

# **INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ**

## **INGENIERÍA ELECTRÓNICA**

### **TEMA:**

**DISEÑO DE UN VEHÍCULO NO TRIPULADO TIPO  
CONVERTIBLE CON LA CAPACIDAD DE MONTAR PAREDES**

### **PRESENTA:**

**MINGO RUBIO JOSÉ JUAN  
VELASCO ÁLVAREZ JOSÉ CARLOS**

### **ASESOR:**

**DR. FRANCISCO RONAY LÓPEZ ESTRADA**

**TUXTLA GUTIÉRREZ CHIAPAS, ENERO 2017**

# Índice general

Introducción. . . . .	4
Justificación. . . . .	4
Objetivos. . . . .	5
General. . . . .	5
Específicos. . . . .	5
<b>Capítulo I: Antecedentes.</b>	<b>6</b>
1.1 Instituto Tecnológico de Tuxtla Gutiérrez. . . . .	6
1.2 Localización. . . . .	7
1.3 misión y visión. . . . .	8
1.3.1 Misión. . . . .	8
1.3.2 Visión. . . . .	8
<b>Capítulo II: Metodología.</b>	<b>9</b>
2.1 Estado del arte . . . . .	9
2.2 Estimación de posición y movimiento angular en el espacio 3D . . . . .	11
2.3 Diseño del sistema de control. . . . .	11
2.4 Relación entre la producción de empuje y el peso del vehículo. . . . .	12
2.5 Implementación de los algoritmos de estimación de orientación y del sistema de control en el vehículo. . . . .	13
<b>Capítulo III: Partes del vehículo y construcción.</b>	<b>14</b>
3.1 Diseño de las piezas del vehículo implementando solidworks. . . . .	14
3.2 Descripción y función de los componentes. . . . .	21
3.2.1 radiocontrol. . . . .	21
3.2.2 Receptor de radiocontrol. . . . .	23

3.2.2 Arduino nano. . . . .	24
3.2.3 Raspberry pi 3. . . . .	25
3.2.4 ESC(circuito controlador de velocidad). . . . .	26
3.2.5 Motor brushless. . . . .	27
3.2.6 Servomotor. . . . .	28
3.2.7 Batería y BEC. . . . .	30
3.3 Diagrama a bloques del sistema y vehículo ensamblado. . . . .	30
Conclusiones. . . . .	31
Anexos. . . . .	32

# Índice de figuras

1.	Instalaciones del edificio Z(izquierda),laboratorio de automatización y control(derecha) . . . . .	7
2.	Ubicación del Instituto Tecnológico de Tuxtla Gutiérrez . . . . .	7
3.	Robot VertiGo . . . . .	10
4.	B-Flying Car . . . . .	11
5.	EMISORA FLYSKY FS-TH9X-B 2.4Ghz. 9Canales . . . . .	21
6.	Receptor FrSky V8FR-II 2.4Ghz de 8 canales (Versión HV) . . . . .	23
7.	Diagrama esquemático de un Arduino nano . . . . .	24
8.	Diagrama esquemático de una Raspberry pi 3 . . . . .	25
9.	ESC BLHeli 30 A, OPTO . . . . .	26
10.	Especificaciones ESC BLHeli 30 A, OPTO . . . . .	26
11.	Motor brushless T-Motor Air Gear 2213/920kv . . . . .	27
12.	Eficiencia del T-Motor Air Gear 2213/920kv . . . . .	27
13.	Servomotor Tower Pro SG90 . . . . .	28
14.	Servomotor Tower Pro MG996R . . . . .	29
15.	diagrama a bloques del funcionamiento del sistema . . . . .	30
16.	Prototipo ensamblado . . . . .	31

## **Introducción.**

Las aplicaciones de los vehículos inteligentes se han ido diversificando de acuerdo a las necesidades que se presentan actualmente, es por ello que en este trabajo se presenta el diseño y creación de un vehículo con la capacidad de desarrollar distintos tipos de actividades, en este caso tener la capacidad de volar, montar paredes verticales y tener una modalidad de rodar en el suelo, con la posibilidad de dotarlo con cámaras y sistemas FPV (o First Person View, por sus siglas en inglés), para tener una perspectiva en primera persona del movimiento del mismo; aunque solamente se ha logrado terminar por completo la parte de vehículo terrestre controlado remotamente mediante radio-control, los alcances que un sistema como este puede tener son muchos ya que pueden ser desde simple recreación hasta supervisión del estado de contenedores verticales ya sean de alimentos o combustibles al instrumentarlo con los sensores necesarios para dicha tarea; es conveniente mencionar la implementación de software y hardware libres en la realización de este proyecto ya que de esta manera las posibilidades para seguir trabajando y mejorando las cualidades del vehículo están totalmente abiertas.

## **Justificación.**

La razón principal por la cual se realizó este proyecto fue para crear un vehículo dinámico que pudiera desempeñar su trabajo (desplazarse en el espacio) sin importar si el entorno es el suelo, una pared o el espacio aéreo, la creación de éste vehículo puede tener diferentes aplicaciones ya que sus cualidades lo permitirían totalmente al tener la capacidad de desarrollar tres tareas, de esta forma podría tener las aplicaciones de un cuadrotor, que actualmente es ampliamente usado por camarografos, también de vehículo terrestre con capacidad de montar paredes y analizar estas mismas con la instrumentación necesaria en busca de grietas o imperfecciones; con un sistema totalmente abierto de hardware y software que productos comerciales casi no ofrecen hoy en día.

## Objetivos.

### General.

- Diseñar y desarrollar un vehículo autónomo con capacidad para pasar de vehículo terrestre a vehículo escalador de pared.

### Específicos.

- Estudio del estado del arte.
- Estimación de posición y movimiento angular en el plano 3D.
- Diseño del sistema de control.
- Investigar la relación entre la producción de empuje y el peso del vehículo.
- Diseñar las piezas del vehículo en solidworks.
- Implementación de los algoritmos de estimación de orientación y del sistema de control en el vehículo.

# Capítulo I: Antecedentes.

## 1.1 Instituto Tecnológico de Tuxtla Gutiérrez.

En los años 70's, llegó al estado de Chiapas el movimiento nacional de extensión educativa para la Educación, con la intervención del gobierno del estado de Chiapas ante la federación. Esta gestión dio lugar a la creación del Instituto Tecnológico Regional de Tuxtla Gutiérrez (ITRTG), hoy Instituto Tecnológico de Tuxtla Gutiérrez (ITTG). El 23 de agosto de 1971, el Gobernador del Estado, Dr. Manuel Velasco Suárez, colocó la primera piedra de lo que pronto sería el centro educativo de nivel medio superior principal de la entidad. El 22 de octubre de 1972, con una infraestructura de dos edificios con ocho aulas, dos laboratorios y un taller en construcción abre sus puertas el Instituto Tecnológico Regional de Tuxtla Gutiérrez con las carreras de técnico en motores de combustión interna, en electricidad, en químico laboratorista y en máquinas y herramientas. En 1974 comenzó el nivel superior, con las carreras de Ingeniería Industrial en Producción e Ingeniería Bioquímica de Productos Naturales. Y fue hasta 1987 que se abrió la carrera de Ingeniería en Electrónica. El proyecto a presentar se realizó en el área de posgrado del instituto en el edificio Z, específicamente en el laboratorio de Control y Automatización.



Figura 1: Instalaciones del edificio Z(izquierda),laboratorio de automatización y control(derecha)

## 1.2 Localización.

El Instituto Tecnológico de Tuxtla Gutiérrez está ubicado en Carretera Panamericana Km. 1080, en la ciudad de Tuxtla Gutiérrez, Chiapas, México.

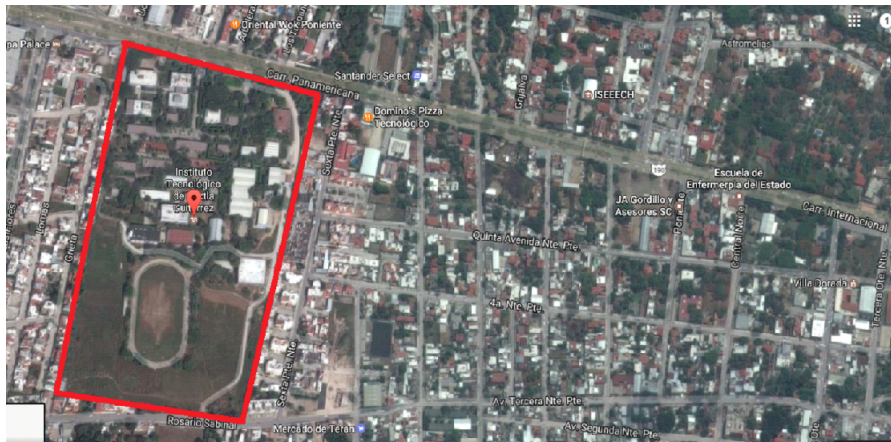


Figura 2: Ubicación del Instituto Tecnológico de Tuxtla Gutiérrez



## **1.3 misión y visión.**

### **1.3.1 Misión.**

Formar de manera integral profesionistas de excelencia en el campo de la ciencia y la tecnología con actitud emprendedora, respeto al medio ambiente y apego a los valores éticos.

### **1.3.2 Visión.**

Ser una institución de excelencia en la educación superior tecnológica del sureste, comprometida con el desarrollo socioeconómico sustentable de la región.

# Capítulo II: Metodología.

Para lograr los resultados obtenidos fue importante tener avances notorios en cada uno de los objetivos específicos; en este capítulo se describirán detalladamente los logros obtenidos en cada uno de estos.

La metodología empleada para el desarrollo de este proyecto se basó en la investigación de las fuerzas de empuje de los motores brushless para poder determinar el diseño adecuado de la estructura del vehículo, teniendo en cuenta el peso del mismo. Para lograr lo anterior fue necesario conocer las características eléctricas así como mecánicas de los motores brushless a utilizar, como la potencia y el empuje generado con cierto tipo de propelas.

## 2.1 Estado del arte

En este apartado se realizó la documentación que necesitaríamos para realizar todas las funciones que desempeñaría el vehículo, se encontraron publicaciones muy interesantes las cuales nos ayudaron a entender el movimiento de un vehículo de dos ruedas utilizando un encoder en cada rueda para ubicar su posición en el espacio bidimensional [1], la implementación de IMU's, Inertial Measurement Unit o unidad de movimiento inercial para la ubicación en el espacio 3D y ver su comportamiento aplicando modelos matemáticos para obtener mediciones más precisas ya que éste tipo de dispositivos necesita enviar datos en tiempo real para poder ser procesados.[2].

De igual forma es sumamente necesario conocer los alcances de este tipo de sistemas no tripulados, ya sea para la investigación o inspección remota de zonas geográficas[3]

Como se menciona anteriormente el procesamiento de la información debería realizar las funciones que el proyecto requiere, para ello se recopiló información de sistemas embebidos capaces de analizar grandes cantidades de información, para ello se recurrió

a la implementación de una placa Raspberry pi 3 ya que es mucho mas poderosa que una placa Arduino que es con la que se trabaja mayormente en el instituto, en todo aspecto, como la velocidad de procesamiento, almacenamiento de información, y es muy accesible ya que se le puede cargar un sistema operativo(Raspbian) basado en Linux; en este caso, el entorno de programación mas amigable para el usuario es el programa Python.

La inspiración para este proyecto fue el robot VertiGo de Disney Research Zurich y la Escuela Politécnica Federal de Zurich, un robot que escala paredes e incluye la transición del suelo a la pared, este modelo no vuela, así que nuestra propuesta ha sido mejorar el diseño haciendo que aparte de ser terrestre y escalar paredes también pueda volar.

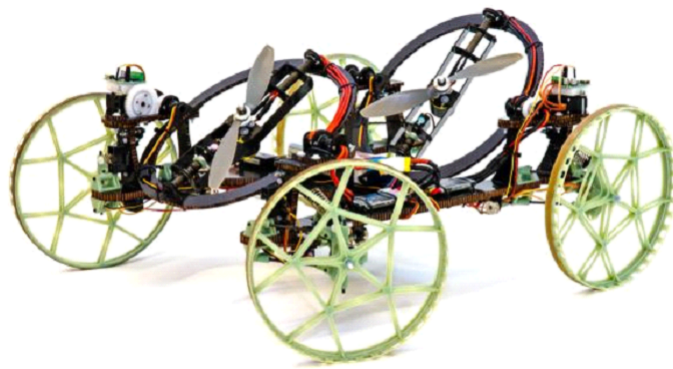


Figura 3: Robot VertiGo

También se tomó la idea de las ruedas con engranaje planetario del modelo B-Flying Car de la marca B GO BEYOND, al principio se pensaba dejar los motores brushless dentro de las ruedas, pero al ir dibujando el diseño nos encontramos con el problema de que las propelas chocarían con los AROS propulsores, así que solo quedaron las ruedas.



Figura 4: B-Flying Car

## 2.2 Estimación de posición y movimiento angular en el espacio 3D

Se trabajó con una unidad de movimiento inercial o IMU, ésta es un dispositivo electrónico que tiene incorporados acelerómetros giroscopios y magnetómetros los cuales nos permiten ubicar en el espacio un dispositivo, saber su dirección velocidad y posición; también es posible saber los ángulos de inclinación en cada uno de los tres ejes del espacio tridimensional (Pitch en el eje X, Roll en el eje Y, Yaw en el eje Z)[5].

El resultado mas destacado fue la obtención de datos del movimiento angular de la IMU en el eje X, con lo que se logró estabilizar mediante un control PID un PVTOL.

## 2.3 Diseño del sistema de control.

En este apartado se utilizaron instrumentos de medición como el osciloscópio, para saber si las señales con las que se estaba trabajando tenían que modificarse para de esa manera empezar a trabajar en el diseño del programa que controlaría todo el sistema.

Se desarrolló un programa para controlar las funciones básicas del vehículo(desplazarse como vehículo terrestre) y se utilizo el lenguaje de programación Python.

## 2.4 Relación entre la producción de empuje y el peso del vehículo.

Esta es una regla empírica descrita para evaluar la motorización adecuada para un modelo determinado, fue definida por el gurú americano del vuelo eléctrico, Keith Shaw y es la más empleada al momento de elegir los elementos electrónicos del vehículo. Se estima que la potencia consumida por el motor, necesaria para mantener un avión en vuelo, ha de ser como mínimo de 100 watts por kilogramo del peso total del modelo. Para vuelo de sport, que permite hacer trepadas lentas y maniobras de entrenamiento se estima una relación potencia peso de 120-130W/Kg. Por último, para realizar vuelo acrobático se precisa 180-200W/Kg. En la actualidad, las marcas que fabrican motores eléctricos para aeromodelismo han optado por incluir en las tablas de especificaciones de sus modelos el empuje generado con relación a la potencia. Para este proyecto se utilizaron un par de motores de la marca T-Motors modelo 2213 de 920 kv, la potencia ganerada por estos motores con una batería de 11.1 V al 100 % de aceleracion es de 108.78W.

Según la regla de Potencia-Peso, con estos motores podríamos impulsar un vehículo de aproximadamente 2 Kg, ya que la batería empleada entrega 11.1 V de tensión.


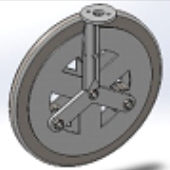
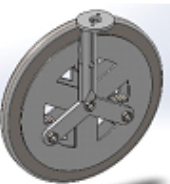
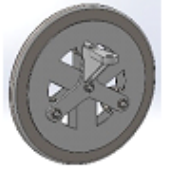
## **2.5 Implementación de los algoritmos de estimación de orientación y del sistema de control en el vehículo.**

En esta fase se procedió al armado del vehículo y la comprobación de todo el sistema.


# Capítulo III: Partes del vehículo y construcción.


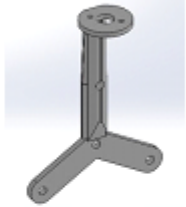
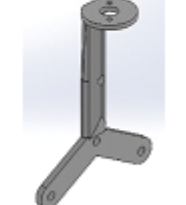
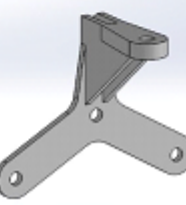
## 3.1 Diseño de las piezas del vehículo implementando solidworks.

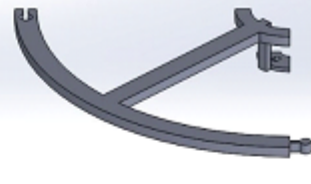
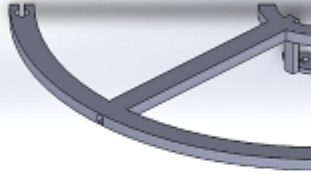
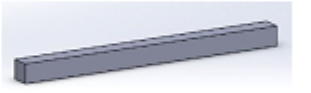
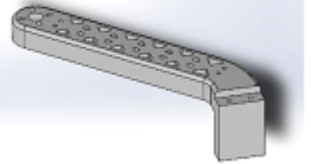
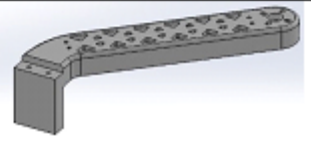
Las siguientes tablas muestra cada una de las piezas del diseño, incluyendo los ensambles que se utilizaron para el ensamble final.

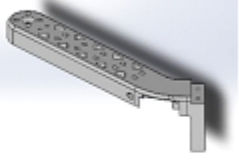
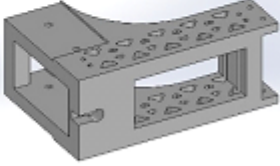
Cantidad	Ensamble	Sub-ensamble	Pieza	Descripción	Imagen
	Ensamble Final del prototipo			Este es el modelo totalmente ensamblado, los motores no se incluyen como parte del ensamble, solo están para mejorar la estética del mismo.	
1		Ensamble de Ruedas 1		Este es uno de los 3 ensambles de las ruedas, más adelante se expondrán cada una de las piezas que lo conforman.	
1		Ensamble de Ruedas 2		Este es el segundo de los ensambles de las ruedas, la diferencia con el primero es la alineación de los agujeros para atornillarse al brazo.	
2		Ensamble de Ruedas 3		El último de los ensambles de ruedas, la diferencia es que el soporte está diseñado para fijarse al servomotor del sistema de dirección.	

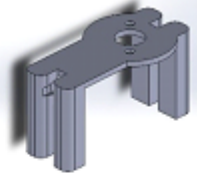
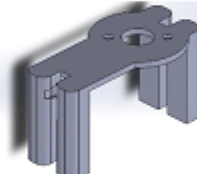
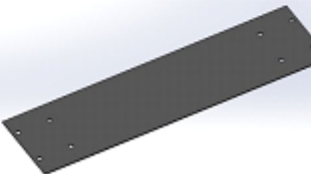




Cantidad	Ensamble	Sub-ensamble	Pieza	Descripción	Imagen
2		Ensamble de ARO		Este ensamble es el soporte de los motores propulsores, está constituido por 2 pares de piezas que se ajustan al motor.	
4			Rueda	Esta pieza forma parte de los ensambles de ruedas, es la que tiene contacto con el suelo y tiene dientes en la parte interna para engranar con la siguiente pieza	
12			Engrane Central	Este engrane forma parte del engranaje planetario de las ruedas, son 3 por cada una.	
4			Soporte externo de engranes	Este soporte es el que hace que los ejes de los engranes centrales sean fijos, para que la rueda gire libremente.	

Cantidad	Ensamble	Sub-ensamble	Pieza	Descripción	Imagen
4			Soporte Interno de engranes	Este soporte es la tapa del engranaje planetario, sujeta los ejes de los engranes para después ajustarse con unas tuercas.	
1			Soporte de Rueda 1	La última pieza del ensamblaje de ruedas.	
1			Soporte de Rueda 2	Como se había mencionado, estas piezas difieren en la alineación de los agujeros para los tornillos.	
2			Soporte de Rueda 3	Y este soporte es para las ruedas delanteras, se fija al eje del servomotor.	

Cantidad	Ensamble	Sub-ensamble	Pieza	Descripción	Imagen
4			Parte 1 del ARO	Forma parte del ensamblaje del ARO que sujeta a los motores propulsores.	
4			Parte 2 del ARO	Esta pieza difiere de la primera porque tiene un hueco cuadrado para darle lugar al eje.	
4			Eje de ARO	Este eje no fue impreso como las demás piezas, fue recortado de un eje de 12 pulgadas de la marca VEX Robotics.	
1			Brazo 1	Estas 6 piezas son parte fundamental de la estructura del prototipo, son las que dan forma al diseño y sostienen todas las piezas secundarias.	
1			Brazo 2		

Cantidad	Ensamble	Sub-ensamble	Pieza	Descripción	Imagen
1			Brazo 3		
1			Brazo 4		
2			Cuerpo Hembra		
2			Cuerpo Macho		

Cantidad	Ensamble	Sub-ensamble	Pieza	Descripción	Imagen
1			Soporte de Servo 1	Esta pieza es la que sostiene al servomotor utilizado para el sistema de dirección, va ajustado al brazo.	
1			Soporte de Servo 2	Al igual que los primeros dos soportes de ruedas estos dos se diferencian por la alineación de los agujeros de los tornillos y también se ajustan a los brazos.	
2			Placa Central	Esta es la placa de circuito impreso a la que se le conectarán los componentes electrónicos incluida la batería, el diseño y los componentes se mostrarán en otro apartado.	

Cantidad	Ensamble	Sub-ensamble	Pieza	Descripción	Imagen
2			Engrane de Servo	Esta pieza se sujeta a los servomotores que harán girar a los AROS del sistema de propulsión	
2			Engrane de ARO	Esta pieza se ensambla al eje del ARO para engranar con el servomotor, la relación de engranes es de 1:2 lo que hace que el giro del servomotor se duplique.	

## 3.2 Descripción y función de los componentes.

A continuación se describe la función de cada uno de los dispositivos implementados en el vehículo y una pequeña descripción de cada uno.

### 3.2.1 radiocontrol.



Figura 5: EMISORA FLYSKY FS-TH9X-B 2.4Ghz. 9Canales

Esta es la última versión de la emisora FLYSKY 2.4Ghz 9ch. La electrónica ha sido mejorada con respecto a la antigua generación, mejorando el aislamiento a posibles interferencias y dotándola de mejor rendimiento. Los nuevos receptores no requieren el uso de satélites, evitando el incremento en peso y cableado de la instalación de los mismos. Incorpora todas las funciones de las versiones anteriores, con mejor rendimiento.

Está dotada de una pantalla LCD de 128x64 pixeles, de gran ayuda durante la programación.

Dispone de capacidad para almacenar hasta ocho modelos distintos, tanto veleros, como aviones rc y helicópteros rc. Una de las ventajas del sistema de radio 2.4GHz es la ausencia de interferencias entre radios en el campo de vuelo, y un notable incremento de la tolerancia a interferencias externas, algo que nos permite volar con más seguridad y concentrarnos en el vuelo en sí. Igualmente, se reducen considerablemente las posibles interferencias producidas por el ESC, motor o cualquier otro elemento electrónico de

nuestro modelo. La señal es puramente digital, por lo que no existen glitches ni los típicos problemas de los sistemas tradicionales de 35Mhz.

Es importante mencionar que éste transmisor contiene cuatro canales que pueden ser configurables al gusto del usuario, en el caso de este vehículo fueron ocupados tres de estos canales para la dirección (izquierda, derecha), aceleración y ángulo de inclinación para ajustar la dirección de impulso.

Radio: 2.4GHz.
Canales: 8ch ppm
Pantalla LCD 128x64 pixeles.
Memoria para 8 modelos.
Modos de vuelo: Helicóptero/avión/velero.
Alimentación: C.C. 12V.

### 3.2.2 Receptor de radiocontrol.



Figura 6: Receptor FrSky V8FR-II 2.4Ghz de 8 canales (Versión HV)

La tecnología FRISKY ACCST (Tecnología Avanzada de Continuo Cambio de Canal) que incorpora este receptor permite la total seguridad en la transmisión de radio, eliminando cualquier tipo de posibilidad de interferencia externa con otros equipos de radio u otro tipo de interferencia, gracias al cambio entre 80 canales que realiza 100 veces en un segundo.

Voltaje de entrada: 3.0 ~ 16V (Versión HV )
Consumo de energía: 30 mA
Resolución: 3072
22ms de latencia:
Número de canales: 8
Rango: 1,5km de 2,5 km
Dimensiones: 44 x 24 x 14 mm

Cabe mencionar que la transmisión se da mediante PPM (Modulación por Posición del Pulso) es una forma parecida a la Modulación por Ancho de Pulso (PWM) por sus siglas en inglés, la diferencia es que en la modulación PPM no cambia la amplitud ni el ancho del pulso, de esta forma dependiendo de la señal moduladora (control remoto) ésta se vera afectada. De esta manera si ésta señal se esta recibiendo constantemente mediante un dispositivo es posible tratar ésta señal y saber en que momento ofrece un mínimo de retardo (valor mínimo de la señal PPM) o un máximo de retardo (valor máximo de la señal PPM)[4].



### 3.2.2 Arduino nano.

Como se menciona anteriormente para que la señal PPM proveniente del radiocontrol pueda ser procesada por el sistema de control de todo el vehículo(Raspberry pi 3), es necesario transformarla a valores numéricos fáciles de interpretar por la Raspberry pi 3, para ello se utilizó un arduino nano, ya que el entorno de desarrollo de Arduino permite usar un comando llamado 'PulseIn()' que sirve para conocer un pulso alto o bajo y devuelve el valor en micro segundos entre cada pulso, de esta manera si el valor de los pulsos PPM es alto, la función 'PulseIn()' nos devuelve un valor numérico en micro segundos alto, y si el valor es bajo, nos devuelve un valor numérico bajo.

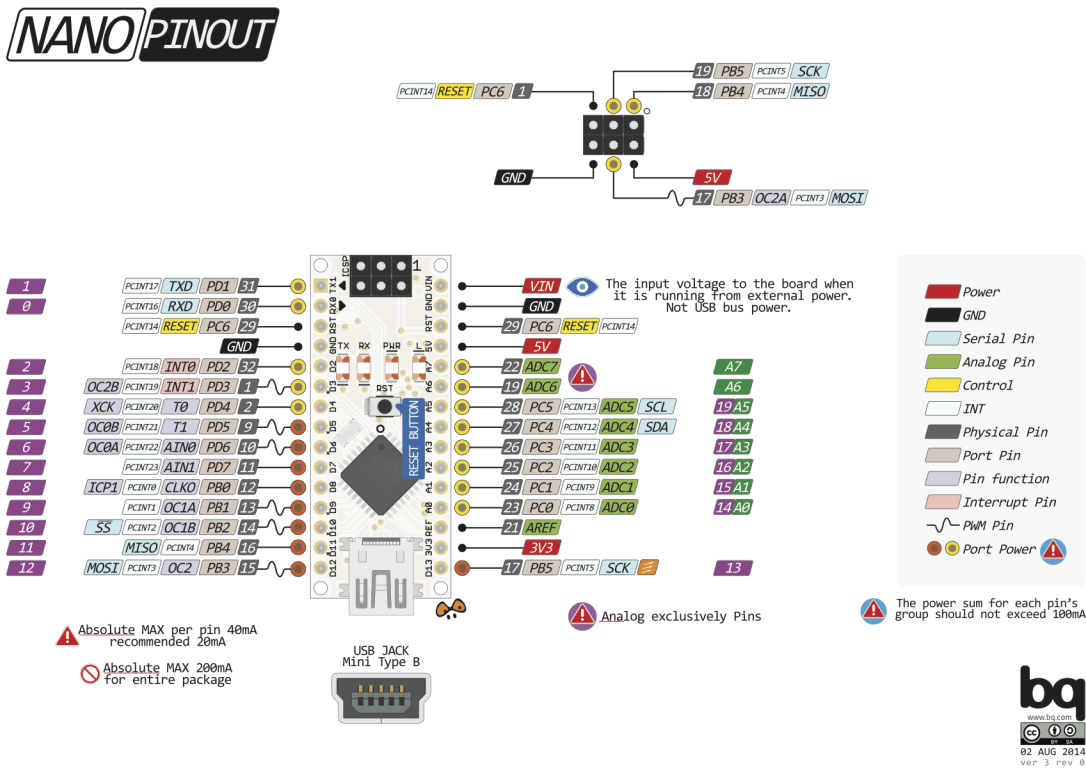


Figura 7: Diagrama esquemático de un Arduino nano

En la imagen anterior se pueden observar con claridad cada uno de los pines con los que cuenta un Arduino nano y los propósitos para los que puede ser destinado cada uno de ellos, en nuestro caso para identificar los pulsos de PPM se usaron pines digitales ya que lo único que se debía identificar era el tiempo en micro segundos que había entre un pulso alto y otro para cada uno de los canales que se usaron.

### 3.2.3 Raspberry pi 3.

Como se menciona anteriormente los tres canales empleados para hacer funcionar todo el sistema ya pueden ser procesados por la Raspberry pi 3, al conectarle el Arduino nano mediante los puertos USB y habilitando la lectura de datos mediante el puerto serial.

Es importante mencionar que la programación en Python es relativamente confusa, ya que trabajando directamente en la consola al crear una línea que ejecuta una acción y pasar a la siguiente línea la anterior se ejecuta, para ello se ocupó un programa llamado Geany, que nos permite realizar un programa completo, guardarlo y ejecutarlo después en la consola de la Raspberry pi 3.

Teniendo la señal adecuada para ser procesada se pasó a realizar el programa en Python, en este programa se hizo una igualación de los máximos y mínimos de todas las señales que la Raspberry pi 3 recibía con los máximos y mínimos que los dispositivos de salida (servomotores y ESC's) necesitaban.

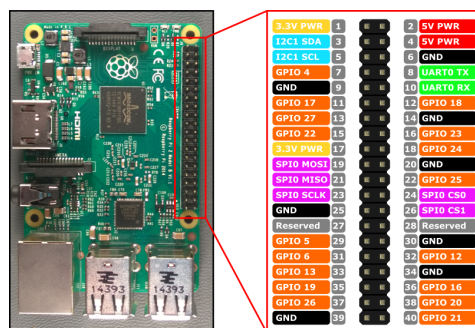


Figura 8: Diagrama esquemático de una Raspberry pi 3

### 3.2.4 ESC(circuito controlador de velocidad).



Figura 9: ESC BLHeli 30 A, OPTO

La gama BLHeli utiliza un procesador que ejecuta el firmware SiLabs BLHeli actualizado para incorporar varias mejoras y características adicionales, tales como las tasas de actualización más altas que dan mucho más rápida respuesta del acelerador lineal en comparación con los CES regulares no Multitrotor.

Modelo	Corriente continua (A)	Burst actual (a) 10 seg	Las células LiPo	Longitud (mm)	Ancho (mm)	Altura (mm)	Peso (g)	Tipo BEC	salida BEC	Programable
BLHeli - 6A	6	8	1-2	22	13	5,5	6	Lineal	0.8A / 5V	sí
BLHeli - 12A	12	15	2-4	42	20	8	11	Lineal	1A / 5V	sí
BLHeli - 20A	20	25	2-4	52	26	7	28	Lineal	2A / 5V	sí
BLHeli - 25A	25	30	2-4	52	26	7	28	Lineal	2A / 5V	sí
BLHeli - 30A	30	40	2-4	52	26	7	28	Lineal	2A / 5V	sí
BLHeli - 30A OPTO	30	40	2-6	67	26	10	25	N / A	N / A	sí
BLHeli - 40A UBEC	40	50	2-6	73	28	12	41	Cambiar	3A / 5V	sí
BLHeli - 50A UBEC	50	60	2-6	73	28	12	41	Cambiar	3A / 5V	sí
BLHeli - 60A UBEC	60	80	2-6	73	36	12	63	Cambiar	5A / 5V	sí
BLHeli - 80A UBEC	80	100	2-6	86	38	12	81	Cambiar	5A / 5V	sí

Figura 10: Especificaciones ESC BLHeli 30 A, OPTO

Estos ESC's reciben señal PWM de la tarjeta Raspberry pi 3 para mover adecuadamente los motores brushless, entre mayor sea el PWM mayor sera la velocidad de los motores y viceversa, un dato curioso de estos ESC's es que no poseen BEC (circuito eliminador de batería) por sus siglas en inglés, esto quiere decir que no pueden aportar energía a otros dispositivos que se alimentan con 5 volts, es por ello que se necesita implementar un BEC externo para alimentar los dispositivos que lo requieran.

### 3.2.5 Motor brushless.



Figura 11: Motor brushless T-Motor Air Gear 2213/920kv

Un motor eléctrico sin escobillas o motor brushless es un motor eléctrico que no emplea escobillas para realizar el cambio de polaridad en el rotor, este tipo de motores son de alto rendimiento, por lo tanto necesitan un alto consumo de potencia, en el caso de los motores empleados en este proyecto cada motor consume aproximadamente 10 A a un voltaje de 11.1 V.

hay una relación que define a este tipo de motores, las literales 'Kv' que no es mas que la cantidad de vueltas que da el motor a máxima potencia por voltio que es aplicado de la batería al ESC.

Item No.	Volts (V)	Prop	Throttle	Amps (A)	Watts (W)	Thrust (g)	RPM	Efficiency (g/W)
AIR 2213 KV920	11.1	T 9545	50%	2	22.2	240	4400	10.81
			65%	3.8	42.18	386	5900	9.15
			75%	5.5	61.05	490	6900	8.03
			85%	7.2	79.92	594	7800	7.43
			100%	9.8	108.78	722	8300	6.64
	12		50%	2.3	27.6	278	4800	10.07
			65%	4.4	52.8	445	6300	8.43
			75%	6.2	74.4	568	2200	7.63
			85%	8.1	97.2	679	8100	6.99
			100%	10.9	130.8	813	8900	6.22
	14.8		50%	3.3	48.84	403	5700	8.25
			65%	6.2	91.76	636	7600	6.93
			75%	8.4	124.32	786	8600	6.32
			85%	10.7	158.36	907	9500	5.73
			100%	14.3	211.64	1084	10200	5.12

Figura 12: Eficiencia del T-Motor Air Gear 2213/920kv

Una de las tres señales emitidas por el control remoto, adecuada por el Arduino nano y procesada por la Raspberry pi 3 es la encargada de mover a dos de estos motores brushless, y es mediante el empuje generado por ellos que el vehículo puede desplazarse libremente, entre menor sea la señal emitida menor sera la señal PWM recibida por los ESC's y por lo tanto menor sera la velocidad de los motores, de lo contrario si la señal es mayor, mayor sera la velocidad de los mismos.

### 3.2.6 Servomotor.

Un servomotor es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición. Está conformado por un motor, una caja reductora y un circuito de control.

Los servomotores hacen uso de la modulación por ancho de pulsos (PWM) para controlar la dirección o posición de los motores de corriente continua. La mayoría trabaja en la frecuencia de los 50 hertz, la electrónica dentro del servomotor responderá al ancho de la señal modulada.

En este proyecto se usaron dos tipos de servomotores.



Figura 13: Servomotor Tower Pro SG90

Peso: 9 g
Tamaño: 22 x 11.5 x 27 mm
Velocidad de funcionamiento (4.8V sin carga): 0.12sec / 60 grados
Puesto de par (4.8V): 17,5 oz / in (1,2 kg / cm)
Rango de temperatura: de -30 a +60 grados C
Muerto Ancho de banda: 7usec
Tensión de funcionamiento: 3.0-7.2 Voltios



Figura 14: Servomotor Tower Pro MG996R

Voltaje de operación: 4.8 V a 7.2 V
Velocidad de operación: 0.17 s/60 <sup>o</sup> (4.8 V), 0.14 s/60 <sup>o</sup> (6 V)
Torque detenido: 9.4 kgf.cm (4.8 V ), 11 kgf.cm (6 V)
Con doble cojinete
Ángulo de rotación: 120 <sup>o</sup> aprox.
Banda muerta: 5 micro segundos
Peso: 55 g

Otra de las señales restantes es la encargada de mover a dos de los pequeños servomotores Tower Pro SG90, que a su vez se encargan de mover las estructuras circulares que soportan a los motores brushless para poder ajustar la dirección de empuje de éstos y mover a todo el vehículo, por otra parte para la dirección del vehículo se usaron dos servomotores Tower Pro MG996R que mueven las llantas delanteras del prototipo y permiten direccionar a la izquierda o derecha dependiendo de la señal emitida por el control remoto, en el caso de los servo mores empleados en el direccionamiento del impulso entre mayor sea la señal emitida por el control remoto mayor perpendicularidad

existirá entre las estructuras circulares que soporta los motores brushless con respecto al suelo (el estado inicial de estas estructuras es completamente paralelas al suelo) y en el caso de la dirección, entre menor es la señal, las ruedas se inclinarán mas a la izquierda y si la la señal va incrementando, se inclinarán mas hacia la derecha.

### 3.2.7 Batería y BEC.

Para la alimentación del prototipo se utilizó una batería LiPo(polímero de litio) de tres celdas que aporta un voltaje de 11.1V y un amperaje de 5.2A, destacaremos la implementación del circuito BEC o circuito eliminador de batería que se encarga de recibir el voltaje directamente de la batería y reducirlo a 5V para energizar al receptor, Arduino nano, Raspberry pi 3, y servomotores ya que de ser conectados directamente a la batería se quemarían.

### 3.3 Diagrama a bloques del sistema y vehículo ensamblado.

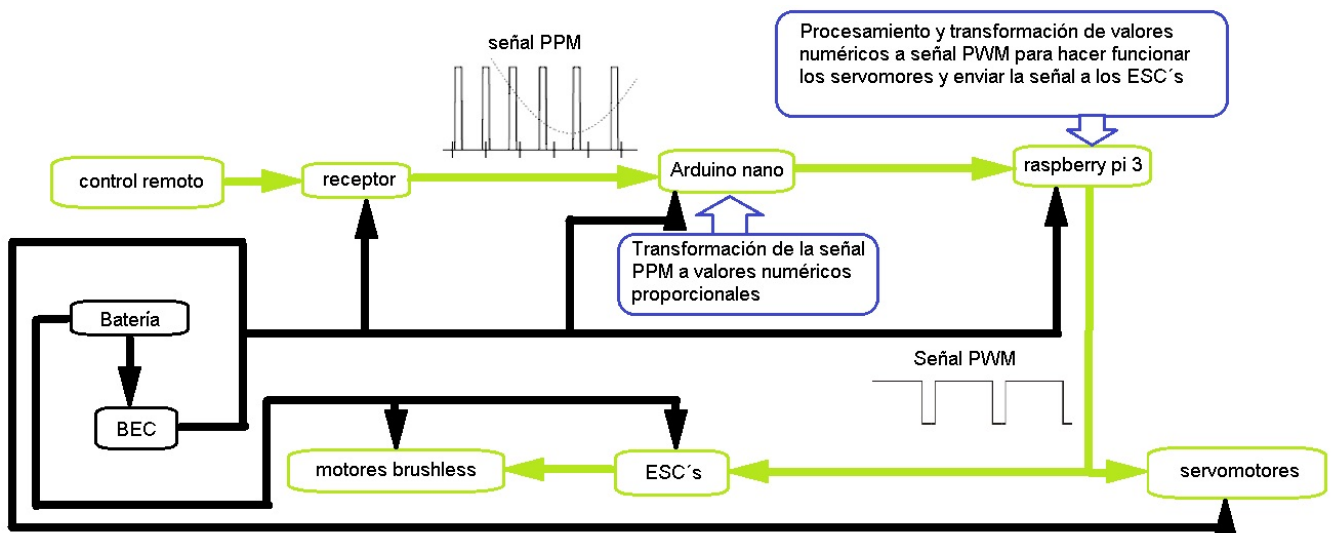


Figura 15: diagrama a bloques del funcionamiento del sistema

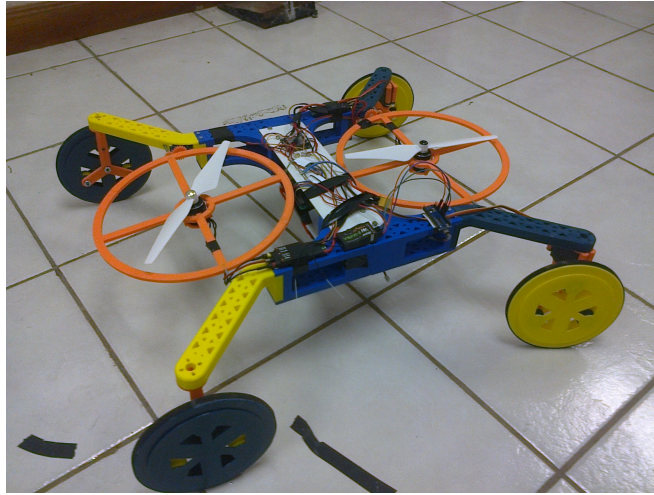


Figura 16: Prototipo ensamblado

## Conclusiones.

Se puede concluir que a pesar de que el proyecto no logró todas las metas propuestas, se lograron avances muy significativos ya que es totalmente funcional para ser usado como vehículo terrestre, es importante destacar que fue la primera vez que se trabajó con una tarjeta Raspberry pi 3, ya que mayormente se ha trabajado con la tarjeta Arduino; y también la utilización de softwares como Solidworks ya que fue de mucha ayuda para el diseño de toda la estructura del vehículo, también destacar la utilización de softwares secundarios que nos permitieron manipular una impresora 3D que tenemos en el laboratorio en el que estuvimos trabajando y softwares de diseño de circuitos como lo es Proteus 8.1.

Una de las grandes ventajas es que ya que el sistema se desarrolló en base a software y hardware libres se pueden hacer mejoras futuras tanto al programa de control como a los sistemas físicos.

También es importante proponer mejoras futuras, como pueden ser; lograr que el vehículo monte paredes e implementar una mejor programación para lograr visualizar el desplazamiento que realice el vehículo y graficarlo en tiempo real en una PC.



## Anexos.

codigo utilizado por el Arduino nano para convertir la señaer PPM a valores numericos.

```
double h;  
double h2;  
double h3;  
void setup()  
{  
  pinMode(6,INPUT);  
  pinMode(5,INPUT);  
  pinMode(4,INPUT);  
  Serial.begin(57600);  
}  
void loop()  
{  
  h = (((pulseIn(5,HIGH)-1050)*(12.23-4.7))/(1900-1050))+4.7;  
  h2 = (((pulseIn(6,HIGH)-1055)*(9.9-4.5))/(1904-1055))+4.5;  
  h3 = (((pulseIn(4,HIGH)-1060)*(12.23-4.7))/(1904-1060))+4.7;  
  Serial.print(hola);  
  Serial.print(",");  
  Serial.print(hola2);  
  Serial.print(",");  
  Serial.print(hola3);  
  Serial.println();  
}
```

Código empleado en python para hacer funcionar todo el sistema.

```
import serial import time
import RPi.GPIO as GPIO
GPIO.setwarnings(False)
from decimal import *
from time import sleep
GPIO.setmode(GPIO.BOARD)
GPIO.setup(11, GPIO.OUT)
GPIO.setup(12, GPIO.OUT)
GPIO.setup(13, GPIO.OUT)
GPIO.setup(15, GPIO.OUT)
GPIO.setup(16, GPIO.OUT)
GPIO.setup(18, GPIO.OUT)
pwm = GPIO.PWM(11,50)
pwm1 = GPIO.PWM(12,50)
pwm2 = GPIO.PWM(13,50)
pwm3 = GPIO.PWM(15,50)
pwm4 = GPIO.PWM(16,50)
pwm5 = GPIO.PWM(18,50)
pwm1.start(0)
pwm2.start(0)
pwm3.start(0)
pwm4.start(0)
pwm5.start(0)
pwm.start(0)
ser=serial.Serial('/dev/ttyUSB0', 57600)
#ser.readline()
#ser.readline().split(' \ ')
#ser.readline().split(' \ ')[0]
while 1:
d=float (Decimal (ser.readline().split(',')[0]))
e=float (Decimal (ser.readline().split(',')[1]))
f=float (Decimal (ser.readline().split(',')[2]))
```

```
if (d+e+f>120.25):
    pwm.ChangeDutyCycle(4.7)
    pwm1.ChangeDutyCycle(4.7)
    pwm2.ChangeDutyCycle(4.5)
    pwm3.ChangeDutyCycle(4.5)
    pwm4.ChangeDutyCycle(4.7)
    pwm5.ChangeDutyCycle(4.7)
    print 'no se puede2'
else: pwm.ChangeDutyCycle(d)
    pwm1.ChangeDutyCycle(d)
    pwm2.ChangeDutyCycle(e)
    pwm3.ChangeDutyCycle(e)
    pwm4.ChangeDutyCycle(f)
    pwm5.ChangeDutyCycle(f)
    print(f)
    pwm.stop()
    GPIO.cleanup()
```

# Bibliografía

- [1] Ernesto Fabregas, Gonzalo Farias, Sebastián Dormido, María Guinaldo, José Sánchez, Sebastián Dormido B., Platform for Teaching Mobile Robotics, Universidad Nacional de Educacion a Distancia, Madrid, España.
- [2] Alexandra Moutinho, Miguel Figueiroa, José Raul Azinheira, Attitude Estimation in  $SO(3)$ : A Comparative UAV Case Study, LAETA, IDMEC; Instituto Superior Técnico, Universidade de Lisbon, Portugal.
- [3] Jorge Mario Cote Corredor, Andrés Felipe Moreno Pineda, Diseño de Control Robusto de Velocidad de Motores Brushless para Robótica Aerea, Universidad Nacional de Colombia, Junio(2010).
- [4] Sistemas de comunicación digitales y analógicos 7ed Leon W. Couch, II editorial Pearson.
- [5] David Ferrer Calatayud, Adquisición de datos IMU en un sistema embebido, Universitat Politecnica de Valencia.