2.2 se presenta las características del módulo ESP8266

**Tabla 2. 2** Características del módulo ESP8266

<b>Módulo ESP8266</b>	
CPU	Tensilica L106 de 32 bits
Protocolo soportado	802.11 b/g/n
WIFI Direct (P2p)	Soft Access Point
Protocolo integrado	TCP/IP
Regulador de manejo de energía	Integrado en el módulo
Corriente	80 micro amperios
Sensor de Temperatura	Integrado
Consumo en modo de baja energía	Menor a 10 micro amperios
Voltaje	3 - 3.6 voltios



**Figura 2. 4** ESP8266

#### **2.4.4. Puente H L298N**

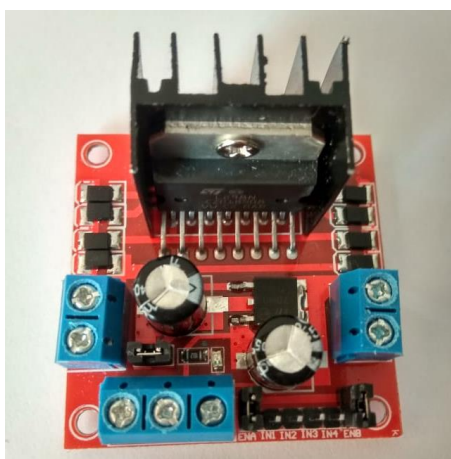
Es un módulo que permite controlar dos motores de corriente en DC o moto reductores que no sobrepasen una corriente de dos Amperios, este dispositivo permite controlar el sentido de giro y la velocidad de los motores mediante señales TTL, que ingresan a través de micro controladores como las tarjetas de desarrollo de Arduino, Rasoberry PI, lauchpads de Texas Instrument. El L298N también puede alimentar a un circuito adicional

ya que tiene una salida de 5 voltios, el inconveniente de este pin es que no puede entregar más de 50mA, para el presente circuito no se utiliza. (López Razo, 2018) Los elementos que contiene el modulo son:

- Conectores Output A y Output B son las salidas para la conexión de los dos motores.
- Conectores Input, los cuales son terminales de control.
- Regulador de voltaje LM7805.
- Jumper de habilitación para utilizar el regulador LM7805.

**Tabla 2. 3** Tabla de verdad de los motores

<b>TABLA DE VERDAD DEL MOTOR A</b>			
<b>ENA</b>	<b>IN 1</b>	<b>IN 2</b>	<b>DESCRIPCIÓN</b>
0	N/A	N/A	Motor A está apagada
1	0	0	Motor A se detiene (frenado)
1	0	1	El motor A está encendido y gira en sentido contrario
1	1	0	Motor A está encendido y gira hacia delante
1	1	1	Motor A se detiene (frenado)
<b>TABLA DE VERDAD DEL MOTOR B</b>			
<b>ENB</b>	<b>IN 3</b>	<b>IN 4</b>	<b>DESCRIPCIÓN</b>
0	N/A	N/A	Motor B está apagada
1	0	0	Motor B se detiene (frenado)
1	0	1	El motor B está encendido y gira en sentido contrario
1	1	0	Motor B está encendido y gira hacia delante
1	1	1	Motor B se detiene (frenado)



**Figura 2. 5** Puente H L298N

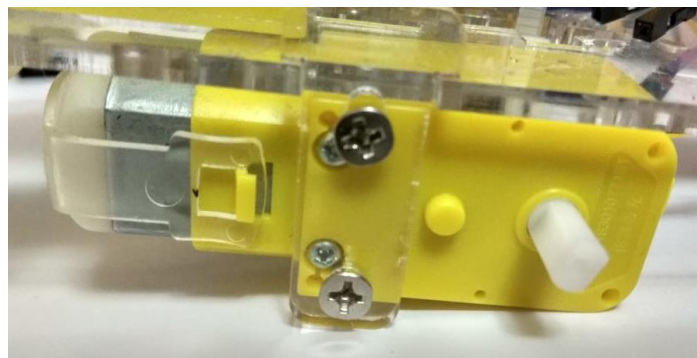
## 2.5. Actuadores

Son los dispositivos que transforman la energía eléctrica en un proceso automatizado, mediante la administración de sus movimientos con el controlador, en este proyecto se utiliza el NodeMCU8266. Los actuadores que se van a utilizar son moto reductores para el movimiento del vehículo y los servomotores dan el movimiento de la cámara.

### 2.5.1. Motor Reductor

Este tipo de elementos poseen motores conectados directamente a ellos, con un sistema de engranaje que les permite disminuir la velocidad de giro, con el propósito de aumentar su fuerza mecánica y también de cambiar el sentido de giro, son utilizados en una gran variedad de proyectos de baja, mediana y alta capacidad.

En el proyecto se va a utilizar como mecanismo de movimiento a las ruedas del vehículo a escala, para así controlar el giro de las llantas.



**Figura 2. 6** Motor reductor

### 2.5.2. Servomotores

Un servomotor es muy utilizado en la robótica, consta de un motor de corriente continua, un sistema de engranajes que permite que la velocidad de giro disminuya, pero aumente en fuerza mecánica los diversos tipos de proyectos, tiene un potenciómetro que da

el giro y finalmente un circuito de control el cual recibe los pulsos PWM para dar los grados de giro del servomotor.

EL dispositivo que se va a utilizar es el SG90, el cual tiene un ancho de banda de 20ms y que permite tener un ángulo de giro de 0 a 180 grados, los cuales dependerán del ancho de pulso que identifique el sistema de control, para calcular el tiempo que requiere girar a un determinado grado es mediante la siguiente expresión:

$$t = 1 + \frac{\text{grados}}{180^\circ} \quad \text{Ec. 1}$$

Los servomotores permitirán en el proyecto que la cámara FPV que va a estar instalado sobre ellos pueda girar tanto vertical como horizontalmente, con un ángulo de giro que el usuario podrá controlar dentro del rango permitido que es de 0 a 180 grados.

## **2.6. Sistema FPV (Visión en primera persona)**

Las gafas FPV son utilizadas para visión en primera persona, esto da la sensación a la persona de controlar el equipo de estar dentro del aparato manejándolo, son muy utilizados en Drones, carros a escala, entre otros.

Este tipo de sistema de visión permite que el operador cree un ambiente más realista en el cual genere más emociones y sensaciones al poder observar lugares de difícil acceso.

El sistema FPV (First Person View) está compuesto por una cámara que será montada en el equipo, un transmisor de video que mediante señales de radiofrecuencia transmitirá las imágenes y sonidos al receptor, las gafas que presentan un monitor incorporado que permitirá visualizar la trayectoria del equipo en tiempo real.

Cada uno de estos dispositivos existen en el mercado con una gran variedad de marcas y modelos, pero todos presentan casi las mismas características y funciones, por lo que para

el presente proyecto se decidió escoger los equipos con base a los valores económicos que poseen.

### 2.6.1. Cámara FPV

Las cámaras son la parte más importante sobre el sistema FPV, ya que ellas captarán las imágenes del entorno en el que se encuentre el equipo, las características de visualización dependerán del tipo de sensor que posea, puede ser un CMOS o CCD. Las cámaras que poseen los sensores CMOS tienen un valor más económico con respecto a las segundas, esto se debe a que el CMOS no controla en gran calidad los cambios de luz del ambiente. Pero en condiciones normales cualquiera de las dos es una gran elección.

La cámara que se va a utilizar es la Mobius Action Cam gran angular C2 la cual presenta en la tabla 2.4:

**Tabla 2. 4** Características técnicas de la cámara mobius

<b>Mobius Action Cam gran angular C2</b>	
Grabación de audio y video	1080P HD
VAFO	131 grados
FOV	1080p_5-30fps, 720p_5-30fps, WVGA_5-30fps
Resolución	2304 x 1536, 1920 x 1080, 1280 x 720
Resolución de visualización de TV	4:3 o 16:9
Sonido de Película	Apagado y tres niveles de volumen
Tarjeta de almacenamiento	Hasta 32 GB no incluido en el equipo
Uso de tiempo	80 minutos
Dimensiones	5 x 2.5 x 2.5
Peso	37 gramos



**Figura 2. 7** Mobius action cam

### 2.6.2. Gafas FPV

Una de las partes más importantes dentro de un sistema FPV es el receptor de imágenes en tiempo real, que para este caso son las gafas FPV, las cuales brindan gran comodidad y percepción del entorno en el que se está inmerso el vehículo transmisor. Son utilizados en múltiples proyectos terrestres y aéreos.

Está formada principalmente por un receptor de video, una pantalla donde se recibe la señal de las imágenes, unas baterías de alimentación que dependerá del modelo y marca del fabricante y por ende el tiempo que puede ser utilizado y finalmente la carcasa o cubierta que se sujeta al rostro del usuario.

Existe en el mercado una gran variedad de equipos con similares características, lo que cambia es las dimensiones físicas y modelos de distintas marcas en el mercado.

En este proyecto se va a utilizar unas gafas FPV modelo Eachine EV800, para ello a continuación en la tabla 2.5 se presenta las características del modelo escogido.

**Tabla 2. 5** Características técnicas de las gafas FPV Eachine EV800

<b>Gafas FPV Eachine EV800</b>	
Pantalla	LCD de 5 pulgadas
Resolución de Pantalla	800 x 480
Receptor	Súper sensible de 5,8 GHz con 40 CH
Búsqueda de frecuencias	Automático
Batería	Interna
Voltaje de Batería	3,7 voltios
Corriente de Batería	2000 mAh
Tiempo de uso	3,5 horas
Peso	375 gramos
Dimensiones	180 x 145 x 82 mm





**Figura 2. 8** Gafas FVP

### 2.6.3. Transmisor FPV

El transmisor es un dispositivo encargado de enviar la información, desde el origen de manera inalámbrica hacia el receptor que está en las gafas FPV. Las frecuencias que se pueden utilizar ya que se encuentran libres son las de 2.4 GHz y 5.8 GHz, el inconveniente de 2.4 GHz es que casi todos los equipos de conexión inalámbrica utilizan esta frecuencia, por lo que genera interferencias en la señal transmitida, por lo que se utiliza la de 5.8 GHz. Los canales son frecuencias que ya están predefinidas por su uso libre y que constan de 10 bandas y 80 canales para transmisión como recepción. En el presente proyecto se utilizó el transmisor Eachine TS832 FPV 5.8 GHz de 48 canales disponibles.



**Figura 2. 9** Transmisor Eachine TS832

TS832 Instruction Manual					
MODEL:TS832 5.8G 600mW 48 Channels AV Transmitter Module					
Modulate		Wideband FM Modulate			
Video Format		NTSC/PAL			
Characteristics		Value			Units
		Min.	Typ.	Max.	
1	Output Impedance	---	50	---	Ohm
2	Output Power	27	27.5	28	dBm
Frequency Range	Channel Frequency				
FR1	CH1	---	5865	---	MHz
	CH2	---	5845	---	MHz
	CH3	---	5825	---	MHz
	CH4	---	5805	---	MHz
	CH5	---	5785	---	MHz
	CH6	---	5765	---	MHz
	CH7	---	5745	---	MHz
	CH8	---	5725	---	MHz
FR2	CH1	---	5733	---	MHz
	CH2	---	5752	---	MHz
	CH3	---	5771	---	MHz
	CH4	---	5790	---	MHz
	CH5	---	5809	---	MHz
	CH6	---	5828	---	MHz
	CH7	---	5847	---	MHz
	CH8	---	5866	---	MHz
FR3	CH1	---	5705	---	MHz
	CH2	---	5685	---	MHz
	CH3	---	5665	---	MHz
	CH4	---	5645	---	MHz
	CH5	---	5885	---	MHz
	CH6	---	5905	---	MHz
	CH7	---	5925	---	MHz
	CH8	---	5945	---	MHz
FR4	CH1	---	5740	---	MHz
	CH2	---	5760	---	MHz
	CH3	---	5780	---	MHz
	CH4	---	5800	---	MHz
	CH5	---	5820	---	MHz
	CH6	---	5840	---	MHz
	CH7	---	5860	---	MHz
	CH8	---	5880	---	MHz
FR5	CH1	---	5658	---	MHz
	CH2	---	5695	---	MHz
	CH3	---	5732	---	MHz
	CH4	---	5769	---	MHz
	CH5	---	5806	---	MHz
	CH6	---	5843	---	MHz
	CH7	---	5880	---	MHz
	CH8	---	5917	---	MHz
FR6	CH1	---	5474	---	MHz
	CH2	---	5492	---	MHz
	CH3	---	5510	---	MHz
	CH4	---	5528	---	MHz
CH5	---	5546	---	MHz	
CH6	---	5564	---	MHz	
CH7	---	5582	---	MHz	
CH8	---	5600	---	MHz	
4	Operating Voltage	7.0	12	16	V
5	Supply Current	---	220	---	mA
6	Operating Temperature	-10	---	+85	°C
7	Video Band Width	0	---	8.0	MHz
8	Audio Carrier Frequency	---	6.5	---	MHz
9	Video Input Level	0.8	1.0	1.2	Vp-p
10	Video Input Impedance	---	75	---	Ohm
11	Audio Input Level	0.5	---	2.0	Vp-p
12	Audio Input Impedance	---	10K	---	Ohm
13	Weight	---	22	---	g
14	Antenna Connector	SMA Connector			
15	Dimension(L×W×H)	54×32×10mm			

Figura 2. 10 Bandas de frecuencia del TS832

Este transmisor va conectado a la cámara FPV del vehículo a escala, lo que permite transmitir cada uno de sus movimientos directamente a las gafas las cuales tienen un Receptor con las mismas características de canales de transmisión dentro de los 5.8 GHz de frecuencia, esto logra un óptimo desempeño al sistema. Se presenta a continuación en la tabla 2.6 las especificaciones principales del transmisor TS832.

Tabla 2. 6 Características técnicas del transmisor FPV Eachine TS832

Transmisor Eachine TS832 FPV	
Ganancia de la Antena	2 dB
Frecuencia	5,8 GHz
Potencia de transmisión	600 mW
Entrada de Alimentación	(7,4 - 16)voltios
Formato de video	NTSC/PAL
Corriente de trabajo	220 mA a 12 Voltios
Peso	22 gramos
Dimensiones	54x32x10 mm sin antena
Número de canales	48 Canales

## **2.7. Programas**

En el diseño del circuito impreso, la programación en Arduino del NodeMCU8266, el diseño del prototipo de la carcasa del vehículo y la creación de la aplicación en Android, es necesaria la utilización de programas específicos de cada uno de los diseños, para lo cual se describe a continuación los softwares escogidos.

### **2.7.1. Software SolidWorks**

Es un programa utilizado para diseñar cualquier tipo objeto sólido, con el objetivo de ser modelado de manera real y si es necesario imprimir o construir según lo amerite. SolidWork es de diseño CAD 3D (Diseño Asistido por Computador en 3D), tiene una interfaz sencilla e intuitiva que incluye cinco líneas de productos:

- Tiene herramientas de diseño en el que se puede modelar y ensamblar.
- Se puede generar planos en formato de 2D.
- Las herramientas de simulación pueden evaluar el producto.
- Permite evaluar el impacto en el sector medioambiental.
- Gestiona y desarrolla un método de diseño para cualquier tipo de sistema.

El programa SolidWork se utiliza con el fin de crear el diseño de la carcasa del vehículo para luego ser impresa en 3D, el material que se utiliza en la impresión es un filamento de plástico llamado ABS (Acrilonitrilo Butadieno Estireno), es el más utilizado en el mercado ya que admite el pulido con acetona y mejora su presentación.

### **2.7.2. App Inventor2**

Es un ambiente de desarrollo que crea aplicaciones para dispositivos móviles con sistema operativo Android, en su funcionamiento no se requiere de la instalación de ningún programa adicional en la PC, lo que se necesita es la navegación en Internet y un teléfono

celular tipo Smartphone, el programa es totalmente en línea y posee su propio servidor con lo que permite guardar todos los trabajos en línea.

Se basa en una de línea de desarrollo visual por bloques de construcción, tiene dos ventanas de trabajo, la primera es el diseño de ventana, en la cual se construye la pantalla de presentación para el usuario con todos los requisitos de ingreso de botones, imágenes, videos, conectividad, etc.

La segunda ventana de trabajo es la de bloques, en la que se ingresan las ordenes que se requieran para que la aplicación las ejecute, permite la configuración de la comunicación entre el dispositivo móvil y múltiples equipos remotos.

### **2.7.3. Arduino IDE (Entorno de desarrollo integrado)**

Es un entorno de desarrollo abierto que permite programar o configurar un dispositivo que sea compatible a este lenguaje, se base en el lenguaje C, tiene predefinido librerías que periten mejorar las líneas de comando en el proceso de construcción.

Es de código abierto, es decir que se puede crear cualquier tipo de programación sin seguir un camino específico, la placa NodeMCU8266 es compatible este tipo de lenguaje que puede ser diseñado para comunicar los diferentes módulos de todo el sistema.

### **2.7.4. Proteus**

Es un programa de la PC que permite el diseño en todas sus etapas para la construcción de los circuitos electrónicos, con una gran librería de elementos que permiten un óptimo desempeño en la fabricación de los mismos, entre sus principales ventajas está el diseño esquemático electrónico, la programación de ciertos controladores, la simulación del circuito con todos los elementos que lo conforman, con el fin de poder exportar un diseño de la construcción del circuito impreso. Las ventanas de trabajos soportan las siguientes funciones:

- Diseño del circuito electrónico llamado ISIS
- Diseño del circuito impreso llamado ARES
- Simulación del circuito y de la programación
- Visualización del circuito impreso con todos sus elementos
- Pruebas de funcionamiento

El programa Proteus permite el diseño del circuito regulador de voltaje step-down para el proyecto, en el cual se puede simular su funcionamiento.

## **2.8. Análisis de costos para el desarrollo del proyecto**

En el análisis de costos del proyecto es necesario tomar en cuenta los objetivos específicos planteados, ya que servirán de base para investigar los elementos que sean acordes a las necesidades del proyecto, se busca como primer paso las características y funciones requeridas, con el fin de examinar en el mercado las diferentes opciones que presenten las mismas especificaciones y el valor económico más adecuado.

### **2.8.1. Placa de control para el circuito**

Para el circuito se debe escoger una placa capaz de trabajar dentro de la programación del IDE de Arduino y que además cuente con una comunicación WIFI.

Dentro de las opciones del mercado electrónico se encuentra las placas de Arduino y las placas NodeMCU, por lo cual se realiza una comparación de los dos dispositivos tanto en sus especificaciones como en sus costos.

La primera opción es usar un Shield Ethernet conectado a una placa de Arduino, así el sistema puede ser administrado desde la red de Internet, pero para este proyecto lo que se busca es una comunicación inalámbrica mediante WIFI, por lo que se descarta éste módulo.

La segunda opción es utilizar la placa de Arduino YUN, dentro de sus principales características esta que tiene la misma cantidad de pines que Arduino Uno y para éste proyecto tiene incorporado un módulo WIFI, pero el costo del dispositivo es alto por lo que se buscara otra opción.

La opción más idónea dentro del mercado fue utilizar la placa NodeMCU 8266 ya que tiene una gran cantidad de pines, sus dimensiones son adecuadas para el sistema y sin duda la mejor ventaja es que viene incorporado el módulo WIFI ESP8266.

En la tabla 2.7 se presenta una comparación entre los precios que están en el mercado electrónico.

**Tabla 2. 7** Especificaciones técnicas del Arduino YUN

<b>COMPARACIÓN DE PRECIOS</b>			
	<b>NodeMCU8266</b>	<b>Arduino YÚN</b>	<b>Arduino ETHERNET</b>
<b>Valor</b>	10 dolares	43 dolares	16 dolares

Las dos placas cumplen con las condiciones requeridas para el proyecto, pero el NodeMCU8266 tiene dos ventajas importantes, la una es que las dimensiones físicas son más pequeñas que la YUN y además el costo es mucho menor, ya que la placa Arduino está en 43 dólares en promedio y el NodeMCU está en 10 dólares.

### **2.8.2. Cámara para el sistema FPV**

En el mercado existe una gran variedad de cámaras, las cuales cumplen funciones similares, dentro de ellas se destacan dos tipos de dispositivos de video, ya que las dimensiones físicas que presentan son adecuadas para el proyecto.

En la tabla 2.8 se presenta una comparación de los precios entre Mobius Action Cam C2 y la GoPro Hero 3.

**Tabla 2. 8** Comparación de las cámaras mobius y GoPro

<b>COMPARACIÓN DE PRECIOS</b>		
	<b>Mobius Action Cam</b>	<b>GoPro Hero 3</b>
<b>Valor</b>	72 dolares	200 dolares

Se escogió la cámara mobius action cam C2 ya que cumple todas las condiciones requeridas de resolución de video, dimensiones físicas pequeñas y el precio que es mucho menor a la GoPro.

### 2.8.3. Transmisor para el sistema FPV

Para determinar cuál es el transmisor que se requiere en el sistema, se seleccionó a los que trabajan en 5.8 GHz de frecuencia, además de que tenga una alimentación mayor a los 7 voltios, que sean de pequeñas dimensiones físicas.

Entre toda una gama de dispositivos se preselecciono a 2 de ellos, el primero es el TS832 y el segundo el TS353, en la tabla 3.5 se presenta una comparación entre los dos transmisores para seleccionar en base a todos los requisitos específicos y también por el costo que tiene en el mercado. En la tabla 2.9 se compara los precios de los transmisores TS832 y TS353.

**Tabla 2. 9** Comparación de los transmisores TS832 y TS353

<b>COMPARACIÓN DE PRECIOS</b>		
	<b>Transmisor TS832 FPV</b>	<b>Transmisor TS353 FPV</b>
<b>Valor</b>	10 dolares	30 dolares

Para el sistema FPV selecciono el transmisor TS832, cuenta con mejores especificaciones técnicas en general que el TS353 y además su costo inferior.

### 2.8.4. Gafas para el sistema FPV

En la selección de las gafas FPV se tomó en cuenta los aspectos técnicos de cada uno y el costo que representa comprarlos, se escogió 2 modelos para comparar el AOMWAY V1 y el EV800, en la tabla 2.10 se presenta las especificaciones que tiene cada uno.

**Tabla 2. 10** Comparación entre las gafas EV800 y AOMWAY V1

<b>COMPARACIÓN DE PRECIOS</b>		
	<b>Gafas FPV Eachine EV800</b>	<b>Gafas AOMWAY V1</b>
<b>Valor</b>	71 dolares	300 dolares

Se escogió las gafas EV800, éstas poseen características similares a la AOMWAY pero tiene dos ventajas, la primera es que tiene una batería incluida en el producto, lo cual genera un ahorro en el presupuesto ya que se puede comprar una batería adicional y la segunda es el precio del producto que está por debajo.

Se escogió las gafas EV800, éstas poseen características similares a la AOMWAY pero tiene dos ventajas, la primera es que tiene una batería incluida en el producto, lo cual genera un ahorro en el presupuesto y la segunda es el precio del producto.

### 2.8.5. Presupuesto del proyecto

Luego de haber seleccionado los elementos de todo el circuito, se procede a realizar el presupuesto del proyecto total, para el prototipo del vehículo a control remoto con visión en primera persona. En la tabla 2.11 se presenta el presupuesto total del proyecto.

**Tabla 2. 11** Presupuesto total del proyecto

<b>Presupuesto Total del Proyecto</b>				
<b>Nombre</b>	<b>Especificación</b>	<b>Cantidad</b>	<b>Valor Unitario(\$)</b>	<b>Valor Total(\$)</b>
Placa de control	NodeMCU8266	1	10	10
Puente H	L298N	1	4,5	4,5
Base para el vehículo	Chasis de acrílico	1	22	22
Motores	Moto reductores	2	4,25	8,5
Cables	Macho - Hembra	15	0,1	1,5
Cables	Hembra - Hembra	15	0,1	1,5
Circuito Impreso	Regulador de voltaje	1	4	4
Case para placa de control	Case para NodeMCU	1	4	4
Batería	Lipo 2S a 1500mAh	1	29	29
Servomotores	SG90	2	4	8
Estructura para servos	Estructura SG90	1	5	5
Gafas para FPV	EV800	1	71	71
Trasmisor para FPV	TS832	1	10	10
Cámara para FPV	Action Cam C2	1	72	72
Carcasa del vehículo	Impresión en 3D	1	1	70
<b>Total del presupuesto</b>				<b>321</b>



## 2.9. Análisis de tiempo

En el análisis de tiempo se realiza una descripción del cronograma realizado durante las fases que comprenden el desarrollo completo del proyecto, en la primera etapa se define los elementos electrónicos y mecánicos del vehículo a control remoto con FPV y software, para ello se buscó en el mercado cuales son los dispositivos electrónicos que cumplen con las especificaciones técnicas y luego escoger en función del costo beneficio cual es el idóneo para el proyecto. Se tomó catorce días, del sábado 28 de abril al viernes 11 de mayo del 2018.

La siguiente etapa está dedicada en el diseñar el hardware para el vehículo a control remoto, se busca los diferentes tipos de conexiones que pueden darse, su tiempo de duración es de 21 días, desde el sábado 12 de mayo al viernes 1 de junio del 2018. En la etapa de implementación de un sistema de comunicación WIFI entre vehículo y sistema de control remoto, es la parte más importante del proyecto ya que se programa el controlador del carro mediante el programa Arduino, se realiza desde el sábado 23 de junio al viernes 27 de Julio del presente año, con una duración de 35 días.

Para la construcción del sistema de visión en primera persona FPV, se implementa la conexión entre el transmisor, las cámaras y las gafas, con el fin de transmitir las imágenes que se encuentra en el entorno del vehículo, la duración es desde el 28 de Julio al 20 de agosto del 2018, con una duración de 24 días.

Las etapas finales corresponden a las pruebas de validación y funcionamiento con la elaboración del trabajo escrito que concluyo hasta el 25 de agosto del 2018. El cronograma completo se presenta en la figura 2.11.

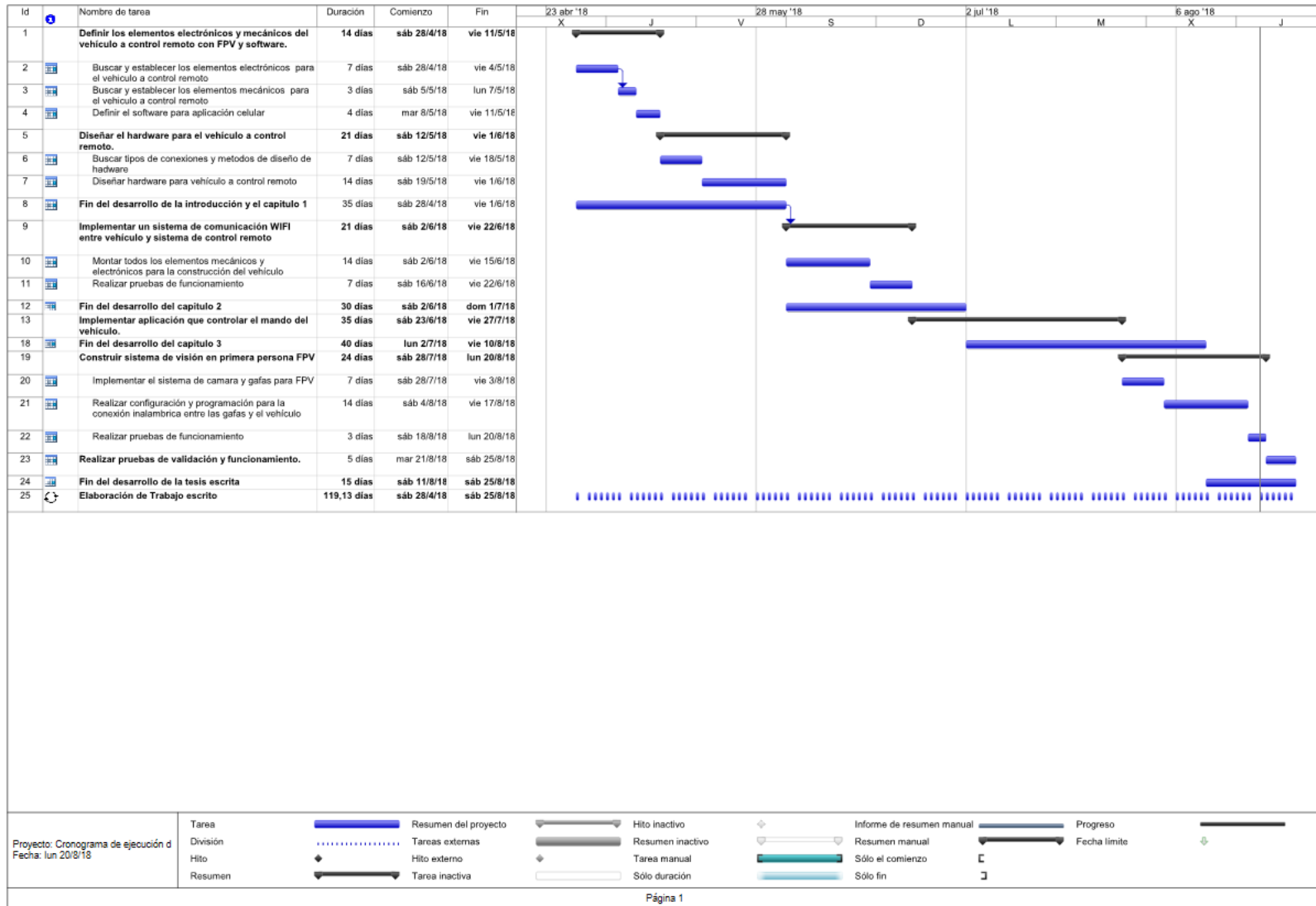


Figura 2. 11 Cronograma de actividades

## CAPITULO III

### IMPLEMENTACIÓN

En el último capítulo del trabajo de grado se presenta el desarrollo e implementación del proyecto de una manera detallada, por lo que se ha dividido en el desarrollo e implementación de la aplicación del control remoto, la fabricación del circuito impreso correspondiente al regulador de voltaje, armado del prototipo de vehículo, armado de la base para los servomotores que permitirán la movilidad de la cámara, montaje de todos los elementos en la presentación final.

#### 3.1. Diseño de fuente para los servomotores

Para el correcto funcionamiento de los servomotores se va a requerir de un regulador de voltaje, ya que la alimentación de los servos es de 5 voltios y la fuente de alimentación que se va a utilizar es de 7.4 voltios, por lo que se necesita realizar una simulación del circuito eléctrico con el fin de verificar si funcionaria antes de ser implementado.

Para la simulación se utilizó el programa PROTEUS, en el cual se puede ingresar todos los elementos que formaran parte del circuito impreso, los elementos utilizados en el regulador de voltaje son:

- Integrado LM7805
- Diodo 1N4007
- Capacitor 10uF
- Resistencia 330  $\Omega$
- LED

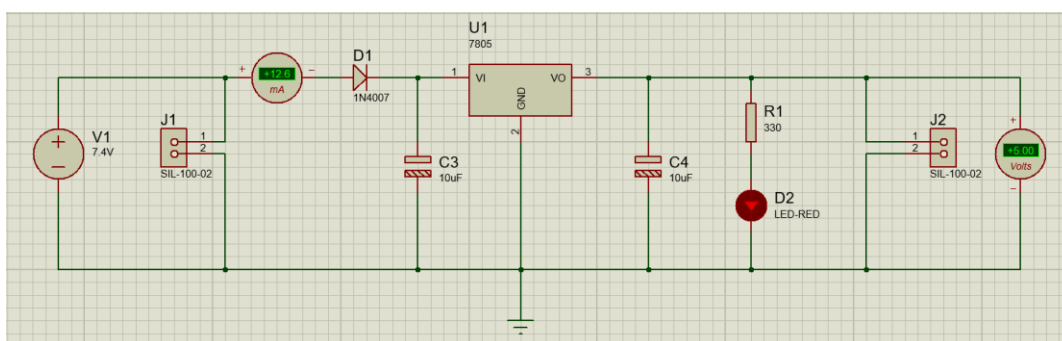
Los cálculos de la resistencia que se utiliza como protección del LED en el circuito de color rojo, tienen un consumo característico de 0.02 amperio, a continuación se presenta los valores:

$$R = \frac{V}{I} \text{ Ec. 2}$$

$$R = \frac{5 \text{ voltios}}{0.02 \text{ amperios}} = 250 \text{ ohmios}$$

En la hoja técnica del fabricante indica que la resistencia recomendada es de 330 ohmios.

El capacitor que está ubicado después del integrado es utilizado para mejorar la respuesta transitoria del voltaje de salida. En la figura 3.1 se presenta la Simulación en Proteus ISIS de todo el circuito armado del regulador de voltaje.



**Figura 3. 1** Diagrama circuital LM7805

Luego de verificar en ISIS que el circuito está en un correcto funcionamiento se procede a ponerlo en la versión de ARES de circuito impreso del programa Proteus con todas las pistas de cobre, esto permitirá imprimir y montar sobre la baquelita con todos los elementos del circuito, esto se explicará a detalle en el capítulo cuatro. En la figura 3.2 se presenta la simulación en ARES.

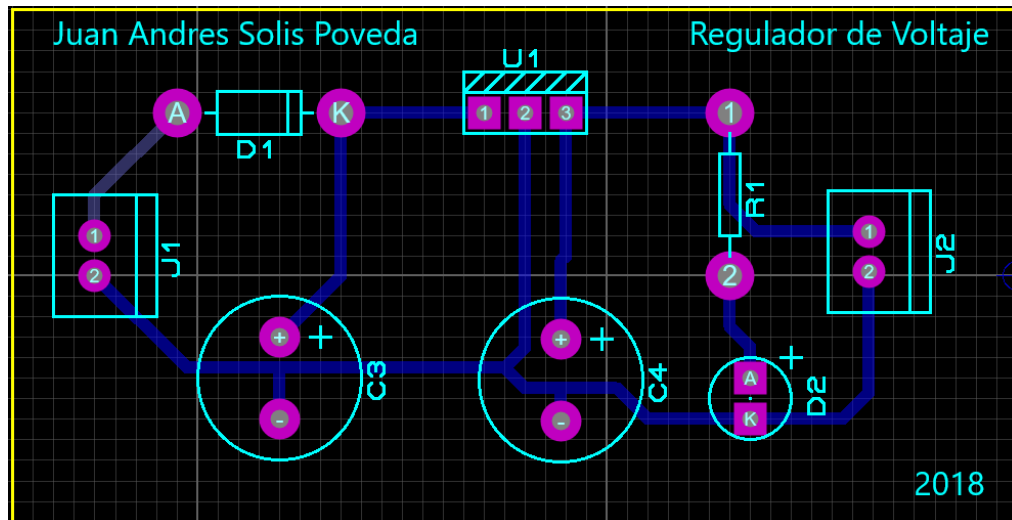


Figura 3. 2 Placa del circuito LM7805

Finalmente, en el mismo programa se puede tener una visualización más real de cómo va a quedar el circuito con todos los elementos ya soldados. En la figura 3.3 se presenta la simulación del circuito.



Figura 3. 3 Placa del circuito LM7805

### 3.2. Dimensionamiento de la fuente de alimentación del circuito

Antes de seleccionar la fuente de alimentación se requirió un análisis de los consumos de corriente de cada uno de los elementos que conforman el circuito con ayuda de un multímetro, además de cuál es el voltaje necesario de los elementos para su correcto funcionamiento.

**Tabla 3. 1** Medición del consumo de corriente máxima del circuito

Nombre	Voltaje(v)	Corriente(mA)	Potencia(W)
NodeMCU8266	5	110	0,55
Regulador de voltaje	7,4	12,6	0,09324
Motor Driver L298N	7,4	165	1,221
2 Servomotor SG90 Horizontal	5	1200	4
Transmisor TS832	7,4	175	1,295
<b>TOTAL</b>		<b>1662,6</b>	<b>7,15924</b>

En la tabla 4.1 se presenta los valores de voltaje, corriente y potencia que servirán como referencia para escoger la fuente que alimentará a todo el proyecto.

Luego de establecer el consumo de corriente total del circuito, se busca en el mercado una batería que pueda solventar la cantidad de amperaje que se requiere y que pueda ajustarse al presupuesto económico.

Para este proyecto se tomó la decisión de utilizar una batería LIPO de 7.4 voltios a 1.5 Amperios de 35C, aunque a simple vista pareciera que no se ajusta al consumo de corriente expuesto en la tabla 3.1 no es así, ya que los 35C son la capacidad de descarga que tiene la batería es decir se le debe multiplicar la corriente de 1.5 Amperios por el valor de C. El fabricante recomienda que se debe hacer el cálculo con un 70 al 80 por ciento del valor de C con el fin de garantizar el correcto funcionamiento, por lo que se va a calcular con 25C.

La batería tiene 1.5 AH que es la cantidad de corriente que puede suministrar en una hora, ese valor debe multiplicarse por 25C que es la máxima corriente que puede suministrar la fuente, pero a un tiempo mucho más corto. Para calcular cual será el tiempo máximo de alimentación del circuito se realiza de la siguiente manera:

$$\text{Corriente máxima} = 1.5A \times 25 = 37.5 A$$

$$C = \frac{1.67A \times 25C}{37.5A} = 1.114$$

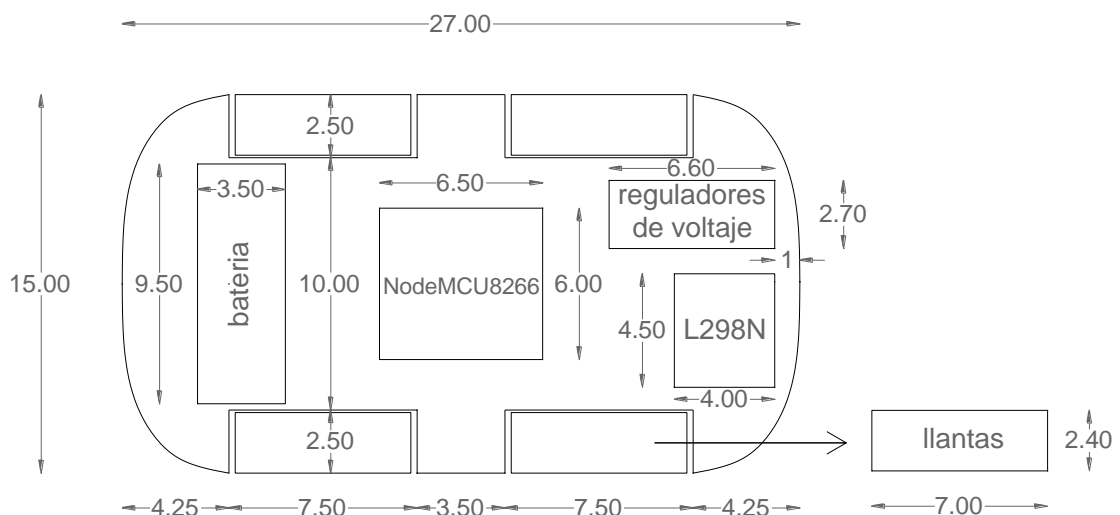
$$\text{Tiempo de funcionamiento} = \frac{60 \text{ minutos}}{1.114} = 53.85 \text{ minutos}$$

El valor obtenido es de alrededor de 53 minutos es el tiempo máximo de funcionamiento que tiene todo el circuito con la batería que se va a utilizar.

### 3.3. Diseño del montaje de los elementos del vehículo

El diseño del montaje del vehículo se refiere a la forma en la que están ubicados los elementos de todo el circuito, para el desarrollo del prototipo se emplea dos plantas divididas por bases de acrílico, las cuales están unidas por postes metálicos con ajustes de tuercas y tornillos.

En la primera planta se presenta la batería LiPo, el controlador NodeMCU866, el circuito reductor de voltaje, y el módulo L298n al cual está conectado a los motores reductores con sus respectivas llantas. En la figura 3.4 se presenta a continuación la distribución de los elementos.

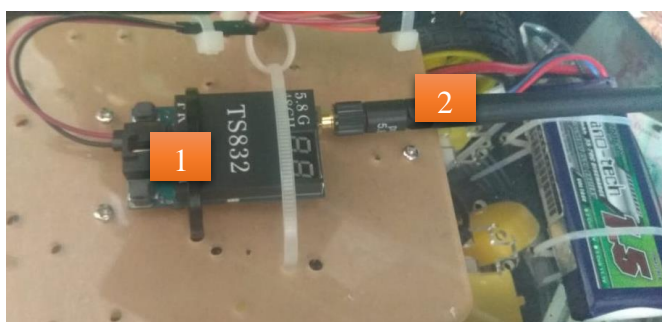


**Figura 3. 4** Diseño del montaje de los elementos del vehículo

Los elementos están dispuestos sobre la base de acrílico de la siguiente manera:

- Batería LiPo
- Moto reductores
- Llantas
- NodeMCU8266
- Regulador de voltaje
- Módulo L298N

El cableado requerido para el funcionamiento del proyecto se encuentra detallado en los anexos.

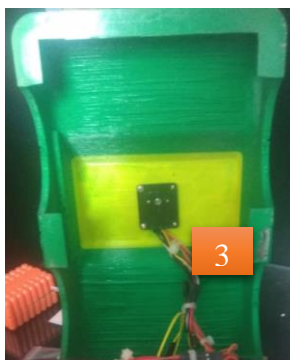


**Figura 3. 5** Diseño del montaje segunda planta

En la figura 4.9 está la segunda base de acrílico en ella se encuentra ubicados los dos siguientes elementos:

- 1. Transmisor de video
- 2. Antena del transmisor de video

Finalmente, en la figura 4.9 se presenta la carcasa del vehículo, en la cual está sujeto la base del sistema de servomotores.

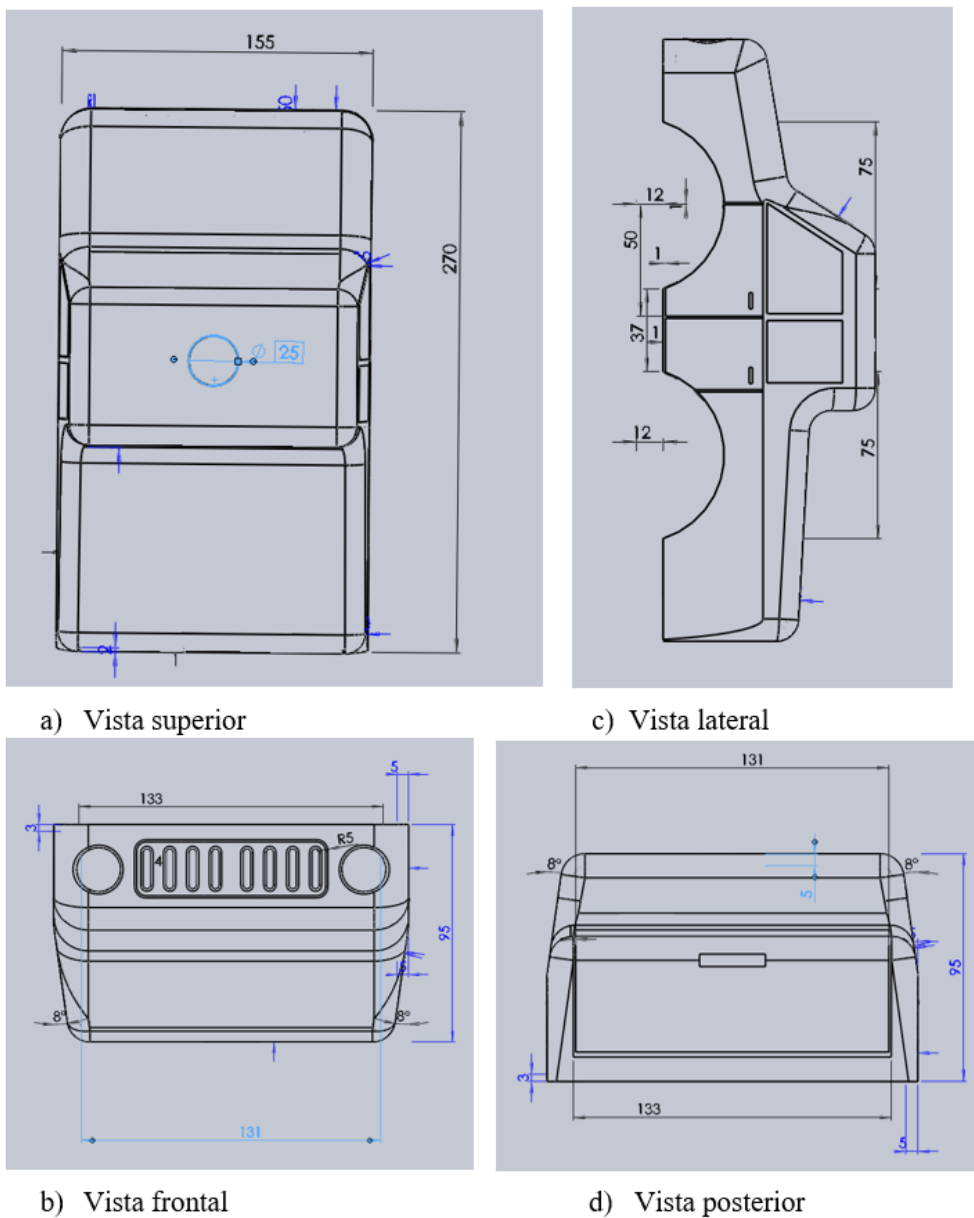


**Figura 3. 6** Diseño del montaje de los servomotores



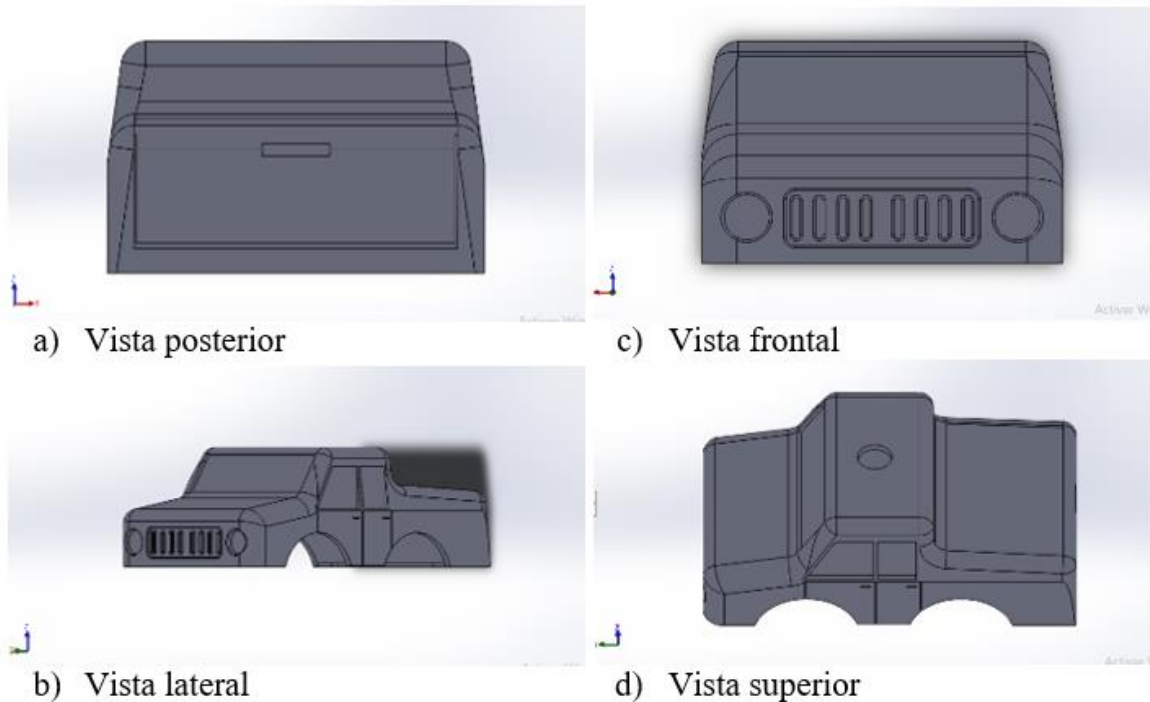
### 3.4. Diseño de la cubierta del vehículo

En el diseño de la carcasa del vehículo a control remoto se utilizó el programa SolidWorks el cual permite diseñar en CAD 3D (Diseño Asistido por Computadora en 3D), para realizar el modelo del proyecto se tomó las medidas exactas de la base, en la altura se tiene que tomar en cuenta los dos niveles que va a tener el vehículo y la altura a la que va a ir montado los servomotores. Como primer paso es necesario diseñar la estructura en 2D de la carcasa del vehículo para tener una idea general de su modelo. En la Figura 3.7 se presenta el diseño de varias de sus vistas.



**Figura 3. 7** Diseño de cubierta del vehículo a) b), c), d)

En el segundo paso se procede a dar profundidad con el ancho requerido del vehículo, para ello se presenta a continuación una serie de imágenes en 3D de varias vistas de la carcasa ya concluida en la figura 3.8.



**Figura 3. 8** Diseño final de la cubierta del vehículo en 3D a), b), c), d)

### 3.5.Diseño de la aplicación para el sistema

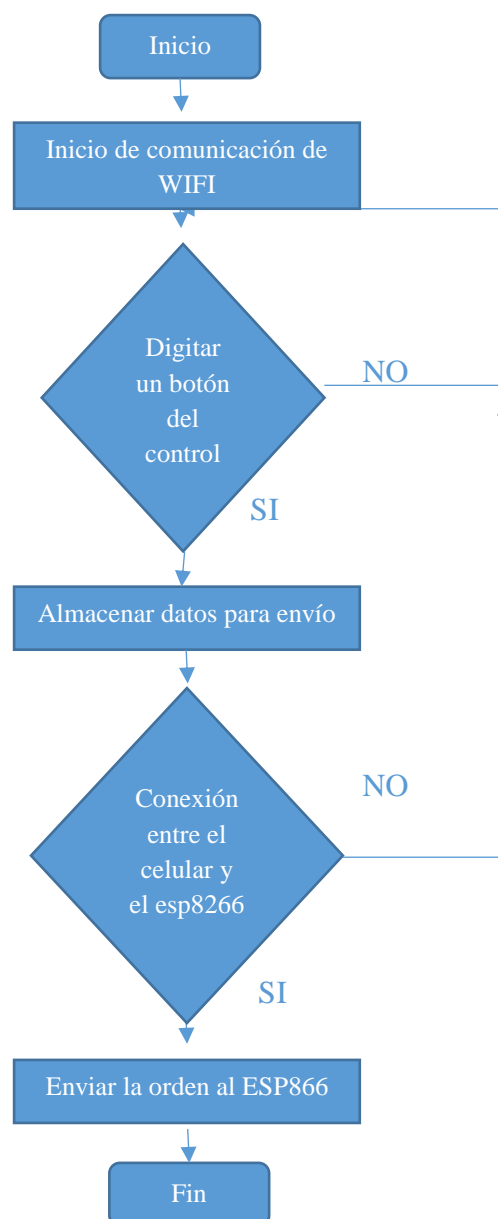
El desarrollo de la aplicación está enfocado en ser el control remoto de todo el sistema, para ello se requiere de un diagrama de flujos en el cual se especifique los pasos de comunicación de la App.

La aplicación se inicializa con el encendido de la misma y la activación de la comunicación WIFI entre la App y el controlador, estos se comunican mediante una conexión punto a punto, para lo cual el celular es el servidor y el controlador su único usuario.

En el control de mando que se despliega en la aplicación el usuario tienen diferentes botones que le permiten dar una dirección al movimiento de giro de las llantas y además de

dar la rotación a los servomotores para que pueda tener un mayor ángulo de visión a la cámara instalada sobre él.

Luego de que digita el usuario una o varias de ellas, estos se almacenan momentáneamente hasta que se verifique si existe la comunicación entre el teléfono celular y el NodeMCU8266, si está conectado se envía la información al controlador, pero si no existe esa conexión se envía nuevamente a la espera de que el usuario digite algún botón nuevamente. En la figura 3.9 se presenta el diseño de la aplicación para el sistema.

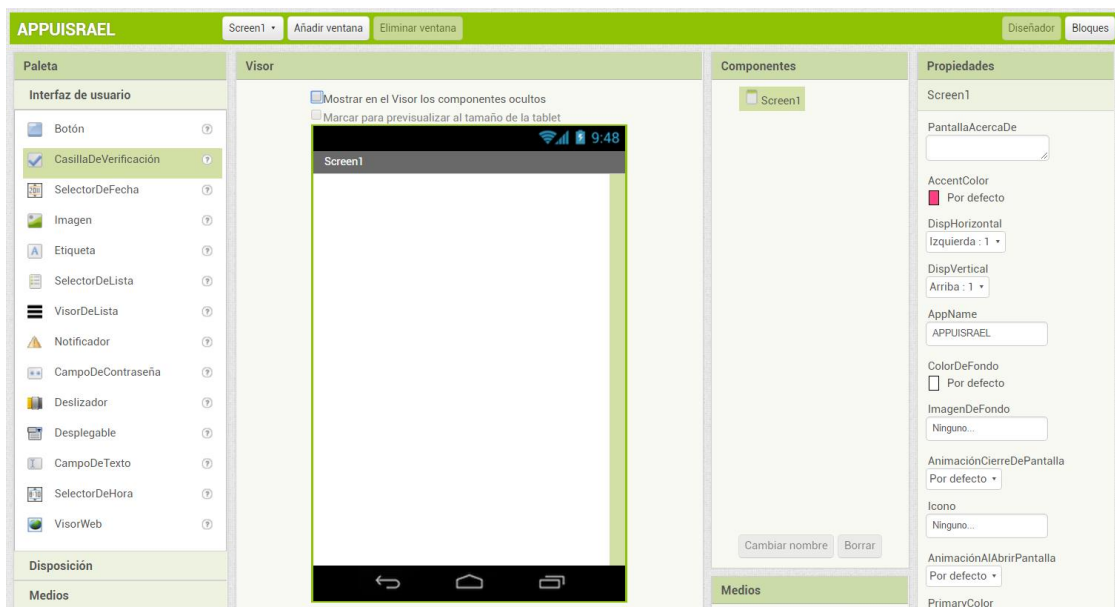


**Figura 3. 9** Diseño de la aplicación para el sistema

### 3.6.Desarrollo de la aplicación para el control del vehículo

Para el desarrollo de la aplicación se utilizará la interfaz de App Inventor 2, se encuentra en línea y el único requisito es tener una cuenta en Gmail, la interfaz tiene dos ventanas de trabajo, la primera es llamada Diseñador el cual permite crear la pantalla la cual tendrá la de usuario con cada uno de los botones de acceso y demás recursos que el usuario podrá manipular. La segunda ventana se llama de Bloques, Es la que realizar la programación necesaria en la que pueda comunicarse la aplicación con el módulo WIFI.

La ventana de diseño está dividida en:



**Figura 3. 10** Ventada de diseño de App Inventor2

- **Paleta:** permite agregar una serie de elementos como botones, imágenes, texto, notificadores, deslizadores y demás complementos para que el usuario pueda manipular la aplicación.
- **Visor:** en esta parte de la ventana es donde se añaden los recursos seleccionados de la Paleta y la posibilidad de ver de una manera real de cuál será la presentación final de la aplicación.
- **Componentes:** permite visualizar cuales son todos los elementos añadidos al visor de la paleta y cuál es el orden jerárquico al que pertenecen.

- **Propiedades:** son todos los recursos disponibles para editar los elementos que van a pertenecer a la aplicación, como: ubicación, desplazamiento, color, títulos, etc.

En la ventana correspondiente a diseño, los elementos que se utilizaron de la paleta son los botones, imágenes y texto que pertenecen a la interfaz de usuario, disposición vertical, horizontal y de tabulación que pertenece a la disposición, la comunicación web que se encuentra en conectividad. A continuación, se presenta la figura 3.11 que corresponde a la Paleta.

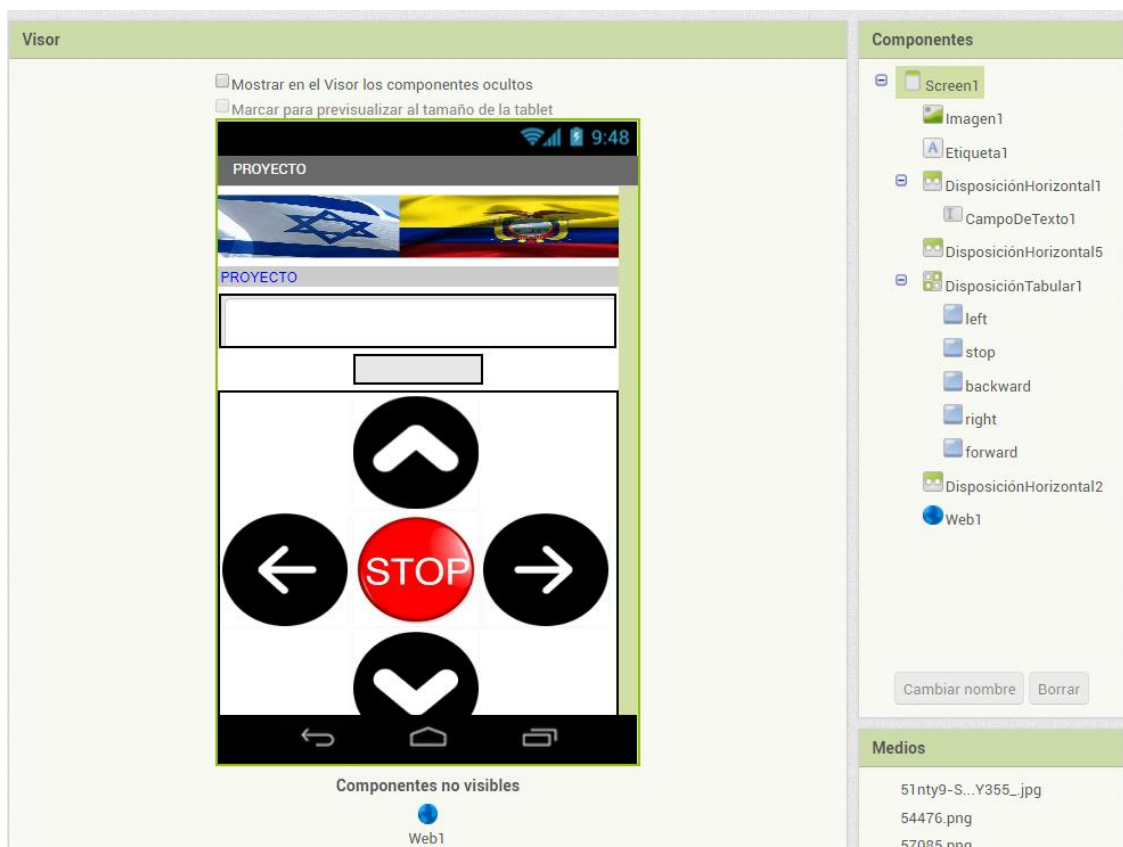


**Figura 3. 11** Opciones de paleta en App Inventor2

Como primer paso se realizó una aplicación que permita controlar las llantas del vehículo, para familiarizarse con la interfaz, en la figura 3.12 se presenta el diseño.

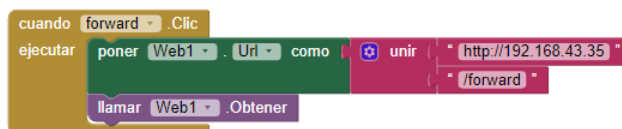
En la parte de componentes se encuentra todos los elementos que se utilizaron de la Paleta y cuál es el orden en el que van ubicados, los iconos que llevan el nombre de left, right, stop, forward, backward son los botones de comando que permitirán que se mueva las llantas en todas las direcciones. Adicionalmente se añade la conectividad web para la comunicación con el módulo WIFI instalado en el vehículo.

La ventana de bloques tiene dos sub interfaces, la primera llamada de bloques contiene todas las posibles opciones de programaciones que están enlazados con la barra de componentes de la ventana de diseño, la segunda es el divisor el cual agrega los bloques y ordenar los comandos a utilizar según el requerimiento del proyecto.



**Figura 3. 12** Control remoto para el movimiento de giro de las llantas

Para la ventana de bloques se realizará una explicación de la cada uno de los elementos utilizados, a continuación, en los anexos se encuentra toda la programación de la aplicación del teléfono móvil celular, a continuación, se hace referencia a los bloques utilizados:



**Figura 3. 13** Programación en bloques para el vehículo

- Los bloques de color naranja se encuentran dentro de cada uno de los botones que se añadieron, es decir los botones left, stop, backward, right y forward. Permiten que cuando sea presionado un botón en específico por el usuario se ejecute una función que se especifica al añadir más bloques sobre él. Es el primer elemento que se debe añadir.

- En el siguiente paso el bloque de color verde se encuentra dentro del dispositivo web1, al hacer clic sobre él se desplegará un submenú y se selecciona e indica en la gráfica anterior. Éste bloque permitirá que se comunique la señal WIFI del módulo instalado en el vehículo con la aplicación mediante la tarjeta WIFI del celular.
- El bloque de color morado permitirá que se llame al bloque web1 para obtener su petición.
- La opción de color rosado unirá la IP del módulo WIFI del vehículo y el botón con el nombre especificado en el bloque en un solo sistema de comunicación, por lo que la conexión será un éxito entre los dos dispositivos.

Para saber cuál es la IP del módulo WIFI ESP8266 se requiere de un programa en Android que permita extraer mediante comunicación serial. En la figura 4.5 se presenta el programa realizado en Arduino IDE.

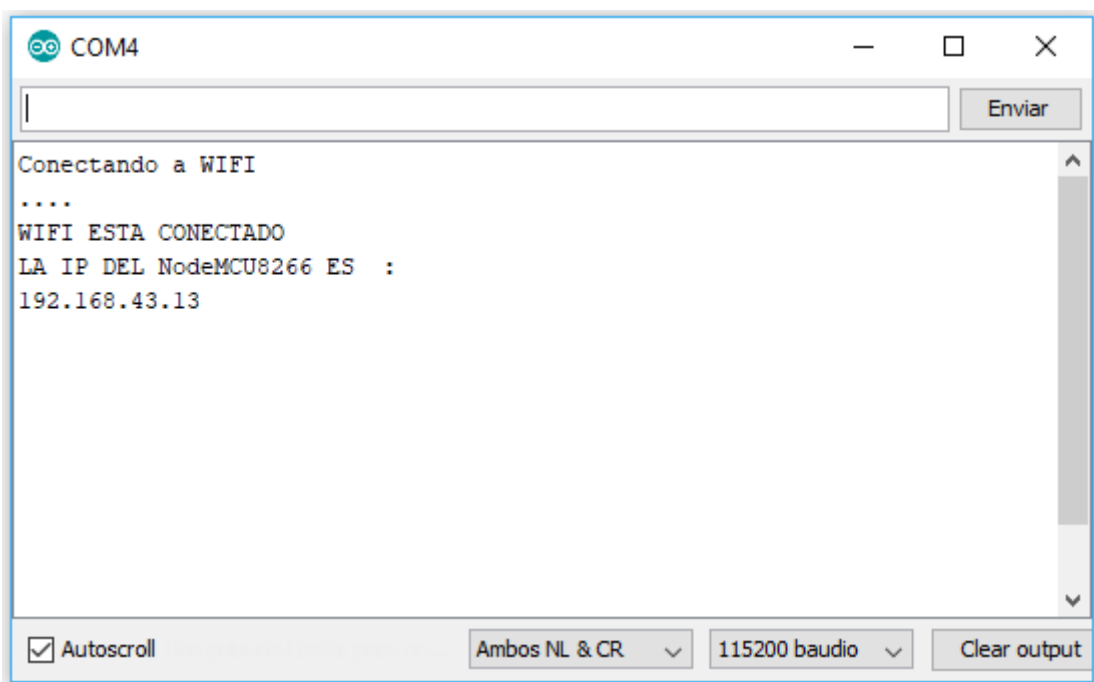
Para comprender las líneas de comando a continuación se especifica que se añadieron en la programación en la que se verifica cual es la dirección IP:

- **Serial.begin ():** Permite establecer la velocidad de los datos en bits por segundo (baudios) para la transmisión de los datos en serie hacia el computador.
- **Serial.println ():** Permite imprimir datos con código ASCII que el usuario pueda entender.
- **WiFi.begin ():** Inicializa la configuración de red de la biblioteca de WIFI e indica cual es el estado actual, en el paréntesis puede ir el usuario y contraseña para conectarse a la red indicada.
- **While ():** Es un tipo de bucle que se ejecuta continuamente mientras se cumpla lo que se indica entre paréntesis en su cabecera, caso contrario se saldrá de su ejecución.
- **WiFi.status ():** Indica cual es el estado de conexión de la red WIFI.
- **WL\_CONNECTED:** Dentro de WiFi.status () permite estar asignado cuando está conectado a una red WIFI.
- **Serial.print ():** Envía una cadena de caracteres o un número a un puerto de serie definido.

- **WiFi.localIP ():** Obtiene la dirección IP de la red WIFI.

En la primera sección se incluye todas las librerías del ESP8266, se crea el servidor y cliente para la programación, se ingresan la contraseña y usuario del celular como punto de acceso y se declara las funciones en el void setup ().

En la segunda sección se procede a solicitar la IP del módulo mediante la comunicación WIFI y se imprime en el puerto serial indicado. Finalmente, en la figura 3.14 se presenta el puerto serial con la dirección IP solicitada que pertenece al módulo ESP8266, en los anexos se presenta toda la programación.



**Figura 3. 14** Pantalla de visualización de la dirección IP

### 3.7.Desarrollo de la aplicación para el control de los servomotores

Luego de realizar la aplicación capaz de controlar el movimiento del vehículo se requiere realizar la implementación de la aplicación del control de los movimientos de los servomotores SG90, mediante un diseño de pantalla más formal y óptima para la presentación del proyecto total.



Para ello se cambian las imágenes de los botones, posición y título de la aplicación, en la ventana de diseño no se agrega ningún icono adicional a los ya explicados, por lo que se presenta a continuación en la figura 3.15 la ventana del visor, que pertenece a la ventana principal de diseño en App Inventor 2.

No se puede visualizar todos los iconos en la pantalla ya que se encuentran minimizados los que se encuentran ubicados en el centro.



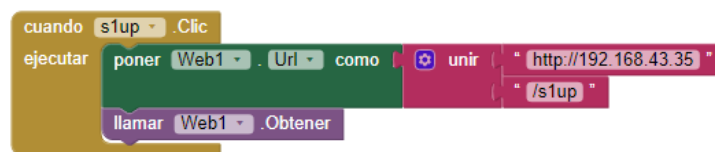
**Figura 3. 15** Diseño completo de la aplicación

En la ventana de bloques, se crea la estructura de las funciones, que permitan la comunicación entre la aplicación instalada en el celular y los servomotores, para el control del movimiento de giro en cada uno de ellos.

A continuación, se especifican los bloques ingresados para el control de los servomotores:

- Los bloques de color naranja permiten ejecutar una función definida cuando son solicitados por el usuario en la aplicación, es decir se digita un botón en específico. Cada uno de ellos se encontrará dentro del submenú de botones en el lado izquierdo de la ventana de bloques.
- Los botones de color verde permiten la comunicación entre los servomotores controlados por el módulo ESP8266 y la aplicación instalada en el celular.
- Los bloques de color morado llaman a la función que controla el bloque Web1 para que se inicialice.
- Los bloques de color rosado permiten la comunicación mediante el reconocimiento de la dirección IP del módulo ESP8266 y la aplicación con el respectivo nombre del botón.

En la figura 3.16 se presenta el diagrama de bloques correspondiente a la aplicación que controla los servomotores que permitirán el giro de la cámara en dos direcciones.



**Figura 3. 16** Programación en bloques para los servomotores

Finalmente se presenta la aplicación con la ventana que será visible para el usuario, que permita controlar tanto al movimiento de los servomotores como del vehículo.

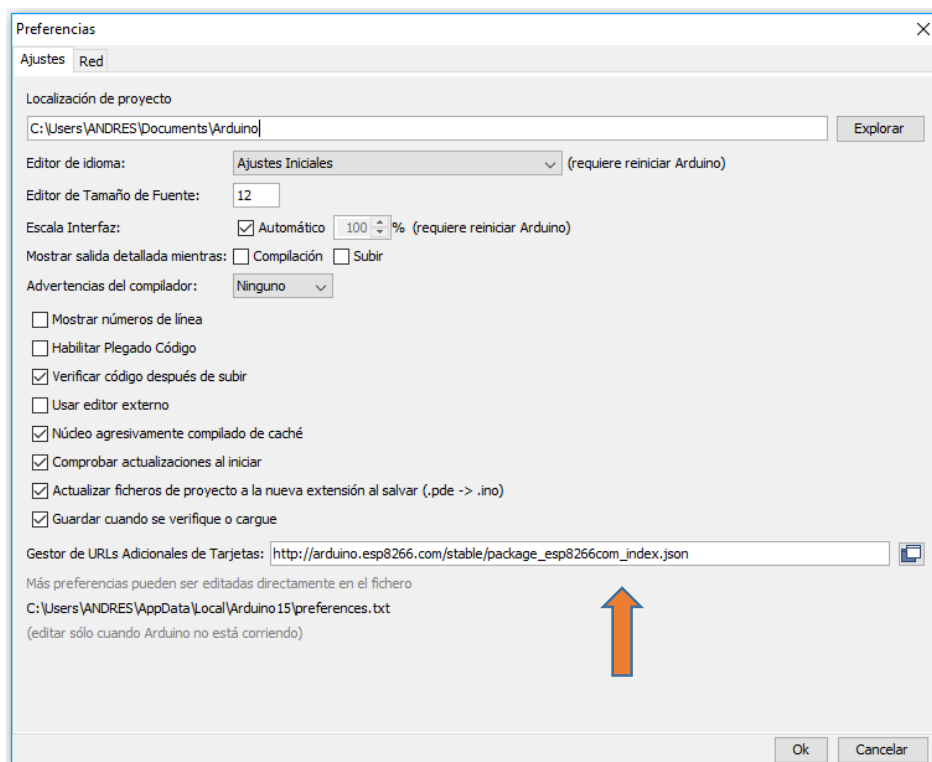


**Figura 3. 17** Programa final realiza en APP INVENTOR

### 3.8. Programación del NodeMCu8266 para vehículo a control remoto

En el control del módulo NodeMCU8266, se utilizará la programación de Arduino, para ello se requiere de una previa explicación de los comandos utilizados, como de la librería específica del módulo.

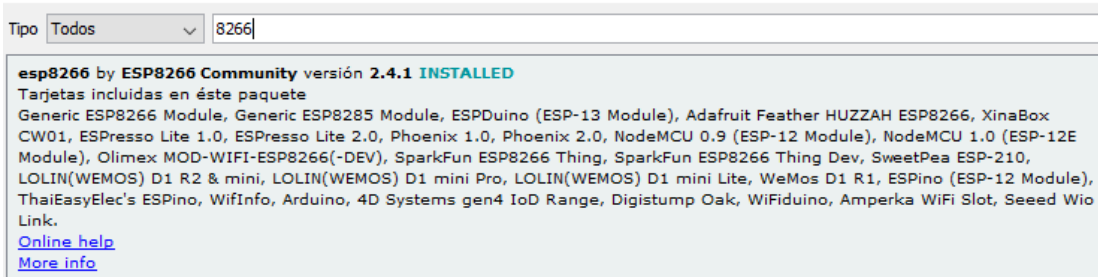
El primer comando que se requiere es la librería del módulo WIFI que contiene el NodeMCU8266 el cual está en el repositorio de GitHub, el cual es un link que se debe agregar en el programa de Arduino como se indica en la figura 3.18 a continuación.



**Figura 3. 18** Ingreso de enlace para agregar la librería ESP8266 en Arduino

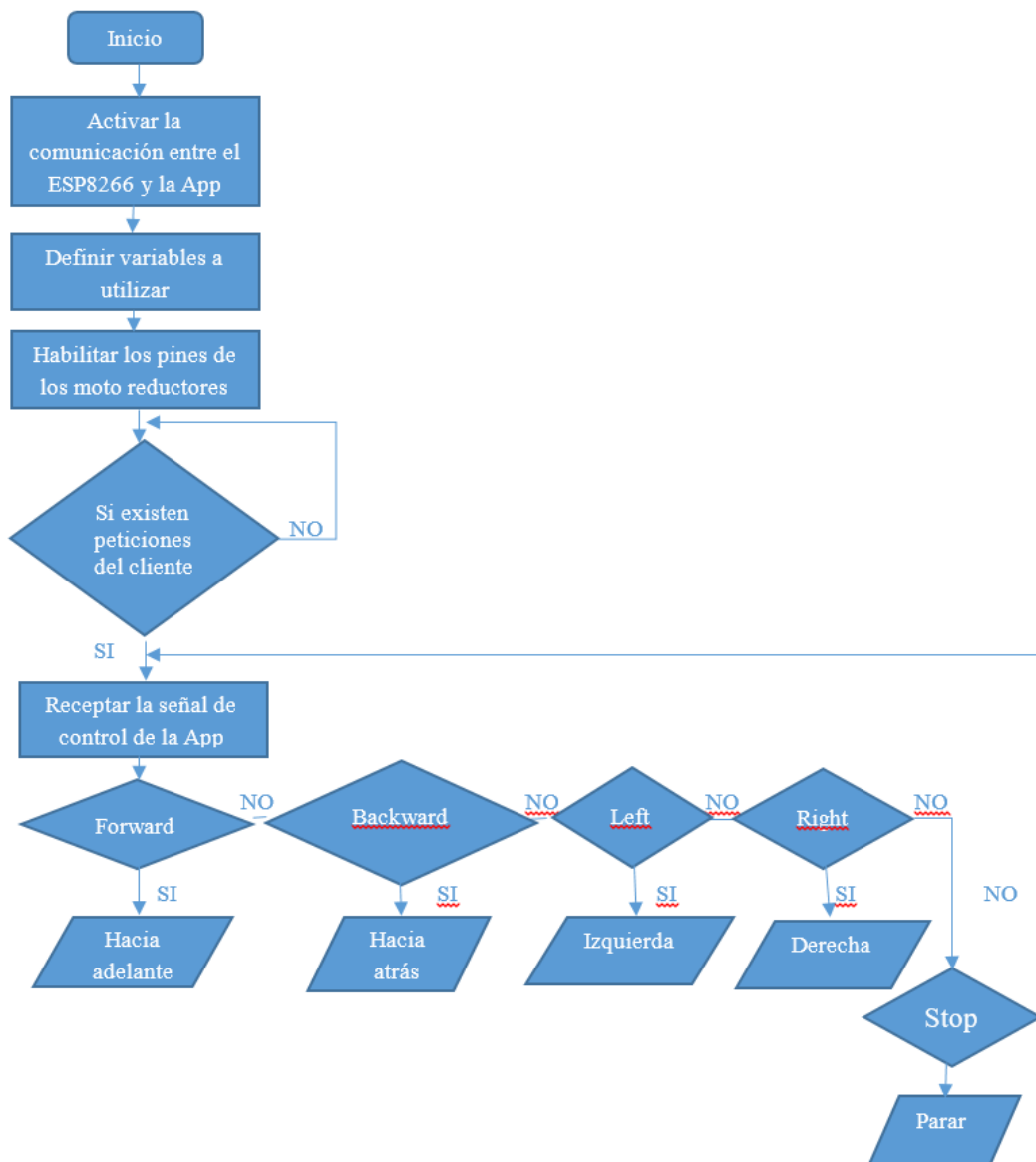
Esto permitirá la compatibilidad de la librería llamada ESP8266WIFI.h con la programación de Arduino.

En el gestor de tarjetas ya se puede seleccionar el ESP8266 con todas sus librerías para su programación y configuración; en la figura 3.19 se presenta el instalador de la tarjeta.



**Figura 3. 19** Descarga de la librería ESP8266

Para explicar cuál es el método utilizado en la programación sobre el NodeMCU8266 se expone a continuación en la figura 3.20 el diagrama de flujo.



**Figura 3. 20** Diagrama de flujo para los motores reductores

El programa inicia con los comandos de comunicación y las librerías de la conexión inalámbrica entre el ESP8266 y la aplicación, luego se definen las variables de que requieren las líneas de programación. En el siguiente punto se habilitan los pines que serán utilizados para el control de los motores reductores, si existe una petición del cliente se receptará la señal que fue enviada desde la aplicación, cada una de estas señales son identificadas según su orden de llegada, si es hacia adelante, atrás, izquierda, derecha o detenerse, estos serán ejecutados por el controlador y los motores van a girar en el sentido en el que se los ordena, caso contrario espera a que se envíe una nueva señal de control hacia el módulo ESP8266.

### 3.8.1. Programación para el movimiento del vehículo

Para la programación en el software IDE (Entorno de Desarrollo Integrado) de Arduino se va a explicar los comandos que se utilizaron en el movimiento del vehículo a control remoto.

- **WifiClient:** Crea un cliente que puede conectarse a una dirección IP y un puerto de Internet especificado.
- **WifiServer ():** Crea un servidor que escucha las conexiones entrantes en un puerto especificado.
- **Const Char:** Comando que se utiliza para almacenar un valor de manera constante en un Byte de memoria.
- **String:** Es un tipo de datos que se usa para guardar cadenas de caracteres y enviarlos mediante un puerto.
- **Int:** Tipo de variable entero que almacena caracteres en 2 Bytes.
- **Void setup ():** Es una de las partes más importantes de toda la programación, tiene toda la configuración que contiene las declaraciones de las variables, es la primera función que se ejecuta en el programa y lo hace en una sola vez, en ella se encuentra la configuración o inicialización de los pinMode, comunicaciones en serie, etc.

- **Void loop ():** Esta función se ejecuta luego de setup (), es la que contiene las instrucciones que se requiere hacer y lo ejecuta de una manera cíclica, es decir en pasos consecutivos.
- **pinMode (pin, mode):** Este comando se utiliza en la parte de configuración del setup () y sirve para configurar el modo de trabajo de un pin que puede ser INPUT (entrada) o como OUTPUT (salida).
- **Server.begin ():** Indica al servidor que comience a escuchar las conexiones entrantes.
- **Server.available ():** Después de que ya se inicializo el servidor, éste comprueba periódicamente si existe peticiones de un cliente, si es afirmativo la propiedad server.available devolverá una entrada a la petición de un cliente, en caso contrario será cero, para ello se pone el comando return.
- **digitalWrite (pin, value):** Envía al pin predefinido como OUTPUT (salida) el valor de High (alto) o LOW (bajo), es decir en 1 o 0 lógico, el pin se puede definir como una constante o variable.
- **readStringUntil:** Lee los caracteres del buffer en una cadena, para éste caso lee los caracteres que se guardaron en String.
- **request.remove ():** Permite eliminar una parte específica de un String, puede ser utilizado con uno o dos argumentos.
- **length ():** Devuelve la longitud de la cadena en caracteres.

Una vez conocida la función de cada uno de los comandos a utilizar en la programación para el movimiento del carro a control remoto, se procede a indicar las líneas de programación en cada una de sus secciones.

En la primera sección del programa se especifica la librería del ESP8266 para que se habilite todas sus líneas de comando, además de crear un servidor el cual será encargado de recibir las peticiones del cliente. Para la conexión entre el módulo NodeMCU8266 y la aplicación instalada en el celular se requiere de especificar el usuario y contraseña de la red WIFI de interconexión de los dispositivos, en este caso se decidió como usuario el nombre Redmi y la contraseña uisraelxxxx, se define la variable data en la que guarde todos los caracteres que recibe de la aplicación, se procede a crear las variables de tipo entero de los

pinos de conexión entre el NodeMCU8266 y el Motor Driver, se inicializa los pinos de los motores del vehículo y finalmente se inicia la comunicación del servidor.

En la segunda sección del programa se ingresan las líneas de comando para que el servidor reciba la información enviada del cliente desde la App, en los cuales cada figura de la aplicación instalada en el celular corresponde a un una orden que indica al NodeMCU8266 si debe ir hacia adelante, hacia atrás, a la izquierda o derecha, lo cual se utiliza el condicional If en la programación.

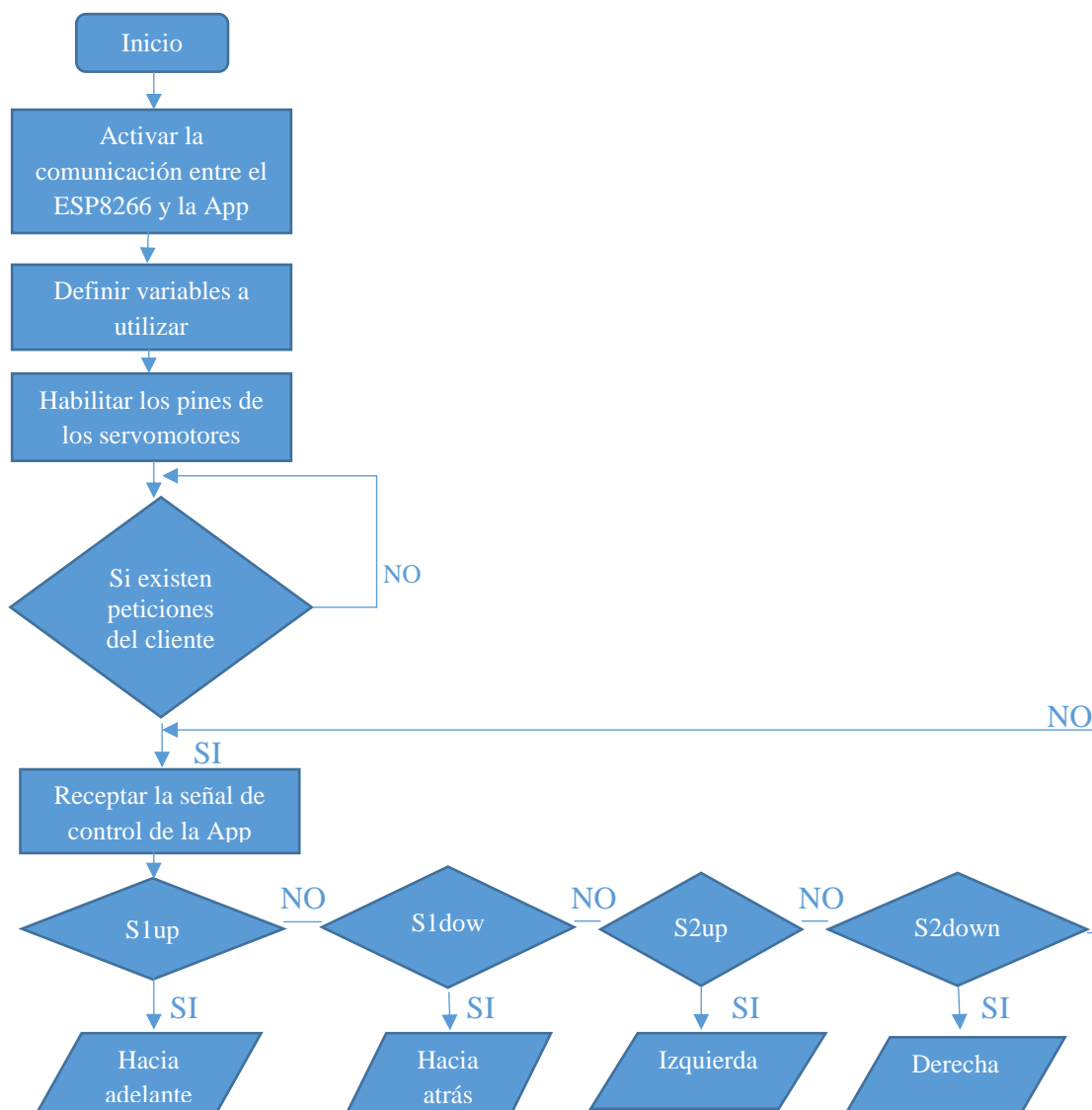
En la última sección del programa se realiza las líneas de comando que permita el vehículo a escala a control remoto pueda ir la dirección que indique la aplicación del celular con la instrucción digitalWrite en cada uno de los motores, además las líneas de programa para que el NodeMCU8266 sea capaz de recibir los datos desde la aplicación. Toda la programación se encuentra en los anexos.

### **3.8.2. Programación de los servomotores para movimiento de la cámara.**

Al igual que en la programación de los motores reductores se presenta un diagrama de flujo de la programación de los servomotores.

En la aplicación se presenta una comunicación que inicia con la activación de conexión vía WIFI entre el ESP8266 y la App, se definen las variables que se requieren durante la programación, en el paso siguiente se habilita los pinos de los servomotores que están conectados al controlador para que puedan ser utilizados.

Si existe una petición de comunicación se recibe la señal que fue enviada desde la aplicación, en las siguientes líneas de programación se establece cual es la petición, si es que el servomotor uno o dos se debe mover hacia arriba o hacia abajo, caso contrario se espera una nueva petición de la aplicación. En la figura 3.21 se presenta el diagrama de flujo para los motores reductores.



**Figura 3. 21** Diagrama de flujo para los motores reductores

Para la programación de los servomotores se requirió añadir comandos que permitan la comunicación y control de los motores con el NodeMCU8266 que son expuestos a continuación:

- **Servo.h:** Proporciona la librería para servomotores.
- **Attach ():** Comando utilizado para definir en qué pin del controlador va a ir conectado el servomotor.
- **Write ():** Escribe datos en un puerto, la información viaja en una longitud de uno a varios bytes.



A continuación, se presenta una explicación de las líneas de comando que se utilizó para la comunicación entre los servomotores SG90 y el NodeMCU8266.

En la primera parte se añade la librería de los servomotores el cual permite una óptima comunicación con el dispositivo, se ingresan las variables para los servomotores como s1 y s2, se ingresa dos variables de tipo entero con un valor de 90, todas estas líneas de comando se agregan a las ya explicadas anteriormente.

Para habilitar los puertos de comunicación entre los servomotores y el NodeMCU se utiliza el comando attach y en el paréntesis se especifica el puerto, como parte de la inicialización del programa se decide que los servomotores estén a 90 grados al encender el circuito, esto se lo hace mediante la instrucción write, cuando los datos ingresados indican que se mueva el servomotor hacia arriba o hacia abajo, el NodeMCU lo interpreta y realiza el giro que fue solicitado mediante el programa que están con el condicional If, finalmente se agregan los comandos con el fin de recibir los datos desde la aplicación.

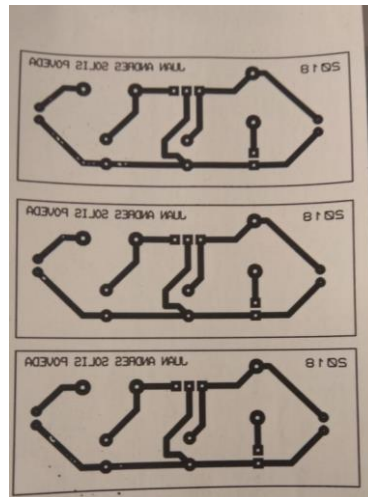
En la última sección de las líneas de comando se ingresa la configuración para el rango de movimiento del servomotor que va de 0 a 180 grados de giro, donde a la variable var se le suma 5 grados o se resta la misma cantidad de grados por cada envío de señal desde la aplicación instalada en el celular.

Se debe tener en cuenta que una mala programación en el alcance máximo del ángulo de giro, en sentido positivo o negativo, posibilitará que los engranes se puedan destruir y por consiguiente el dispositivo se quede en desuso. La programación completa se presenta en los anexos del documento.

### **3.9. Implementación de la fuente de alimentación para los servomotores**

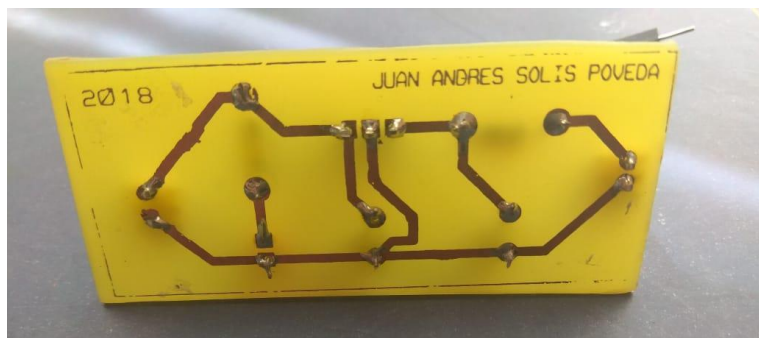
Para la implementación del regulador de voltaje se requiere un circuito impreso el cual ya fue diseñado y simulado en el capítulo tres, luego de haber adquirido todos los elementos que lo conforman se realiza el siguiente procedimiento:

- Como primer paso se debe imprimir el formato PCB (Placa de Circuito Impreso) del simulador Proteus, ya que posee las pistas o guías de las conexiones del circuito, para ello se debe imprimir en papel termotransferible, en la figura se presenta la impresión.
- Se debe cortar a la baquelita a las dimensiones que tiene la hoja impresa con el PCB, luego con la ayuda de una plancha casera se procedió a trasferir las pistas que estan en el papel hacia la baquelita, se debe tener mucho cuidado para que no se pierda ninguna conexión del circuito.

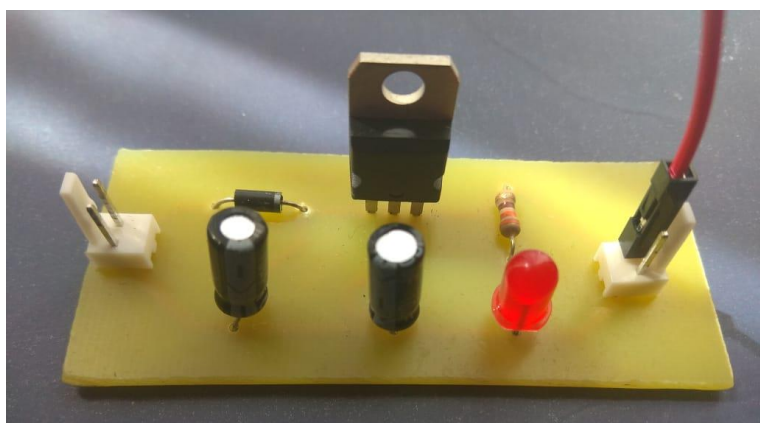


**Figura 3. 22** Diseño de circuito impreso

- Para remover el cobre que no va a ser utilizado en el circuito es necesario sumergir la placa en ácido cloruro férrico mezclado con agua caliente, esto permitirá que en pocos minutos pueda retirarse el exceso de cobre.
- Luego de limpiar la placa y quitar los residuos se procede a soldar cada uno de los elementos del circuito, se debe tener en cuenta cual es la posición correcta y su polaridad ya que puede provocar un daño en toda la placa, en la 3.23 se presenta el circuito soldado, y en la figura 3.24 se presenta el circuito ya montado.



**Figura 3. 23** Elementos soldados a la placa



**Figura 3. 24** Implementación del circuito regulador de voltaje

### 3.10. Implementación del vehículo a control remoto

Para la implementación del vehículo a control remoto se decidió realizarlo sobre una estructura de acrílico con llantas de plástico recubiertas de caucho, en las llantas fueron montados los motores y ubicados por debajo de toda la estructura.

Los pines de conexión están ubicados según la tabla 3.2, los pines de alimentación se describen en la tabla 3.3, se debe tomar en cuenta que la fuente LiPo también alimenta al regulador de voltaje, pero este disminuye a cinco voltios la alimentación que es suministrada a los servomotores. En el transmisor TS832 se encuentra los pines de video y tierra que son conectados directamente a la cámara.

**Tabla 3. 2** Distribución de pines

DISTRIBUCIÓN DE PINES		
NodeMCu8266	L298n	Servomotores
D1		PIN1(naranja)
D2		PIN1(naranja)
D5	IN4	
D6	IN3	
D7	IN2	
D8	IN1	

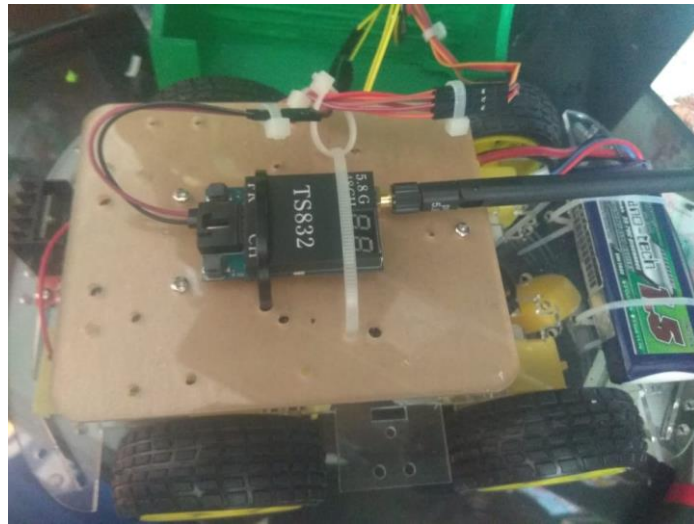
**Tabla 3. 3** Distribución de pines de alimentación

ALIMENTACIÓN DE LOS DISPOSITIVOS				
Fuente LiPo(7,4v)	Regulador de voltaje(5v)	NodeMCU8266	Servomotores	Transmisor TS832
VCC		VCC		VCC
GND		GND		GND
	VCC		VCC	
	GND		GND	

Sobre la cubierta se encuentran sujetos con tornillos y tuercas todos los elementos requeridos del circuito, se ha requerido colocar los dispositivos en dos niveles separados por postes metálicos que permiten una correcta división y ajuste entre las dos plantas. En la primera planta se encuentra el NodeMCU8266, el Motor Driver L298N, el regulador de voltaje Step- Down, el case para el NodeMCU y la batería LiPo. A continuación, se presenta en la figura 3.25 el armado de la primera planta con las conexiones del cableado requerido en el circuito.

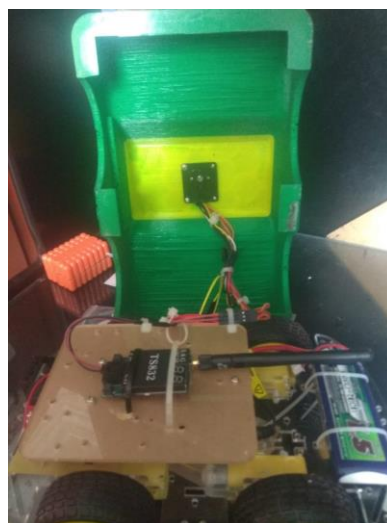
**Figura 3. 25** Armado de la primera planta del vehículo

En la segunda planta se encuentra el transmisor TS832, se lo ubico en el siguiente nivel ya que tiene incorporado la antena transmisora y debe estar muy bien ubicada para que pueda transmitir la señal de video al receptor que se encuentra en las gafas sin ningún problema. En la figura 3.26 se presenta la colocación del transmisor en el segundo nivel del vehículo.



**Figura 3. 26** Armado de la segunda planta del vehículo

En el siguiente paso se coloca los servomotores y el cableado en la carcasa del vehículo que fue impresa en 3D, con la ayuda de amarras se sujeta todos los cables para que tenga una distribución correcta en el armado del circuito. En la figura 3.27 se presenta el armado del servomotor y los cables que lo conectan.



**Figura 3. 27** Armado del servomotor en la cubierta del vehículo

Finalmente se procede a colocar la carcasa del vehículo a control remoto sobre la estructura base, con ayuda de pernos se sujeta con precisión y se procede a verificar que el equipo este correctamente armado. En las figuras 3.28 y 3.29 se presenta la estructura total del circuito armado del vehículo a control remoto controlado por una App con visión en primera persona.



**Figura 3. 28** Proyecto completado, vista lateral



**Figura 3. 29** Proyecto completado, vista frontal

### 3.11. Pruebas de funcionamiento

Para la validación del proyecto es necesario realizar un análisis de los resultados obtenidos de las pruebas de funcionamiento, del producto final, el proyecto está dividido en 3 etapas.

La primera etapa es la comunicación y control de la aplicación instalada en el celular con el vehículo mediante el módulo WIFI, en la tabla 3.4 se presenta los resultados obtenidos en días de funcionamiento.

**Tabla 3. 4** Análisis de funcionamiento del vehículo con la App

<b>Análisis de funcionamiento del vehículo con la App</b>					
<b>Fecha</b>	<b>Tiempo de uso(minutos)</b>	<b>Número de errores</b>			
		<b>D1(10m)</b>	<b>D2(20m)</b>	<b>D3(30m)</b>	<b>D4(40m)</b>
23/7/2018	52	0	1	1	2
25/7/2018	51	0	0	1	3
27/7/2018	53	0	2	1	2
30/7/2018	53	0	0	2	4
1/8/2018	50	0	1	2	4
3/8/2018	52	0	0	1	2
6/8/2018	51	0	0	0	2
8/8/2018	52	0	1	1	1
10/8/2018	54	0	2	1	3

En la segunda etapa se presenta las pruebas realizadas entre la aplicación instalada en el celular y los servomotores en la tabla 3.5.

**Tabla 3. 5** Análisis de funcionamiento de los servomotores con la App

<b>Análisis de funcionamiento de los servomotores con la App</b>					
<b>Fecha</b>	<b>Tiempo de uso(minutos)</b>	<b>Número de errores</b>			
		<b>D1(10m)</b>	<b>D2(20m)</b>	<b>D3(30m)</b>	<b>D4(40m)</b>
30/7/2018	51	0	1	1	1
1/8/2018	50	0	0	1	4
3/8/2018	54	0	1	2	1
6/8/2018	50	0	1	0	3
8/8/2018	53	0	1	2	3
10/8/2018	54	0	2	1	4
13/8/2018	51	0	1	3	3
15/8/2018	52	0	0	1	2
17/8/2018	51	0	1	2	1

Finalmente, el último análisis de resultados se presenta el circuito completo en la tabla 3.6, es decir la aplicación instalada en el celular que controla a todo el sistema instalado en el vehículo y con FPV.

**Tabla 3. 6** Análisis de funcionamiento del circuito completo

<b>Análisis de funcionamiento del circuito completo</b>					
<b>Fecha</b>	<b>Tiempo de uso(minutos)</b>	<b>Número de errores</b>			
		<b>D1(10m)</b>	<b>D2(20m)</b>	<b>D3(30m)</b>	<b>D4(40m)</b>
31/7/2018	50	0	0	1	2
2/8/2018	49	0	2	1	3
4/8/2018	52	0	1	1	4
7/8/2018	51	0	1	3	3
9/8/2018	53	0	2	1	2
11/8/2018	51	0	1	2	1
14/8/2018	50	0	1	2	3
16/8/2018	50	0	2	3	4
17/8/2018	52	0	1	1	2

El análisis indica que durante todas las pruebas de funcionamiento, el circuito completo funcionó correctamente, las pruebas se realizaron a la misma distancia para verificar que las condiciones de prueba no cambien, se tomó el tiempo con el cronómetro del celular, la columna que indica errores en las tablas es si existió durante las pruebas, un problema de comunicación del mando con el equipo, es decir que si el usuario digito una tecla en el control y no hubo respuesta, ésta se toma como error.

### **3.12. Análisis de resultados**

Las pruebas realizadas con el proyecto concluido, dan como resultado que, al aumentar la distancia de recorrido del vehículo con respecto al control de mando, las señales de control se demoran en llegar en el orden de los milisegundos, por lo que no se ejecutan la tarea a un gran trayecto de inmediato, al ser el número de errores muy pocos se descartan, ya que en su gran mayoría fueron satisfactorios.

Se realizaron pruebas por separado, primero para la comunicación de los motores reductores con la aplicación y luego la App con los servomotores, en las dos pruebas de



funcionamiento se verifico que la imagen dentro de la distancia tomada en la ejecución no perdió la recepción del video a las gafas en ninguna de las demostraciones.

La distancia está limitada por el dispositivo telefónico celular utilizado y por la interferencia que puede darse por la utilización de la banda de frecuencia no licenciada. La distancia máxima a la que el equipo trabaja en óptimas condiciones está en cuarenta metros como máximo.

## CONCLUSIONES

- Se definieron los elementos y dispositivos a utilizar de las diferentes etapas de armado y construcción del prototipo, además del tipo de programación y el software para el diseño de la carcasa del vehículo.
- Se realizó el esquema electrónico del hardware del vehículo con la ayuda del simulador Proteus, para desarrollar el diagrama circuital y también el PCB (Placa de Circuito Impreso) que permitió realizar el circuito impreso.
- Se presentó una aplicación en App Inventor 2 para celulares con sistema operativo Android, que permitió la comunicación y control del vehículo y los servomotores, mediante la programación del módulo WIFI.
- Se implementó un sistema de comunicación WIFI, con la programación del módulo NodeMCU8266, para el control de los servomotores y el control de giro de las llantas.
- Se realizó un sistema de visión en primera persona denominado FPV, para que las imágenes captadas por la cámara, instalada sobre servomotores, sean enviadas al emisor TS832 y transmitidas a las gafas.
- Se construyó un regulador de voltaje Step-Down, ya que los servomotores trabajan a una tensión de 5 voltios, por lo que se realizó un circuito impreso con el integrado 7805, para bajar de 7.4 voltios a la alimentación requerida por el dispositivo.
- Se fabricó la carcasa del vehículo a control remoto, tiene el diseño de una camioneta y fue realizada en el programa SolidWork, al ser un formato de imagen compatible con las impresoras 3D se obtuvo una cubierta ideal para la presentación del proyecto.

## RECOMENDACIONES

- En la alimentación del circuito completo, se requiere tomar las mediciones de consumo de corriente de cada uno de los elementos y dispositivos; éstas mediciones permitirán escoger cual es la fuente de alimentación correcta y suficiente para el proyecto.
- Los elementos y dispositivos que componen el proyecto, no se pueden alimentar con la misma fuente ya que son los servomotores trabajan a 5 voltios, por lo que es fundamental conocer las especificaciones técnicas para prevenir daños.
- Al momento de construir el regulador de voltaje, es necesario tener cuidado al realizar el circuito impreso ya que se debe verificar que las pistas estén completas y no cruzadas.
- La estructura base del vehículo y la carcasa deben tener orificios para la ventilación del circuito, ya que sufren daños al sobrecalentarse.
- El momento de programar en el NodeMCU8266 a los servomotores es necesario poner un límite de giro, ya que si supera los 180 grados y el controlador sigue las órdenes para que se mueva, dañara los engranes y por consiguiente al servo.
- En el momento de la conexión del circuito entre todos los elementos que lo conforman se debe siempre tener en cuenta que el cable de tierra de alimentación de la fuente debe ser común para todos, ya que si no es así puede existir una variación de voltaje diferente a los demás.
- Al momento de instalar la cámara FPV, no es recomendable que esté conectada a la alimentación del circuito, ya que el dispositivo se recalienta; la cámara posee una batería interna que permite una duración de 3.5 horas que es más que suficiente para el uso en el proyecto.

## BIBLIOGRAFÍA

- Candela, F., & Corrales, J. (Septiembre de 2007). *Servomotores*. Obtenido de <http://www.aurova.ua.es/previo/dpi2005/docs/publicaciones/pub09-ServoMotores/servos.pdf>
- Del Valle, L. (2018). *Guía* . Obtenido de [https://drive.google.com/file/d/0B0-h\\_HkOmksCYk1uQk1JbVE4SEU/view](https://drive.google.com/file/d/0B0-h_HkOmksCYk1uQk1JbVE4SEU/view)
- García, A., Bueno, C., Muñoz, R., & Albares, J. (2017). *Guía de Mantenimiento y Reparación de Drones*. Madrid: Paraninfo.
- Hernández, R., Fernández, C., & Batista, M. (2010). *Metodología de la Investigación*. México: McGraw Hill Educación.
- Huidrobo Moya, J., & Roldán Martínez, D. (2016). *Comunicaciones en redes WLAN*. Creaciones Copyright.
- Huidrobo, J., & Roldan, D. (2005). *Comunicaciones de Redes WLAN*. Creaciones COPYRIGHT.
- Kerlinger, F. (1982). *Fundamentos de la Investigación del Comportamiento*. México: Nueva Editorial Interamericana.
- López Razo, M. J. (2018). *Guía de Mantenimiento y Reparación de Drones*. Madrid: Paraninfo.
- Prabhu, C. (2004). *Bluetooth Technology*. New Delhi: PHI.
- Reyes Cortés , F. (2011). *Robótica*. México: Alfaomega.
- Reyes, F. (2011). *Robótica Control de Robots Manipuladores*. México: Alfaomega.
- Savant, C., Roden, M., & Carperter, G. (2000). *Diseño Electrónico, Circuitos y Sistemas*. Pearson Educación.
- Schneider, B. (s.f.). *A Guide to LiPo Batteries*. Obtenido de [http://www.rcaces.org/RC%20Aces%20News%20Letter/A\\_Guide\\_to\\_LiPo\\_Batteries.pdf](http://www.rcaces.org/RC%20Aces%20News%20Letter/A_Guide_to_LiPo_Batteries.pdf)
- Schwartz, M. (2016). *Internet of Things with ESP8266*. Birmingham: Packt Publishing.
- Tran, P. (2017). *SolidWorks Basic Tools 2018*. SDC Publication.
- Zabala, G. (2007). *Robótica Guía Teórica y Práctica*. Banfield: USERS expess.

## **ANEXOS**

# **PROYECTOS DE REPOSITARIOS DE UNIVERSIDADES Y ESCUELAS POLITÉCNICAS**

## **DISEÑO E IMPLEMENTACIÓN DE UN ROBOT HEXÁPODO**

El presente proyecto consiste en un diseño electromecánico que pretende solventar incidencias que ocurren en las cámaras de registro de la empresa Telconet. Para el efecto, inicialmente se realizó un estudio de campo que permitió conocer los procedimientos que realizan los técnicos para el ingreso a las cámaras de registro, actividades que realizan dentro de las cámaras de registro tales como; revisión de cables de fibra óptica, acometidas, instalación y desmontaje de cables de obra, creación de mangas y fusiones. También se realiza revisiones de la infraestructura civil. Dentro de todos los procedimientos descritos en la cámara de registro se evidenciaron inconvenientes, como riesgos de seguridad al momento del ingreso y la salida de los técnicos de Telconet e incomodidad al realizar algún trabajo dentro de la cámara.

Condiciones ambientales adversas dentro de la cámara de registro principalmente como presencia de gas, baja iluminación, altas temperaturas y humedad. Con la construcción de un robot zoomórfico con transmisión de video y control remoto mediante comunicación inalámbrica, a través de una interfaz gráfica en un Smartphone con sistema operativo Android, se pretende evitar o minimizar los inconvenientes y posibles riesgos identificados para los técnicos que trabajan diariamente en la revisión física de las cámaras de registro, dando la alternativa de revisar la cámara de registro desde afuera, para asegurar a la personas que van a ingresar.

De esta manera se podrán identificar los factores nocivos para el ser humano y también se realizará la revisión de la infraestructura de la cámara de registro. Toda la información referente a la programación o código fuente del hardware estará adjuntada en Anexos al final de la documentación del proyecto. Palabras Clave: Diseño, Robot, Hexápodo, Inspección, Cámaras, Fibra, Telconet.

## **ESTUDIO, DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ROBOT EXPLORADOR Y ENRUTADOR DE CABLE.**

En el presente documento se muestra el diseño e implementación de un robot explorador y enrutador de cable, este proyecto tiene como finalidad ayudar a técnicos que realizan cableado estructurado, a enrutar cable en lugares en donde no es posible el acceso. Este prototipo consta de las herramientas necesarias para guiar cable y además monitorear la tarea que se está efectuando en tiempo real. En el capítulo 1 se detalla los antecedentes del problema que se va a resolver y los objetivos que se tiene que alcanzar.

En el capítulo 2 se muestra toda la teoría requerida para el desarrollo del robot, detallando las características generales de los componentes usados. En el capítulo 3 se explica el estudio y diseño del sistema, lo que permite el montaje y la implementación del robot, con las explicaciones necesarias.

En el capítulo 4 se muestra el análisis de las pruebas realizadas bajo los parámetros establecidos; además se presenta un análisis FODA del sistema implementado; y finalmente se presenta una lista de costos de los equipos utilizados en el proyecto. En el capítulo 5 se presentan las conclusiones y recomendaciones sacadas del proyecto.

## **CONSTRUCCIÓN DE UN ROBOT MÓVIL HÍBRIDO OMNIDIRECCIONAL.**

En este apartado se describe la construcción y las características técnicas de funcionamiento de los actuadores, sensores, tarjetas de control y módulos especiales a emplear en el robot explorador híbrido omnidireccional. El robot está compuesto por una plataforma móvil omnidireccional y un brazo robótico de 5 grados de libertad con una pinza, los cuales se los puede utilizar de manera individual si así se lo requiere.

El prototipo está basado en la imagen de un robot industrial conocido como los Kuka youBot, el robot tiene como característica de manejarse sobre superficies planas con desniveles de hasta el 10% de su longitud horizontal. El robot omnidireccional será controlado inalámbricamente, con la manipulación directa de un operador que contará con el apoyo de un software amigable, el cual permitirá un fácil manejo del robot. La

construcción del robot, parte de un diseño previamente dimensionado para la ubicación correcta de los elementos electrónicos, el hardware se realizó en impresión en 3D y el diseño fue realizado en el software Inventor. Se realizará las respectivas pruebas de funcionamiento para verificar su viabilidad, además el robot pretende servir de soporte en aplicaciones robóticas en el campo de la investigación dentro de la carrera de la ingeniería electrónica.

## **DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE ROBOT MÓVIL CON CONTROL AUTÓNOMO O RADIOFRECUENCIA E IMAGEN EN TIEMPO REAL.**

Este prototipo de robot se pone en marcha y mediante sensores reconoce obstáculos en escenarios para el cual ha sido definida su programación. Este robot móvil sirve para la experimentación e investigación en ambientes reales, ya que posee características similares a los robots de aplicación industrial o comercial. Tiene tres modos de operación, mediante un potenciómetro y la variación de voltaje se puede seleccionar en qué modo desea que trabaje el prototipo sea este como ROBOT AUTÓNOMO SUMO, ROBOT AUTÓNOMO EXPLORADOR y en cualquiera de estos modos mencionados si se manipula el transmisor del control de radio frecuencia el prototipo interrumpe su modo de operación en el que se encuentra. Una vez interrumpida su operación en cualquiera de los modos este procede a recibir las señale de RF y pasa a trabajar o funcionar en su tercer modo de operación que es el de CONTROLADO POR RF. Además en cualquiera de los 3 modos de operación puede trabajar con la imagen en tiempo real ya que tiene un interruptor independiente para encender la mini cámara inalámbrica y podremos visualizar en el ordenador la imagen transmitida por el prototipo.

## **DISEÑO Y CONSTRUCCIÓN DE UN ROBOT MÓVIL EXPERIMENTAL 8X8 ARTICULADO PARA SALVAR OBSTÁCULOS**

Se realizó el diseño e implementación del robot móvil experimental 8x8 articulado para salvar obstáculos; el robot cuenta con dos vagones unidos mediante una unión de tipo par cilíndrico, posee dos sensores infrarrojos para la detección de obstáculos ubicados frente a las ruedas delanteras articulada, que se suben de manera independiente; su sistema de locomoción es mediante transmisión por rodillos de fricción; cuatro motores de corriente



continua son los cuales brindan la fuerza de tracción hacia el par de ruedas ubicadas a cada lado de los vagones, cuatro servomotores que sirven como actuadores de los mecanismos articulados los cuales cumplen la función de levantar las ruedas móviles y de esta manera poder salvar de mejor forma los obstáculos; dos circuitos de control de motores los cuales activan los motores DC y controlan el sentido de giro (adelante o atrás) para movilizar el robot para poder explorar su entorno y un circuito de control principal, el cual recibe la información de los sensores y el control remoto, los procesa y envía las señales de pulsos (PWM) a los servomotores y los motores DC. El control Remoto envía mediante cable información sobre el posicionamiento, mediante el movimiento del mismo.

## **DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO ROBÓTICO MÓVIL TELEOPERADO INALÁMBRICAMENTE PARA INSPECCIÓN EN LUGARES DE ALTO RIESGO**

El presente trabajo de titulación detallado consiste en el diseño y construcción de un prototipo robótico móvil para la inspección de lugares de alto riesgo enfocado a la seguridad civil. El robot tiene la capacidad de ser maniobrado inalámbricamente por medio de un control remoto y transmitir audio y vídeo en tiempo real. El objetivo del desarrollo de este proyecto es la inspección y reconocimiento del entorno en sitios de posibles reuniones delictivas por medio del robot, mientras que los agentes del orden monitorizan y teledirigen al robot desde un lugar seguro; de esta manera, salvaguardando sus vidas. Además, se ha diseñado e investigado una geometría especial para la rodadura de la rueda a base de un material de alta deformidad y desplazamiento, con la finalidad de absorber los efectos de caída libre de hasta 2 metros. El alcance máximo en línea de vista es de 260 metros y en interiores es de 90 metros con una autonomía máxima de 50 minutos., lo que le hace un robot espía a corta distancia.

# **//PROGRAMACIÓN EN IDE DE ARDUINO PARA VEHÍCULO A CONTROL REMOTO CON VISIÓN EN PRIMERA PERSONA**

## **// PROYECTO DE TITULACIÓN**

## **// JUAN ANDRÉS SOLÍS POVEDA**

```
#include <Servo.h> // Se incluye la librería para los servomoteres
#include <ESP8266WiFi.h> // Se ingresa la Librería para el NodeMCU ESP8266
WiFiClient client; // Se ingresa la Librería para el NodeMCU ESP8266
WiFiServer server(80); // Se crea un servidor que escucha las conexiones entrantes
Servo s1; // Se ingresa la Librería para el NodeMCU ESP8266
Servo s2; // Se ingresa la variable para el servo 2

int var1=90; // Se crea una variable de tipo entero que tenga un valor de 90 para el servo 1
int var2=90; // Se crea una variable de tipo entero que tenga un valor de 90 para el servo 2

const char* ssid = "Redmi"; // Se define el nombre de usuario de la red a conectarse
const char* password = "uisrael2018"; // Se define la contraseña del usuario de la red a conectarse
String data = ""; // Se define el comando para recibir información desde la aplicación

// Se definir los pines de control del NodeMCU8266 al motor driver L298n
int leftMotorForward = 14; // Conectar GPIO14(D5) a la entrada IN3 del motor driver
int rightMotorForward = 15; // Conectar GPIO15(D8) a la entrada IN1 del motor driver
int leftMotorBackward1 = 12; // Conectar GPIO12(D6) a la entrada IN4 del motor driver
```

```

int rightMotorBackward1 = 13; // Conectar GPIO13(D7) a la entrada IN2 del motor driver

void setup()
{
  s1.attach(4); // Se conecta el pin del servo 1 al puerto GPIO4(D1)
  s2.attach(5); // Se conecta el pin del servo 1 al puerto GPIO5(D2)
  s1.write(90); // El servo 1 se va a inicializar en 90 grados
  s2.write(90); // El servo 2 se va a inicializar en 90 grados

  // Se inicializan los pines de control del motor driver como salidas

  pinMode(leftMotorForward, OUTPUT);
  pinMode(rightMotorForward, OUTPUT);
  pinMode(leftMotorBackward, OUTPUT);
  pinMode(rightMotorBackward, OUTPUT);
  server.begin(); // Se inicia la comunicación del servidor
}

void loop()
{
  // Si el servidor está disponible, se va a ejecutar la función "checkClient"
  client = server.available();
  if (!client) return;
  data = checkClient ();

  //Comandos de comunicación para que la app comande los movimientos del vehículo
  if (data == "forward") MotorForward(); // Si los datos ingresados son "hacia adelante",
  ejecute la función "MotorForward"

```

```

    else if (data == "backward") MotorBackward(); // Si los datos ingresados son "hacia
atrás", ejecute la función "MotorBackward"

    else if (data == "left") TurnLeft(); // Si la información ingresada es "izquierda", ejecute
la función "TurnLeft"

    else if (data == "right") TurnRight(); // Si los datos ingresados son "derecha", ejecute la
función "TurnRight"

    else if (data == "stop") MotorStop(); // Si los datos ingresados son "detenerse", ejecute
la función "MotorStop"

    else if (data == "s1up") s1arriba(); // Si los datos ingresados son "hacia arriba", ejecute
la función "s1up"

    else if (data == "s1down") s1abajo(); // Si los datos ingresados son "hacia abajo", ejecute
la función "s1down"

    else if (data == "s2up") s2arriba(); // Si los datos ingresados son "hacia arriba", ejecute
la función "s2up"

    else if (data == "s2down") s2abajo(); // Si los datos ingresados son "hacia abajo", ejecute
la función "s2down"
}

```

```

// Comandos para el movimiento del vehículo hacia adelante

```

```

void MotorForward(void)
{
    digitalWrite(leftMotorForward,HIGH);
    digitalWrite(rightMotorForward,HIGH);
    digitalWrite(leftMotorBackward1,LOW);
    digitalWrite(rightMotorBackward1,LOW);
}

```

```

void MotorBackward(void)

```

```

{

```

```

// Comandos para el movimiento del vehículo hacia atras

```

```

digitalWrite(leftMotorBackward1,HIGH);
digitalWrite(rightMotorBackward1,HIGH);
digitalWrite(leftMotorForward,LOW);
digitalWrite(rightMotorForward,LOW);
}
void TurnLeft(void)
{
// Comandos para el movimiento del vehículo hacia la izquierda
digitalWrite(leftMotorForward,HIGH);
digitalWrite(rightMotorForward,LOW);
digitalWrite(rightMotorBackward1,LOW);
digitalWrite(leftMotorBackward1,LOW);
}
// Comandos para el movimiento del vehículo hacia la derecha
void TurnRight(void)
{
digitalWrite(leftMotorForward,LOW);
digitalWrite(rightMotorForward,HIGH);
digitalWrite(rightMotorBackward1,LOW);
digitalWrite(leftMotorBackward1,LOW);
}
// Comandos para que el vehículo se detenga por completo
void MotorStop(void)
{
digitalWrite(leftMotorForward,LOW);
digitalWrite(leftMotorBackward,LOW);
digitalWrite(rightMotorForward1,LOW);
}

```

```

digitalWrite(rightMotorBackward1,LOW);
}
//Recibe los datos desde la aplicación
String checkClient (void)
{
while(!client.available()) delay(1);
String request = client.readStringUntil('\r');
request.remove(0, 5);
request.remove(request.length()-9,9);
return request;
}

// Comandos para que el movimiento del servomotor 1 sea hacia arriba
void s1arriba(void)
{
var1=var1+5;
if(var1>=180)
{
var1=180;
}
s1.write(var1);
}

// Comandos para que el movimiento del servomotor 1 sea hacia abajo
void s1abajo(void)
{
var1=var1-5;
if(var1<=0)

```

```
{  
    var1=0;  
}  
s1.write(var1);  
}  
  
// Comandos para que el movimiento del servomotor 2 sea hacia arriba
```

```
void s2arriba(void)
```

```
{  
    var2=var2+5;  
    if(var2>=180)  
    {  
        var2=180;  
    }  
    s2.write(var2);  
}
```

```
// Comandos para que el movimiento del servomotor 2 sea hacia abajo
```

```
void s2abajo(void)
```

```
{  
    var2=var2-5;  
    if(var2<=0)  
    {  
        var2=0;  
    }  
    s2.write(var2);  
}
```



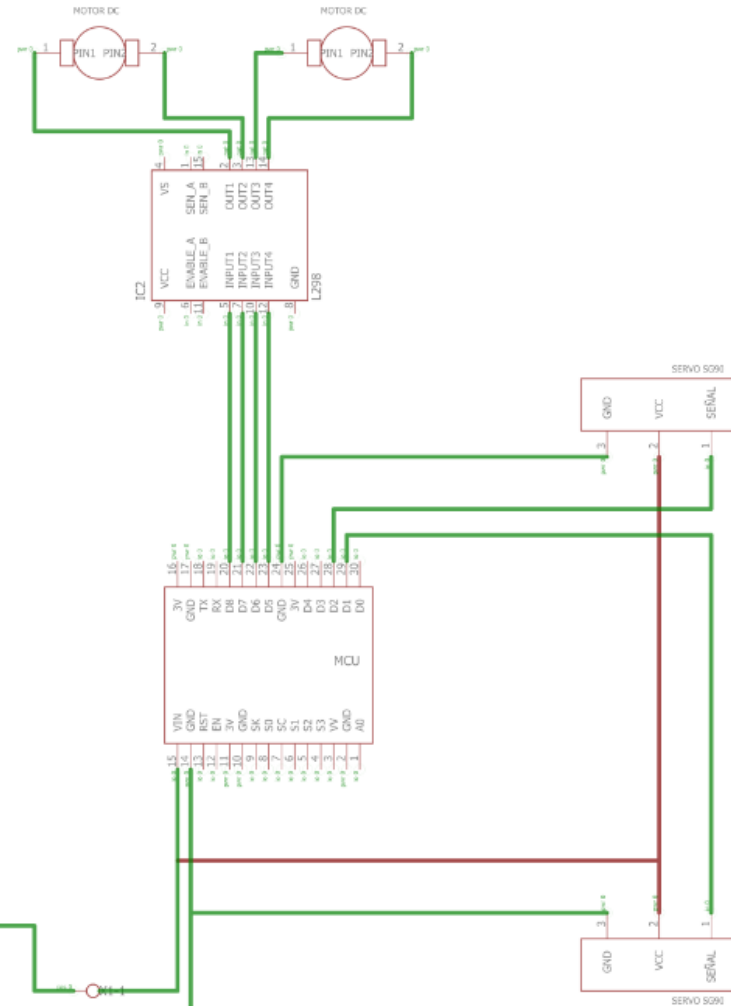
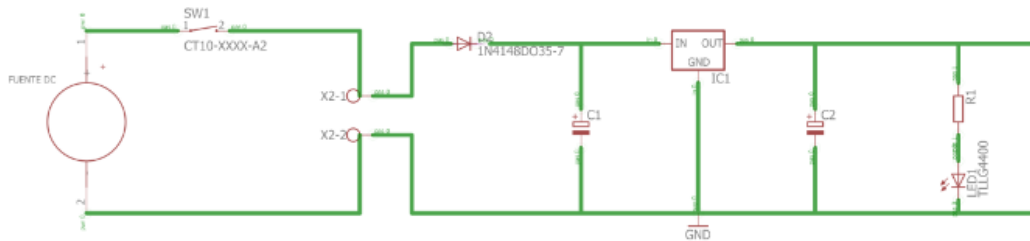


# VEHÍCULO A CONTROL REMOTO CON VISIÓN EN PRIMERA PERSONA

JUAN ANDRÉS SOLÍS POVEDA

## LEYENDA

- SW1: Interruptor
- D2: Diodo
- C1, C2: Capacitores
- IC1: Transistor 7805
- R1: Resistencia
- LED1: Diodo emisor de luz
- IC2: Motor Driver L298N
- MCU: Placa NodeMCU8266
- Servo SG90: Servomotores





**DESARROLLO DE UN VEHÍCULO A CONTROL REMOTO CON  
VISIÓN EN PRIMERA PERSONA.**

**MANUAL DEL USUARIO**

Versión: 1.0

Fecha: 22-08-2018

## CONTENIDO

1. DESCRIPCIÓN DEL CIRCUITO .....	89
1.1. Precauciones antes del encendido del vehículo.....	89
1.2. Recomendaciones para el mantenimiento del vehículo .....	89
2. INTRODUCCIÓN .....	89
2.1. Funcionalidad.....	89
2.2. Componentes.....	90
3. SOFTWARE.....	94
3.1. Diagrama del circuito.....	100
4. SOLUCIÓN DE PROBLEMAS MÁS COMUNES.....	101
5. SOPORTE TÉCNICO .....	102

## LISTA DE FIGURAS

<b>Figura 2.1</b> NodeMCU8266.....	90
<b>Figura 2.2</b> Motor Driver L298N.....	90
<b>Figura 2.3</b> Motor Driver L298N.....	91
<b>Figura 2.4</b> Regulador de voltaje .....	91
<b>Figura 2.5</b> Motor reductor .....	91
<b>Figura 2.6</b> Batería LiPo .....	91
<b>Figura 2.7</b> Cámara FPV .....	92
<b>Figura 2.8</b> Cámara FPV .....	92
<b>Figura 2.9</b> Transmisor FPV .....	93
<b>Figura 2.10</b> Servomotores .....	94
<b>Figura 3.1</b> Reductor de voltaje .....	101
<b>Figura 3.2</b> Diagrama del circuito total.....	101

<b>Universidad</b>	Universidad Tecnológica Israel		
<b>Proyecto</b>	Desarrollo de un vehículo a control remoto con visión en primera persona.		
<b>Entregable</b>	Manual de Usuario		
<b>Autor</b>	Juan Andrés Solís Poveda		
<b>Versión/Edición</b>	1.0	<b>Fecha Versión</b>	22-08-2018
<b>Tutor</b>	Mg. René Ernesto Cortijo Leyva	<b>Nº Total de Páginas</b>	18

## **DESCRIPCIÓN DEL CIRCUITO**

### **Precauciones antes del encendido del vehículo**

- El primer paso es verificar que ningún cable visible al exterior este suelto o desconectado.
- Comprobar que el vehículo, cámara, gafas estén a temperatura ambiente.
- Antes de colocarse las gafas se debe ajustar las amarras que contiene al contorno de la cabeza del usuario.
- No se debe colocar en superficies mojadas, el vehículo no posee esa protección.

### **Recomendaciones para el mantenimiento del vehículo**

- Debe alimentar al circuito con una fuente de alimentación de 7.4 voltios a 1500 mAh recomendables con una batería LiPo 2S con un valor de C mayor o igual a 35.
- Si es necesario remover la cámara, solo se debe retirar el velcro que lo sujeta y no los tornillos de seguridad del servomotor.
- Para cargar la batería de las gafas se puede utilizar un adaptador USB incluido por el fabricante o se puede utilizar una batería LiPo de 3.7 voltios a 300 mAh.
- Si es necesario abrir la cubierta del vehículo, se debe tener mucho cuidado de no retirar ningún elemento sin haber quitado la energía del circuito, en especial la antena del transmisor, ya que se produciría el daño total del componente.

## **INTRODUCCIÓN**

### **Funcionalidad**

Es un vehículos terrestres no tripulado con manejo remoto y con un sistema de vista en primera persona, para lo cual tiene un prototipo (hardware y software) de vehículo con aplicación para procesos de búsqueda y exploración en sitios desconocidos.

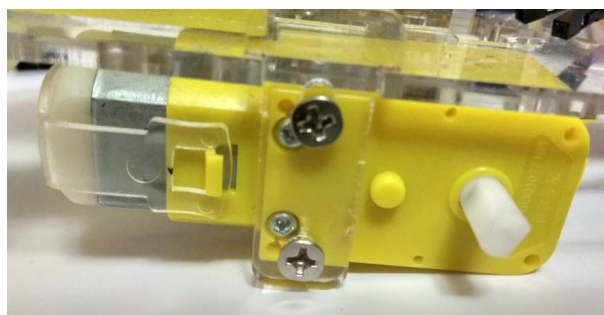
Posee una sistema FPV (First Person View / Vista en Primera Persona), logrando que el usuario tenga una visión más precisa de la ruta, sin que la distancia represente dificultad de manejo y sin afectar la visión del entorno. El vehículo es controlado a través de una aplicación móvil creada en App Inventor.

## Componentes

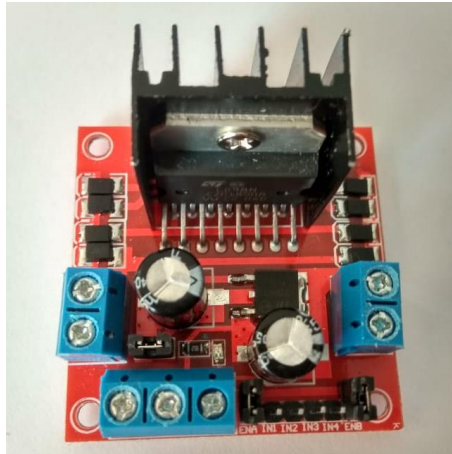
El vehículo a control remoto con visión en primera persona, está dividido en 2 subsistemas, el primero de esta compuesto por los elementos que conforman el circuito de control y movimiento del vehículo y el segundo es el sistema FPV. A continuación se encuentra las figuras correspondientes al primer subsistema.



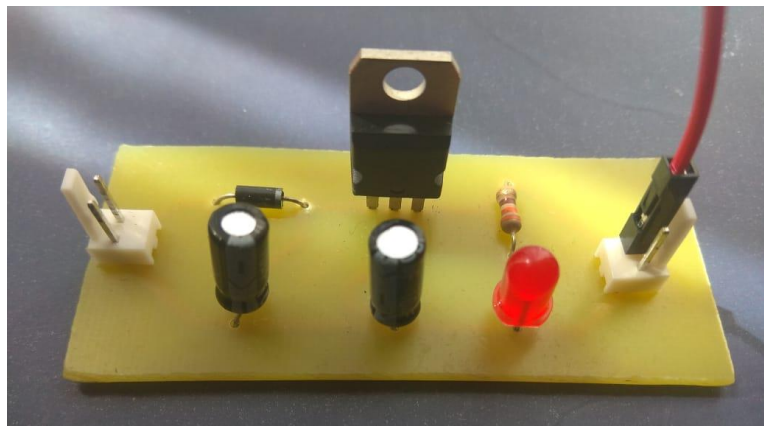
**Figura 2.1** NodeMCU8266



**Figura 2.2** Motor Driver L298N



**Figura 2.3** Motor Driver L298N



**Figura 2.4** Regulador de voltaje



**Figura 2.5** Motor reductor



**Figura 2.6** Batería LiPo



Para el segundo subsistema se presenta a continuación las figuras de cada uno de sus elementos.



**Figura 2.7** Cámara FPV



**Figura 2.8** Cámara FPV

En las gafas existen una serie de botones de mando, permitiendo así que el usuario pueda ingresar a las diferentes opciones que contiene, a continuación explica a continuación:

- Search: Permite buscar automáticamente los canales de comunicación con el transmisor.
- Antena: Es donde se debe colocar la antena receptora.
- On/Off: Botón de encendido y apagado.
- CH+: Busca los canales de manera manual en forma ascendente
- Banda+: Busca las bandas de frecuencia de manera ascendente
- Menú: En la pantalla se despliega el menú para visualizar la banda y canal a la que se encuentra.
- Banda-: Busca las bandas de frecuencia de manera descendente.
- CH-: Busca los canales de manera manual en forma descendente.



**Figura 2.9** Transmisor FPV



**Figura 2.10** Servomotores

## **SOFTWARE**

El software fue implementado sobre el IDE de Arduino, sobre una placa NodeMCU8266, este no debe ser alterado en sus líneas de comando. A continuación se presenta toda la programación.

**//PROGRAMACIÓN EN IDE DE ARDUINO PARA VEHÍCULO A CONTROL  
REMOTO CON VISIÓN EN PRIMERA PERSONA**

**// PROYECTO DE TITULACIÓN**

**// JUAN ANDRÉS SOLÍS POVEDA**

**#include <Servo.h> // Se incluye la librería para los servomotores**

**#include <ESP8266WiFi.h> // Se ingresa la Librería para el NodeMCU ESP8266**

**WiFiClient client; // Se ingresa la Librería para el NodeMCU ESP8266**

**WiFiServer server(80); // Se crea un servidor que escucha las conexiones entrantes**

**Servo s1; // Se ingresa la Librería para el NodeMCU ESP8266**

**Servo s2; // Se ingresa la variable para el servo 2**

**int var1=90; // Se crea una variable de tipo entero que tenga un valor de 90 para el servo 1**

```

int var2=90; // Se crea una variable de tipo entero que tenga un valor de 90 para el servo 2

const char* ssid = "Redmi"; // Se define el nombre de usuario de la red a conectarse

const char* password = "uisrael2018"; // Se define la contraseña del usuario de la red a conectarse

String data = ""; // Se define el comando para recibir información desde la aplicación

// Se definir los pines de control del NodeMCU8266 al motor driver L298n

int leftMotorForward = 14; // Conectar GPIO14(D5) a la entrada IN3 del motor driver
int rightMotorForward = 15; // Conectar GPIO15(D8) a la entrada IN1 del motor driver
int leftMotorBackward1 = 12; // Conectar GPIO12(D6) a la entrada IN4 del motor driver
int rightMotorBackward1 = 13; // Conectar GPIO13(D7) a la entrada IN2 del motor driver

void setup()
{
  s1.attach(4); // Se conecta el pin del servo 1 al puerto GPIO4
  s2.attach(5); // Se conecta el pin del servo 1 al puerto GPIO5
  s1.write(90); // El servo 1 se va a iniiializar en 90 grados
  s2.write(90); // El servo 2 se va a iniiializar en 90 grados

  // Se inicializan los pines de control del motor driver como salidas
  pinMode(leftMotorForward, OUTPUT);
  pinMode(rightMotorForward, OUTPUT);
  pinMode(leftMotorBackward, OUTPUT);
  pinMode(rightMotorBackward, OUTPUT);
  server.begin(); // Se inicia la comunicación del servidor
}

void loop()
{

```

```

// Si el servidor está disponible, se va a ejecutar la función "checkClient"

client = server.available();

if (!client) return;

data = checkClient ();

//Comandos de comunicación para que la app comande los movimientos del vehículo

if (data == "forward") MotorForward(); // Si los datos ingresados son "hacia adelante",
ejecute la función "MotorForward"

else if (data == "backward") MotorBackward(); // Si los datos ingresados son "hacia
atrás", ejecute la función "MotorBackward"

else if (data == "left") TurnLeft(); // Si la información ingresada es "izquierda", ejecute
la función "TurnLeft"

else if (data == "right") TurnRight(); // Si los datos ingresados son "derecha", ejecute la
función "TurnRight"

else if (data == "stop") MotorStop(); // Si los datos ingresados son "detenerse", ejecute
la función "MotorStop"

else if (data == "s1up") s1arriba(); // Si los datos ingresados son "hacia arriba", ejecute
la función "s1up"

else if (data == "s1down") s1abajo(); // Si los datos ingresados son "hacia abajo", ejecute
la función "s1down"

else if (data == "s2up") s2arriba(); // Si los datos ingresados son "hacia arriba", ejecute
la función "s2up"

else if (data == "s2down") s2abajo(); // Si los datos ingresados son "hacia abajo", ejecute
la función "s2down"

}

// Comandos para el movimiento del vehículo hacia adelante

void MotorForward(void)
{

```

```
digitalWrite(leftMotorForward,HIGH);
digitalWrite(rightMotorForward,HIGH);
digitalWrite(leftMotorBackward1,LOW);
digitalWrite(rightMotorBackward1,LOW);
}
void MotorBackward(void)
{

// Comandos para el movimiento del vehículo hacia atras
digitalWrite(leftMotorBackward1,HIGH);
digitalWrite(rightMotorBackward1,HIGH);
digitalWrite(leftMotorForward,LOW);
digitalWrite(rightMotorForward,LOW);
}
void TurnLeft(void)
{

// Comandos para el movimiento del vehículo hacia la izquierda
digitalWrite(leftMotorForward,HIGH);
digitalWrite(rightMotorForward,LOW);
digitalWrite(rightMotorBackward1,LOW);
digitalWrite(leftMotorBackward1,LOW);
}

// Comandos para el movimiento del vehículo hacia la derecha

void TurnRight(void)
```

```
{  
    digitalWrite(leftMotorForward,LOW);  
    digitalWrite(rightMotorForward,HIGH);  
    digitalWrite(rightMotorBackward1,LOW);  
    digitalWrite(leftMotorBackward1,LOW);  
}
```

// Comandos para que el vehículo se detenga por completo

```
void MotorStop(void)
```

```
{  
    digitalWrite(leftMotorForward,LOW);  
    digitalWrite(leftMotorBackward,LOW);  
    digitalWrite(rightMotorForward1,LOW);  
    digitalWrite(rightMotorBackward1,LOW);  
}
```

//Recibe los datos desde la aplicación

```
String checkClient (void)
```

```
{  
    while(!client.available()) delay(1);  
    String request = client.readStringUntil('\r');  
    request.remove(0, 5);  
    request.remove(request.length()-9,9);  
    return request;  
}
```

```
// Comandos para que el movimiento del servomotor 1 sea hacia arriba
```

```
void s1arriba(void)
```

```
{
```

```
    var1=var1+5;
```

```
    if(var1>=180)
```

```
    {
```

```
        var1=180;
```

```
    }
```

```
    s1.write(var1);
```

```
}
```

```
// Comandos para que el movimiento del servomotor 1 sea hacia abajo
```

```
void s1abajo(void)
```

```
{
```

```
    var1=var1-5;
```

```
    if(var1<=0)
```

```
    {
```

```
        var1=0;
```

```
    }
```

```
    s1.write(var1);
```

```
}
```

```
// Comandos para que el movimiento del servomotor 2 sea hacia arriba
```

```
void s2arriba(void)
```

```
{
```

```
    var2=var2+5;
```



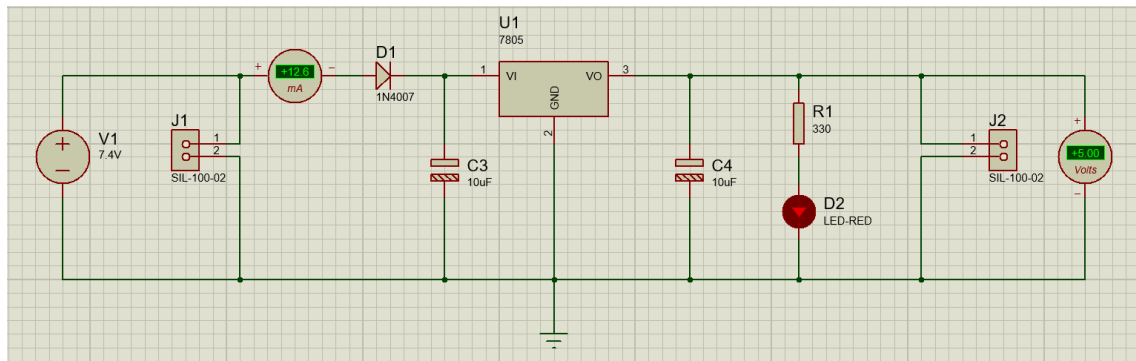
```
    if(var2>=180)
    {
        var2=180;
    }
    s2.write(var2);
}
```

// Comandos para que el movimiento del servomotor 2 sea hacia abajo

```
void s2abajo(void)
{
    var2=var2-5;
    if(var2<=0)
    {
        var2=0;
    }
    s2.write(var2);
}
```

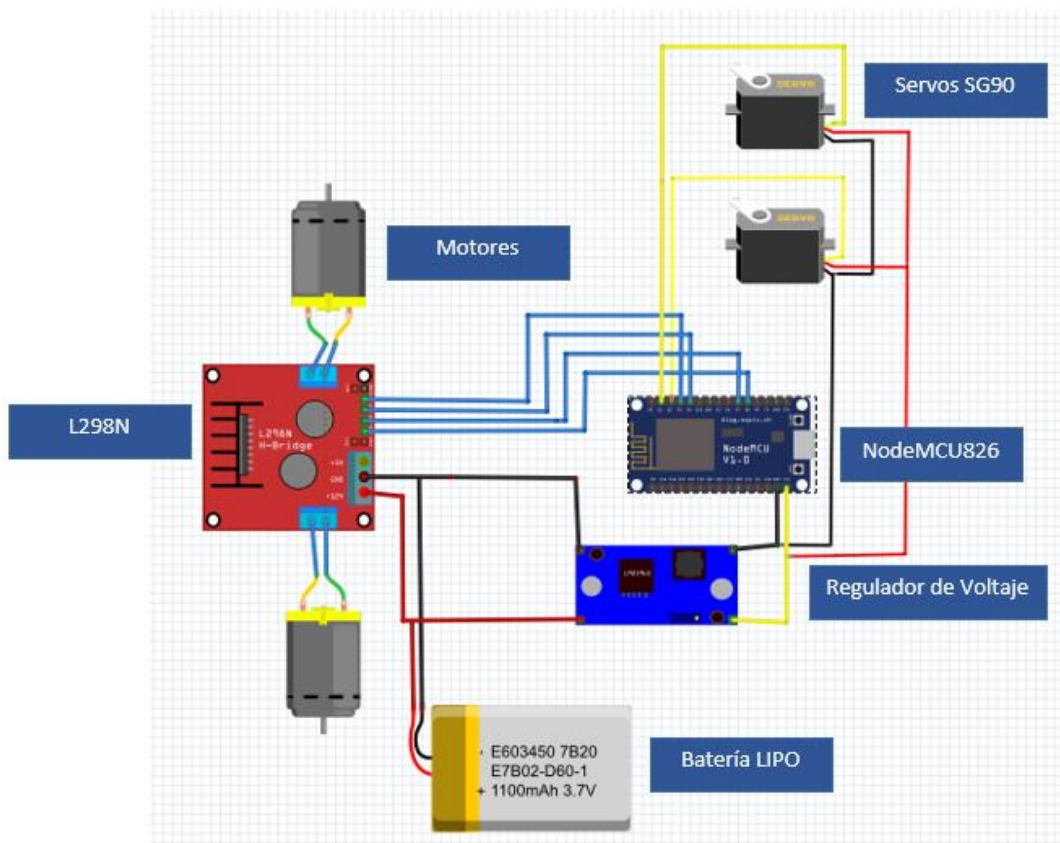
### **Diagrama del circuito**

Los diagramas de circuito, permiten conocer al usuario cuales son las pistas diseñadas para el proyecto. En la figura 3.1 se presenta el diagrama del regulador de voltaje.



**Figura 3.11** Reductor de voltaje

A continuación en la figura 3.2 se presenta el desgrama del armado del circuito total.



**Figura 3.12** Diagrama del circuito total

## SOLUCIÓN DE PROBLEMAS MÁS COMUNES

- No se debe sustituir la fuente de alimentación por una menor a los valores especificados, ya que esto podría desembocar en un daño al circuito si los valores de voltaje son mayores.

- Si no se visualiza ninguna imagen en las gafas, a pesar de que esta encendido, se debe verificar el canal de comunicación, es muy probable que no esté en el mismo del transmisor. Para facilitar el resultado se debe digitar la tecla Search en las gafas para que busque automáticamente la banda y el canal.
- Si se desea grabar el recorrido del vehículo se debe verificar que las Tarjeta de memoria este colocada en la cámara y sobre todo que tenga espacio de almacenamiento.
- Si no se visualiza una imagen en la cámara, otra opción para verificar es ver si la cámara está encendida, ya que el dispositivo de video no se alimenta con la misma fuente del circuito, porque tiene alimentación propia.

## **SOPORTE TÉCNICO**

**Nombre:** Andrés Solís

**Correo:** andres.solis.jasp@gmail.com

# ESP8266EX

## Datasheet



Version 5.8  
Copyright © 2018



# 1.

# Overview

Espressif's ESP8266EX delivers highly integrated Wi-Fi SoC solution to meet users' continuous demands for efficient power usage, compact design and reliable performance in the Internet of Things industry.

With the complete and self-contained Wi-Fi networking capabilities, ESP8266EX can perform either as a standalone application or as the slave to a host MCU. When ESP8266EX hosts the application, it promptly boots up from the flash. The integrated high-speed cache helps to increase the system performance and optimize the system memory. Also, ESP8266EX can be applied to any microcontroller design as a Wi-Fi adaptor through SPI / SDIO or I2C / UART interfaces.

ESP8266EX integrates antenna switches, RF balun, power amplifier, low noise receive amplifier, filters and power management modules. The compact design minimizes the PCB size and requires minimal external circuitries.

Besides the Wi-Fi functionalities, ESP8266EX also integrates an enhanced version of Tensilica's L106 Diamond series 32-bit processor and on-chip SRAM. It can be interfaced with external sensors and other devices through the GPIOs. Software Development Kit (SDK) provides sample codes for various applications.

Espressif Systems' Smart Connectivity Platform (ESCP) enables sophisticated features including fast switch between sleep and wakeup mode for energy-efficient purpose, adaptive radio biasing for low-power operation, advance signal processing, spur cancellation and radio co-existence mechanisms for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation.

## 1.1. Wi-Fi Protocols

- 802.11 b/g/n support
- 2 x Wi-Fi interface, supports infrastructure BSS Station mode / P2P mode / SoftAP mode support
- Hardware accelerators for CCMP (CBC-MAC, counter mode), TKIP (MIC, RC4), WAPI (SMS4), WEP (RC4), CRC
- 802.11n support (2.4 GHz)
- Supports MIMO 1×1 and 2×1, STBC, and 0.4 μs guard interval
- WMM
- UMA compliant and certified
- Antenna diversity and selection (software managed hardware)
- Configurable packet traffic arbitration (PTA) with dedicated slave processor based design provides flexible and exact timing Bluetooth co-existence support for a wide range of Bluetooth Chip vendor.



- Dual and single antenna Bluetooth co-existence support with optional simultaneous receive (Wi-Fi/Bluetooth) capability

## 1.2. Specifications

Table 1-1. Specifications

Categories	Items	Parameters
Wi-Fi	Certification	Wi-Fi Alliance
	Protocols	802.11 b/g/n
	Frequency Range	2.4G ~ 2.5G (2400M ~ 2483.5M)
	Tx Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: -91 dbm (11 Mbps)
802.11 g: -75 dbm (54 Mbps)		
802.11 n: -72 dbm (MCS7)		
Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip	
Hardware	CPU	Tensilica L106 32-bit processor
	Peripheral Interface	UART/SDIO/SPI/I2C/I2S/IR Remote Control
		GPIO/ADC/PWM/LED Light & Button
	Operating Voltage	2.5V ~ 3.6V
	Operating Current	Average value: 80 mA
	Operating Temperature Range	-40°C ~ 125°C
	Storage Temperature Range	-40°C ~ 125°C
	Package Size	QFN32-pin (5 mm x 5 mm)
External Interface	-	
Software	Wi-Fi Mode	Station/SoftAP/SoftAP+Station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Software Development	Supports Cloud Server Development / Firmware and SDK for fast on-chip programming
	Network Protocols	IPv4, TCP/UDP/HTTP/FTP
	User Configuration	AT Instruction Set, Cloud Server, Android/iOS App



## 1.3. Applications

- Home appliances
- Home automation
- Smart plugs and lights
- Mesh network
- Industrial wireless control
- Baby monitors
- IP cameras
- Sensor networks
- Wearable electronics
- Wi-Fi location-aware devices
- Security ID tags
- Wi-Fi position system beacons



# 2. Pin Definitions

Figure 2-1 shows the pin layout for 32-pin QFN package.

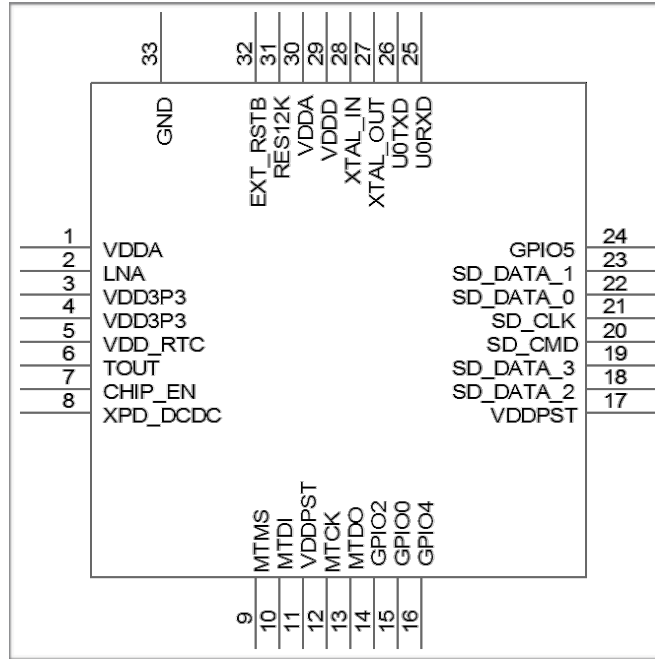


Figure 2-1. Pin Layout

Table 2-1 lists the definitions and functions of each pin.

Table 2-1. ESP8266EX Pin Definitions

Pin	Name	Type	Function
1	VDDA	P	Analog Power 2.5V ~ 3.6V
2	LNA	I/O	RF antenna interface Chip output impedance=39+j6 Ω. It is suggested to retain the π-type matching network to match the antenna.
3	VDD3P3	P	Amplifier Power 2.5V ~ 3.6V
4	VDD3P3	P	Amplifier Power 2.5V ~ 3.6V
5	VDD_RTC	P	NC (1.1V)
6	TOUT	I	ADC pin. It can be used to test the power-supply voltage of VDD3P3 (Pin3 and Pin4) and the input power voltage of TOUT (Pin 6). However, these two functions cannot be used simultaneously.
7	CHIP_PU	I	Chip Enable High: On, chip works properly Low: Off, small current consumed





Pin	Name	Type	Function
8	XPD_DCDC	I/O	Deep-sleep wakeup (need to be connected to EXT_RSTB); GPIO16
9	MTMS	I/O	GPIO 14; HSPI_CLK
10	MTDI	I/O	GPIO 12; HSPI_MISO
11	VDDPST	P	Digital/IO Power Supply (1.8V ~ 3.6V)
12	MTCK	I/O	GPIO 13; HSPI_MOSI; UART0_CTS
13	MTDO	I/O	GPIO 15; HSPI_CS; UART0_RTS
14	GPIO2	I/O	UART Tx during flash programming; GPIO2
15	GPIO0	I/O	GPIO0; SPI_CS2
16	GPIO4	I/O	GPIO4
17	VDDPST	P	Digital/IO Power Supply (1.8V ~ 3.6V)
18	SDIO_DATA_2	I/O	Connect to SD_D2 (Series R: 200Ω); SPIHD; HSPIHD; GPIO9
19	SDIO_DATA_3	I/O	Connect to SD_D3 (Series R: 200Ω); SPIWP; HSPIWP; GPIO10
20	SDIO_CMD	I/O	Connect to SD_CMD (Series R: 200Ω); SPI_CS0; GPIO11
21	SDIO_CLK	I/O	Connect to SD_CLK (Series R: 200Ω); SPI_CLK; GPIO6
22	SDIO_DATA_0	I/O	Connect to SD_D0 (Series R: 200Ω); SPI_MISO; GPIO7
23	SDIO_DATA_1	I/O	Connect to SD_D1 (Series R: 200Ω); SPI_MOSI; GPIO8
24	GPIO5	I/O	GPIO5
25	U0RXD	I/O	UART Rx during flash programming; GPIO3
26	U0TXD	I/O	UART Tx during flash programming; GPIO1; SPI_CS1
27	XTAL_OUT	I/O	Connect to crystal oscillator output, can be used to provide BT clock input
28	XTAL_IN	I/O	Connect to crystal oscillator input
29	VDDD	P	Analog Power 2.5V ~ 3.6V
30	VDDA	P	Analog Power 2.5V ~ 3.6V
31	RES12K	I	Serial connection with a 12 kΩ resistor and connect to the ground
32	EXT_RSTB	I	External reset signal (Low voltage level: active)

**Note:**

*GPIO2, GPIO0, and MTDO are configurable on PCB as the 3-bit strapping register that determines the booting mode and the SDIO timing mode.*

# PRODUCT SPECIFICATION

<b>MODEL: TS832</b>						
<b>5.8G 600mW 32 Channels AV Transmitter Module</b>						
	<b>Modulate</b>		Wideband FM Modulate			
	<b>Video Format</b>		NTSC/PAL			
	<b>Characteristics</b>		<b>Value</b>			<b>Units</b>
			<b>Min.</b>	<b>Typ.</b>	<b>Max.</b>	
<b>1</b>	<b>Output Impedance</b>		---	50	---	Ohm
<b>2</b>	<b>Output Power</b>		27	27.5	28	dBm
<b>3</b>	<b>Frequency range</b>	<b>Channel Frequency</b>				
	<b>FR1 or (A)</b>	<b>CH1</b>	---	5865	---	MHz
		<b>CH2</b>	---	5845	---	MHz
		<b>CH3</b>	---	5825	---	MHz
		<b>CH4</b>	---	5805	---	MHz
		<b>CH5</b>	---	5785	---	MHz
		<b>CH6</b>	---	5765	---	MHz
		<b>CH7</b>	---	5745	---	MHz
		<b>CH8</b>	---	5725	---	MHz
	<b>FR2 or (B)</b>	<b>CH1</b>	---	5733	---	MHz
		<b>CH2</b>	---	5752	---	MHz
		<b>CH3</b>	---	5771	---	MHz
		<b>CH4</b>	---	5790	---	MHz
		<b>CH5</b>	---	5809	---	MHz
		<b>CH6</b>	---	5828	---	MHz
		<b>CH7</b>	---	5847	---	MHz
		<b>CH8</b>	---	5866	---	MHz

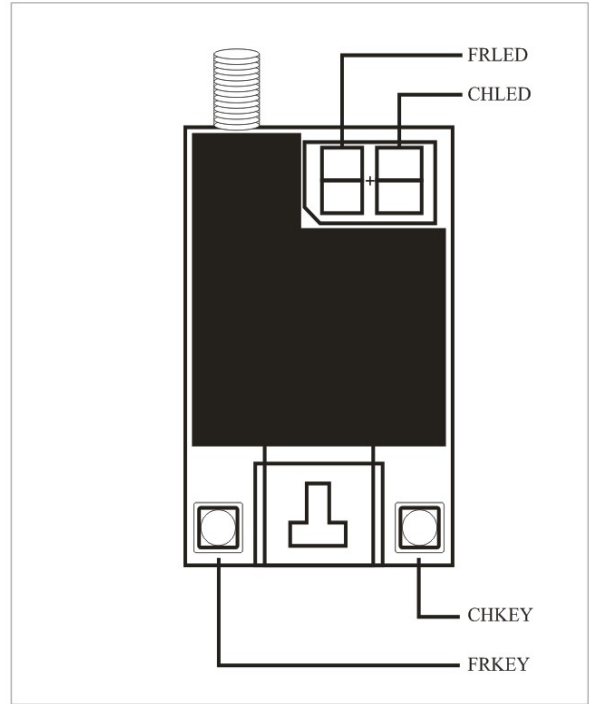
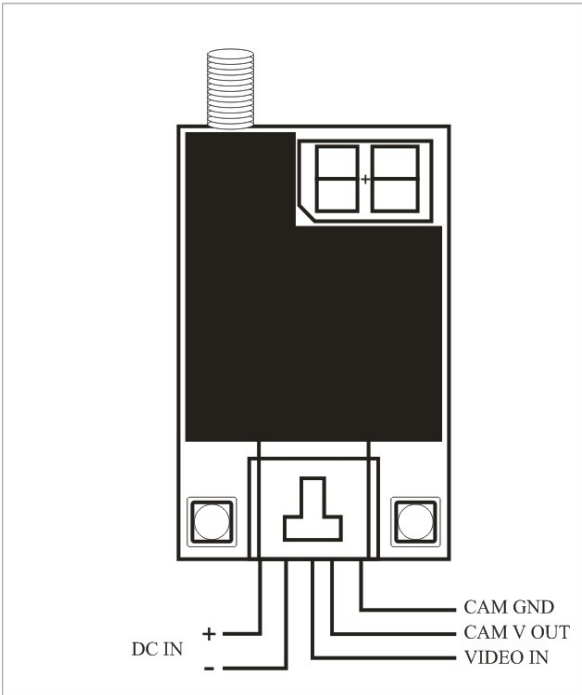
<b>3</b>	<b>FR3 or (E)</b>	<b>CH1</b>	---	5707	---	MHz
		<b>CH2</b>	---	5685	---	MHz
		<b>CH3</b>	---	5665	---	MHz
		<b>CH4</b>	---	5645	---	MHz
		<b>CH5</b>	---	5885	---	MHz
		<b>CH6</b>	---	5905	---	MHz
		<b>CH7</b>	---	5925	---	MHz
		<b>CH8</b>	---	5945	---	MHz
	<b>FR4 or (F)</b>	<b>CH1</b>	---	5740	---	MHz
		<b>CH2</b>	---	5760	---	MHz
		<b>CH3</b>	---	5780	---	MHz
		<b>CH4</b>	---	5800	---	MHz
		<b>CH5</b>	---	5820	---	MHz
		<b>CH6</b>	---	5840	---	MHz
		<b>CH7</b>	---	5860	---	MHz
		<b>CH8</b>	---	5880	---	MHz
<b>4</b>	<b>Operating Voltage</b>	7.0	12	16	V	
<b>5</b>	<b>Supply current</b>	---	220	---	mA	
<b>6</b>	<b>Operating Temperature</b>	-10	---	+85	°C	
<b>7</b>	<b>Video Band Width</b>	0	---	8.0	MHz	
<b>8</b>	<b>Audio carrier Frequency</b>	---	6.5	---	MHz	
<b>9</b>	<b>Video Input Level</b>	0.8	1.0	1.2	Vp-p	
<b>10</b>	<b>Video Input Impedance</b>	---	75	---	Ohm	
<b>11</b>	<b>Audio Input Level</b>	0.5	---	2.0	Vp-p	
<b>12</b>	<b>Audio Input Impedance</b>	---	10K	---	Ohm	
<b>13</b>	<b>Weight</b>	---	22	---	g	
<b>14</b>	<b>Antenna Connector</b>	<b>SMA Connector</b>				
<b>15</b>	<b>Dimensions (L x W x H)</b>	54x32x10mm				

**Picture**



**Connect:**

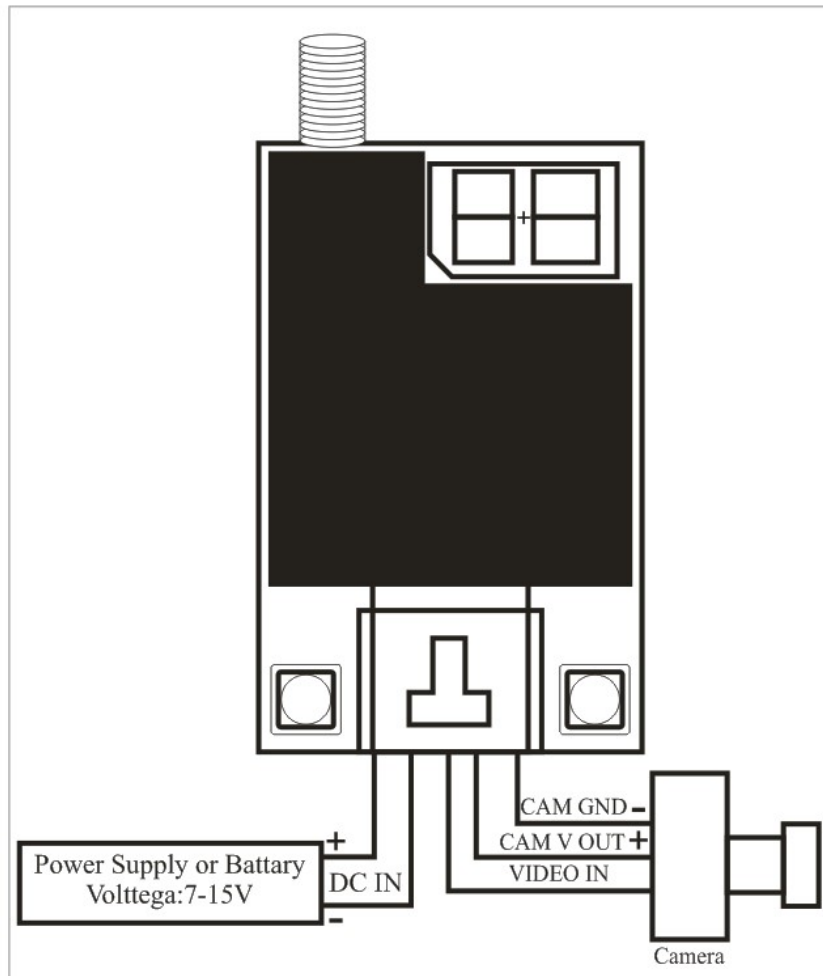
**Channel Switch**



**Frequency and channel frequency table**

FR \ CH		CH							
		CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8
FR	FR1	5865M	5845M	5825M	5805M	5785M	5765M	5745M	5725M
	FR2	5733M	5752M	5771M	5790M	5809M	5828M	5847M	5866M
	FR3	5705M	5685M	5665M	5645M	5885M	5905M	5925M	5945M
	FR4	5740M	5760M	5780M	5800M	5820M	5840M	5860M	5880M

## Application examples

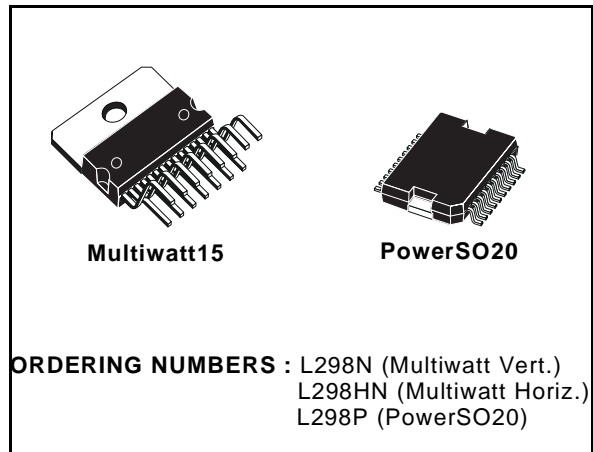


## DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

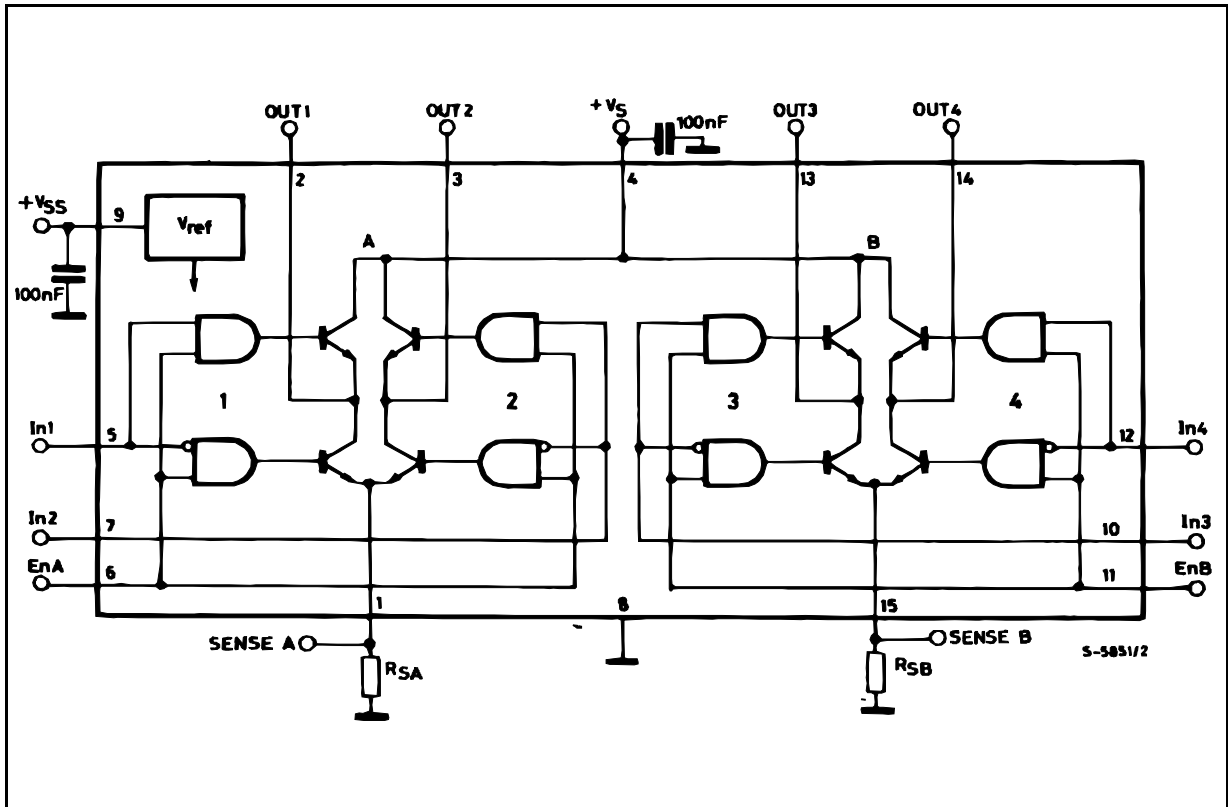
### DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

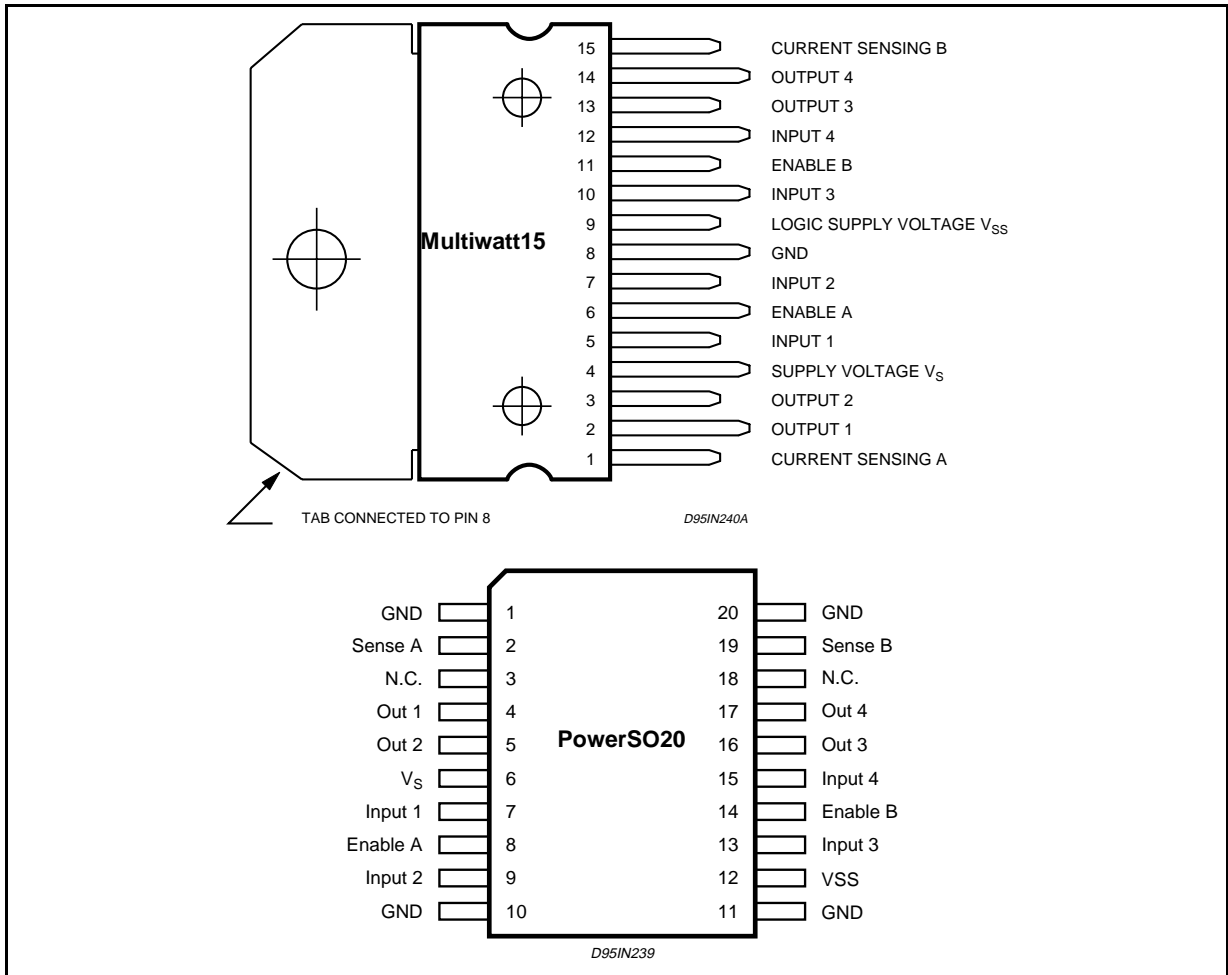
### BLOCK DIAGRAM



**ABSOLUTE MAXIMUM RATINGS**

Symbol	Parameter	Value	Unit
$V_S$	Power Supply	50	V
$V_{SS}$	Logic Supply Voltage	7	V
$V_I, V_{En}$	Input and Enable Voltage	-0.3 to 7	V
$I_O$	Peak Output Current (each Channel)		
	- Non Repetitive ( $t = 100\mu s$ )	3	A
	- Repetitive (80% on -20% off; $t_{on} = 10ms$ )	2.5	A
	-DC Operation	2	A
$V_{sens}$	Sensing Voltage	-1 to 2.3	V
$P_{tot}$	Total Power Dissipation ( $T_{case} = 75^\circ C$ )	25	W
$T_{op}$	Junction Operating Temperature	-25 to 130	$^\circ C$
$T_{stg}, T_j$	Storage and Junction Temperature	-40 to 150	$^\circ C$

**PIN CONNECTIONS (top view)**



**THERMAL DATA**

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th j-case}$	Thermal Resistance Junction-case	Max.	-	3	$^\circ C/W$
$R_{th j-amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	$^\circ C/W$

(\*) Mounted on aluminum substrate



# LM78XX/LM78XXA

## 3-Terminal 1A Positive Voltage Regulator

### Features

- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

### General Description

The LM78XX series of three terminal positive regulators are available in the TO-220 package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.

### Ordering Information

Product Number	Output Voltage Tolerance	Package	Operating Temperature
LM7805CT	±4%	TO-220	-40°C to +125°C
LM7806CT			
LM7808CT			
LM7809CT			
LM7810CT			
LM7812CT			
LM7815CT			
LM7818CT			
LM7824CT			
LM7805ACT	±2%		0°C to +125°C
LM7806ACT			
LM7808ACT			
LM7809ACT			
LM7810ACT			
LM7812ACT			
LM7815ACT			
LM7818ACT			
LM7824ACT			



## Block Diagram

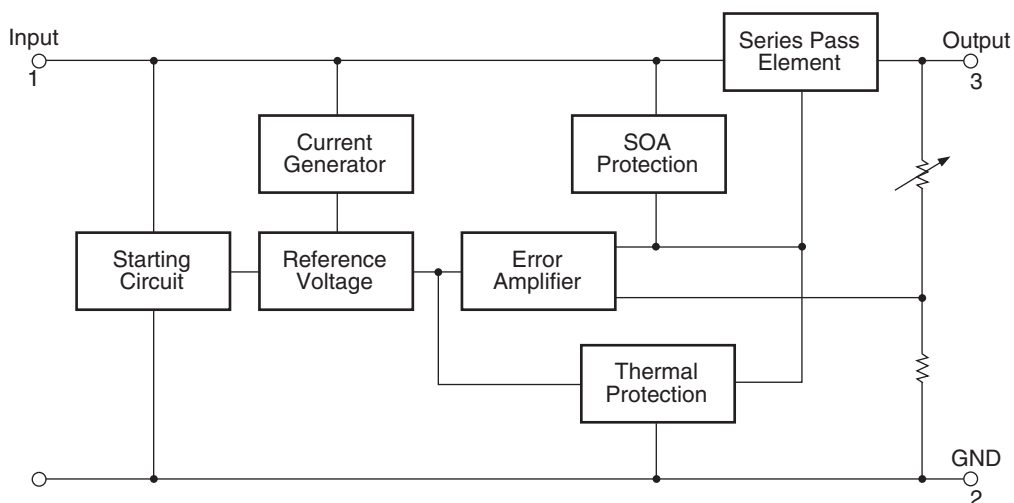


Figure 1.

## Pin Assignment

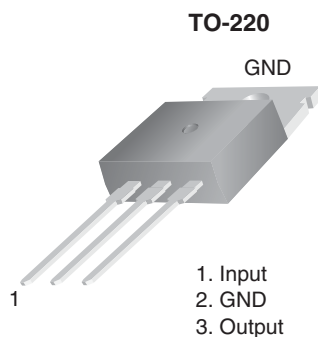


Figure 2.

## Absolute Maximum Ratings

Absolute maximum ratings are those values beyond which damage to the device may occur. The datasheet specifications should be met, without exception, to ensure that the system design is reliable over its power supply, temperature, and output/input loading variables. Fairchild does not recommend operation outside datasheet specifications.

Symbol	Parameter	Value	Unit	
$V_I$	Input Voltage	$V_O = 5V$ to $18V$	35	V
		$V_O = 24V$	40	V
$R_{\theta JC}$	Thermal Resistance Junction-Cases (TO-220)	5	$^{\circ}C/W$	
$R_{\theta JA}$	Thermal Resistance Junction-Air (TO-220)	65	$^{\circ}C/W$	
$T_{OPR}$	Operating Temperature Range	LM78xx	-40 to +125	$^{\circ}C$
		LM78xxA	0 to +125	
$T_{STG}$	Storage Temperature Range	-65 to +150	$^{\circ}C$	

Typical Performance Characteristics

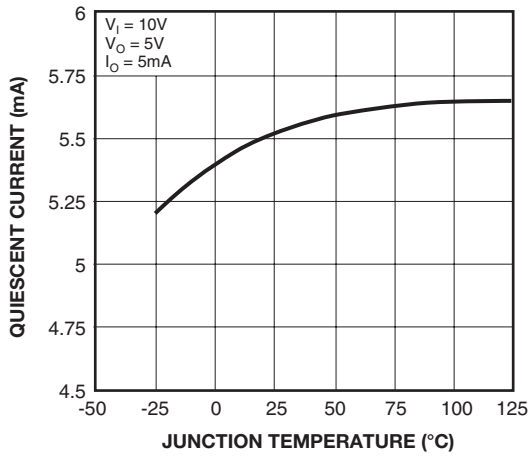


Figure 3. Quiescent Current

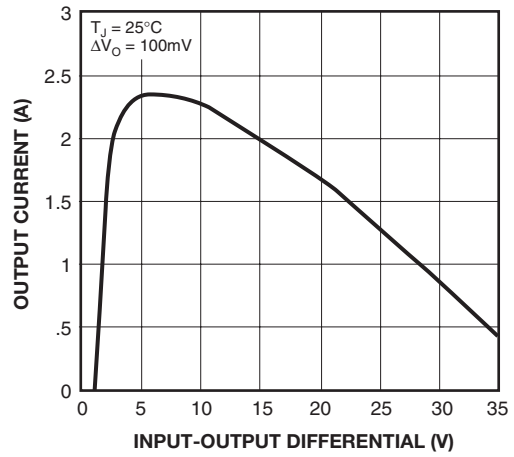


Figure 4. Peak Output Current

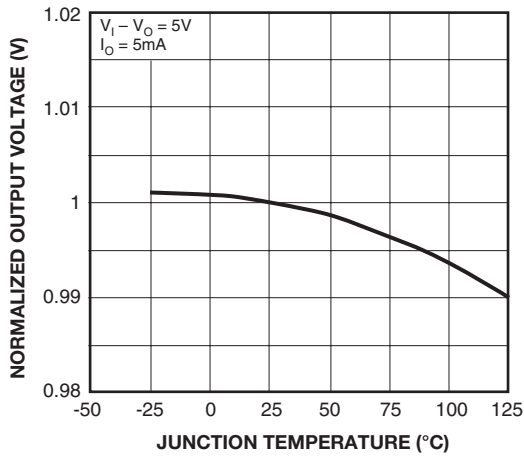


Figure 5. Output Voltage

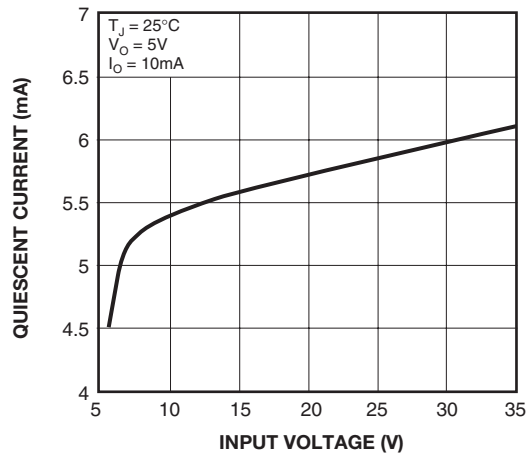


Figure 6. Quiescent Current

## Typical Applications

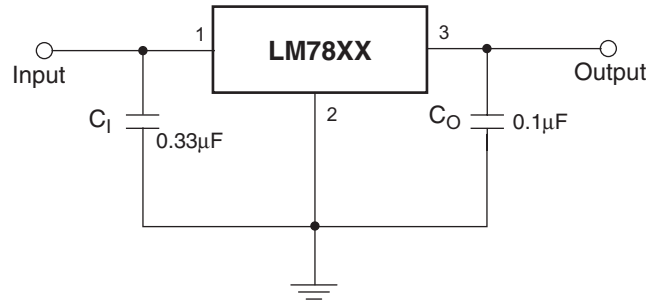


Figure 7. DC Parameters

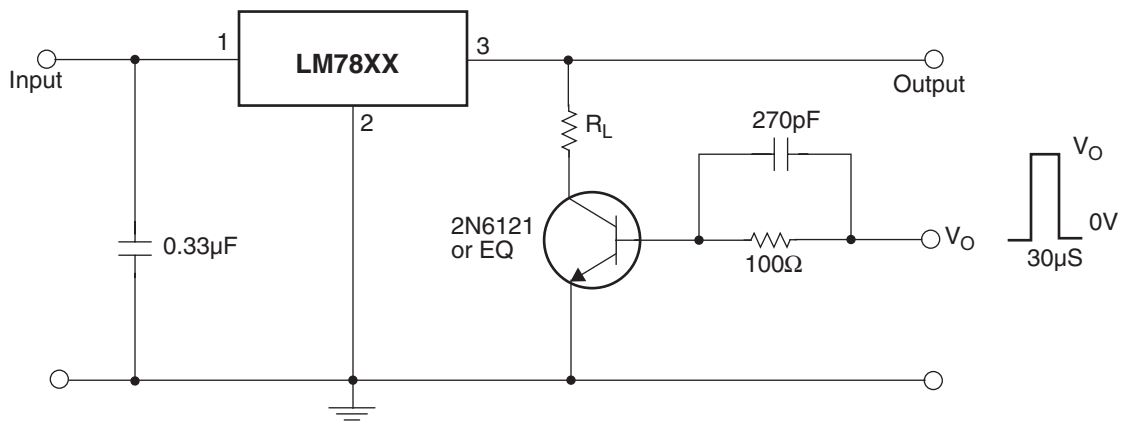


Figure 8. Load Regulation

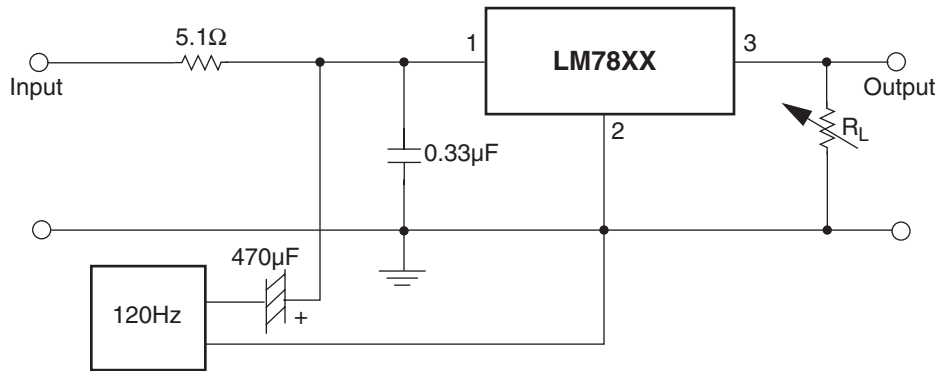


Figure 9. Ripple Rejection

## MANUAL DE INSTRUCCIONES DE MOBIUS ACTIONCAM



La imagen superior muestra una descripción de las distintas funciones.

Los dos disipadores de calor pueden llegar a calentarse mucho durante la operación. Esto es normal, pero así todo nunca debe dejarse la cámara encendida cuando la temperatura exterior excede de 47 °C (por ejemplo, en un coche aparcado a pleno sol). Si deseas usar la cámara en condiciones de mucho calor, deberías asegurarte de que los disipadores están refrigerados adecuadamente, por ejemplo con aire frío directo a los disipadores, o usando un ventilador para disipar el calor de las placas metálicas. Las altas temperaturas también podrían dañar la batería LiPo. Si usas la cámara en condiciones de mucho calor con alimentación externa, por ejemplo sobre el salpicadero de un coche, habría que considerar sustituir la batería con un supercondensador. Éste se puede comprar a la mayoría de suministradores que venden la Mobius. Un supercondensador permite a la cámara guardar correctamente el final de las grabaciones al ser desconectada de la fuente exterior. Un supercondensador completamente cargado tiene suficiente energía para mantener el RTC (reloj interno) en funcionamiento durante varios días sin corriente externa. Sólo lleva unos cuantos minutos cargar completamente el supercondensador.

► Este manual no cubre la sustitución de la batería interna o de la lente. Si necesitas hacer ésto, acude por favor a las instrucciones detalladas que puedes encontrar en <http://www.rcgroups.com/forums/showpost.php?p=25170910&postcount=4>.

► Este manual asume que la cámara lleva el último firmware disponible a la fecha de su escritura. La funcionalidad puede variar con versiones anteriores o posteriores.

## Tarjetas

A día de hoy, es posible usar con la cámara cualquier tarjeta micro SD, micro SDHC o micro SDXC con una capacidad de hasta 64Gb. Es muy probable que tarjetas de mayor capacidad también sean soportadas, pero aún no están disponibles. Para evitar problemas de compatibilidad con las tarjetas, usa siempre tarjetas de calidad con una velocidad de clase 4 o superior. La cámara soporta tarjetas formateadas con FAT, FAT32 o exFAT. Debido a razones técnicas, no se puede actualizar el firmware si tu tarjeta ha sido formateada con exFAT. Cuando se usa la función de formateo desde la propia cámara (ver más abajo) las tarjetas de capacidad superior a 32Gb se formatean con exFAT. El resto de tarjetas se formatearán como FAT/FAT32. Los sistemas operativos más antiguos, como XP-64 bit y Windows 2003, no soportan el sistema de archivos exFAT a no ser que se instalen controladores adicionales. A continuación puede encontrarse una lista de controladores exFAT: <http://search.microsoft.com/enus/DownloadResults.aspx?q=KB955704>.

► Deberían usarse siempre tarjetas de marcas reconocidas. Tarjetas de fabricantes no reconocidos y de baja calidad, así como algunas de clase 2 (o menos), pueden consumir más de 20mA que las tarjetas de calidad. Esto puede llevar a ruidos indeseados en la grabación de audio.

## Cables

La cámara requiere de un cable USB estándar para cargarse y conectarse a la corriente externa para la grabación continua. Un cable USB estándar se usa también para conectar la cámara al PC para configurar los parámetros o acceder a la cámara como disco duro externo o cámara web.

La salida de audio/vídeo requiere un cable especial en el que el pin #4 está cortocircuitado con el pin #5 (masa). Cuando el firmware detecta el pin #4 conectado a masa pondrá la señal de vídeo en el pin #2 y la señal de audio en el pin #3. Se pueden pedir cables especiales en forma de arnés o los puedes hacer tú mismo usando un conector mini USB de 5 pins. También se puede modificar un cable OTG (On The Go) de un smartphone que ya tiene los pins #4 y #5 cortocircuitados en el conector mini USB. La señal de audio está únicamente disponible en modo reproducción. El cable “especial” de la cámara #16 no funciona con esta cámara. No se debe utilizar bajo ninguna circunstancia ningún otro cable “especial”, en particular el cable “especial” suministrado con la cámara #11. Usar este cable destruirá la cámara sin posibilidad de reparación. Así que no uses un cable “especial” de la cámara #11 bajo ninguna circunstancia.

## Salida de vídeo

El cable opcional de salida de vídeo (ver abajo) tiene que conectarse siempre con la cámara en espera o apagada. Si el cable se conecta o desconecta durante la grabación, la grabación se parará y la cámara entrará en modo en espera. La cámara nunca será reconocida como disco duro externo cuando el cable de vídeo está conectado.

## Antes de empezar

Antes de usar la cámara por primera vez, asegúrate de que está completamente cargada. Consulta el apartado “Carga”. Una batería nueva, totalmente cargada, dura unos 80 minutos.

Para grabar o reproducir contenido, toma como referencia la imagen de arriba e inserta en la ranura de la tarjeta SD, una tarjeta micro SD boca abajo con los contactos hacia delante.

## Botón de encendido

Presiona el botón de encendido y enciende la cámara. El LED azul se enciende durante 1 segundo antes de que el firmware comience a cargarse. Después de 3 segundos el LED amarillo se encenderá indicando que la cámara está preparada para su uso (modo en espera). Si la característica “Power On Delay” está activada” presiona el botón de encendido durante 4 segundos hasta que se encienda el LED amarillo.

- La característica “Power On Delay” (encendido retardado) es útil para prevenir el encendido inadvertido de la cámara.
- Si la cámara está configurada con “Auto Power” en “On” y “Record Set” en “On”, la cámara comenzará automáticamente a grabar en Modo 1.
- Para apagar la cámara, primero para cualquier grabación presionando el botón de disparo y luego presiona el botón de encendido durante al menos medio segundo (algunas versiones de firmware requieren presionar el botón durante 2 segundos). La cámara se apagará.

## Botón de modo

El botón de modo permite elegir entre 4 modos diferente. Cada pulsación del botón avanza al modo siguiente indicado por el color del LED.

- Modo de vídeo 1 (LED amarillo). Se puede configurar individualmente los cuadros por segundo, la resolución, AOV y rotación.
- Modo de vídeo 2 (LED azul). Se puede configurar individualmente los cuadros por segundo, la resolución, AOV y rotación.
- Modo foto (LED rojo) que incluye fotos en timelapse.
- Modo reproducción (LED rojo) sólo funciona cuando se conecta el cable especial de salida audio/vídeo.

## Botón de disparo

Presiona el botón de disparo para comenzar o terminar una grabación de vídeo, hacer una fotografía o iniciar/parar un archivo estando en modo reproducción. El modo reproducción está disponible únicamente cuando el cable de salida audio/vídeo está conectado.

## Indicadores LED de estado

Debido a la variedad de funciones de la cámara, es importante entender el significado de los LEDs. Los siguientes estados LED son para una cámara no conectada al PC.

LED amarillo continuo:

- La cámara está en modo de vídeo 1 y está en espera.

LED amarillo parpadeando (1 segundo encendido, 1 segundo apagado)

- La cámara está en modo de vídeo 1 y está grabando vídeo. El indicador “Video Recording” (grabación de vídeo) debe estar activado en la configuración.

LED azul continuo:

- La cámara está en modo de vídeo 2 y está grabando vídeo. El indicador “Video Recording” (grabación de vídeo) debe estar activada en la configuración.

LED rojo continuo:

- La cámara está en modo foto o modo reproducción (si un cable audio/vídeo está conectado) y está en espera.

LED rojo parpadea una vez:

- La cámara ha tomado una fotografía.

LED verde continuo:

- La batería está cargando.

## LED posterior

Cuando la cámara está encendida, el LED posterior parpadeará de una a tres veces para indicar la carga de la batería. Tres parpadeos indican batería llena, mientras que un parpadeo indica que la carga de la batería es baja. Los parpadeos son rápidos. Parpadeos más lentos extenderían innecesariamente el tiempo para que la cámara entrase en modo de espera. Si la cámara está configurada para mostrar el indicador de grabación (“Recording Indicator”), entonces el LED posterior parpadeará en secuencia con el LED principal durante la grabación. El LED posterior no parpadeará cuando se ha configurado el modo fotografía para tomar fotos en timelapse con un intervalo de tiempo inferior a 2 segundos.

## Conectando la cámara al PC

Asegúrate de que la cámara esté apagada si quieres acceder a ella como un dispositivo externo. Conecta siempre directamente la cámara al PC. No uses un hub USB.

## Carga

- Conecta un extremo del cable USB suministrado a la cámara.
- Conecta el otro extremo a una fuente de 5V DC (cargador de pared, PC o batería externa).
- Cuando el LED amarillo se encienda, presiona brevemente el botón de encendido para apagar la cámara.

NOTA: La carga cuando se conecta a puerto USB de un PC puede ser activada o desactivada usando el programa de configuración. Cuando se conecta a un smartphone, deberías apagar esta funcionalidad.

Durante la carga, el LED verde estará encendido, y se apagará cuando la batería esté completamente cargada.

La carga dura unas 2,5 horas para una batería completamente descargada con la cámara apagada y conectada a un cargador USB externo. Las baterías no están cubiertas por la garantía. La carga está limitada a unos 140mA, lo que ofrece la mejor protección a largo plazo para la batería. Si se carga por debajo de la corriente nominal o usando un hub USB sin fuente externa de alimentación, la carga tardará más tiempo. Para conseguir resultados óptimos, el voltaje de carga debería ser de +5V. Usa sólo cargadores de calidad capaces de suministrar 1000mA (1A) y un voltaje estable de al menos 5V. Los cargadores de baja calidad pueden no ser capaces de suministrar suficiente intensidad y puede provocar un fallo prematuro de la batería o, en el peor de los casos, destruir incluso la cámara.

## **Formateo**

Las aplicaciones de formateo suministradas por el sistema operativo no son siempre de confianza. Además no optimizan el rendimiento de las tarjetas SD. Para un rendimiento óptimo de la tarjeta de memoria se recomienda usar siempre el formateo de la propia cámara. Las instrucciones para usar esta herramienta se encuentran en el punto “Formateo de cámara”. Otra excelente herramienta de formateo es “SD Fomatter”, que se puede descargar en [www.sdcard.org/downloads/formatter\\_4](http://www.sdcard.org/downloads/formatter_4). Esta aplicación ha sido desarrollada específicamente para formatear tarjetas SD.

## **Apagado automático desde modo en espera**

Si la cámara está en modo de espera (Modo de vídeo 1, Modo de vídeo 2 o modo foto) debes hacer una foto o comenzar a grabar un vídeo en 30\* segundos. La cámara automáticamente se apagará tras 30 segundos de inactividad.

\*Este valor puede modificarse usando el programa de configuración. El apagado automático también puede ser desactivado.

Cuando el cable especial de audio/vídeo se conecta, el apagado automático se desactiva. Esto previene la pérdida de señal durante su uso como FPV cuando el tiempo de modo en espera termina.

## **Encendido de la cámara**

El encendido de la cámara puede ser configurado como “rápido” o “temporizado”. En el modo rápido la cámara se enciende inmediatamente al presionar el botón de encendido. En modo temporizado, presiona y mantén el botón de encendido durante 4-5 segundos hasta que el LED amarillo se encienda y permanezca encendido. Independientemente de si la cámara está configurada en modo rápido o temporizado, el LED azul se encenderá durante un segundo seguido por el LED amarillo. Cuando el LED amarillo permanece encendido, la cámara está preparada para su uso.



- Si se ha activado la función “Auto Power-On Record” (grabación en encendido), la cámara comenzará automáticamente a grabar en Modo 1 al presionar el botón de encendido o conectar una fuente de alimentación externa. Si el LED de grabación ha sido desactivado, el LED amarillo destella 3 veces y se apaga, indicando que la grabación ha comenzado. De lo contrario el LED amarillo destella lentamente a intervalos de dos segundos indicando que la cámara está grabando.
- Si en la cámara se ha desactivado la opción “Auto Record” (auto grabación), la cámara entrará en el modo de vídeo 1 (LED amarillo) y permanecerá en el modo de espera hasta que se presione un botón o termine el tiempo de la opción “Auto-Power-Off” (auto apagado).

## **Apagado de la cámara**

Presionar y mantenerle botón de encendido durante al menos medio segundo y soltarlo (algunas versiones de firmware requieren presionar el botón durante dos segundos). El LED rojo parpadeará tres veces y la cámara se apagará.

## **Cambio de modo**

La cámara puede cambiar entre diferentes modos presionando el botón de modo. Cada pulsación del botón cambia al siguiente modo. Modo vídeo 1 – Modo vídeo 2 – Modo foto – Modo reproducción (sólo si está conectado el cable audio/vídeo) – Modo vídeo 1, etc.

## **Grabar en modo vídeo 1 o 2 (LED amarillo o azul)**

### **Comenzar la grabación de vídeo y proteger contra escritura archivos importantes**

- Presionar brevemente el botón de disparo
  - ▶ Si el LED de grabación ha sido desactivado, el LED amarillo o azul parpadeará 3 veces y se apagará, indicando que la grabación ha comenzado.
  - ▶ Si el LED de grabación ha sido activado, el LED amarillo o azul comenzará a parpadear a intervalos de dos segundos indicando que la cámara está grabando.
- El archivo actual puede ser protegido contra la escritura presionando el botón de modo durante la grabación.

### **Detener la grabación**

- Presiona brevemente el botón de disparo. La cámara regresará al modo de espera.

## Haciendo una foto (LED rojo)

- Presiona brevemente el botón de disparo. El LED rojo parpadeará indicando que se ha hecho una fotografía.

## Modo de reproducción

- Asegurarse de haber conectado el cable de audio/vídeo entre la cámara y un monitor externo, y de que haya insertada una tarjeta.
- Encender la cámara si no lo estaba ya.
- Presionar el botón de modo hasta que se encienda el LED rojo (modo foto).
- Presionar el botón de modo una vez más. El LED rojo permanecerá encendido, pero la cámara estará en modo reproducción y el primer cuadro del primer archivo encontrado en la tarjeta se mostrará en el monitor.
- ► Si no hay archivos en la tarjeta se mostrará el mensaje “Empty” (vacío)
- Presionar el botón de disparo para reproducir o pausar el archivo.
- Presionar el botón de modo para pasar al siguiente archivo.
- Presionar el botón de encendido para salir del modo reproducción y apagar la cámara.
- ► No es posible salir del modo de reproducción si no es apagando la cámara.

## Modo disco extraíble

- Asegúrate de que una tarjeta de memoria está insertada y la cámara está apagada.
- Conecta la cámara al PC usando un cable USB estándar. El LED verde puede encenderse indicando que la batería está cargando.
- El LED azul se encenderá durante un segundo y entonces se encenderá el LED amarillo. Después de un tiempo el LED amarillo se apagará.
- La cámara está ahora en modo disco extraíble y un nuevo disco duro extraíble debería aparecer como disponible en tu ordenador. Esto puede tardar hasta 30 segundos.
- ► No es posible entrar en el modo de disco extraíble cuando se usa un cable de salida de vídeo (pins#4 y #5 puestos a tierra).

## Modo cámara web

Hay dos formas de entrar en el modo cámara web:

Método 1:

- Conectar la cámara como disco extraíble (ver punto anterior)
- Presionar el botón de disparo para entrar en modo cámara web.
- ► Si es la primera vez que se conecta la cámara como cámara web, Windows instalará automáticamente los drivers necesarios.

- ► Desconectar la cámara o presionar el botón de disparo para salir del modo cámara web.

Método 2:

- Quitar la tarjeta de memoria
- Conectar la cámara al PC usando un cable USB estándar.
- El LED amarillo parpadeará a intervalos de dos segundos indicando que no hay tarjeta insertada. Esto es normal.
- ► Si es la primera vez que se conecta la cámara como cámara web, Windows instalará automáticamente los drivers necesarios.
- La cámara está ahora en modo cámara web.
- Abre tu software favorito de cámara web, como AMCap. Si la pantalla está negra, el software no está configurado correctamente. Asegúrate de que el programa muestra “USB PC camera” como dispositivo seleccionado para la captura de vídeo. Si esta opción no se muestra en la lista de dispositivos, reinicia el software de la cámara web y compruébalo de nuevo.
- Desconecta la cámara o presiona el botón de disparo para salir del modo cámara web.

► Nota: Windows XP tiene integrado un visor de cámara web. La Mobius debería ser visible en el explorador (My Computer) como “USB Video Device”. Esta funcionalidad no existe en Windows 7.

## Formateo interno

La cámara tiene un formateador integrado que puede usarse para formatear la tarjeta SD. Esta función es mejor usarla utilizando mSetup.exe (ver más abajo), pero también puede hacerse como sigue:

- Conecta la cámara como disco extraíble (ver arriba)
  - Presionar y mantener presionado el botón de modo. NO SOLTAR el botón hasta completar los siguientes pasos.
  - Desconecta la cámara del ordenador.
  - El LED amarillo parpadeará unos cuantos segundos seguido de tres parpadeos del LED rojo, después de lo cual la cámara se apagará. Debes esperar hasta ver parpadear tres veces el LED rojo.
  - Soltar el botón de modo.
- Las tarjetas con una capacidad superior a 32Gb serán formateadas como exFAT. El resto de tarjetas serán formateadas como FAT32. Ver la nota a cerca de usar exFAT en sistemas operativos más antiguos en la descripción.

## Configuración de la cámara

Para modificar la configuración de la cámara en Windows, siempre se recomienda usar el programa mSetup.exe. Este programa se comunica directamente con la cámara y hace que la configuración sea una tarea muy sencilla.

Sin embargo, también es posible cambiar manualmente los parámetros de configuración editando el archivo “SYSCFG.TXT”, el cual debe ser colocado en el directorio raíz de la tarjeta, para que la cámara lo encuentre y cargue los nuevos parámetros. El archivo “SYSCFG.TXT” debe estar en formato ASCII, y no debe ser Unicode. Si no sabes lo que es un directorio raíz, o no sabes lo que es un archivo ASCII o editar manualmente este tipo de archivos, usa únicamente mSetup.exe.

Un archivo “SYSCFG.TXT” debe ser generado primero por la cámara. Los parámetros de la cámara son auto descriptivos. Es importante no cambiar la secuencia entre corchetes y no configurar combinaciones inválidas. Por ejemplo, nunca combinar un archivo más largo de 5 minutos con la opción de bucle continuo. Si introduces combinaciones o valores inválidos, la cámara no funcionará según lo esperado. Es mucho más seguro usar el programa de configuración ya que siempre configurará la cámara correctamente. Si deseas mantener la fecha y hora actual, sustituye los caracteres numéricos de los corchetes por ?.

El archivo “SYSCFG.TXT puede generarse automáticamente como sigue:

1. Desconecta la cámara del PC y apágala.
2. Presiona y mantén el botón de modo. **NO SUELTES el botón hasta el paso 4.**
3. Presiona y mantén el botón de encendido hasta que el LED rojo parpadee tres veces y la cámara se apague. Esto tardará unos 5 segundos. Durante el proceso, los LEDS azul y amarillo pueden parpadear.
4. Suelta el botón de modo y el de encendido.

El archivo “SYSCFG.TXT” debería estar ahora en el directorio raíz de la tarjeta de memoria. Ahora puedes editar el archivo según lo indicado arriba y cargar la nueva configuración en la cámara usando exactamente el mismo método descrito en los pasos del 1 al 4. Una vez completado, el archivo “SYSCFG.TXT” será automáticamente borrado del directorio raíz de la tarjeta.

## Usando el software para configurar la cámara.

mSetup.exe configura automáticamente la cámara con los parámetros que desees. Se pueden encontrar instrucciones detalladas en el programa pulsando el botón “Help” (ayuda).

## Actualización de firmware.

Normalmente, no es necesario actualizar el firmware. Únicamente debe ser actualizado si tienes problemas con el firmware actual o el nuevo firmware añade alguna funcionalidad que necesitas y que no está disponible en tu versión actual.

La última versión de firmware puede ser descargada e instalada directamente desde el programa de configuración o instalada manualmente descargando el firmware de <http://www.rcgroups.com/forums/showpost.php?p=25170908&postcount=3>.

La actualización del firmware es un proceso fácil y seguro, pero deberías tener en cuenta que no es posible actualizar el firmware usando tarjetas SD con una capacidad superior a 32Gb.

El firmware sólo puede actualizarse usando tarjetas SD formateadas con FAT32 (tarjetas de capacidad inferior a 64Gb). La actualización del firmware es una rutina de bajo nivel ejecutada antes de la carga del archivo de sistema exFAT.

#### **Actualización automática del firmware**

Usa mSetup (Tools) para actualizar automáticamente el firmware. Es con diferencia, el método más sencillo.

#### **Actualización manual del firmware**

**NO PRESIONES NINGÚN BOTÓN durante el proceso de actualización.**

Hay dos métodos para actualizar el firmware:

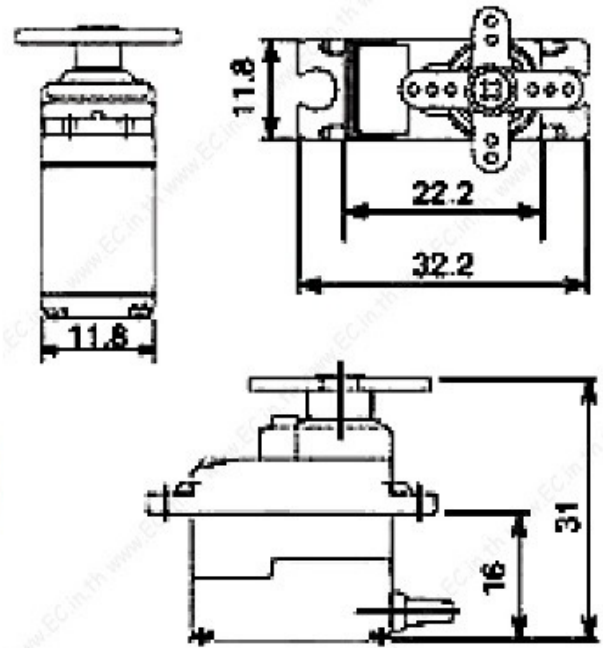
##### ***Método 1 sin pulsar ningún botón:***

- Apagar la cámara
- Conectar la cámara al ordenador y esperar hasta que la tarjeta sea reconocida como disco extraíble.
- Copiar el archivo del firmware en el directorio raíz de la tarjeta. Todos los archivos de firmware se llaman FWTLCAM.BIN. Este nombre no debe ser modificado.
- Desconectar la cámara del ordenador y esperar hasta que se apague.
- Esperar hasta que los LEDS se apaguen.
- Conectar la cámara al PC. El LED azul se encenderá durante unos 13 segundos y entonces comenzará a parpadear. ESPERA.
- Después de otros 7 segundos el LED azul se apagará. ESPERA.
- Después de un segundo o dos el LED amarillo se encenderá. ESPERA.
- El archivo del firmware será borrado automáticamente.
- La cámara automáticamente se conectará como disco extraíble.
- Tras 4 segundos el proceso se completará y el LED amarillo se apagará.
- La cámara está ahora preparada para usar como dispositivo USB.

##### ***Método 2 pulsando el botón de encendido:***

- Apagar la cámara
- Conectar la cámara al ordenador y esperar hasta que la tarjeta sea reconocida como disco extraíble.
- Copiar el archivo del firmware en el directorio raíz de la tarjeta. Todos los archivos de firmware se llaman FWTLCAM.BIN. Este nombre no debe ser modificado.
- Desconectar la cámara del ordenador y esperar hasta que se apague.
- Esperar hasta que los LEDS se apaguen.
- Enciende la cámara y ESPERA.
- El LED azul se encenderá durante unos 13 segundos y comenzará a parpadear. ESPERA.
- Después de otros 7 segundos el LED azul se apagará. ESPERA.
- Después de un segundo o dos el LED amarillo se encenderá indicando que el firmware ha sido automáticamente borrado.
- La cámara está ahora en modo de espera y lista para usar.

# SG90 9 g Micro Servo

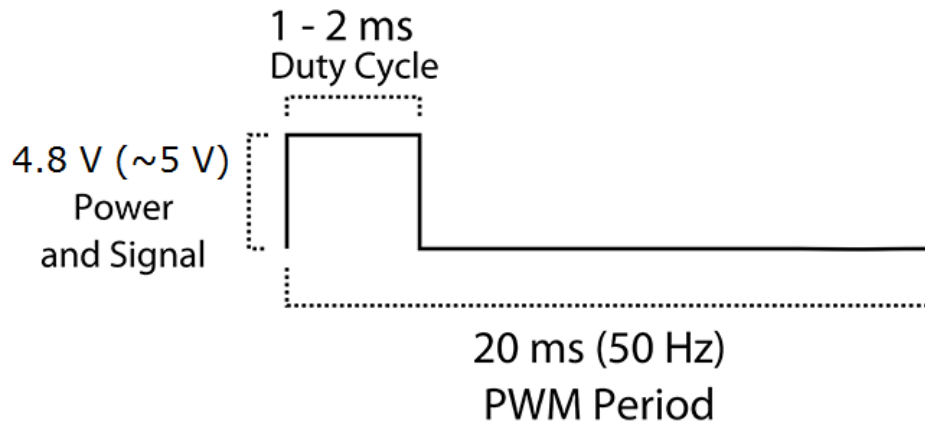
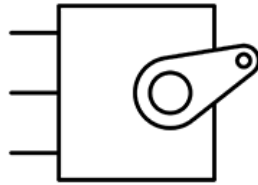


Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but *smaller*. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

## Specifications

- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf·cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10  $\mu$ s
- Temperature range: 0 °C – 55 °C

PWM=Orange (  $\square$  )  
Vcc = Red ( + )  
Ground=Brown ( - )



Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, "-90" (~1 ms pulse) is all the way to the left.



# Plagiarism Checker X Originality Report

**Similarity Found: 9%**

Date: miércoles, agosto 22, 2018

Statistics: 1238 words Plagiarized / 13256 Total words

Remarks: Low Plagiarism Detected - Your Document needs Optional Improvement.

---

RESUMEN En el presente proyecto se desarrolla un vehículo a control remoto con visión en primera persona, lo cual le da al usuario la sensación de estar manejándolo de forma directa ya que la imagen se proyecta en tiempo real. El prototipo está enfocado a ser un vehículo no tripulado, comandado mediante una aplicación instalada sobre la plataforma Android con tecnología WIFI para controlar el movimiento de giro de las llantas del vehículo en todas las direcciones, además de manipular la rotación de la cámara a través de servomotores en sentido horizontal y vertical.

La cámara forma parte de un sistema FPV (Visión en Primera Persona), que está conectada a un transmisor, permitiendo así enviar en tiempo real todas las imágenes que estén en el entorno del vehículo y sean captadas por el lente del dispositivo de video; esta información es transmitida a unas gafas que tienen incorporado un receptor que se encuentre en el mismo canal de comunicación logrando que el usuario tenga una imagen del ambiente circundante del vehículo.

Palabras claves: FPV, NODEMCU, ESP8266, servomotor, App Inventor2, LiPo



*Handwritten signature and date: 22/08/2018*



## DECLARACIÓN Y AUTORIZACIÓN

Yo, Juan Andrés Solís Poveda, CI 1720235876 autor del trabajo de graduación:

**DESARROLLO DE UN VEHÍCULO A CONTROL REMOTO CON VISIÓN EN PRIMERA PERSONA.**, previo a la obtención del título de **INGENIERO EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES** en la UNIVERSIDAD TECNOLÓGICA ISRAEL.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de difundir el respectivo trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de graduación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Quito, 18 de septiembre del 2018

Atentamente.



**Juan Andrés Solís Poveda**  
C.I. 1720235876