



**UNIVERSIDAD SAN FRANCISCO DE QUITO**

**Colegio de Ciencias e Ingeniería**

**Diseño y construcción de simulador para vehículo teledirigido**

**Diego Roberto Vaca Delgado**

**Omar Aguirre, MSc., Director de Tesis**

Tesis de grado presentada como requisito  
para la obtención del título de Ingeniero Electrónico

Quito, Enero del 2014

Universidad San Francisco de Quito

Colegio de Ciencias e Ingeniería

## HOJA DE APROBACIÓN DE TESIS

### **Diseño y construcción de simulador para vehículo teledirigido**

Diego Roberto Vaca Delgado

Omar Aguirre, MSc.  
Director de la tesis

---

Bernard Herrera, Ing.  
Miembro del Comité de Tesis

---

Luis Caiza, MSc.  
Miembro del Comité de Tesis

---

Omar Aguirre, MSc.  
Director del programa

---

Ximena Córdova, Phd.  
Decana de la Escuela de Ingeniería

---

**Quito, Enero del 2014**

## © DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído la Política de Propiedad Intelectual de la Universidad San Francisco de Quito y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo de investigación quedan sujetos a lo dispuesto en la Política.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo de investigación en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Firma: \_\_\_\_\_

Nombre: Diego Roberto Vaca Delgado

C. I.: 1002325130

Fecha: Quito, enero 2014

## RESUMEN

El presente trabajo de investigación tiene como finalidad describir la construcción y el proceso de diseño para desarrollar una estructura mecatrónica capaz de simular los movimientos, fuerzas y efectos gravitacionales que se producen en tiempo real a través de un vehículo pequeño con cámara a bordo, el cual es controlado inalámbricamente por medio de un volante para videojuegos en una interfaz con un computador (laptop), mediante la programación con software de ingeniería.

La finalidad del proyecto propuesto es poder destacar la importancia que tiene la simulación de los efectos que se pueden recrear de un vehículo móvil en la superficie que se desplaza de forma real, contribuyendo de esta manera a futuros desarrollos en sistemas dentro de la robótica móvil y en general para proyectos relacionados con sistemas de simulación física real, para principalmente despertar el interés por el estudio de ramas de ingeniería como son la electrónica y mecánica. Para ello, se describirá a detalle todo el proceso de manufactura del simulador, así como también todos los materiales implementados en la parte mecánica, electrónica y telecomunicación con su respectiva descripción de funcionamiento y posibles soluciones para el mejoramiento del prototipo, ya que al ser un proyecto de bajo costo, tiene limitaciones en cuanto al tamaño y material utilizado. Sin embargo, se presentará el diseño más adecuado que cumpla a cabalidad con el objetivo propuesto, siguiendo una estricta secuencia de pruebas de funcionamiento durante su desarrollo, para luego integrar conjuntamente los módulos y circuitos electrónicos con los componentes mecánicos, que finalmente serán programados por medio de software con interfaz gráfica para un manejo fácil y amigable al usuario.

## ABSTRACT

The present research is intended to describe the construction and design process to develop a mechatronic structure capable of simulating movements, forces and gravitational effects that occur in real time through a small vehicle with onboard camera, which is wirelessly controlled by a video gaming steering wheel interfaced to a computer (laptop) programmed with software engineering.

The purpose of this project is to highlight the importance of the simulation's effects that can be recreated from a moving vehicle on the surface that moves in a real way, thus contributing to future developments in systems within the mobile robotics generally for projects related to actual physical simulation systems, mainly to arouse interest in the study of engineering branches such as electronics and mechanics. For this, all the manufacturing process of the simulator will be described, as well as all materials implemented in the mechanical, electronic and telecommunication areas with a complete description of the operation and possible solutions for improvement of the prototype. As being an low cost project it has limitations on the size and material used. However, the most suitable design shall comply fully with this objective following a strict sequence of performance testing during development and then integrating the electronic and mechanical components together, which eventually will be programmed through graphical user interface software for easy operation and user friendly.

## TABLA DE CONTENIDO

<b>Resumen</b> .....	4
<b>Abstract</b> .....	5
<b>INTRODUCCIÓN AL PROBLEMA</b> .....	12
Antecedentes .....	12
El problema .....	13
Hipótesis .....	13
Pregunta(s) de investigación .....	14
Contexto y marco teórico .....	14
Propósito de estudio .....	15
Significado de estudio .....	16
Propósito de estudio .....	16
Definición de términos .....	17
<b>OBJETIVOS</b> .....	23
<b>CAPITULO 1: Diseño y plan de desarrollo</b> .....	24
I) División mecánica .....	24
1. Chasis .....	24
1.1 Estructura base .....	26
1.2 Estructura móvil .....	28
2. Vehículo teledirigido .....	28
2.1 Estructura bases principales .....	30
2.1 Funcionamiento .....	33
3. Accesorios adicionales .....	35
3.1 Pantalla LCD .....	35
3.2 Asiento .....	35
3.3 Control de mando .....	36
II) División electrónica .....	36
1. Control del simulador .....	36
1.1 Actuadores .....	37
1.2 Fuente de energía .....	38
1.3 Módulo de control .....	40
1.4 Módulo de Potencia .....	41
2. Control del vehículo .....	42
2.1 Transmisor y receptor .....	42
2.2 Sensor de movimiento .....	44
2.3 Cámara inalámbrica .....	46
<b>CAPITULO 2: Construcción y montaje</b> .....	49
I) Simulador .....	49
1. Estructura y chasis .....	49
2. Montaje de motores .....	54
3. Montaje y descripción de circuitos .....	60
II) Vehículo .....	65
1. Circuito selector .....	65

2. Módulo de transmisión y recepción .....	67
3. Sensor de movimiento.....	69
4. Esquema de conexiones .....	70
5. Montaje de la cámara .....	72
<b>CAPITULO 3: Programación y funcionamiento .....</b>	<b>74</b>
I) Programación.....	74
1. Configuración de módulos .....	74
1.1 Arduino .....	74
1.2 Xbee .....	77
2. Interfaz gráfica .....	81
2.1 Interfaz con el volante.....	81
2.2 Control de dirección, aceleración y arrancador del motor .....	85
2.3 Lectura del acelerómetro.....	89
2.4 Control de los motores del simulador .....	90
II) Funcionamiento.....	92
1. Esquema .....	92
2. Programa principal.....	93
3. Ejecución y manejo .....	99
<b>ESPECIFICACIONES .....</b>	<b>101</b>
Simulador .....	101
Vehículo .....	102
Costos.....	102
<b>CONCLUSIONES.....</b>	<b>103</b>
<b>REFERENCIAS .....</b>	<b>105</b>
<b>ANEXOS .....</b>	<b>107</b>



## TABLAS

Tabla 1: Características del motor DC .....	37
Tabla 2: Características de la fuente de poder .....	39
Tabla 3: Características del microcontrolador .....	40
Tabla 4: Características del puente H.....	42
Tabla 5: Características módulo Xbee/Xbee Pro .....	43
Tabla 6: Características acelerómetro MMA7361 .....	45
Tabla 7: Modo de operación puente H.....	64
Tabla 8: Descripción de pines del módulo Xbee Pro .....	67
Tabla 9: Especificaciones del simulador.....	101
Tabla 10: Especificaciones del vehículo.....	102
Tabla 11: Costos .....	102

## FIGURAS

Figura 1: Simulador simple de manejo .....	16
Figura 2: Tubos PVC .....	17
Figura 3: Perfil de estantería .....	17
Figura 4: Circuito puente H.....	18
Figura 5: Acelerómetro en un joystick.....	19
Figura 6: Módulo arduino uno .....	20
Figura 7: Estructura de servomotor.....	22
Figura 8: Potenciómetro.....	22
Figura 9: Grados de libertad de movimiento.....	24
Figura 10: Simulador de 2DOF con tubos de metal.....	25
Figura 11: Simulador de 2DOF con tubos de plástico .....	26
Figura 12: Bosquejo de la estructura base .....	27
Figura 13: Bosquejo de la estructura móvil .....	28
Figura 14: Vehículo teledirigido .....	29
Figura 15: Estructura y partes del vehículo.....	30
Figura 16: Amortiguador .....	31
Figura 17: Servomotor .....	31
Figura 18: Motor de 3.4cc a combustión .....	32
Figura 19: Arrancador electrónico del motor a combustión .....	32
Figura 20: Tanque de combustible.....	32
Figura 21: Vehículo en funcionamiento .....	33
Figura 22: Sistema de inyección de combustible.....	34
Figura 23: Televisor HDTV de 27" .....	35
Figura 24: Asiento deportivo para automóvil .....	35
Figura 25: Logitech momo racing wheel .....	36
Figura 26: Motor DC de limpia parabrisas .....	37
Figura 27: Fuente de poder .....	38
Figura 28: Módulo arduino uno .....	39
Figura 29: Puente H .....	41

Figura 30: Módulos Xbee Pro .....	43
Figura 31: Acelerómetro de 3 ejes MMA7361 .....	44
Figura 32: Cámara HD GoPro Hero3 .....	46
Figura 33: Transmisor de video HD EP3207 .....	47
Figura 34: Base (vista lateral).....	49
Figura 35: Base (vista frontal).....	50
Figura 36: Abrazaderas metálicas .....	50
Figura 37: Rueda giratoria .....	50
Figura 38: Movimiento lateral .....	50
Figura 39: Base móvil horizontal.....	51
Figura 30: Rueda giratoria grande.....	51
Figura 41: Base giratoria.....	51
Figura 42: Base móvil vertical .....	52
Figura 43: Acople con la base vertical.....	52
Figura 44: Asiento montado.....	53
Figura 45: Movimiento vertical y lateral .....	53
Figura 46: Estructura completa .....	53
Figura 47: Montaje de la pantalla .....	54
Figura 48: Montaje completo.....	54
Figura 49: Modificación de giro de los motores.....	55
Figura 50: Modificación en velocidad de giro del motor.....	55
Figura 51: Pruebas de los motores .....	56
Figura 52: Montaje del motor vertical .....	56
Figura 53: Polea con piñon y guia cadena .....	57
Figura 54: Montaje del motor horizontal .....	57
Figura 55: Resortes de tensión .....	57
Figura 56: Mecanismo completo.....	58
Figura 57: Chasis pintado .....	58
Figura 58: Bandas elásticas con tubo de llanta .....	59
Figura 59: Chasis y estructura completa.....	60
Figura 60: Fuente de alimentación y circuitos .....	60
Figura 61: Diagrama de conexión de los circuitos de control de los motores .....	61
Figura 62: Diagrama del circuito del módulo de control para servomotor .....	62
Figura 63: Adaptación de potenciómetros en los motores .....	63
Figura 64: Disipador de calor y ventiladores de 12V .....	64
Figura 65: Diagrama del circuito de selección.....	66
Figura 66: Placa del circuito selector.....	66
Figura 67: Jumpers de selección de modo del shield xbee .....	68
Figura 68: Base y soportes del módulo en el vehículo.....	69
Figura 69: Montaje del acelerómetro .....	69
Figura 70: Configuración de pines del acelerómetro .....	70
Figura 71: Esquema de conexión de módulos en el vehículo .....	73
Figura 72: Montaje total de módulos en el vehículo .....	72
Figura 73: Cámara HD y módulo transmisor.....	72
Figura 74: Cámara montada en el vehículo.....	73
Figura 75: Software arduino .....	75
Figura 76: Programación principal del arduino .....	76
Figura 77: Programador Xbee USB .....	78
Figura 78: Ventana principal del programa X-CTU .....	78

Figura 79: Comprobación de comunicación con el módulo Xbee.....	79
Figura 80: Parámetros de configuración del módulo Xbee.....	79
Figura 81: Pantalla principal del Labview 2012 .....	81
Figura 82: Programación para el control del volante .....	82
Figura 83: Ejecución del programa de interfaz con el volante.....	83
Figura 84: Programación para el arrancador, dirección y aceleración del vehículo .....	85
Figura 85: Configuración del bloque “init” .....	86
Figura 86: Interfaz gráfica de la programación del arrancador, dirección y aceleración.....	88
Figura 87: Programación de los motores .....	89
Figura 88: Interfaz gráfica de lectura del acelerómetro .....	90
Figura 89: Programación de los motores del simulador .....	90
Figura 90: Interfaz gráfica del control de los motores del simulador .....	91
Figura 91: Esquema de funcionamiento de todo el sistema.....	92
Figura 92: Programa de control principal del sistema .....	93
Figura 93: Inicialización del módulo inalámbrico .....	94
Figura 94: Programación de acondicionamiento de la señal del sensor.....	94
Figura 95: Programación de señal del volante .....	95
Figura 96: Cierre del puerto de lectura del módulo .....	96
Figura 97: Inicialización del puerto para el simulador.....	96
Figura 98: Programación para sensibilidad de posición neutral del simulador .....	97
Figura 99: Bloque “in range and coerce” .....	97
Figura 100: Bloque “select” .....	98
Figura 101: Bloque de control para el volante .....	98
Figura 102: Condiciones de “stop” para el loop .....	99
Figura 103: Interfaz gráfica del programa principal del sistema .....	100

## INTRODUCCIÓN AL PROBLEMA

El término simulador se lo acredita a un sistema físico o virtual por el cual se puede reproducir efectos, resultados, sensaciones y experiencias físicas que pueden llegar a pasar en la realidad con un determinado margen de certeza y error<sup>1</sup>. Dentro de los sistemas de simulación física encontramos gran variedad de simuladores para autos, aviones, submarinos e inclusive naves espaciales, en donde se busca recrear los movimientos, fuerzas gravitacionales y efectos que el móvil puede llegar a experimentar en situaciones reales, por lo que existe gran diversidad de estructuras mecánicas que cumplen a cabalidad con este objetivo, siendo las más comunes, los simuladores para vehículos sobre ruedas.

### **Antecedentes**

El estudio e investigación de sistemas móviles de simulación ha ido tomando territorio cada vez más dentro del campo de la robótica móvil, debido a su gran importancia en la exploración y reconocimiento de territorio poco habitable cuando se realizan pruebas con vehículos teledirigidos. Para que este tipo de vehículos realicen una exploración exhaustiva de un determinado territorio es necesario obtener la mayor información posible en cuanto a condiciones climáticas, geográficas y superficiales del entorno en donde se realizará la exploración. Por ello es determinante que se pueda simular físicamente de algún modo las condiciones terrenales por donde circula el vehículo para que de esta manera se pueda tener una mejor idea del trazado del camino que éste vaya a tener y así evitar percances que puedan impedir al vehículo su movimiento y reconocimiento de trayectoria.

---

<sup>1</sup> [1]

## **El problema**

Uno de los principales problemas que podemos encontrar dentro del ámbito de simulación de vehículos teledirigidos, es que solo existen simuladores virtuales, es decir, simulación digital mediante computadoras, ya que técnicamente a través de la simulación virtual se puede tener una gran aproximación del comportamiento de un móvil, con lo cual es posible ahorrar mucho tiempo en el proceso de diseño y construcción del mismo. Sin embargo, los resultados del comportamiento del vehículo podrían ser optimizados aun más si es que existiese también una simulación física a través de una estructura mecánica que pueda recrear los movimientos a partir de la simulación virtual, que por consiguiente podría influir de manera significativa, por ejemplo en las etapas de “prueba y error” de un prototipo de un robot móvil al momento de hacerlo funcionar en condiciones extremas, con lo cual se podría obtener mayor información para modificar su estructura y diseño.

## **Hipótesis**

Actualmente existen gran variedad de mecanismos mediante los cuales se reproducen las simulaciones físicas de los automóviles, siendo muy comunes encontrarlos en las grandes empresas de fabricación automotriz, alcanzando costos muy elevados con lo cual su adquisición se torna en un factor muy limitante ante alguna compañía o centro educativo con limitaciones económicas, por lo cual la presente investigación se centrará principalmente en la fabricación de un simulador de bajo costo a partir de materiales para construcción residencial y de módulos electrónicos simples, que conjuntamente serán capaces de recrear hasta un 70% del entorno físico del movimiento de un vehículo pequeño con sensores controlado inalámbricamente, esto quiere decir que se podrá simular con gran aproximación los movimientos verticales y horizontales que se puedan

experimentar durante su desplazamiento, además de fuerzas motrices (choques) y de aceleración variada (velocidad).

### **Pregunta(s) de investigación**

¿Cómo y hasta qué punto es posible construir un simulador de movimientos de bajo costo para un vehículo teledirigido a partir de materiales simples como los que son utilizados para construcción doméstica residencial?

¿Cómo se realizará la comunicación y monitoreo en tiempo real del vehículo para simular sus movimientos?

### **Contexto y marco teórico**

Dentro de las posibles opciones para la construcción y diseño del simulador tenemos que principalmente tomar en cuenta el tipo de material que utilizará el chasis de la estructura principal, siendo esta la parte más relevante del diseño ya que a partir de este punto podremos determinar el resto de componentes que vayamos a utilizar. Entre los materiales para el chasis tenemos opciones como tubos de aluminio, hierro o plástico reforzado. Luego de haber escogido el tipo de material del que se construirá el chasis será necesario escoger el tipo de motores que harán posible el movimiento del chasis, pudiendo ser estos motores eléctricos DC o AC tomando en cuenta sus características como son el tamaño, consumo de corriente, torque y revoluciones por minuto. Los motores eléctricos necesitarán módulos electrónicos para su manejo en donde se dividirán dos etapas, que son la etapa de control y la etapa de potencia, en donde igualmente dependerá su elección de acuerdo al tipo de motores que se vaya a utilizar y también el tipo de fuente de alimentación que proporcionarán la energía eléctrica necesaria para su funcionamiento, con lo que finalmente la estructura y diseño del chasis estará concluida.

Para el vehículo teledirigido se tienen también numerosas opciones como los carros RC eléctricos o a combustión en donde solo nos concentraremos en su tamaño y diseño que estos tengan descartando su circuitería interna, ya que se necesitará modificar sus conexiones eléctricas y estructura para poder acoplar la mini cámara de video que transmitirá en tiempo real las imágenes del desplazamiento del móvil. Así mismo los módulos electrónicos con sensores que vayamos a utilizar tendrán que ser encajados dentro del vehículo para poder monitorear sus movimientos de manera inalámbrica, pudiendo ser esta a través de protocolos de comunicación conocidos como bluetooth o WiFi.

Finalmente para poder integrar y controlar todo el sistema será necesario la programación por medio de software, con el que se creará una interfaz gráfica para el usuario para saber el estado de los sensores y controles manuales del vehículo; para este efecto se utilizará un volante de video juegos para computador el cual permitirá girar, acelerar y frenar el vehículo, recalcando que se podrá visualizar el video transmitido por la cámara inalámbrica por medio de una pantalla LCD montada sobre el chasis del simulador.

### **El propósito del estudio.**

Con el proyecto propuesto se llegará a simular los movimientos y fuerzas físicas que experimenta un vehículo teledirigido mientras es conducido a través de cualquier tipo de terreno, consiguiendo con esto tener una gran aproximación de la sensación a escala real que el móvil experimenta, debido al adecuado diseño estructural de la plataforma e implementación y programación de componentes mecatrónicos.

### **El significado del estudio.**

El presente trabajo se fundamenta principalmente en la limitación económica, por lo que con los resultados esperados lo llegarán a convertir en un proyecto sobresaliente y casi único dentro del ámbito de simulación móvil, que puede contribuir al desarrollo de estructuras de simulación más complejas de bajo presupuesto para diferentes tipos de vehículos o robots móviles y también a la implementación de este tipo de simuladores para que puedan ser adaptados a diferentes necesidades que se requiera, como por ejemplo para simulación en recorrido de vehículos reales, aviones, barcos, trenes, etc... como medio de entrenamiento para futuros pilotos.

### **Definición de términos**

#### ***Simulador***

Un simulador es una máquina que reproduce el comportamiento de un sistema en ciertas condiciones, lo que permite que la persona que debe manejar dicho sistema pueda entrenarse. Los simuladores suelen combinar partes mecánicas o electrónicas y partes virtuales que le ayudan a simular la realidad<sup>2</sup>, como se muestra en la *figura 1*.



Figura 1: Simulador simple de manejo<sup>3</sup>

---

<sup>2</sup> [2]

<sup>3</sup> [3]



### ***Tubería PVC***

Los tubos PVC son elaborados a partir de un material derivado del plástico que son comúnmente utilizados para conexiones de agua en las instalaciones y construcciones domiciliarias e industriales (*figura 2*). Existen diferentes tipos de tubería de acuerdo a su rigidez, espesor y tamaño de acuerdo a la aplicación.



Figura 2: Tubos PVC<sup>4</sup>

### ***Perfiles de estantería***

Son tiras de hierro o metal en forma de “L” alargados que son utilizados como perfiles en estructuras comúnmente utilizadas en bodegas e instalaciones domiciliarias con la particularidad de tener orificios redondos y oblicuos a lo largo de su superficie. La *figura 3* nos muestra un ejemplo de estos instrumentos:



Figura 3: Perfil de estantería<sup>5</sup>

---

<sup>4</sup> [4]

<sup>5</sup> [5]

### ***Puente H***

Un puente H es un módulo electrónico que puede controlar la dirección de giro de uno o dos motores DC dependiendo su configuración. El esquema básico (*figura 4*) de su circuito es el siguiente:

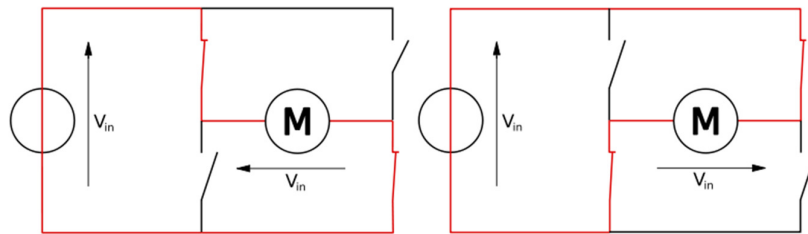


Figura 4: Circuito puente H<sup>6</sup>

### ***Fuerza G***

La fuerza G no es una medida de fuerza sino una medida intuitiva de aceleración. Está basada en la aceleración que produciría la gravedad terrestre en un objeto cualquiera en condiciones ideales (sin atmósfera u otro rozamiento). Una aceleración de 1G igual a la gravedad estándar, que es de 9.80665 metros por segundo cuadrado (m/s<sup>2</sup>).

La fuerza G para un objeto es de 0G en cualquier ambiente sin gravedad, como una caída libre o un satélite orbitando la tierra y de 1G a cualquier objeto estacionario en la superficie de la tierra al nivel del mar<sup>7</sup>. Las fuerzas G pueden ser mayores a 1, como en una montaña rusa, en una centrifugadora o en un cohete.

### ***Acelerómetro***

Es un instrumento que nos permite medir aceleraciones (fuerzas G) determinadas por el peso, a partir de un nivel de referencia propio del dispositivo.

<sup>6</sup> [6]

<sup>7</sup> [7]

Dentro de la electrónica existen varios tipos de acelerómetros que nos permiten sentir movimientos, siendo los más conocidos los que se utilizan en los teléfonos celulares (smartphones) y mandos de control en consolas de videojuegos<sup>8</sup> (figura 5).



Figura 5: Acelerómetro en un joystick<sup>9</sup>

### ***Arduino***

Es un prototipo electrónico programable conformado por un microprocesador con entradas y salidas analógicas / digitales, el cual fue desarrollado para implementación en funciones básicas de proyectos electrónicos y con lenguaje de programación con software de código libre, que requiere un nivel básico de aprendizaje. Existen varios modelos de este prototipo con diferentes características diferenciadas principalmente por el microprocesador que utiliza, tamaño y funciones que realiza, siendo el más popular y utilizado el arduino UNO, con un tamaño relativamente pequeño, como se ve en la figura 6:

---

<sup>8</sup> [8]

<sup>9</sup> [9]

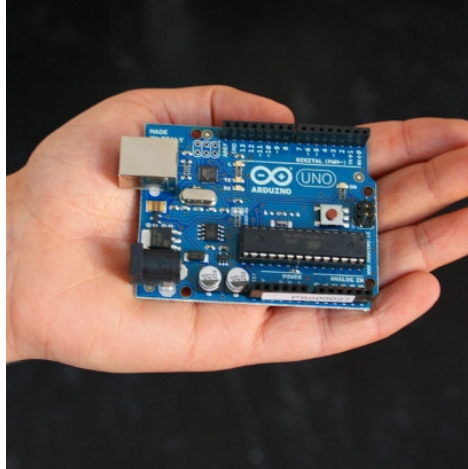


Figura 6: Módulo arduino UNO<sup>10</sup>

### ***HDMI***

High-Definition Multimedia Interface o HDMI, (interfaz multimedia de alta definición), es una norma de audio y vídeo digital cifrado sin compresión apoyada por la industria para que sea el sustituto del euroconector. HDMI provee una interfaz entre cualquier fuente de audio y vídeo digital como podría ser un sintonizador TDT, un reproductor de Blu-ray, un tablet PC, un ordenador (Microsoft Windows, Linux, Apple Mac OS X, etc.) o un receptor A/V, y monitor de audio/vídeo digital compatible, como un televisor digital (DTV)<sup>11</sup>.

### ***Wi-Fi***

(Wireless Fidelity) es un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica. Los dispositivos habilitados con Wi-Fi, tales como: un ordenador personal, una consola de videojuegos, un smartphone o un reproductor de audio digital, pueden conectarse a Internet a través de un punto de acceso de red inalámbrica.

---

<sup>10</sup> [10]

<sup>11</sup> [11]

Dentro de los diversos tipos de Wi-Fi, basados en un estándar IEEE 802.11 encontramos que los estándares IEEE 802.11b, IEEE 802.11g e IEEE 802.11n tienen aceptación internacional ya que la banda de frecuencia de 2.4 GHz está disponible casi universalmente, con una velocidad de hasta 11 Mbit/s, 54 Mbit/s y 300 Mbit/s, respectivamente.

Actualmente se encuentra también el estándar IEEE 802.11a, conocido como WiFi 5, que opera en la banda de 5 GHz y que disfruta de una operatividad con canales relativamente limpios. La banda de 5 GHz al ser recientemente habilitada, no tiene problemas de interferencias con otros protocolos y tecnologías de comunicación y su alcance es algo menor que el de los estándares que trabajan a 2.4 GHz, debido a su alta frecuencia de trabajo que limita su espacio de transmisión<sup>12</sup>.

### ***Servomotor***

Un servomotor es un dispositivo similar a un motor DC, que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición, siendo posible también el control de su velocidad por medio de un circuito electrónico interno<sup>13</sup>. Posee un potenciómetro adjuntado al engranaje de giro del eje principal, el cual proporciona una señal de retroalimentación para determinar la posición de giro. Existen diferentes tipos de servomotores en cuanto al tamaño, la capacidad de giro, velocidad y torque, que están determinados por el sistema de engranajes que posee internamente (*figura 7*).

---

<sup>12</sup> [12]

<sup>13</sup> [13]



Figura 7: Estructura servomotor<sup>14</sup>

### ***Potenciómetro***

Es un elemento electrónico que tiene la característica de variar su resistencia desde un valor de cero hasta el valor final. Existen muchos tipos de potenciómetros, pudiendo ser digitales o analógicos, pero los más comunes son los analógicos que poseen una pequeña perilla para variar su resistencia manualmente, como se muestra en la *figura 8*:



Figura 8: Potenciómetro<sup>15</sup>

---

<sup>14</sup> [14]

<sup>15</sup> [15]

## OBJETIVOS

- Diseñar y construir una estructura mecánica que pueda reproducir los movimientos laterales y verticales de un vehículo teledirigido, tratando también de recrear fuerzas de impacto y de aceleración.
- Implementar módulos y sensores electrónicos para monitorear los movimientos de un vehículo teledirigido.
- Implementar y programar módulos de comunicación inalámbrica a través de protocolos de radiofrecuencia para el control del vehículo teledirigido mediante un volante (joystick) para videojuegos.
- Implementar el uso de una cámara de video inalámbrica para que pueda transmitir en tiempo real la trayectoria y desplazamiento del vehículo teledirigido a través de una pantalla LCD montada sobre la estructura mecánica diseñada.
- Programar por medio de interfaz gráfica la simulación de los movimientos del vehículo teledirigido para que sean reproducidos en la estructura mecánica diseñada.
- Realizar las pruebas necesarias para optimizar al máximo el sistema.

## CAPITULO 1: Diseño y plan de desarrollo

### I) División mecánica

#### 1. Chasis

Para el diseño del chasis, hay que destacar un factor importante en los simuladores, como es el grado de libertad que estos tienen en su estructura y movimiento. Los grados de libertad en una estructura se refieren a los movimientos en el espacio tridimensional que se puedan realizar, siendo estos movimientos verticales, laterales, longitudinales, horizontales, rotativos y de translación<sup>16</sup> (figura 9).

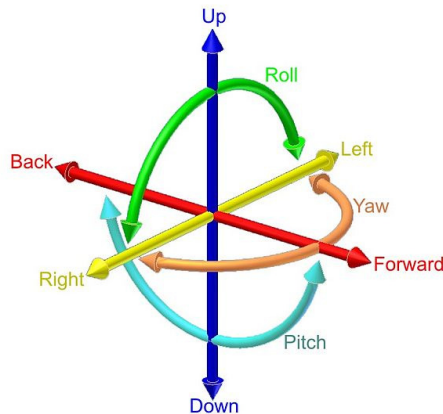


Figura 9: Grados de libertad de movimiento<sup>17</sup>

Si pudiésemos representar todos estos movimientos en una estructura, entonces se diría que esta tiene 6 grados de libertad ó 6 DOF (degrees of freedom). Para el presente diseño y aplicación se determinó que una estructura de 2 DOF que represente los movimientos laterales y verticales es suficiente para poder también representar los

<sup>16</sup> [16]

<sup>17</sup> [17]



movimientos del vehículo y con la correcta programación poder tener también una simulación aproximada de fuerzas de aceleración e inercia.

Para la construcción del chasis se utilizarán tubos PVC de diferentes diámetros y longitud, ya que son tubos fáciles de recortar y unir, usando adaptadores de diferentes tipos para poder fácilmente construir el esquema mecánico que se va a diseñar, lo que será económicamente beneficioso al ser un material fácil de modificar y ensamblar. También es importante señalar que este material es muy liviano, con lo cual hará factible la utilización de motores con menos potencia para que puedan mover la estructura, y que también facilita enormemente el transporte del simulador una vez que se haya terminado.

Existen diseños de simuladores de 2DOF en el internet, de los cuales se tomó referencia, como son los siguientes representados en la *figura 10 y 11*:



Figura 10: Simulador de 2DOF con tubos de metal<sup>18</sup>

---

<sup>18</sup> [18]



Figura 11: Simulador de 2DOF con tubos de plástico<sup>19</sup>

Dentro de estos diseños, es posible dividir su estructura en dos partes:

- Estructura base
- Estructura móvil

### ***1.1 Estructura base.***

La estructura base es la parte principal del diseño de todo el chasis, ya que a partir de esta podremos determinar el tamaño de todo el simulador. Hay que tomar en cuenta que esta estructura deberá soportar todo el peso de la estructura móvil, pantalla LCD, asiento y la persona en el simulador, por lo que necesitará tener soportes de apoyo suficientes en sus ejes y perfiles para que los tubos de PVC no sean vulnerables a doblarse o romperse.

---

<sup>19</sup> [19]

A continuación se presenta un bosquejo dibujado a mano de la estructura base:

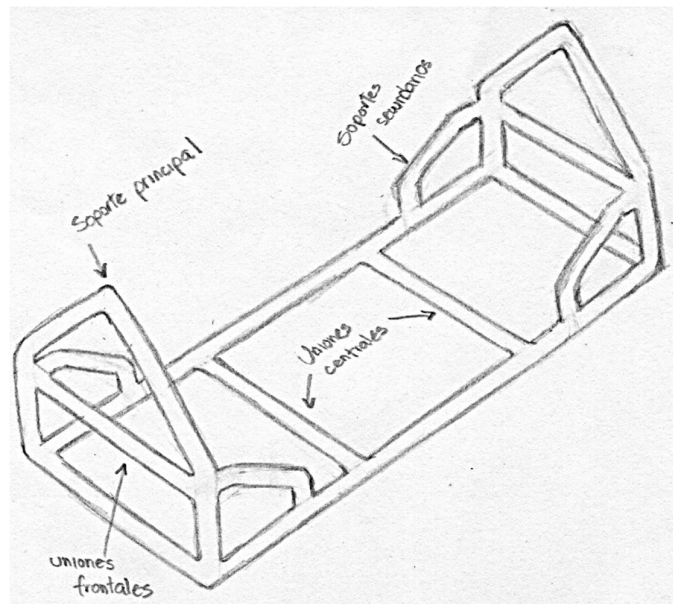


Figura 12: Bosquejo de la estructura base

En la *figura 12*, el soporte principal tiene forma de triángulo ya que se encargará de sostener la estructura móvil a través de un sujetador con rulimán, el cual podrá tener un movimiento lateral de hasta 45 grados en cada lado. Las uniones frontales impedirán que el soporte principal colapse en caso de que se someta a un peso mayor del que pueda tolerar y de la misma manera los soportes secundarios sirven de refuerzo para los perfiles verticales. Finalmente, hay que tomar en cuenta que se deberá tratar de reducir las medidas al máximo posible, ya que los tubos de plástico pierden rigidez si sus medidas aumentan, por lo que las medidas de longitud y altura serán detalladas posteriormente en la fase de construcción, al igual que el tipo de tubo PVC que se utilizará para esta estructura.

### 1.2 Estructura móvil.

Esta estructura será la encargada de realizar los movimientos laterales y verticales del simulador, y será montada sobre la estructura base junto con los demás accesorios y componentes como son el asiento, la pantalla LCD y el volante con sus pedales.

De la misma manera se presenta un dibujo aproximado de la estructura móvil en la *figura 13*:

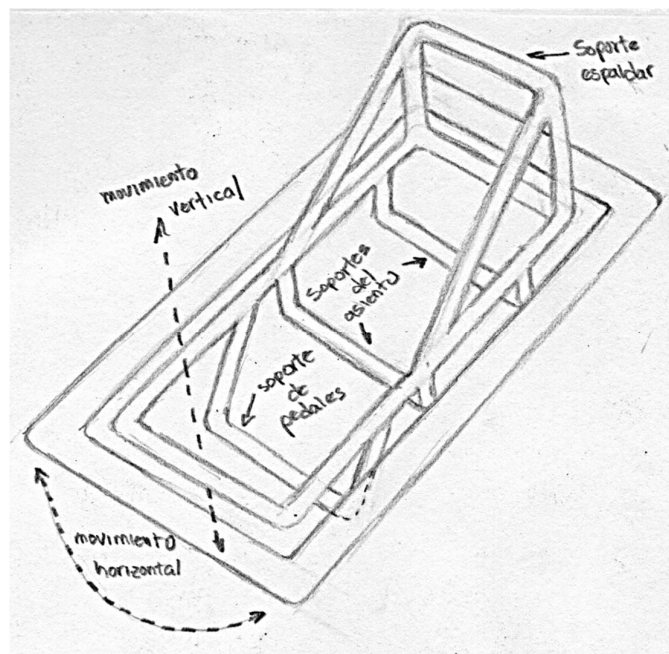


Figura 13: Bosquejo de la estructura móvil

Los tubos que conforman el contorno de esta estructura y que realizarán los movimientos laterales, tendrán mayor diámetro debido a que su longitud es mayor que el resto de uniones, y de esta manera se podrá evitar que se rompan.

## 2. Vehículo teledirigido

Dentro de las numerosas opciones en la elección del vehículo que deberá ser controlado y a la vez transmitir las señales de los movimientos para el simulador, se ha

escogido un carro a escala 1/10 a radio control (*figura 14*) que funciona por medio de un motor a combustión, el cual tiene las siguientes características:



Figura 14: Vehículo teledirigido<sup>20</sup>

Tipo: Truggy

Escala: 1 / 10

Longitud: 17.75 pulgadas (450.85mm)

Anchura: 13,5 pulgadas (342.9mm)

Distancia entre ejes: 11,75 a 11,875 pulgadas (298.45 - 301.625mm)

Peso: 6,2 libras (2.812kg)

Chasis: Negro, aluminio anodizado

Tipo de neumáticos: caucho, 3.2 pl

Motor: Losi 3,4 cc

Combustible: Nitrometano

Radio: DX3S Spektrum 2.4GHz DSM Tecnología con telemetría.

Baterías: 7.4V LiPo

Cargador: LiPo Cargador de pared

Cuerpo: Pre-pintado y decorado (carcaza plástica)<sup>21</sup>

---

<sup>20</sup>

La descripción básica de este vehículo se ha dividido en las siguientes secciones que son:

- Estructura y partes principales
- Funcionamiento

### 2.1 Estructura y partes principales.

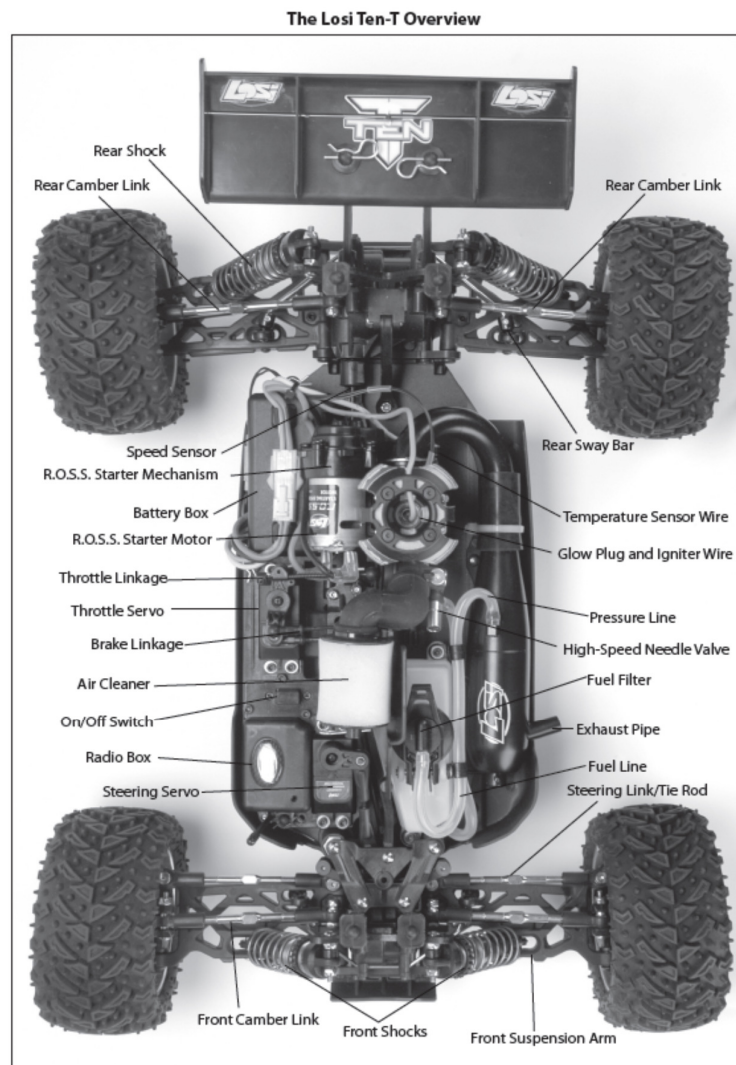


Figura 15: Estructura y partes del vehículo<sup>22</sup>

<sup>21</sup> [21]

<sup>22</sup> [22]

El vehículo cuenta con tracción en las cuatro ruedas (AWD) y cuatro amortiguadores de alto impacto con resortes regulables (*figura 16*), capaces de soportar caídas de hasta 2 metros de altura.



Figura 16: amortiguador

La dirección y aceleración están controlados por dos servomotores (*figura 17*) de 3.3V con un giro máximo de 60 grados y con un torque de 5.0kg/cm.



Figura 17: Servomotor

El motor funciona con combustible tipo Nitrometano y tiene un solo pistón con 15.30mm de recorrido. El bloque tiene una cilindrada de 3.4cc y potencia de 1.8cv que puede girar hasta 30.000 rpm (*figura 18*), lo suficiente para que el vehículo alcance hasta un máximo de 80 Km/h en superficie plana.



Figura 18: Motor de 3.4cc a combustión

El arrancador del motor es electrónico, activado por medio de un relé que proporciona energía suficiente a través de un batería que acciona un pequeño motor DC (*figura 19*) con una caja de engranajes acoplada al eje de giro del motor a combustión.

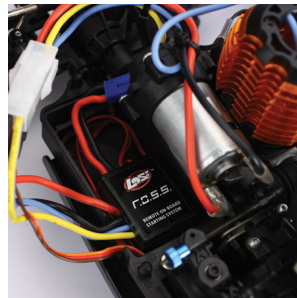


Figura 19: Arrancador electrónico del motor a combustión

El tanque de combustible tiene una capacidad de 75cc (*figura 20*), lo cual permite al motor del vehículo funcionar por aproximadamente 10 minutos, dependiendo del tipo de velocidades y aceleraciones a las cuales se someta.



Figura 20: Tanque de combustible



## ***2.2 Funcionamiento.***



Figura 21: Vehículo en funcionamiento

Para proporcionar energía eléctrica suficiente al sistema de encendido, dirección y aceleración del vehículo, este cuenta con una batería recargable de 7.4V, la cual canaliza la potencia necesaria a todos los componentes a través de un relé de protección contra sobrecargas, que también está conectado a un módulo electrónico en donde se conectan las terminales para los servomotores de dirección y aceleración. En este módulo también se encuentra el receptor inalámbrico del radio control del vehículo, el cual tiene un alcance de recepción de señal de hasta 800 metros a campo abierto, pero que no será necesario detallar ya que este sistema inalámbrico será desacoplado para poder integrar los nuevos módulos que utilizará el vehículo y que serán descritos posteriormente en el presente trabajo. Sin embargo, se fabricará un pequeño circuito que permita la selección en cuanto al control del vehículo, pudiendo elegir entre el control por medio del radiocontrol de fábrica ó el control por medio del sistema de simulación a través del computador.

El motor a combustión se pone en marcha mediante el arrancador electrónico, que se acciona al presionar el botón de encendido en el radiocontrol durante algunos segundos,

que conjuntamente acciona una pequeña bujía ubicada dentro de la cámara del pistón, en donde por medio de una pequeña cañería, se inyecta el combustible que es regulado por medio de un filtro con ajuste manual, para calibrar el estado de encendido.



Figura 22: Sistema de inyección de combustible

El combustible es inyectado al motor por medio de una pequeña manguera conectada al tanque (*figura 22*), que a su vez tiene otra manguera conectada al tubo de escape, el cual al expulsar los gases del motor, estos ejercen presión de aire por esta sección, y consecuentemente impulsan el combustible del tanque hacia el motor, con lo que se consigue que este mecanismo actúe a manera de “bomba de gasolina”. El motor es refrigerado con aire por medio de un radiador de aluminio que está acoplado sobre el bloque, para el cual existe una pequeña ventana en la carcasa del vehículo para que la corriente de aire que se genera conforme este avanza, circule a través del radiador evitando el sobrecalentamiento del motor. El eje de giro principal del motor, transfiere su potencia mediante un pequeño embrague con volante de inercia, hacia un sistema diferencial conectado a las cuatro ruedas, con lo que se consigue el movimiento del vehículo.

### 3. Accesorios adicionales

#### 3.1 Pantalla LCD



Figura 23: Televisor HDTV de 27" Norcent<sup>23</sup>

La pantalla que montaremos en el simulador para visualizar el video transmitido por el vehículo, será un televisor LCD de alta definición de 27 pulgadas (*figura 23*).

#### 3.2 Asiento



Figura 24: Asiento deportivo para automóvil<sup>24</sup>

---

<sup>23</sup> [23]

<sup>24</sup> [24]

En el simulador se montará este tipo de asiento (*figura 24*), ya que posee perfiles laterales que ayudan al cuerpo para mantener una posición fija y no resbalar al costado.

### 3.3 Control de mando



Figura 25: Logitech Momo racing wheel<sup>25</sup>

Este volante es un joystick para video juegos de computador (*figura 25*), que incluye pedales de aceleración y freno, para que de esta manera podamos manejar el vehículo como si fuese uno de verdad.

## II) División Electrónica

### 1. Control del simulador

Para el control electrónico de los movimientos del simulador, primero es necesario escoger los tipos de actuadores que se encargarán de ejercer el movimiento vertical y lateral del mismo, y luego los módulos programables que realizarán el control de dichos actuadores.

---

<sup>25</sup> [25]

### 1.1 Actuadores

Como actuadores se entiende a los dispositivos mecánicos, electromecánicos, neumáticos ó hidráulicos que pueden ejercer algún tipo de fuerza sobre otro objeto a través del movimiento de su mecanismo. Para nuestro objetivo, se ha determinado que los motores de corriente continua (DC) son el tipo de actuador que más se acoplan al tipo de función que realizarán para mover verticalmente y lateralmente la estructura móvil del simulador.

El tipo de motor que se utilizará es un motor DC (*figura 26*), que se utilizan para mover los limpia parabrisas de automóviles, ya que económicamente hablando son muy eficaces y tienen muy buenas prestaciones como son:



Figura 26: Motor DC de limpia parabrisas<sup>26</sup>

Technical parameter		
Rating Voltage	$U_s$	12V      24V
Rating Power	$P_s$	50W
No Load Current	$I_s$	2.0/2.5A      1.5/2.5A
No Load Speed	$n_s$	45/65 ± 5 RPM
Load Current	$I_b$	6.5/7.5A      3.5/5A
Load Speed	$n_b$	40/60 ± 5 RPM
Stall current	$I_A$	30A <sub>max</sub> 15A <sub>max</sub>
Stall Torque	$M_A$	29NM(min)
Protection Class		IP33
Weight	$W$	2kg

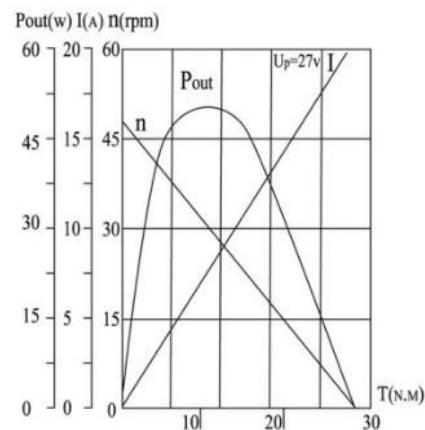


Tabla 1: Características del motor DC<sup>27</sup>

<sup>26</sup> [26]

Dentro de las características mostradas en la *tabla 1*, cabe resaltar su gran capacidad de torque de 29NM (newtons/metro) con lo que puede alcanzar hasta 65 rpm dependiendo de la fuerza a la que se someta. Con esto podemos asegurar que los movimientos del simulador tendrán una buena aproximación en cuanto a la respuesta de las señales, en el momento de sincronizarlos con los sensores del vehículo teledirigido. En la gráfica podemos ver que el consumo de corriente aumenta de acuerdo a la cantidad de torque ejercido, alcanzando una potencia máxima de aproximadamente 50 watts, por lo que es relevante para determinar el tipo de fuente de poder que utilizaremos como alimentación de energía, ya que los motores estarán casi siempre ejerciendo su máxima potencia debido al gran peso que necesitarán mover.

### ***1.2 Fuente de energía***

La fuente de energía o alimentación (*figura 27*), se encargará de suministrar la corriente necesaria a los dos motores DC, por ello requerirá de mucha potencia y de protecciones contra sobrecargas, y debido a esto, se eligió una fuente de poder para computador de escritorio que presenta las siguientes características:



Figura 27: Fuente de poder

---

<sup>27</sup> [27]

Specifications	Agiler AGI PS 800W
<b>Max Output</b>	<b>800 Watts</b>
<b>+5 Volt Rail:</b>	40 amps
<b>+12 Volt1 Rail:</b>	40 amps
<b>+12 Volt2 Rail:</b>	40 amps
<b>+3.3 Volt Rail:</b>	24 amps
<b>-5 Volt Rail:</b>	.5 amps
<b>-12 Volt Rail:</b>	.5 amps
<b>+5 Volt SB:</b>	2 amps

Tabla 2: Características de la fuente de poder<sup>28</sup>

Los motores utilizarán un suministro de 12 voltios y se estima que consuman hasta un máximo de hasta 15 amperios, por lo que la fuente de poder es más que suficiente ya que cuenta con varios canales que proporcionan 12, 5 y 3.3V (*tabla 2*), tomando en cuenta que el canal de 12 voltios puede suministrar hasta 40 amperios de corriente.

### ***1.3 Módulo de control***

El controlador de los motores será un módulo Arduino UNO (*figura 28*), que es un prototipo de fácil programación que ofrece salidas y entradas analógicas presentando las siguientes características:

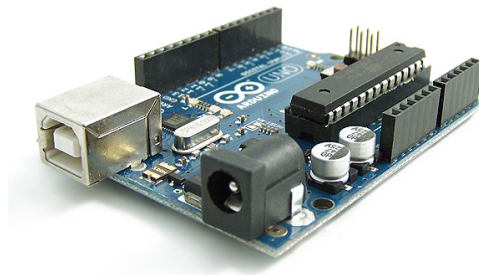


Figura 28: Módulo arduino UNO

---

<sup>28</sup> [28]

- 14 entradas/salidas digitales, de los cuales 6 pueden ser usados como salidas PWM
- Posee 6 entradas analógicas
- Los pin 0 y 1 pueden funcionar como RX y TX serial.
- Un oscilador de crystal de 16 MHz
- Conector USB
- Jack de poder
- Conector ICSP
- Botón de Reset

Características del microcontralor:

<b>Microcontroller</b>	<b>ATmega328</b>
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega 328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Tabla 3: Características microcontrolador<sup>29</sup>

En este módulo utilizaremos dos entradas analógicas (*tabla 3*) para monitorear las señales de retroalimentación de los motores, para que puedan ser programadas de tal manera que puedan detectar la posición de giro del eje del motor. También se utilizará tres salidas digitales para producir las señales PWM (pulse width modulation) para controlar los dos motores y el sistema de arranque del vehículo por medio del puerto USB del computador. Cabe mencionar que en el vehículo se montará un segundo módulo arduino

---

<sup>29</sup> [29]



para que pueda ser acoplado con el módulo de transmisión y que este procese también la señal del sensor de movimiento y transmita los datos hacia el computador. En resumen, tendremos un módulo arduino conectado a la computadora para controlar el simulador y otro módulo montado en el vehículo para su control inalámbrico.

#### ***1.4 Módulo de potencia***

Debido a que el control de potencia de los motores, será regulado por medio de señales PWM del arduino, es necesario la implementación de un módulo que permita el giro en ambas direcciones de los motores, para lo cual se utilizará el siguiente circuito conocido como “puente H” (*figura 29*) y que presenta las siguientes características:

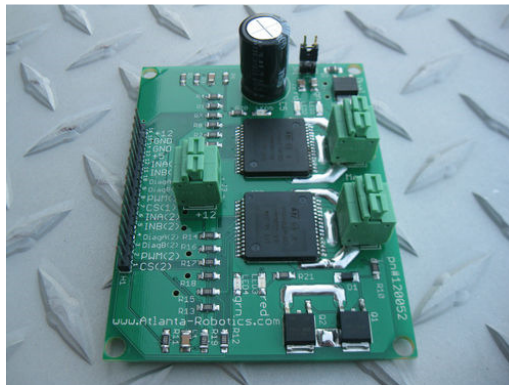


Figura 29: Puente H

## Features

Type	$R_{DS(on)}$	$I_{out}$	$V_{CCmax}$
VNH2SP30-E	19m $\Omega$ max (per leg)	30A	41V

- 5V logic level compatible inputs
- Undervoltage and overvoltage shut-down
- Overvoltage clamp
- Thermal shut down
- Cross-conduction protection
- Linear current limiter
- Very low stand-by power consumption
- PWM operation up to 20 kHz
- Protection against loss of ground and loss of  $V_{CC}$
- Current sense output proportional to motor current
- Package: ECOPACK®

Tabla 4: Características del puente H<sup>30</sup>

Como podemos ver en la *tabla 4*, el controlador puede manejar corrientes de hasta 30 amperios, con protección de sobrecargas e inclusive sobrecalentamientos, lo cual lo hace ideal para el manejo de los motores ya que de esta manera no se correrá el riesgo de quemarlos ni de dañar el módulo, gracias a los dos chips VNH2SP30-E de los que se describe las anteriores características.

## 2. Control del vehículo

### 2.1 Transmisor y receptor

Los módulos que se encargarán de transmitir las señales de los movimientos del vehículo y a la vez recibir los comandos del computador para maniobrarlo, deberán ser bidireccionales, es decir que tienen que ser capaces de enviar y recibir datos al mismo tiempo. Los módulos Xbee Pro (*figura 30*) son módulos inalámbricos programables que

---

<sup>30</sup> [30]

son bidireccionales y son compatibles con el arduino, presentando las principales características:



Figura 30: Módulos Xbee Pro

Specification	XBee	XBee-PRO
<b>Performance</b>		
Indoor/Urban Range	Up to 100 ft (30 m)	Up to 300 ft. (90 m), up to 200 ft (60 m) International variant
Outdoor RF line-of-sight Range	Up to 300 ft (90 m)	Up to 1 mile (1600 m), up to 2500 ft (750 m) international variant
Transmit Power Output (software selectable)	1mW (0 dBm)	63mW (18dBm)* 10mW (10 dBm) for International variant
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 bps - 250 kbps (non-standard baud rates also supported)	1200 bps - 250 kbps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
<b>Power Requirements</b>		
Supply Voltage	2.8 – 3.4 V	2.8 – 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)	250mA (@3.3 V) (150mA for international variant) RPSMA module only: 340mA (@3.3 V) (180mA for international variant)
Idle / Receive Current (typical)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	< 10 $\mu$ A	< 10 $\mu$ A
<b>General</b>		
Operating Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector, RPSMA Connector	Integrated Whip, Chip or U.FL Connector, RPSMA Connector
<b>Networking &amp; Security</b>		
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	PAN ID, Channel and Addresses

Tabla 5: Características del módulo Xbee/ Xbee Pro<sup>31</sup>

<sup>31</sup> [31]

La gran ventaja de estos módulos es que son de fácil programación y cuentan con software propio, lo cual permitirá la interacción con el arduino y sus puertos, pudiendo de esta manera integrar los dos módulos para que puedan transmitir las señales bidireccionalmente. Como podemos ver en la *tabla 5*, otra de las principales características, es que tienen una señal de alcance de hasta 1600 metros a campo abierto, a una frecuencia de 2.4 GHz, con la ventaja de poder establecer una conexión punto a punto entre los módulos, y de esta manera evitar que exista interferencia con otras señales externas y a su vez, que la señal de los módulos no interfiera con otras.

## ***2.2 Sensor de movimiento***

El sensor encargado de monitorear los movimientos y fuerzas de inercia, será un acelerómetro de tres ejes, el cual puede medir las fuerzas que ejerce el peso en los tres planos X, Y y Z. El acelerómetro que se utilizará es el MMA7361 (*figura 31*) y tiene las siguientes características:

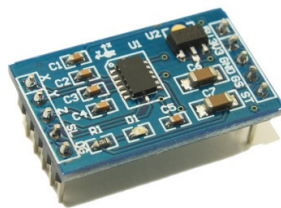


Figura 31: Acelerómetro de 3 ejes (MMA7361)

### Features

- 3mm x 5mm x 1.0mm LGA-14 Package
- Low Current Consumption: 400  $\mu$ A
- Sleep Mode: 3  $\mu$ A
- Low Voltage Operation: 2.2 V – 3.6 V
- High Sensitivity (800 mV/g @ 1.5g)
- Selectable Sensitivity ( $\pm$ 1.5g,  $\pm$ 6g)
- Fast Turn On Time (0.5 ms Enable Response Time)
- Self Test for Frefall Detect Diagnosis
- 0g-Detect for Frefall Protection
- Signal Conditioning with Low Pass Filter
- Robust Design, High Shocks Survivability
- RoHS Compliant
- Environmentally Preferred Product
- Low Cost

Characteristic	Symbol	Min	Typ	Max	Unit
Operating Range <sup>(2)</sup>					
Supply Voltage <sup>(3)</sup>	V <sub>DD</sub>	2.2	3.3	3.6	V
Supply Current <sup>(4)</sup>	I <sub>DD</sub>	—	400	600	$\mu$ A
Supply Current at Sleep Mode <sup>(4)</sup>	I <sub>DD</sub>	—	3	10	$\mu$ A
Operating Temperature Range	T <sub>A</sub>	-40	—	+85	$^{\circ}$ C
Acceleration Range, X-Axis, Y-Axis, Z-Axis					
g-Select: 0	g <sub>FS</sub>	—	$\pm$ 1.5	—	g
g-Select: 1	g <sub>FS</sub>	—	$\pm$ 6.0	—	g
Output Signal					
Zero-g (T <sub>A</sub> = 25 $^{\circ}$ C, V <sub>DD</sub> = 3.3 V) <sup>(5), (6)</sup>	V <sub>OFF</sub>	1.485	1.65	1.815	V
Zero-g <sup>(4)</sup>	V <sub>OFF, T<sub>A</sub></sub>	-2.0	$\pm$ 0.5	+2.0	mg/ $^{\circ}$ C
Sensitivity (T <sub>A</sub> = 25 $^{\circ}$ C, V <sub>DD</sub> = 3.3 V)					
1.5g	S <sub>1.5g</sub>	740	800	860	mV/g
6g	S <sub>6g</sub>	190.6	206	221.5	mV/g
Sensitivity <sup>(4)</sup>	S <sub>T<sub>A</sub></sub>	-0.0075	$\pm$ 0.002	+0.0075	%/ $^{\circ}$ C
Bandwidth Response					
XY	f <sub>-3dBXY</sub>	—	400	—	Hz
Z	f <sub>-3dBZ</sub>	—	300	—	Hz
Output Impedance	Z <sub>O</sub>	—	32	—	k $\Omega$
0g-Detect	0g <sub>detect</sub>	-0.4	0	+0.4	g

Tabla 6: Características del acelerómetro MMA7361<sup>32</sup>

Como vemos en la *tabla 6*, la velocidad de respuesta al movimiento de 0.5 ms (microsegundos) nos asegura que el sensor será lo suficientemente rápido como para detectar hasta la más mínima variación en el movimiento del vehículo, teniendo también la opción de poder disminuir la sensibilidad de detección de fuerzas de 1.5g a 6g por medio de uno de sus pines, lo que podría servir para disminuir los movimientos bruscos en el simulador en caso de que se requiera.

<sup>32</sup> [32]

El acelerómetro irá montado sobre la parte frontal del vehículo, para que sea más susceptible a los cambios terrenales como subidas o bajadas, y como ya se mencionó, este se conectará al arduino montado en el vehículo, para que sus señales sean enviadas inalámbricamente hacia el computador y puedan ser programadas hacia el simulador.

### 2.3 Cámara inalámbrica

La cámara que se utilizará y se montará en el vehículo es la siguiente:



Figura 32: Cámara HD GoPro Hero3<sup>33</sup>

- Videocámara HD profesional de hasta 120 fps (720p), 60 fps (1080p), 48 fps (1440p) y 240 fps (WVGA)
- Puerto MicroHDMI
- Puerto para tarjeta microSD expandible
- Captura de fotos de 12 megapíxeles con ráfagas de 30 imágenes por segundo
- Se incluyen Wi-Fi integrada y Mando a distancia Wi-Fi
- Micrófonos en estéreo y rendimiento profesional con poca luz
- Carcasa impermeable resistente hasta a 60 m de profundidad

---

<sup>33</sup> [33]

Esta cámara de alta definición (*figura 32*) posee entre sus diversas características el poder grabar con una velocidad de 60 cuadros por segundo con una resolución de 1080p, lo cual nos permitirá visualizar hasta el más mínimo cambio y vibración en la imagen, además de poder visualizar el panorama en un ángulo abierto ya que posee un lente integrado que permite dicha función. También incluye una carcasa reforzada a prueba de agua, lo que protegerá la cámara en caso de que el vehículo sufra un fuerte choque o se vuelque.

Lamentablemente el sistema WiFi que incorpora la cámara para transmisión de imagen no es muy eficaz, ya que fue diseñado para pre-visualizar la imagen solamente al momento de tomar fotos, mas no para que transmita en tiempo real, por lo que tiene un pequeño retraso al momento de visualizar el video. Para resolver este problema, se utilizará un transmisor de video especial para imágenes HD que irá conectado a través del puerto micro HDMI de la cámara para transmitir el video en tiempo real hacia la pantalla LCD.

El transmisor de video (*figura 33*) que se utilizará se describe a continuación:



Figura 33: Transmisor de video HD EP3207<sup>34</sup>

---

<sup>34</sup> [34]

*Transmitter End (TX)*

- Power Input: Micro USB
- Battery Life: Approximately 2.5hrs
- Battery Type: Li-ion battery 1300mAh
- I/O Connector: HDMI Type A(Male)x1, Micro-USB x1, Power switch x1
- HDMI Connector: HDMI 1.4
- Supported Devices: Portable device with HDMI connector.
- Video Formats: 1080p(Full HD), 720(HD), 480P
- Size: 60 x 52 x 15 mm / 2.36 x 2.05 x 0.59 inch

*Receiver End (RX)*

- Power Input: 5.0V/2A, Micro USBx1
- I/O Connector: Micro-USB x1, HDMI Type A(Female)
- Operating Frequency: Wi-fi 802.11n standard 5Ghz
- Size: 56 x 48 x 11 mm / 2.2 x 1.89 x 0.43 inch
- Net Weight: 67 g / 2.36 oz (included receiver and transmitter)<sup>35</sup>

El protocolo WiFi de comunicación que utiliza este dispositivo es el 802.11n, el cual es un estándar que permite la transmisión de datos a una velocidad de 108 Mbps (megabits por segundo), lo cual lo hace relevante al momento de transmitir video en alta definición sin retraso y en tiempo real. El tamaño del dispositivo transmisor y receptor es relativamente pequeño y poseen baterías recargables integradas para una duración de hasta dos horas en la transmisión de video de diferentes resoluciones.

---

<sup>35</sup> [35]



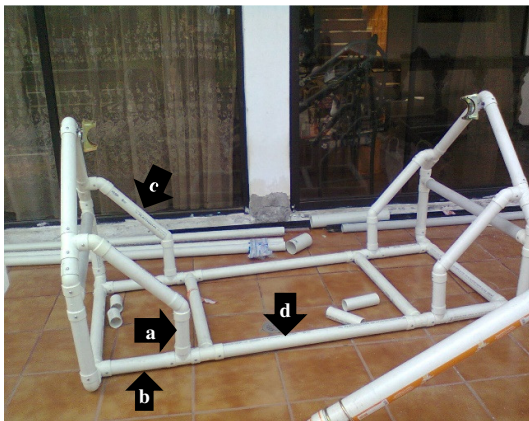
## CAPITULO 2: Construcción y montaje

### I) Simulador

#### 1. Estructura y chasis

La construcción del simulador se basará en el bosquejo ya presentado con algunas modificaciones, del cual para la estructura base se utilizó:

- Tubos PVC para desagüe de 3cm de diámetro
- 6 uniones de 90 grados
- 12 uniones de 45 grados
- 12 uniones tipo “T”



Las medidas de los cortes para la estructura base (figura 34 y 35) son:

- 4 piezas de 17 cm
- 4 piezas de 26 cm
- 4 piezas de 35 cm
- 2 piezas de 80 cm

Figura 34: Base (vista lateral)

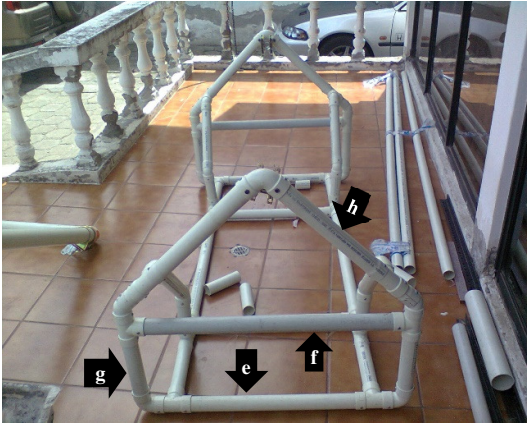


Figura 35: Base (vista frontal)

- e. 4 piezas de 70 cm
- f. 2 piezas de 80 cm
- g. 4 piezas de 27 cm
- h. 4 piezas de 41 cm



Figura 36: Abrazaderas metálicas

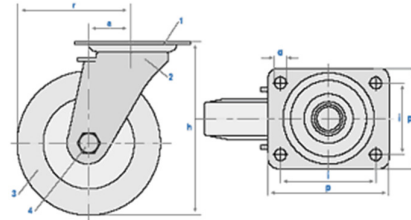


Figura 37: Rueda giratoria

Para que la base móvil se pueda mover lateralmente (*figura 38*), se utilizará ruedas giratorias (*figura 37*) con placa metálica.

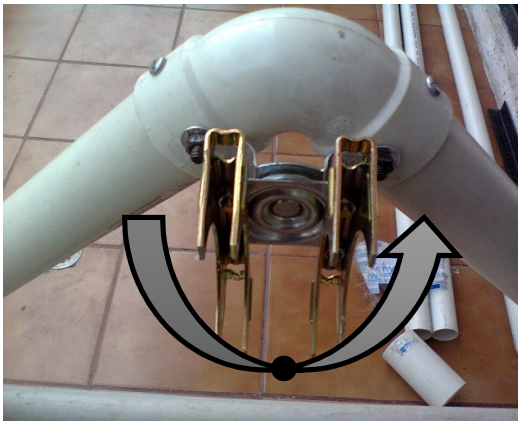
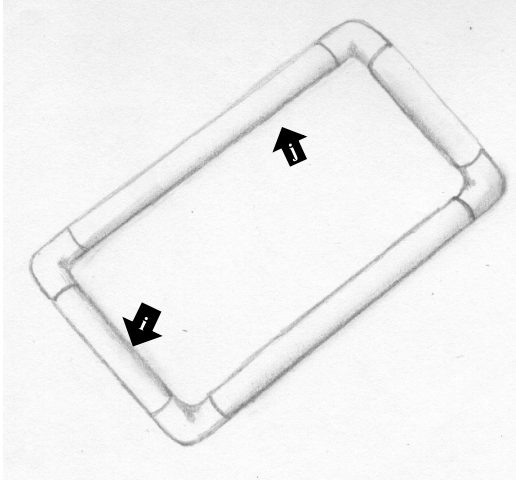


Figura 38: Movimiento lateral

Las ruedas se modificarán de tal manera que las placas que sostienen la rueda serán dobladas hacia afuera y sujetadas con dos pernos a través de la unión en los tubos. La otra placa cuadrada se acoplará también con pernos a dos abrazaderas metálicas (*figura 36*).



Para la base móvil horizontal (*figura 39*) utilizaremos tubos PVC de 9 cm de diámetro y 4 uniones de 90 grados con los siguientes cortes:

- i. 2 piezas de 58 cm
- j. 2 piezas de 145 cm

Figura 39: Base móvil horizontal



Se utilizarán ruedas giratorias de mayor tamaño (*figura 40*) para sujetar la base que ejercerá los movimientos verticales.

Figura 40: Rueda giratoria grande



No se requerirá ninguna modificación en las ruedas giratorias (*figura 41*) a excepción de los orificios para los pernos que sujetaran las abrazaderas y el tubo de la base móvil vertical.

Figura 41: Base giratoria

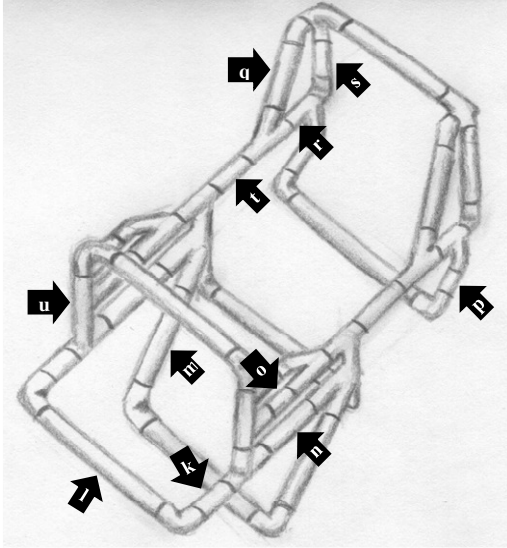


Figura 42: Base móvil vertical

- 12 uniones de 90 grados
- 6 uniones de 45 grados
- 4 uniones en “Y”
- 14 uniones en “T”

La base móvil vertical (*figura 42*) utilizará también tubos de 3cm de diámetro con los siguientes cortes:

- k. 2 piezas de 16 cm
- l. 6 piezas de 40 cm
- m. 2 piezas de 46 cm
- n. 2 piezas de 21 cm
- o. 2 piezas de 11 cm
- p. 2 piezas de 14 cm
- q. 2 piezas de 36 cm
- r. 2 piezas de 19 cm
- s. 2 piezas de 25 cm
- t. 2 piezas de 13 cm
- u. 2 piezas de 10 cm



Figura 43: Acople con la base vertical

Al acoplar la base giratoria con el tubo principal de la base vertical (*figura 43*), es importante determinar el centro de gravedad con el asiento, para equilibrar el peso de la persona con el resto de accesorios que se colocarán.



Figura 44: Asiento montado

La base del asiento (*figura 44*) es de fibra de vidrio hecho a medida, para disminuir el peso y optimizar medidas e irá sujeta con pernos sobre las bases que se dispusieron en el bosquejo del diseño.



Figura 45: Movimiento vertical y lateral

En este punto hay que probar que la estructura soporte el peso suficiente y que sus movimientos son los adecuados, sin que existan obstrucciones ni colapsos por parte de la estructura. El movimiento lateral tendrá un giro de 45 grados hacia ambos lados y el vertical un máximo de 35 grados hacia arriba o hacia abajo (*figura 45*).



Figura 46: Estructura completa

Luego de acoplar la estructura base y móvil, se montará el volante y los pedales, con lo cual se utilizó una base de madera y dos piezas con perfiles de estantería para sostener los pedales (*figura 46*). Estas medidas son de referencia ya que dependerá del tipo de asiento y comodidad del constructor.



Figura 47: Montaje de la pantalla

Nuevamente se utilizaron perfiles de estantería para sujetar la pantalla y de igual manera las medidas son referenciales dependiendo el tipo y tamaño de pantalla que se vaya a utilizar como se ve en la *figura 47*.



Figura 48: Montaje completo

Finalmente, en la estructura base fue necesario adicionar un tubo adicional en la mitad mediante dos uniones tipo “T”, ya que al montar todo el peso, la estructura no tenía mucha estabilidad en esta zona, pero su modificación fue sencilla y sin mayor problema (*figura 48*).

## 2. Montaje de motores

Los motores para limpia parabrisas de autos sólo giran un cierto ángulo repetidas veces y a diferentes velocidades, por lo que es necesario una pequeña modificación interna para lograr que funcionen como un motor DC normal:

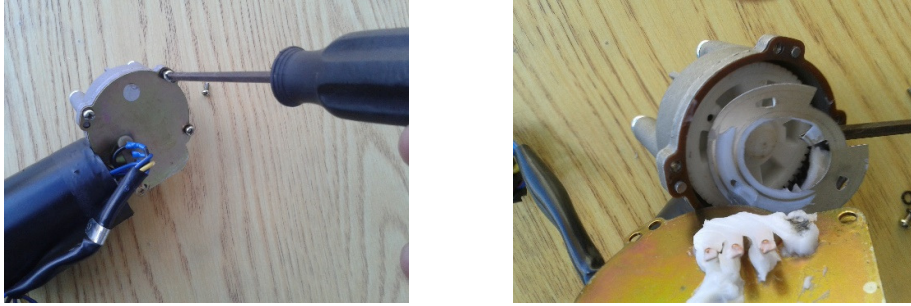


Figura 49: Modificación del giro de los motores

Existen contactos internos que requieren ser desechados para que el motor gire libremente hacia ambos lados al momento de aplicarle corriente. Una vez que se haya modificado el motor (*figura 49*), también será necesario modificarlo nuevamente, ya que se comprobó que el sentido de giro hacia un sentido es más rápido que el otro.



Figura 50: Modificación en velocidad de giro del motor

Para corregir este inconveniente fue necesario girar la tapa del rotor aproximadamente 20 grados en sentido horario (*figura 50*) y se procedió a conectar los motores mediante relés y un pequeño programa a través del arduino conectado al computador únicamente para aplicar señales PWM que permitan girar el motor con distintas velocidades aleatorias, únicamente para comprobar que estas sean las mismas en

ambos sentidos de giro de los motores y de esta manera asegurar el correcto movimiento en el simulador (*figura 51*).

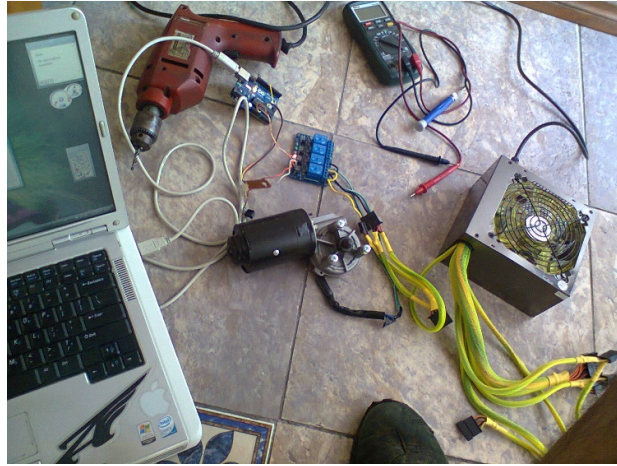


Figura 51: Pruebas de los motores



Figura 52: Montaje motor vertical

El motor para el movimiento vertical se acopló por medio de dos cortes con perfiles de estantería de 94 cm y separados 14 cm uno del otro (*figura 52*). El motor se colocó sobre la parte superior y se fabricó una polea de 9 cm de diámetro para cadenilla de bicicleta. En la parte inferior también se colocó otra polea guía para que el mecanismo logre un circuito cerrado con la cadenilla.



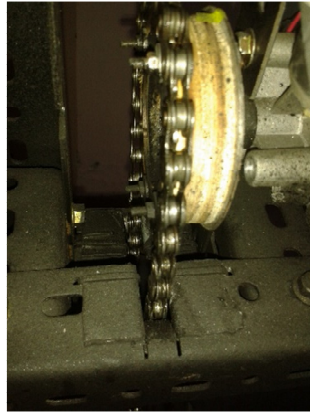


Figura 53: Polea con piñón y guía  
cadena

La cadenilla se encargará de subir y bajar la estructura móvil vertical del simulador, para lo cual fue necesario fabricar un guía cadena con un pedazo de perfil y de esta manera asegurar que la cadenilla no se salga del piñón al realizar los giros como se ve en la *figura 53*.

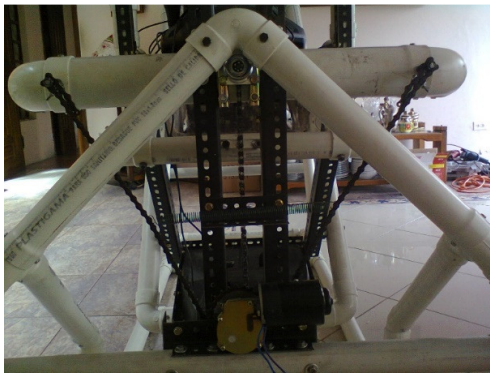


Figura 54: Montaje motor horizontal

El motor para los movimientos laterales (*figura 54*) también utilizará el mismo mecanismo y se lo acopló en la mitad de la estructura base. La cadenilla se sujetó en dos puntos de la estructura móvil lateral por medio de dos pernos.



Figura 55: Resortes de tensión

Para lograr tensión en la cadenilla al momento de realizar los giros del motor, se utilizaron dos resortes metálicos (*figura 55*), y de esta manera asegurar que la cadenilla no se salga del piñón de giro.



Se aseguró la estructura con soportes laterales tanto en la parte superior e inferior. Todo el mecanismo se realizó con perfil de estantería y pernos de 1 cm con tuercas de seguridad para sujetar las diferentes uniones (figura 56).

Figura 56: Mecanismo completo

Luego de tener todo el mecanismo armado, se procedió a pintar todo el chasis de color negro y tapizar el asiento como se muestra en la figura 57:



Figura 57: Chasis pintado

Posteriormente se procedió a realizar las respectivas pruebas de movimiento de las estructuras por medio de conexión directa de los motores hacia la línea de 12V de la fuente de alimentación a través de un switch, y se pudo comprobar que los movimientos de ambas estructuras laterales y verticales cumplieran con su función de giro previsto, pero

se tuvo una deficiente velocidad y torque de los motores debido a que el peso de todo el simulador excedía la capacidad de los motores para que cumplan su función, por ello fue necesario acoplar bandas elásticas (*figura 58*) a los lados de la estructura lateral y en la parte posterior de la estructura móvil vertical, para que de esta manera al momento en que la estructura gire y se complete el ángulo giratorio de ambos movimientos, las bandas elásticas incrementarán la fuerza de retorno para ayudar a que los motores ganen velocidad para que puedan dar el torque adecuado en los movimientos:



Figura 58: Bandas elásticas de tubo de llanta

Las bandas elásticas se colocaron de manera que permanezcan estiradas a la mitad de su capacidad en la posición original del simulador, por lo que al realizar el movimiento lateral o vertical en su ángulo límite de giro, estas alcancen su máximo estiramiento, imposibilitando que se sobrepase el ángulo máximo del movimiento.

Por último se adjuntó con pernos dos perfiles de estantería a los bordes de los tubos de la estructura lateral (*figura 59*), ya que debido al peso estaba propensa a doblarse y perder rigidez:



Figura 59: Chasis y estructura completa

### 3. Montaje y descripción de circuitos

Se modificó el case de la fuente para que pueda ser sujeta con un perfil de estantería atornillado horizontalmente en la parte frontal del simulador (*figura 60*) para que fácilmente los cables de alimentación de energía sean lo más cortos posibles para que se conecten con los dos motores:



Figura 60: Fuente de alimentación y circuitos

Los módulos controladores (arduino y puente H) se atornillaron sobre una placa de acrílico y esta a su vez sobre la parte superior de la fuente de alimentación. El esquema de las conexiones se realizó con el software FRITZING y es el siguiente:

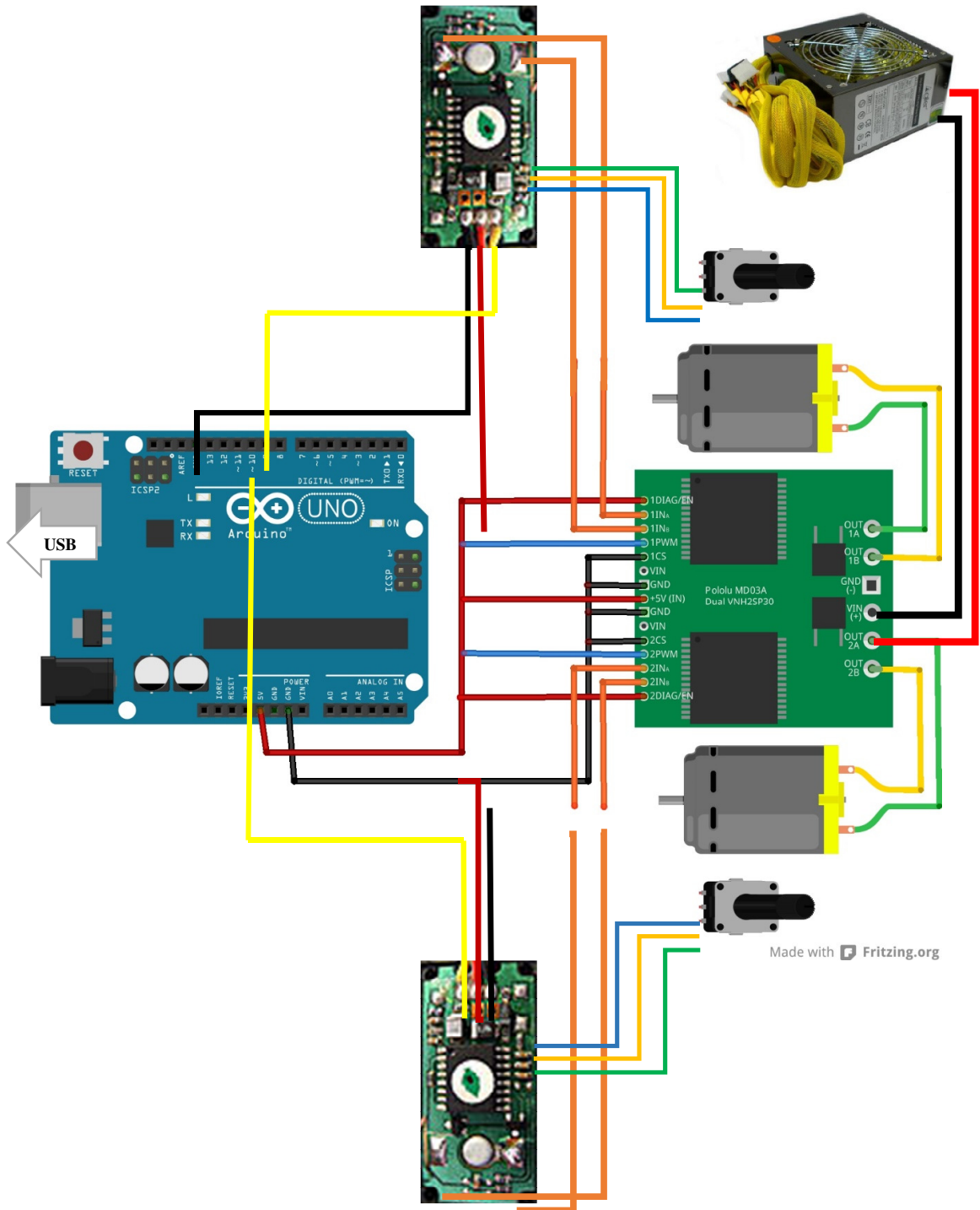


Figura 61: Diagrama de conexión de los circuitos de control para los motores

Como se puede ver en la *figura 61*, se añadió un nuevo circuito que contiene componentes de los que se utiliza para manejar un servomotor pequeño, y con esto

lograremos que los motores DC de parabrisas, se conviertan en servomotores controlados por una simple señal PWM con retroalimentación por medio de los potenciómetros que se instalarán posteriormente.

La alimentación para los circuitos controladores es proporcionada por el módulo arduino mediante los pines de 5V y GND respectivamente, y este irá conectado al PC por medio del puerto USB. Los pines 9 y 10 del arduino, servirán para emitir la señal PWM desde el PC, que indicarán la posición o el ángulo a la que los motores se deben mover, mediante los módulos de servomotor que se acoplarán y que tienen la siguiente configuración (figura 62):

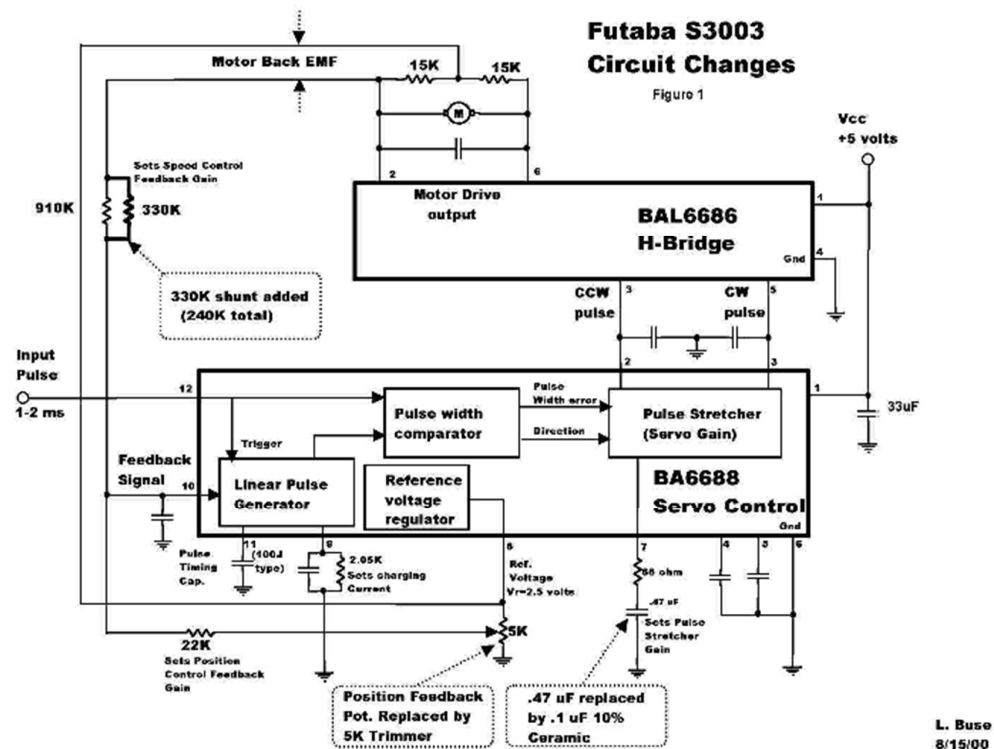


Figura 62: Diagrama del circuito del módulo de control para servomotor

Estos módulos en los servomotores generalmente tienen la misma configuración y no varían, es decir tienen tres líneas de conexión, de las cuales dos de ellas son para la alimentación y la otra es para recibir la señal PWM del controlador que se utilice. La señal

PWM con la que trabaja este circuito es de un rango de 1 hasta 2 ms (milisegundos) y dependiendo del valor que tenga en este rango, indicará al motor para que gire hasta un cierto ángulo. Esto lo podemos observar en el diagrama anterior, ya que la señal PWM ingresa simultáneamente a un generador de pulsos a manera de disparador y a un circuito comparador, los cuales dan como resultado una señal para que sea acondicionada y enviada hacia un puente H que controlará y determinará la dirección en la que se debe mover el motor. El eje de giro del motor está acoplado a un potenciómetro, el cual proporcionará la retroalimentación necesaria para indicar hasta qué punto debe girar el motor, por lo que ingresará al circuito de generación de pulsos para indicar un determinado valor y decidir si se debe seguir emitiendo el proceso para que gire el motor o detenerlo de acuerdo a la señal PWM que se reciba.

Las dos líneas de salida que normalmente se conectan a los motores dc pequeños presentes en cualquier servomotor, se utilizarán como señales de control para el módulo de potencia (puente H) que moverán finalmente los motores DC del simulador, mediante los pines INa y INb del módulo para cada motor, por lo que tendremos 4 conexiones entrantes para este propósito. Los pines de señal de PWM del puente H de potencia, serán conectados a la línea de 5V, ya que de esta manera los motores funcionarán a su máxima velocidad de giro y de igual manera los pines DIAG/EN irán conectados a la misma línea para poder habilitar el control del motor tal como se muestra en la *tabla 7* de funcionamiento del datasheet:

IN <sub>A</sub>	IN <sub>B</sub>	DIAG <sub>A</sub> /EN <sub>A</sub>	DIAG <sub>B</sub> /EN <sub>B</sub>	OUT <sub>A</sub>	OUT <sub>B</sub>	CS	Operating mode
1	1	1	1	H	H	High Imp.	Brake to V <sub>CC</sub>
	L				I <sub>SENSE</sub> = I <sub>OUT</sub> /K	Clockwise (CW)	
0	1			L		H	High Imp.
	0			L	L	High Imp.	Brake to GND

Tabla 7: Modo de operación del puente H<sup>36</sup>

Los pines CS (current sense) son salidas que indican un valor proporcional de la corriente con la que están trabajando los motores, pero en el presente propósito no serán utilizados y se los conectará a la línea de tierra. Finalmente, los pines de salida OUT irán conectadas a los motores, y la fuente de poder alimentará los pines alimentación de los mismos, con lo cual es importante distinguir estas líneas de alimentación ya que el módulo opera con distintos tipos de suministro de energía independiente para el funcionamiento del módulo en sí, y para los motores.

Como ya se mencionó, los motores requerirán proporcionar una señal para obtener la ubicación de su giro, lo que hizo necesario adaptar potenciómetros de 10k Ohms en su eje giratorio, con la ayuda de engranajes y perfiles pequeños de aluminio (*figura 63*):

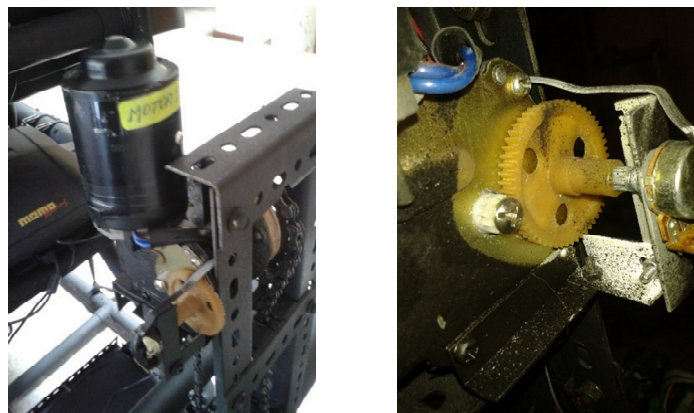


Figura 63: Adaptación de potenciómetros en los motores

<sup>36</sup> [30]



Se utilizaron piñones reductores debido a que el potenciómetro tiene un límite de 270 grados en su giro, y el eje del motor debido al recorrido que debe realizar puede girar hasta 3 vueltas, por ello con el uso de los engranajes se logró que el giro del motor entre en el rango de giro del potenciómetro.

Como paso final, se acopló un disipador de calor y dos ventiladores de 12V (*figura 65*) conectados a la fuente principal para el módulo del puente H, debido a que al ser un módulo de potencia sus chips tienden a sobrecalentarse con el uso continuo, especialmente si se va a trabajar con señales que cambian drásticamente en milésimas de segundo.

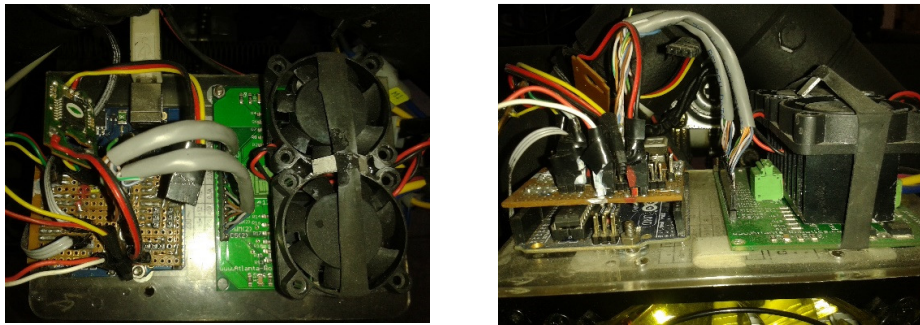


Figura 65: Disipador de calor y ventiladores de 12V

## II) Vehículo

### 1. Circuito selector

En el vehículo teledirigido fue necesario fabricar una pequeña placa PCB para crear un circuito que permita la selección del control del mando entre el PC y el control remoto original, con lo cual se procura no perder la funcionalidad original del mismo. El diagrama del circuito se presenta en la *figura 65*:

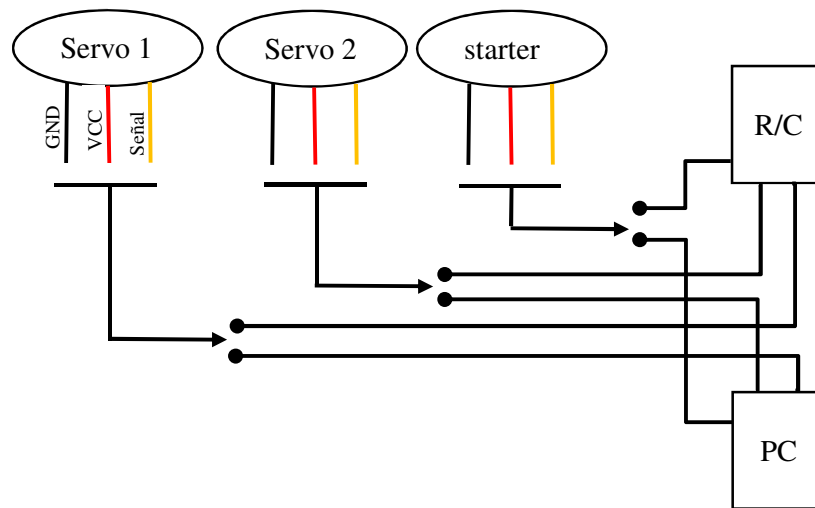


Figura 65: Diagrama del circuito de selección

La placa PCB (*figura 66*) permitirá intercambiar las señales que son emitidas para controlar los servomotores del vehículo entre el PC y el radio control, por medio de “jumpers” y un switch independiente para la alimentación. Adicionalmente se agregó dos líneas con interruptor conectados a una batería de 9V, que servirá para habilitar el funcionamiento del arduino, ya que la batería original del vehículo únicamente puede proporcionar 7V, la cual será la fuente de energía para los servomotores que controlan la dirección y aceleración, además de proporcionar la corriente suficiente que consume el arrancador electrónico del motor a combustión.

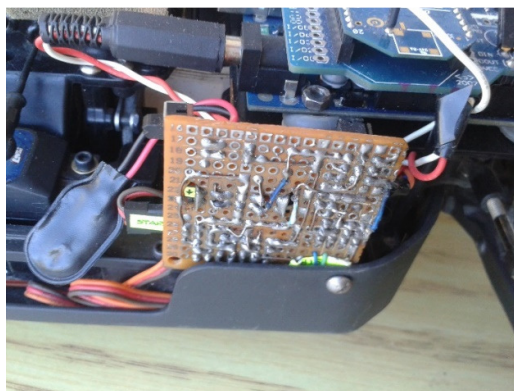


Figura 66: Placa del circuito selector

## 2. Módulo transmisor y receptor

El módulo arduino tiene gran variedad de adaptadores para sus diferentes componentes y sensores para los que es compatible. Estos adaptadores comercialmente se los denomina “shields” y tienen la facilidad de acoplarse a los pines del arduino sin necesidad de hacer ninguna conexión mediante cables o adicionar componentes adicionales, ya que los shields incluyen íntegramente cualquier otro componente adicional para que el sensor o componente para el cual fueron diseñados, funcione correctamente.

El módulo Xbee Pro, será integrado al arduino mediante un shield que permitirá también el uso de todos los pines del arduino incluyendo sus entradas de lectura analógica y salidas digitales, así como también su línea auxiliar para alimentación de 5V.

En *tabla 8* podemos ver una descripción sencilla de todos los pines del módulo Xbee Pro:

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	DO8*	Output	Digital Output 8
5	RESET	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	RTS / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

Tabla 8: Descripción de pines del módulo Xbee Pro<sup>37</sup>

<sup>37</sup> [31]

Se utilizarán únicamente los pines de alimentación y los pines de transmisión y recepción de datos (2 y 3), ya que las entradas analógicas serán utilizadas directamente del arduino para la conexión de las señales del acelerómetro. Debido a que el shield de este módulo es “plug and play” únicamente tendremos que preocuparnos de situarlo en la manera correcta sobre el arduino, tomando en cuenta que el shield nos proporciona dos modos de trabajo que son el modo USB y el modo Xbee. El primero sirve para programar el módulo directamente mediante el PC a través del cable USB, mientras que el segundo modo es necesario cuando el módulo ya está programado y permita la transmisión de los datos inalámbricamente, por lo cual hay que posicionar los jumpers (*figura 67*) de acuerdo al modo en el que se esté operando:

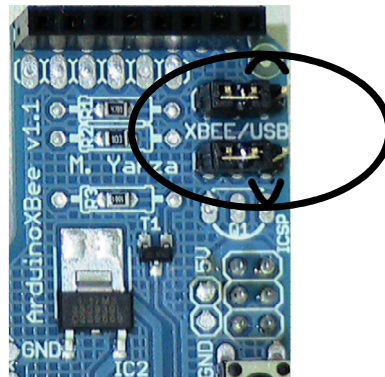


Figura 67: Jumpers de selección de modo del shield Xbee

Los módulos se montarán sobre una base de baquelita (*figura 68*), ya que es un material resistente al calor y fácil de modificar, en donde para su colocación se agregaron pequeños soportes de silicón con el objetivo de reducir la vibración en el módulo e impedir que se produzca cualquier tipo de ruido en las señales monitoreadas:



Figura 68: Base y soportes del módulo en el vehículo

### 3. Sensor de movimiento

El acelerómetro será montado sobre la parte frontal del vehículo (*figura 69*) ya que de esta manera se podrá medir con mayor resolución los movimientos las fuerzas que se produzcan entre el vehículo y la superficie, además de poder amortiguar con mayor eficiencia las vibraciones producidas por el motor, debido a que esta sobre los ejes que sujetan la suspensión de las ruedas delanteras:

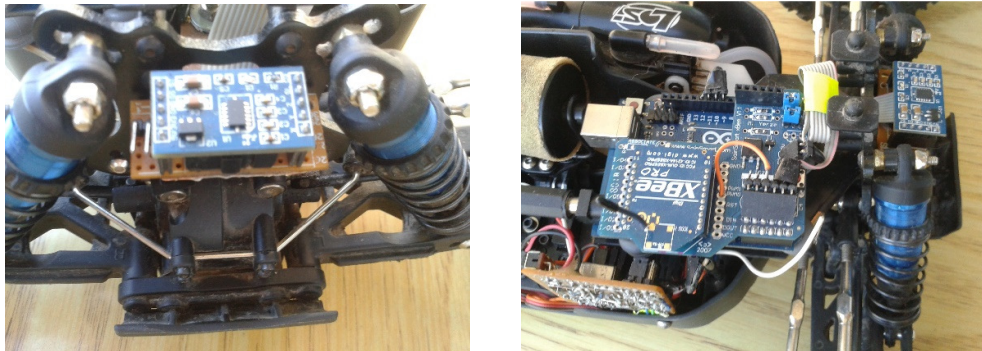


Figura 69: Montaje del acelerómetro

Este sensor tiene un rango de funcionamiento para emitir su señal desde 0V hasta 5V en cada uno de sus ejes (x,y,z) y tiene la siguiente configuración en sus pines, como muestra en la *figura 70*:

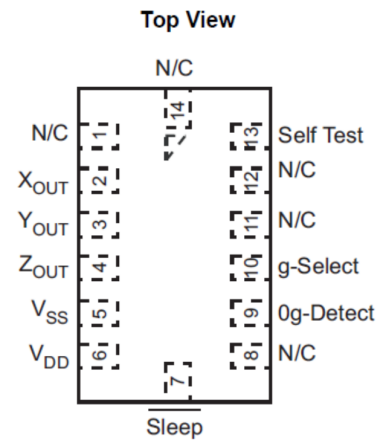


Figura 70: Configuración de pines del acelerómetro<sup>38</sup>

Los pines 2, 3 y 4 irán conectados a las entradas de lectura analógica del arduino para que se pueda monitorear y procesar las señales posteriormente en el PC. El pin 10 lo utilizaremos como pin auxiliar para que sea conectado mediante un jumper a la línea de 5V, en caso de que se requiera disminuir la resolución de los movimientos en el vehículo y el resto de pines a excepción de los pines de alimentación de energía, serán conectados a la línea de tierra o no tendrán conexión.

#### 4. Esquema de conexiones

Se presenta a continuación el esquema completo de las conexiones en el vehículo:

<sup>38</sup> [32]

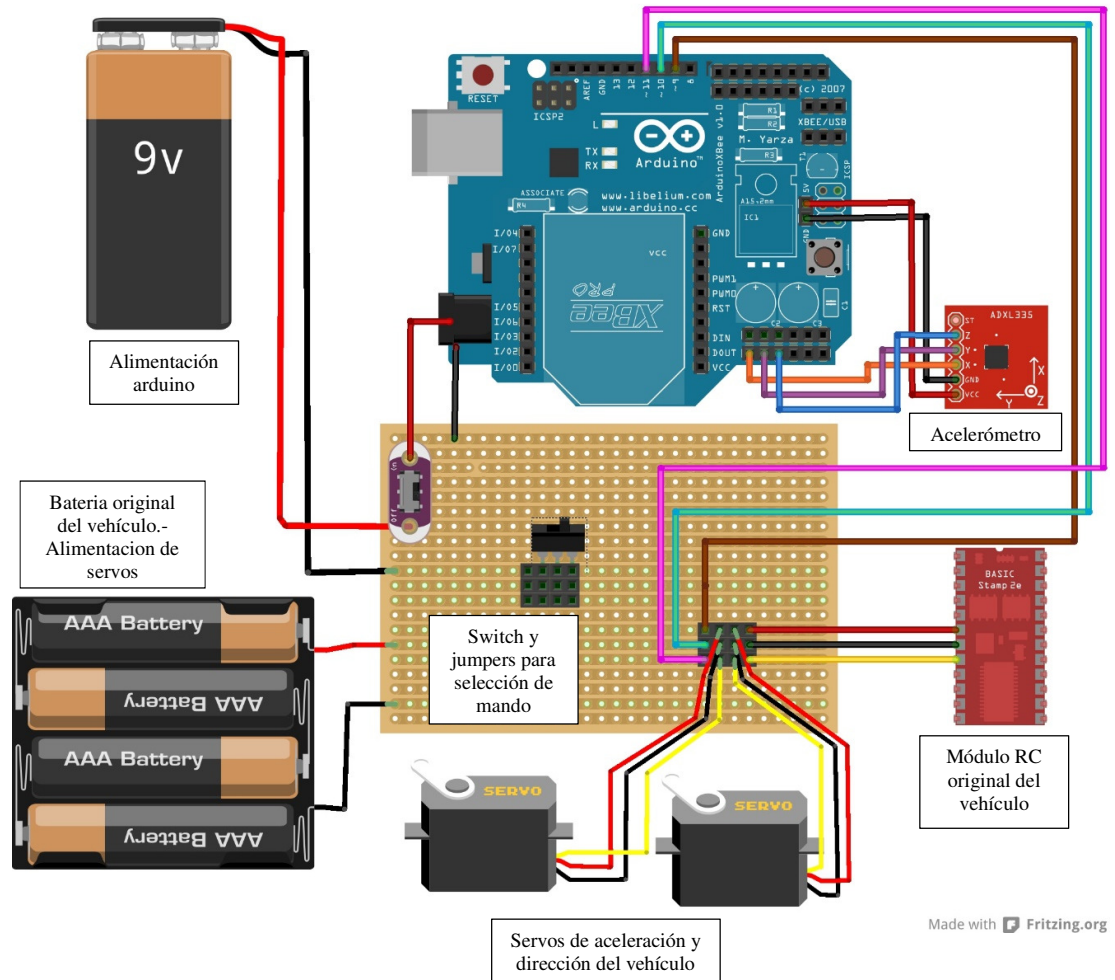


Figura 71: Esquema de conexión de módulos del vehículo

Del esquema de la *figura 71* podemos observar que la alimentación del arduino es a través de la batería de 9V mediante un interruptor. El acelerómetro es energizado mediante las líneas de 5V del shield en el arduino y sus salidas de señal en los ejes de movimiento (x, y z) son conectados a los pines de lectura analógica en el arduino en sus pines A0, A1 y A2 respectivamente. Las salidas PWM del arduino en los pines 9, 10 y 11 permitirán el control de los servos y arranque del motor a combustión y son conectados a los pines de los jumpers para que puedan ser deshabilitados en caso de querer operar el vehículo con el radio control original (*figura 72*).

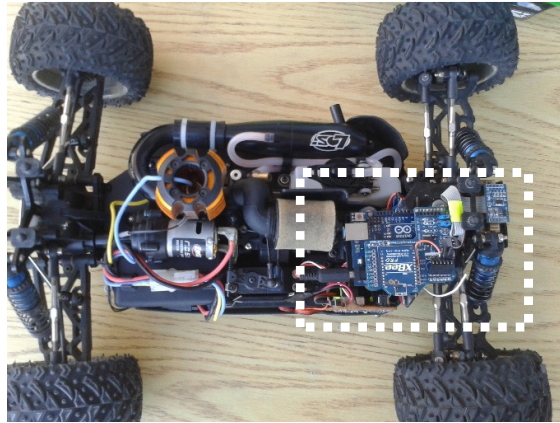


Figura 72: Montaje total de módulos en el vehículo

### 5. Montaje de la cámara

La cámara de alta definición fue sujeta mediante un soporte pegado con cinta de doble faz sobre la parte media de la carcaza del vehículo y acoplada al módulo transmisor de imagen de alta definición mediante una banda elástica, como se ve en la *figura 73*:



Figura 73: Cámara HD y módulo transmisor

La cámara se conecta al módulo transmisor mediante un cable del tipo microHDMI a HDMI ya que tanto la cámara como el módulo poseen puertos diferentes y por ello fue



necesario modificar la carcasa de la cámara para poder conectar el cable, mediante un pequeño orificio en el costado (*figura 74*).



Figura 74: Cámara montada en el vehículo

## CAPITULO 3: Programación y funcionamiento

### I) Programación

En el presente y último capítulo se va a describir la configuración utilizada en cada uno de los módulos y la programación a través de la interfaz gráfica utilizando el software Labview edición 2012, así como también su esquema de funcionamiento y argumentación.

#### 1. Configuración de módulos

Los módulos controladores necesitan ser programados para que puedan interactuar con el software mencionado, para lo cual será necesario tener instalado un paquete de controladores específicos del Labview denominados “toolkits”. El toolkit de arduino está disponible para descarga libre en su página web (<http://www.ni.com>).

##### *1.1 Arduino*

El módulo Arduino UNO tiene software propio disponible en su página web (<http://www.arduino.cc>) de donde descargaremos la última versión compatible con nuestro sistema operativo. Una vez instalado el programa, lo ejecutamos y nos dirigiremos a la pestaña FILE > Open (*figura 75*):

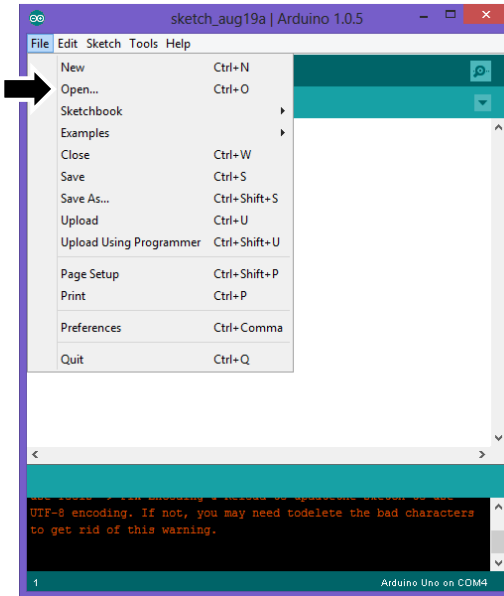
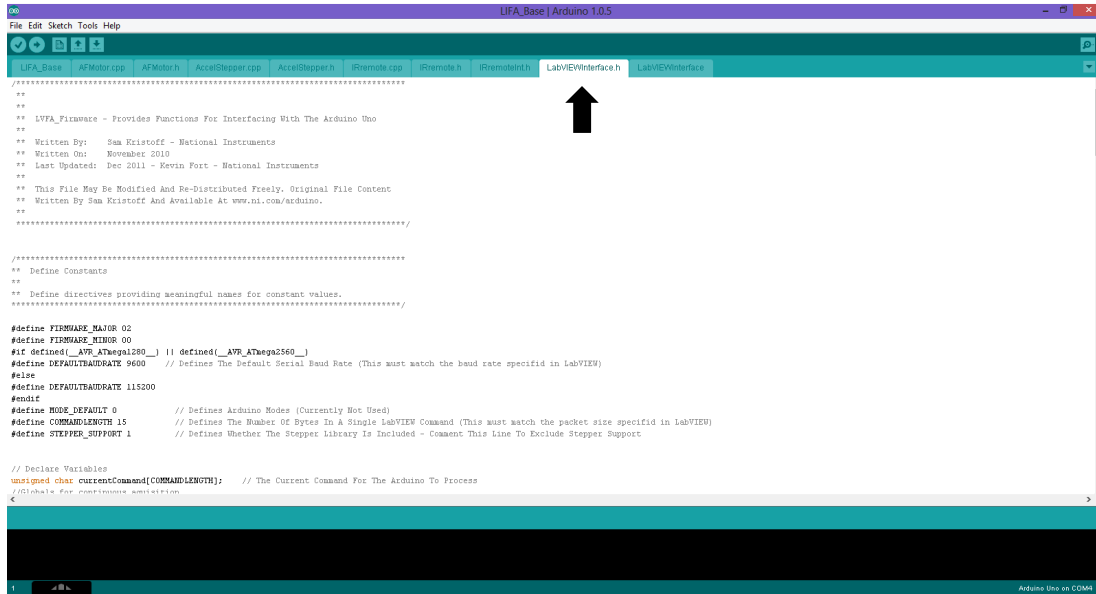


Figura 75: Software arduino

Luego nos dirigiremos al directorio en donde tenemos instalado Labview y abriremos el archivo de configuración para el arduino que por defecto tiene la siguiente dirección:

c:\Program Files\National Instruments\Labview 2012\vi.lib\labview interface for arduino\firmware\LIFA\_Base\Lifa\_base.ino

Los programas del arduino se denominan sketches y pueden tener varias ventanas en las que se muestra el código de programación (*figura 76*):



```

LFA_Base | Arduino 1.0.5
File Edit Sketch Tools Help
LFA_Base AFMotor.cpp AFMotor.h AccelStepper.cpp AccelStepper.h IRremote.cpp IRremote.h IRremoteInt.h LabVIEWInterface.h LabVIEWInterface
*****
**
** LFA_Firmware - Provides Functions For Interfacing With The Arduino Uno
**
** Written By: Sam Kristoff - National Instruments
** Written On: November 2010
** Last Updated: Dec 2011 - Kevin Fort - National Instruments
**
** This File May Be Modified And Re-Distributed Freely. Original File Content
** Written By Sam Kristoff And Available At www.ni.com/arduino.
**
*****

//*****
** Define Constants
**
** Define directives providing meaningful names for constant values.
*****

#define FIRMWARE_MAJOR 02
#define FIRMWARE_MINOR 00
#ifndef defined(__AVR_ATmega2560__)
#define DEFAULTBAUDRATE 9600 // Defines The Default Serial Baud Rate (This must match the baud rate specified in LabVIEW)
#else
#define DEFAULTBAUDRATE 115200
#endif
#define MODE_DEFAULT 0 // Defines Arduino Mode (Currently Not Used)
#define COMMANDLENGTH 15 // Defines The Number Of Bytes In A Single LabVIEW Command (This must match the packet size specified in LabVIEW)
#define STEPPER_SUPPORT 1 // Defines Whether The Stepper Library Is Included - Comment This Line To Exclude Stepper Support

// Declare Variables
unsigned char currentCommand[COMMANDLENGTH]; // The Current Command For The Arduino To Process
//Global for continuous execution

```

Figura 76: Programación principal del arduino

En la pestaña “LabVIEWInterface.h” podremos observar la codificación principal del programa, en donde tendremos en cuenta las variables de velocidad de comunicación “DEFAULTBAUDRATE” en donde podremos cambiar la velocidad de transmisión de datos entre el arduino y el PC, en caso de existir problemas de configuración al correr los futuros programas.

Para la configuración del módulo que estará conectado al PC por medio de cable USB y controlará el simulador, necesitamos mantener la velocidad en 115200 bps. A continuación en la pestaña TOOLS del menú principal seleccionaremos “Board > Arduino Uno” y dentro de la misma pestaña en “serial port” seleccionaremos el puerto COM al que esté conectado el módulo. En caso de no saber a que puerto está conectado nuestro módulo, podremos verificarlo en el panel de control de Windows en la sección de “Device Manager ó Administrador de dispositivos”. Una vez que hayamos seleccionado el puerto, procederemos a subir el programa hacia el módulo presionando las teclas CTRL + U, y podremos observar la verificación y compilación del programa hasta que finalmente se

despliegue el mensaje “done upload”. De esta manera el módulo que controlará los motores del simulador estará listo para ser programado en la interfaz gráfica con Labview.

### ***1.2 Xbee***

Para programar el módulo arduino que está montado en el vehículo y que permitirá la interacción con el Xbee (transmisor / receptor), realizaremos todo el proceso anterior para subir el código que permitirá interactuar al módulo con el software, con la excepción de que en el sketch del programa principal “LabVIEWinterface.h” tendremos que modificar la siguiente parte del código que se encuentra al principio:

```
#define FIRMWARE_MAJOR 02
#define FIRMWARE_MINOR 00
#if defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__)
#define DEFAULTBAUDRATE 57600
#else
#define DEFAULTBAUDRATE 57600
```

En esta parte en el código estamos cambiando la declaración de variables, en donde estamos eligiendo una velocidad de comunicación de 57600 bps (bits por segundo), ya que se comprobó que el Xbee trabaja de manera segura con esta velocidad de transmisión de datos sin que se presenten errores durante la comunicación inalámbrica entre el PC y este módulo.

Para que exista comunicación entre el PC y el vehículo, necesitaremos configurar los dos módulos Xbee que transmitirán la comunicación bidireccional, para lo cual será necesario implementar un programador USB adicional (*figura 76*) para estos módulos:



Figura 77: Programador USB para el Xbee

El Xbee independientemente del arduino, necesita también ser configurado mediante el software propio del módulo, que puede ser descargado en su página web (<http://www.digi.com>) denominado X-CTU. Una vez instalado el programa, lo ejecutaremos y seleccionaremos el puerto al que esté conectado el programador USB con el Xbee y presionamos en el botón “test” en la pantalla que se despliega como en la *figura 78*:

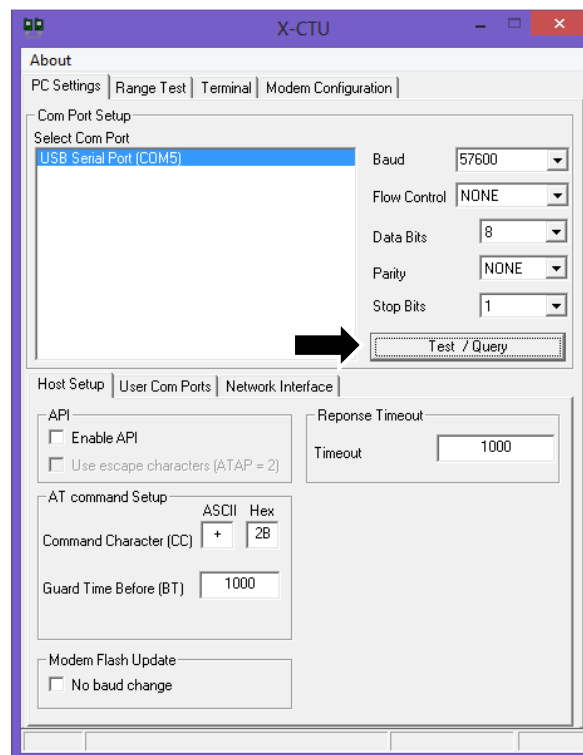


Figura 78: Ventana principal del programa X-CTU

De esta manera nos aseguraremos que el puerto se está comunicando correctamente con el módulo para poder configurarlo. En caso de que exista un error en la comunicación, se tendrá que cambiar indistintamente la velocidad del “Baud” en la correspondiente casilla hasta que se logre comunicar con el módulo (*figura 79*):

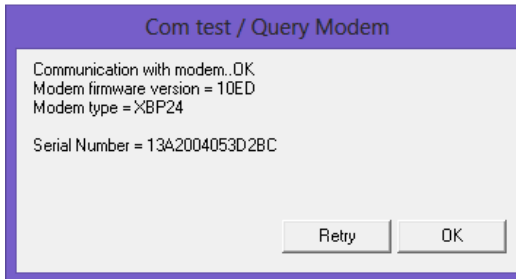


Figura 79: Comprobación de comunicación con el módulo Xbee

Luego nos dirigiremos a la pestaña “Modem configuration” y dentro del recuadro “Modem Parameter and Firmware” presionamos en el botón “read” (*figura 80*) para poder acceder a todos los parámetros del módulo:

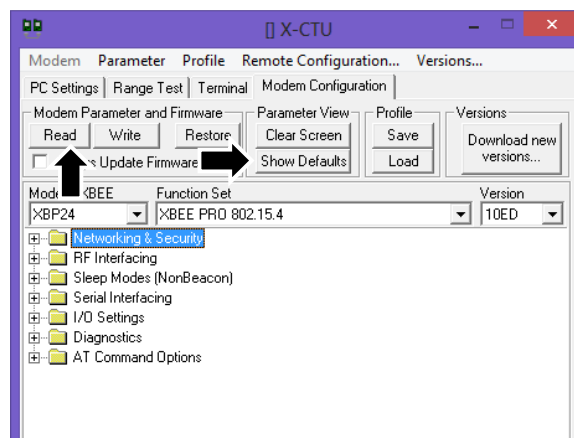


Figura 80: Parámetros de configuración del módulo Xbee

Para asegurar que el módulo no tenga parámetros con distinta configuración, presionaremos el botón “Show Defaults” dentro del recuadro “Parameter View” y entonces procederemos a cambiar lo siguiente:

- Networking & Security

*Pan ID (Personal Area Network)* > sirve para configurar una red única entre dos o más módulos Xbee que tengan el mismo número de red en código hexadecimal, para este caso se utilizó “3332” pudiendo tener un rango entre 0 hasta 0xFFFF.

*Destination address high* > Lee y escribe en los 32 bits más significativos de los 64 bits de la dirección de destino. Si se requiere utilizar únicamente 16 bits se deberá asignar un valor de “0” con un máximo de 0xFFFFFFFF en caso de utilizar 64 bits.

*Destination address low*> Lee y escribe en los 32 bits menos significativos de la dirección de destino con un rango de hasta 0x0000FFFF.

- Serial Interfacing

*Interface data rate* > aquí se definirá el baud rate para la comunicación serial entre los módulos y se lo fijó en 57600 como se pre-estableció en la configuración del arduino.

Para ambos módulos Xbee (PC y vehículo) utilizaremos la misma configuración anterior, con la excepción de que en uno de los módulos deberemos intercambiar la información entre los parámetros de *Destination address low* (DL), es decir que el DL del primer módulo deberá ir en el DL del segundo y viceversa, con esto nos aseguraremos de que el intercambio de información entre los módulos sea codificada y enviada de tal



manera que solo los módulos con el número de serie asignados puedan leerlos y así evitar cualquier tipo de interferencia por parte de cualquier otra señal.

## 2. Interfaz gráfica

La programación de la interfaz gráfica se llevará a cabo mediante el software Labview 2012, para lo cual se ha dividido en distintas etapas los controles y monitoreo de los módulos, que posteriormente serán unificados en una sola pantalla para controlar todo el sistema.

Para comenzar a programar la interfaz, debemos arrancar el programa y crear un VI (visual interface) en blanco, como se ve en la *figura 81*:

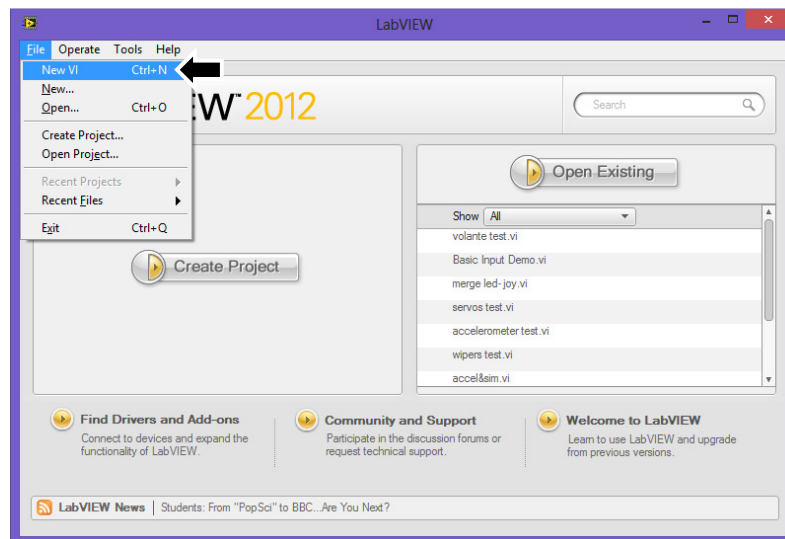


Figura 81: Pantalla principal de Labview 2012

### 2.1 Interfaz con el volante

Utilizando la pantalla de diagrama de bloques (CTRL+E) en el nuevo VI, procederemos a agregar los siguientes elementos (*figura 82*):

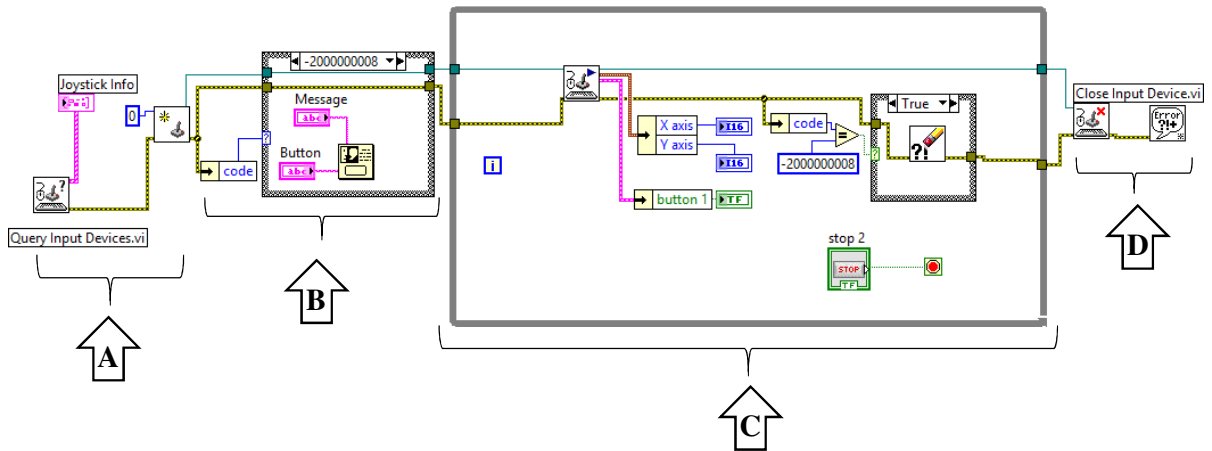


Figura 82: Programación para el control del volante

- A. *Query Input Devices* > Con este bloque podemos inicializar un puerto para dispositivo de entrada. En este caso se utilizará la entrada “joystick”. *Initialize joystick* > Nos permite leer el dispositivo conectado al puerto de entrada y asignarle un número en caso de que exista mas de uno.
- B. Esta sección es opcional, ya que se utiliza el bloque “Case estructura” únicamente para desplegar un mensaje de error en caso de que no se detecte ningún dispositivo conectado. Las condiciones que utiliza el bloque es que si no recibe ningún valor de la secuencia de inicialización del puerto no despliega ningún mensaje, por ende existirá un dispositivo conectado, pero si recibe un valor, detendrá el proceso y mostrará un mensaje de error.

**C. *While loop*** > Este bloque nos permitirá repetir un proceso infinito número de veces hasta que lo detengamos manualmente o exista un error de ejecución.

*Acquire input data* > dentro del loop anterior, la secuencia se inicia con este bloque de adquisición de datos, el cual va a monitorear repetidamente las señales del dispositivo para que puedan ser procesadas. De este bloque se desprenden dos sub-bloques en los que podemos determinar el número de ejes que se requiera utilizar con el dispositivo conectado, así como también en número de botones. Por medio de indicadores numéricos podremos observar el valor del movimiento del eje en el dispositivo conectado con el formato que se le asigne. De igual manera podremos agregar un indicador visual por medio de una variable booleana para indicar cuando se presione algún botón en el dispositivo conectado.

El *case structure* consiguiente, va ligado a la sección B, en la cual se determina si existe algún error de conexión pero se decide continuar la operación en caso de haber reconectado el dispositivo e ignorar la señal de error.

**D. *Close input device*** > Finalmente con este bloque se cierra el puerto de lectura del dispositivo cuando se detenga la ejecución del programa.

*General error handler* > Nos indica si hubo alguna señal de error durante el proceso y su posible causa, pudiendo ser opcional su uso.

Para ejecutar el programa, nos dirigimos a la pantalla de interface visual y presionaremos el botón “Run” en la ventana (figura 83) ó las teclas “CTRL + R”:

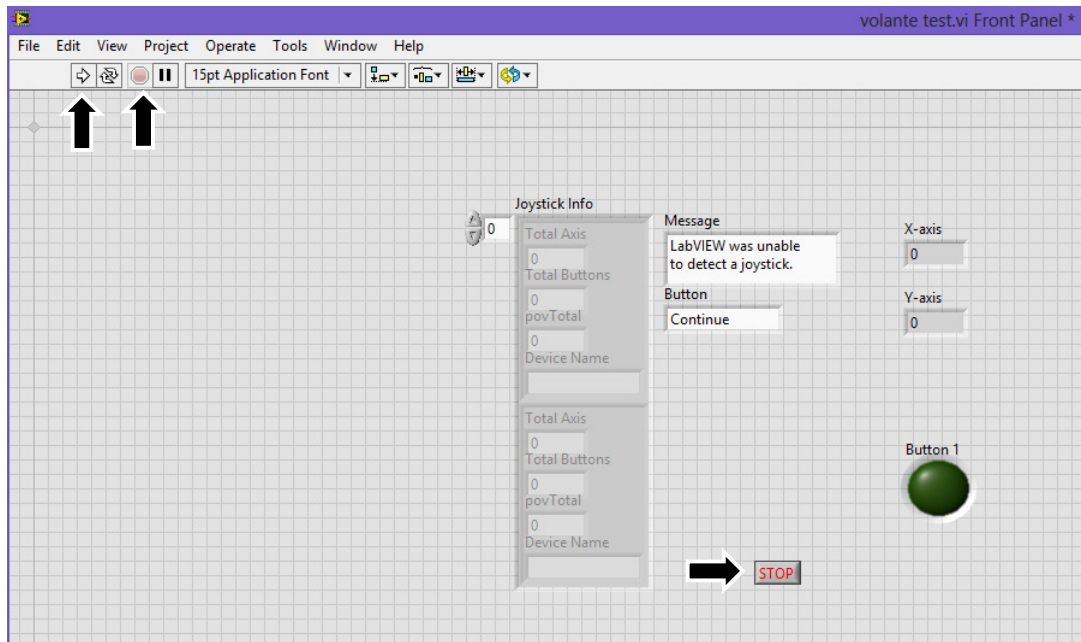


Figura 83: Ejecución del programa de interfaz con el volante

En la ventana de visualización podremos observar el valor numérico de cada eje de las variables asignadas al momento de mover los mandos del dispositivo conectado y también si se presiona el botón en la variable booleana.

Para detener la ejecución presionaremos en el botón “stop” o en el menú principal el botón “abort” en caso de que se requiera forzar la detención del programa en cualquier instancia.

## 2.2 Control de dirección, aceleración y arrancador de motor

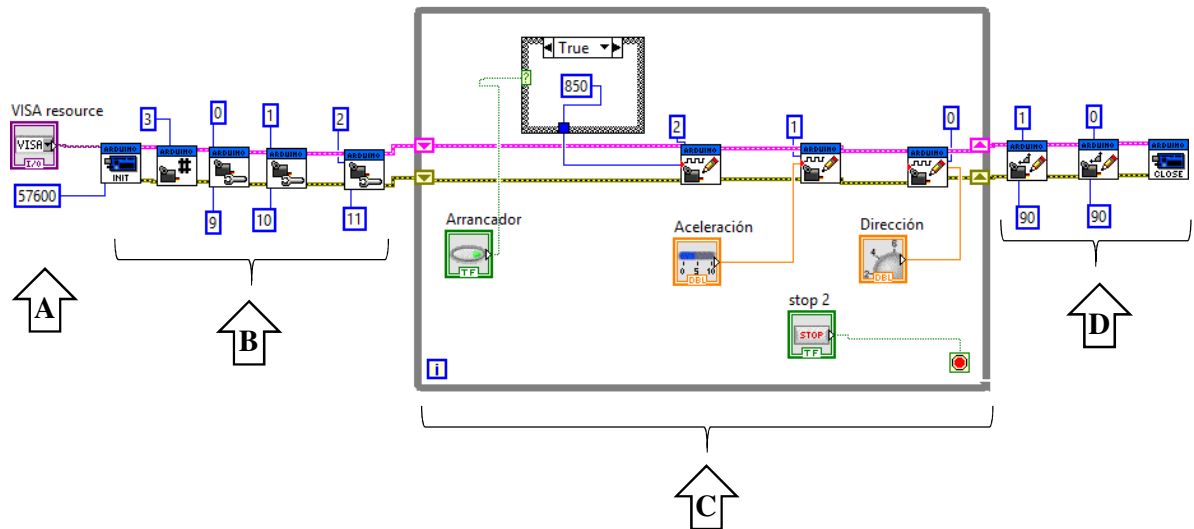


Figura 84: Programación para el control del arrancador, dirección y aceleración del vehículo

- A. *Visa Resource* > Este bloque (*figura 84*) se encarga de gestionar la apertura de un puerto disponible en el computador con el cual vamos a intercambiar información con nuestro dispositivo.
- B. *Init* > Dentro del toolkit de arduino encontraremos todos los bloques necesarios para inicializar y controlar el módulo. El bloque “init” (*figura 84*) sirve para inicializar el dispositivo y pre-configuralo con las siguientes opciones:

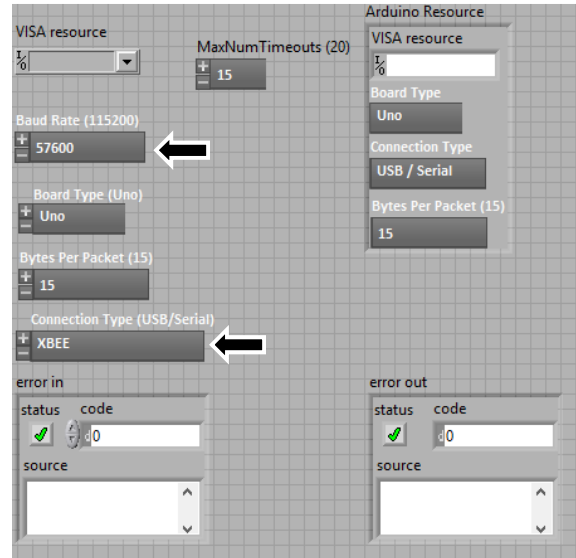


Figura 85: Configuración del bloque “init”

*Set number of servos* > Con este bloque configuraremos el arduino para controlar servomotores. Se reservará un espacio en la memoria de su microcontrolador destinado a la creación de una matriz indexada con un número específico de servomotores, para lo cual debemos agregar una constante conectada al bloque, indicando el número de servomotores que vayamos a utilizar.

*Configure servo* > En este bloque asignaremos un pin con salida de digital del módulo para poder controlar un servomotor. Debido a que el arrancador del motor y los servomotores de aceleración / dirección utilizarán estos pines, tendremos que agregar tres bloques distintos en los cuales por medio de constantes asignaremos un número para cada servomotor empezando desde el número cero, así como el pin de salida digital al que estará ligado.

Para los siguientes dos bloques, de igual manera indicaremos el número de servo ya asignados para el control de dirección y aceleración en el vehículo, pero con mandos de control manuales que tendrán un rango variable de 1000 a 2000, ya que los servomotores trabajan con señales de 1 a 2 ms (milisegundos) para realizar los recorridos de ángulos en sus giros.

C. Dentro del *while loop* agregaremos tres bloques *servo write pulse width* del toolkit de arduino. En estos bloques podemos indicar el número de servo que vamos a utilizar, sin necesidad de especificar el pin de salida, ya que como se mencionó, esto está pre-configurado en la memoria del módulo gracias a los bloques anteriores. Para el primer bloque debemos asignar una conexión con una constante indicando el número de servomotor que vamos a utilizar y el valor de duración del pulso de la señal que por defecto está en microsegundos (*us*); el arrancador del motor fue puesto a prueba y se comprobó que se activa con un pulso de 850(*us*) y como el arrancador debe ser accionado mediante un pulsador, se agregó un botón con variable booleana (con valor verdadero o falso) conectada a un *case structure*, en el cual se asignó una condición de "0" en caso de ser falso (botón desactivado) y un valor de 850 en caso de ser verdadero (botón activado); con esto, podremos accionar el arrancador al momento de presionar un botón en la pantalla de interface visual.

**D.** En esta última sección de bloques tendremos que cerrar el puerto al que está conectado el arduino, pero antes agregaremos dos bloques adicionales (*servo write angle*) para mover los motores a su posición inicial (90 grados) y de esta manera asegurar que la dirección y aceleración en el vehículo no permanezcan en posiciones incorrectas al momento de detener el programa. Por último cerraremos la conexión con el módulo por medio del bloque “*close*”.

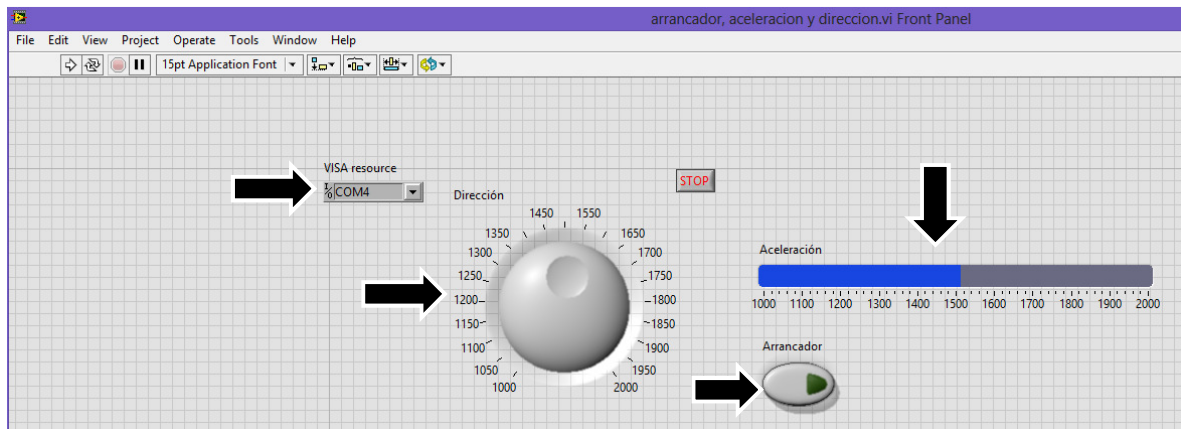


Figura 86: Interfaz visual de la programación del arrancador, dirección y aceleración del vehículo

En la ventana de la interfaz visual (*figura 86*) debemos seleccionar el puerto al que está conectado el módulo antes de ejecutarlo. Al momento de correr el programa, debemos asegurarnos que el módulo del vehículo esté encendido, así como la fuente de alimentación de los servomotores. La dirección será controlada por un control manual en forma de perilla giratoria y la aceleración por medio de una barra de desplazamiento. El arrancador es simplemente un botón pulsador ya que para arrancar el motor solo es necesario oprimir el botón durante unos segundos. No es necesario poner en funcionamiento el motor a combustión del vehículo para realizar las pruebas con estos



controles, ya que fácilmente se puede observar el movimiento de los servos y funcionamiento del arrancador.

### 2.3 Lectura del acelerómetro

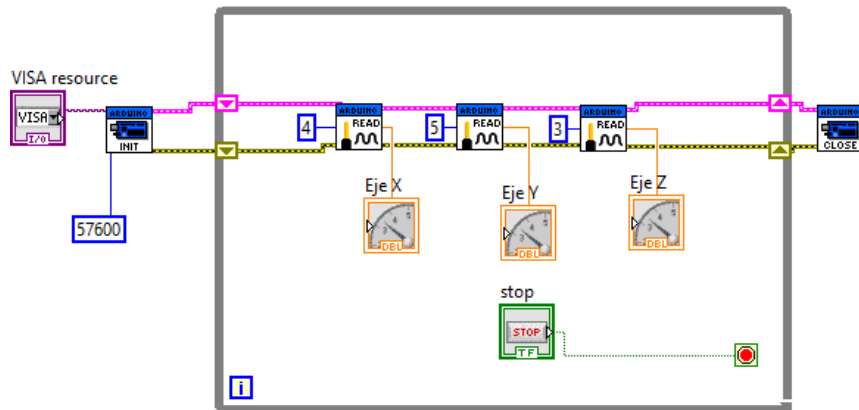


Figura 87: Programación del control de motores de simulador

Para la lectura de las señales de los tres ejes de movimiento del acelerómetro montado en el vehículo, vamos a utilizar tres entradas de lectura analógica del arduino, de manera que inicializaremos el módulo con los bloques ya descritos (*visa resource*, *Init*) y luego dentro del *while loop* agregaremos tres bloques para la lectura analógica que es el *analog read pin* dentro del toolkit de arduino (*figura 86*). En estos bloques asignaremos un pin para cada eje de movimiento del acelerómetro por medio de una constante y se visualizará el valor de su señal por medio de un indicador analógico. Finalmente cerraremos la conexión con el bloque *close* fuera del loop y realizaremos la ejecución del programa por medio de la interfaz visual:

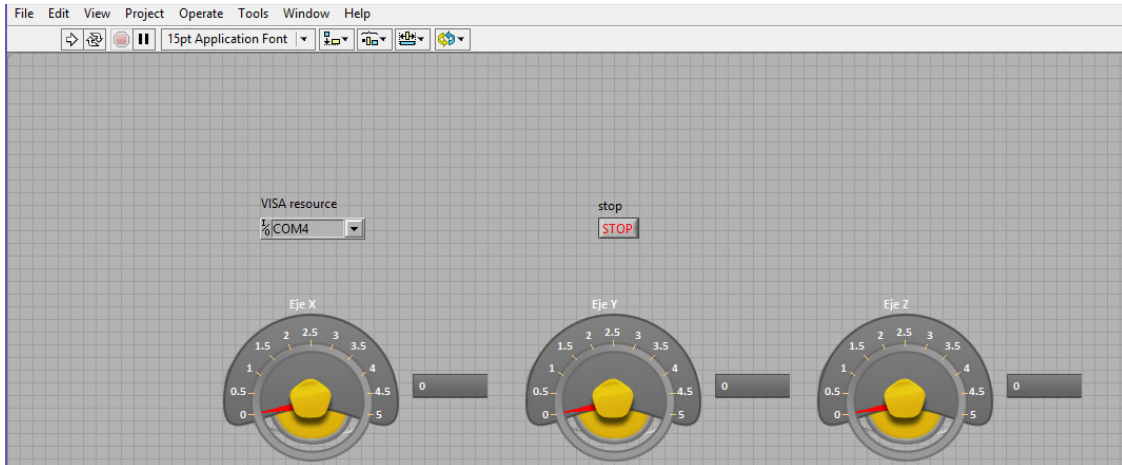


Figura 88: Interfaz visual de lectura del acelerómetro

Al ejecutar el programa (figura 88) tendremos que encender el módulo del vehículo para establecer la comunicación entre el PC y el Xbee. Al momento de mover el vehículo lateralmente o verticalmente podremos observar como cambiar de valor los diferentes ejes de lectura y también si lo agitamos o elevamos de su punto de referencia; esto será importante para filtrar estas señales y programarlas posteriormente para que realicen el correcto movimiento en el simulador.

#### 2.4 Control de motores del simulador

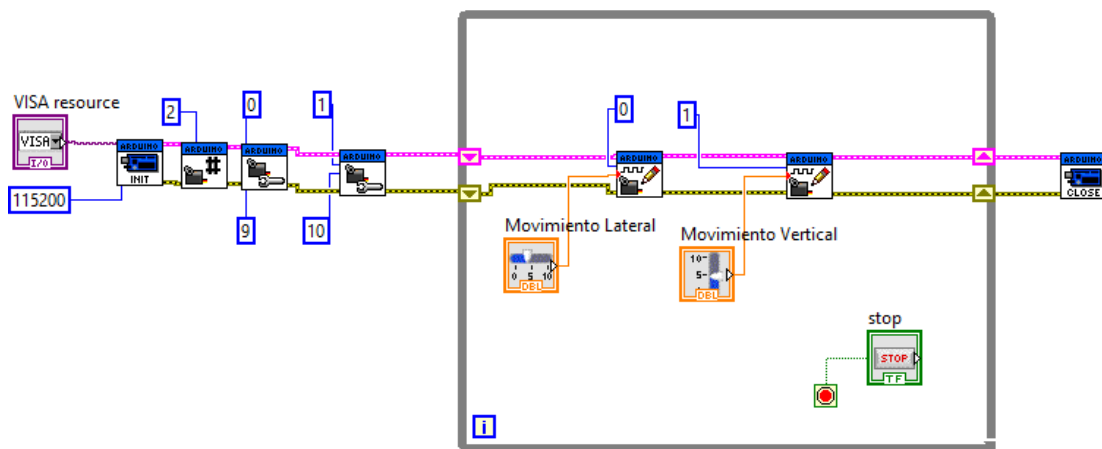


Figura 89: Programación de los motores del simulador

En esta interfaz (*figura 89*) utilizaremos la misma configuración para controlar los servos de dirección y aceleración, ya que gracias a la modificación del hardware en los motores del simulador, estos podrán ser manipulados como servomotores con los bloques de programación del toolkit del arduino.

Se iniciará la lectura del módulo de la misma manera que los programas anteriores y se utilizará los mismos bloques para la configuración de los pines de control de los motores, con la excepción de que utilizaremos una velocidad de 115200 en baud rate, ya que este módulo estará conectado por medio del cable USB al computador, lo cual nos permite una comunicación más fiable.

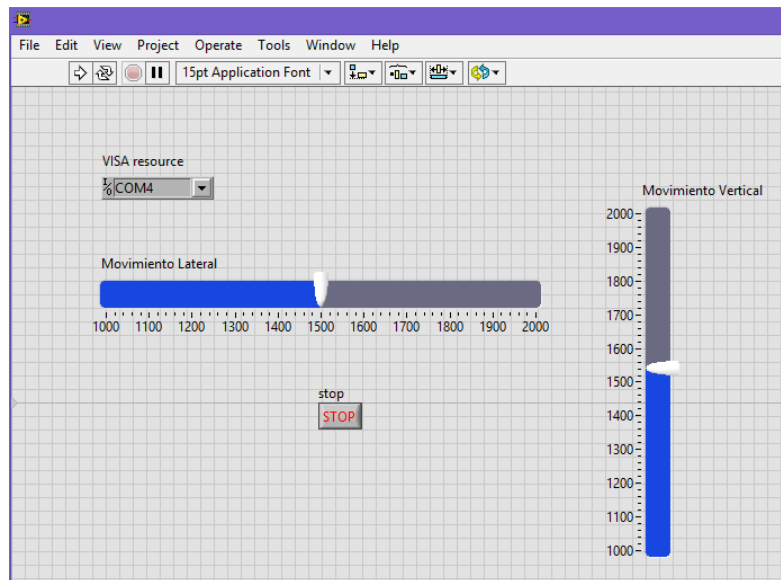


Figura 90: Interfaz visual del control de los motores del simulador

En esta interfaz (*figura 90*) podremos controlar los movimientos en el simulador, por lo cual es recomendable utilizar un contrapeso en el asiento del mismo para obtener un correcto balance en los giros y movimientos verticales. También podremos calibrar correctamente los potenciómetros en los motores para poder colocar el simulador en una

posición central predeterminada y así poder utilizar valores proporcionales al ángulo de giro de cada eje vertical y lateral.

## II) Funcionamiento

### 1. Esquema

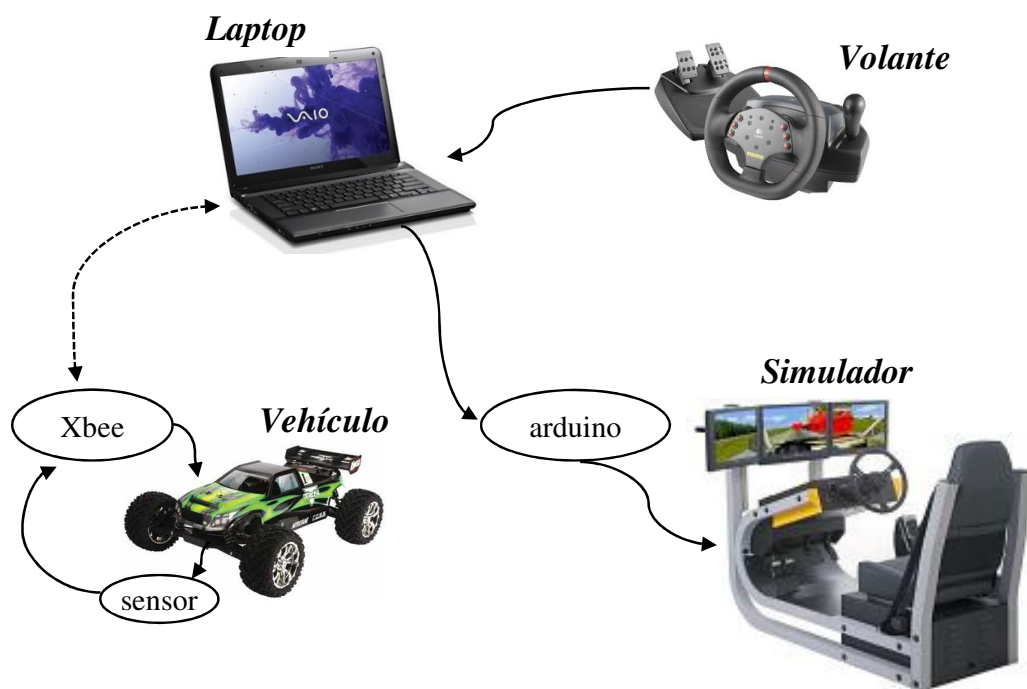


Figura 91: Esquema de funcionamiento de todo el sistema

El sistema unificado (*figura 91*) funcionará de la siguiente manera:

Mediante el volante y los pedales enviaremos las señales de dirección, aceleración y arranque del vehículo hacia el computador, en donde se procesarán a través del software para que estas señales permitan la manipulación de los servomotores del vehículo de manera inalámbrica por medio de los módulos Xbee. A su vez con el movimiento del vehículo, el acelerómetro va a transmitir las señales de fuerzas que se ejercen en los tres

ejes X, Y, Z desde su punto de referencia hacia el Xbee, el cual transmitirá bidireccionalmente estas señales al computador para que al mismo tiempo sean procesadas y acondicionadas para que produzcan un determinado movimiento de los motores en el simulador, que son controlados por medio del arduino y el puente H. De esta manera todo el sistema funciona a manera de “loop” continuo, para lo cual será necesario unificar cada una de las interfaces con los diferentes componentes en los que ya se programó anteriormente su determinada función.

## 2. Programa principal

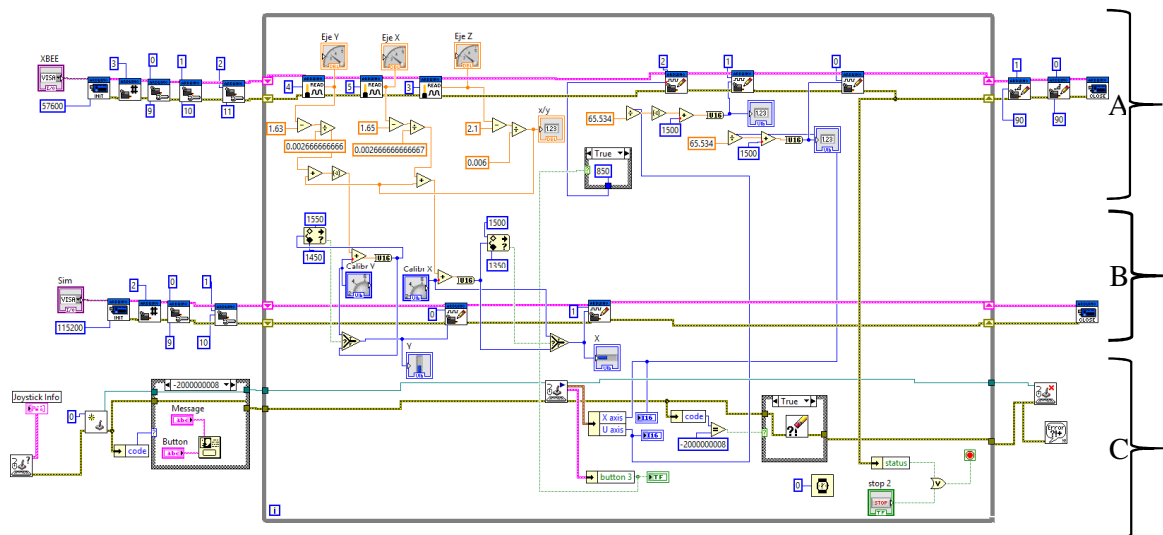


Figura 92: Programa de control principal del sistema

El programa principal (*figura 92*) se compone de la integración de todos los subprogramas creados anteriormente en una sola interfaz para el control unificado de los módulos y elementos que constituye el sistema dentro de un solo loop. El loop se divide en tres secciones:

### A. Control inalámbrico

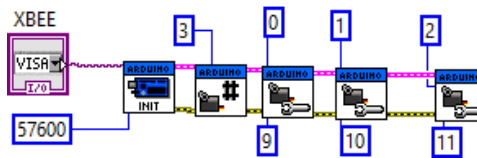


Figura 93: Inicialización del módulo inalámbrico

El puerto del módulo Xbee inalámbrico se inicializará con las configuraciones preestablecidas anteriormente (*figura 93*) en el subprograma de prueba con los pines de escritura 9, 10 y 11 para el control del vehículo.

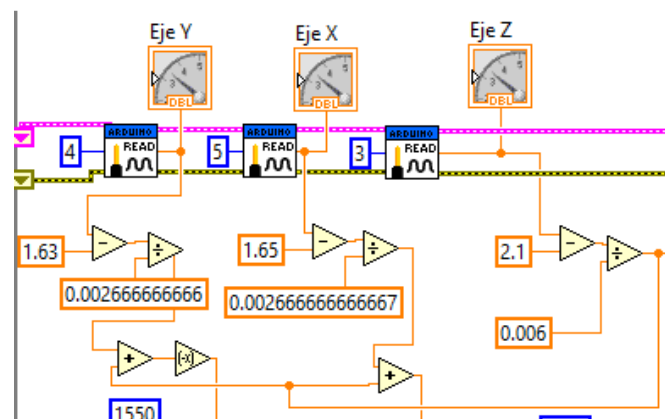


Figura 94: Programación de acondicionamiento de la señal del sensor de movimiento

A continuación se agregarán los bloques de lectura analógica para los tres ejes del acelerómetro, en donde se acondicionará su valor original (*figura 94*) para que pueda ser utilizado posteriormente en el movimiento del simulador.

El eje “Y” del acelerómetro transmitirá los valores de movimiento vertical del vehículo, teniendo como punto central un valor de aproximadamente 1,63V dentro de la

escala de 0 hasta 3, en donde el 0 es el punto en que el vehículo estará en un giro de 90 grados de su punto central y lo mismo para el valor de 3V. Debido a que los motores del simulador están configurados para moverse como servomotores, necesitaremos un rango de valores desde 1000 hasta 2000 (*us*), por lo que se necesita transformar el rango de la señal del acelerómetro a una escala de -500 hasta +500 para que pueda ser sumado y restado al valor de 1500 (*us*) el cual indicará la posición central del movimiento lateral en el simulador. Al valor del eje del acelerómetro se le restará 1.63 para que entre en un rango nulo y luego se dividirá para 0.00266666 que es el número resultado de la operación  $(3-1.63)/500$  con lo cual conseguiremos que la señal entre dentro de la escala requerida.

Para el eje “X” que controlará el movimiento vertical del simulador, realizaremos el mismo proceso para para acondicionar el valor dentro de la escala, y al final del resultado de esta señal obtendremos un valor de -500 hasta +500. Ambos valores (X y Y) recibirán la suma de un valor adicional proporcionado por el eje Z el cual mide las fuerzas de inercia y por ende trabajará en una escala menor (-200 hasta +200) para que no tenga un efecto excesivo en la representación del movimiento de estas fuerzas.

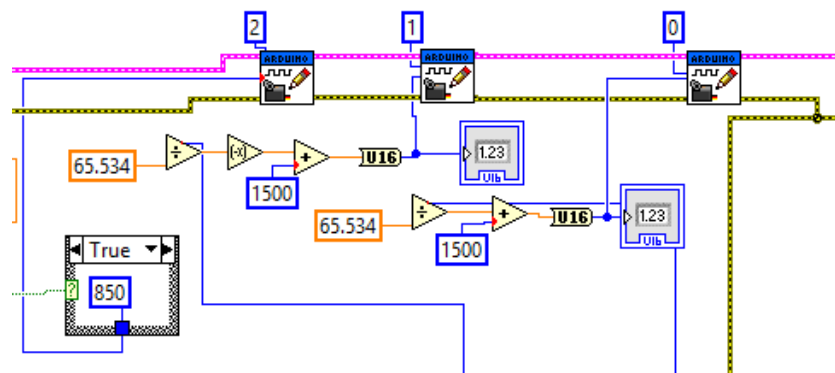


Figura 95: Programación de la señal del volante para controlar el vehículo

Para el movimiento de los servomotores del vehículo utilizaremos el subprograma creado anteriormente con la excepción de que acoplaremos la señal del volante en lugar de un control manual. Para esto también se necesita transformar el valor de la señal de los ejes del joystick para que puedan ser utilizados en una escala de -500 hasta +500. El valor original de ambos ejes del joystick tienen un valor integrador de 16 bits (-32768 hasta +32768) el cual se lo va a dividir para 65,534 que es el número resultado de  $32768/500$ . Con esto lograremos que todos los valores entren en el rango propuesto que posteriormente serán sumados a 1500 que es la posición neutra de ambos servomotores de control de dirección y aceleración. Para el arrancador simplemente utilizaremos una variable booleana proveniente de la señal de cualquier botón del joystick acoplado al bloque de “case structure” descrito anteriormente (figura 95).

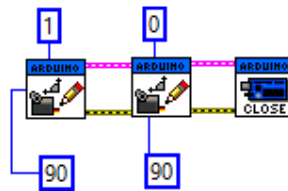


Figura 96: Cierre del puerto de lectura del módulo

Para finalizar, se cerrará el puerto indicando (figura 96) a los servomotores que regresen a su posición neutra, para lo cual utilizaremos los bloques de “*servo write angle*” de la librería del arduino con un valor de 90 grados en cada uno.

#### B. Control del simulador

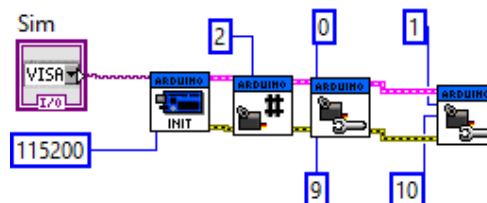


Figura 97: Inicialización del puerto del módulo para el simulador



Para el simulador de igual manera se inicializará el puerto (figura 97) con la configuración preestablecida del programa de prueba.

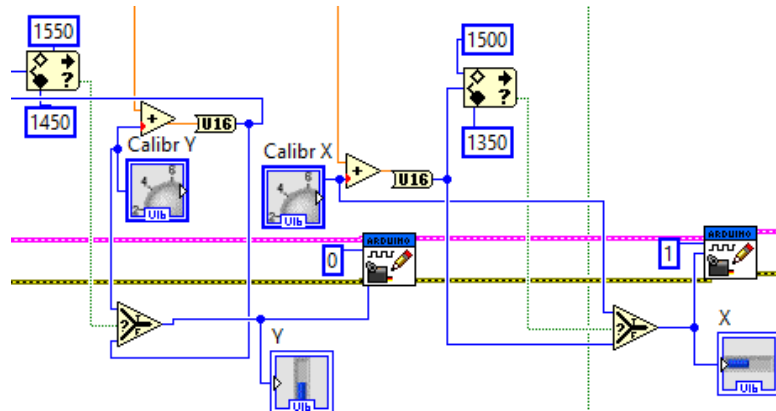


Figura 98: Programación para sensibilidad de posición neutra del simulador

En esta parte (figura 98) utilizaremos los valores de los ejes X y Y que fueron adecuados anteriormente para que puedan ser sumados a un control manual con un valor inicial de 1500. A este valor se lo acoplará a la entrada del bloque “in range and coerce” (figura 99):

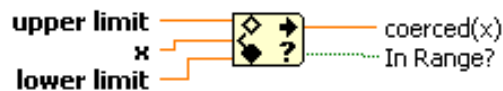


Figura 99: Bloque “in range and coerce”

Con esto se logra obtener una señal booleana (verdadero o falso) cuando la variable de entrada exceda el límite superior o inferior propuestos en el bloque, y de esta manera podremos utilizar esta señal resultante para avisar al bloque “select” (figura 100) cuando puede realizar la escritura hacia el bloque de control del servomotor, ya que este bloque permite seleccionar entre dos señales de acuerdo al valor booleano que se le indique:



Figura 100: Bloque “select”

De esta manera el bloque transmitirá el valor de los ejes del acelerómetro sólo si exceden el valor de los límites del anterior bloque y con esto poder ajustar la sensibilidad con la cual se podrá mover el simulador desde su posición inicial. Hay que notar que el valor inicial de 1500 que es la posición central de ambos ejes, podrá ser calibrado mediante un control manual que varíe desde 1400 hasta 1600, para lograr que el simulador quede centrado en caso de que los potenciómetros de los motores no hayan sido calibrados correctamente.

### C. Control del joystick

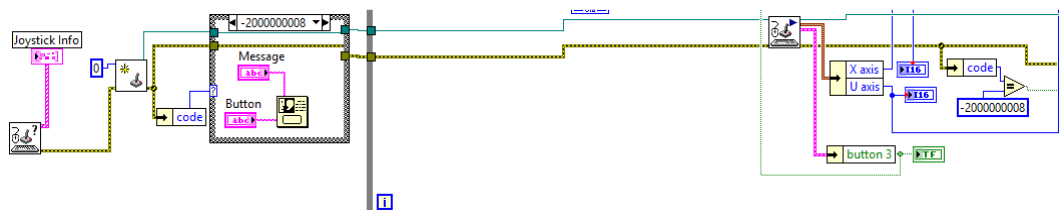


Figura 101: Bloques de control para el volante

Una vez más se integrará el programa (*figura 101*) creado con anterioridad con los mismos bloques de control dentro del loop principal con los demás bloques.

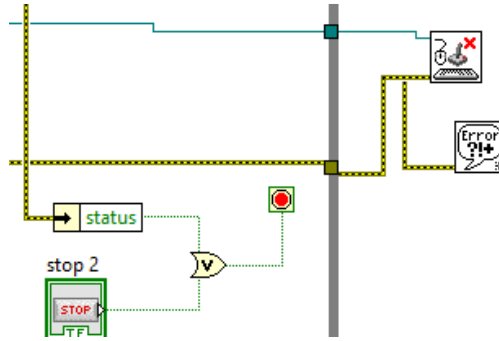


Figura 102: Condiciones de “stop” para el loop

Finalmente para detener la ejecución del loop se agregó una condición entre el botón “stop” y el bloque “*unbundle by name*” (figura 102) para que por seguridad el programa también pueda ser detenido automáticamente por algún error de comunicación que pueda surgir entre el módulo inalámbrico del vehículo y el PC.

### 3. Ejecución y manejo

Para la ejecución del programa principal y poder manipular todo el sistema es necesario proceder con los siguientes pasos:

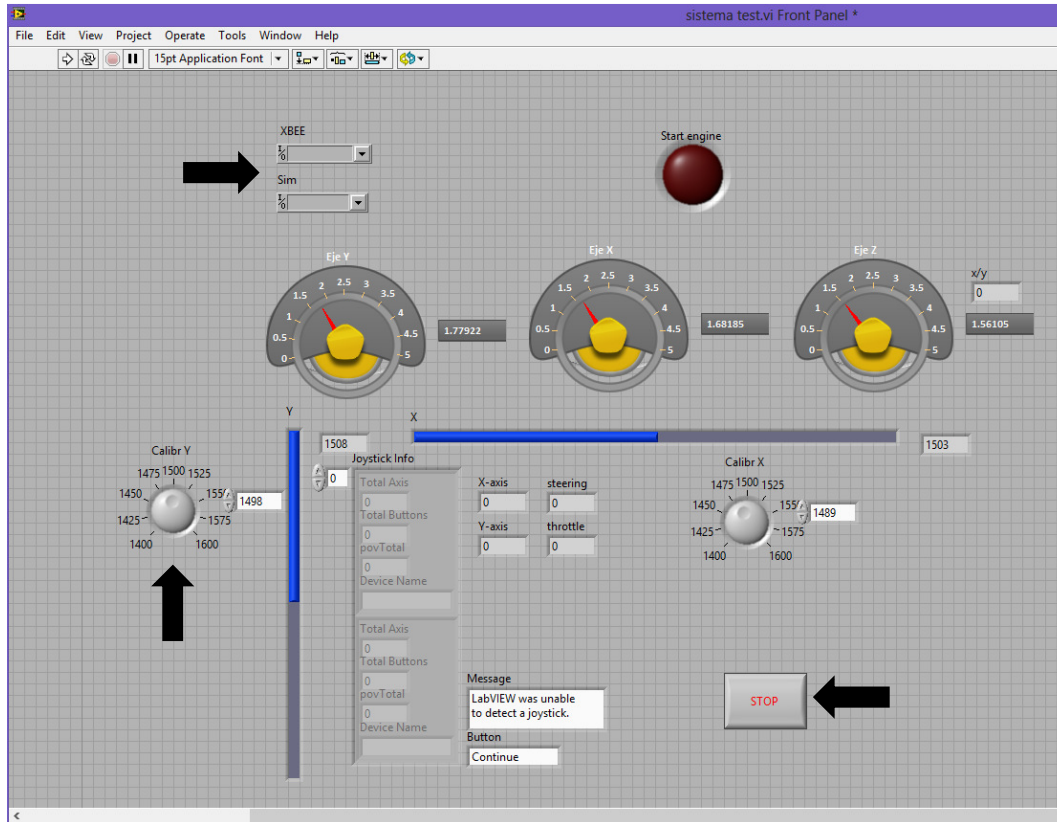


Figura 103: Interfaz gráfica del programa principal del sistema completo

- En la ventana principal de la interfaz gráfica (*figura 103*) se tendrá que asegurar de seleccionar los puertos correspondientes a los módulos del vehículo y simulador.
- Los “switchs” de las fuentes de alimentación del vehículo tendrán que estar en la posición de encendido al igual que la cámara y su módulo de transmisión.
- Al momento de ejecutar el programa, se deberá esperar a que se logre la comunicación entre los módulos del simulador y el vehículo, para lo cual los servomotores se centrarán y la lectura del acelerómetro comenzará a desplegar sus valores iniciales para cada eje.

- Se comprobará el funcionamiento de los servomotores del vehículo mediante el volante y se moverá el vehículo manualmente para comprobar que las señales del sensor sean las correctas.
- A continuación se procederá a encender la fuente de poder del simulador para calibrar su posición con los controles manuales en la pantalla en caso de que se requiera.
- Finalmente se podrá manipular el sistema dando arranque al vehículo presionando durante algunos segundos el botón asignado en el volante.

## ESPECIFICACIONES

### *Simulador:*

Material	Tubería PVC
Tamaño del chasis	175 x 90 x 140 cm
Peso	60 lb
Ejes de movimiento	Lateral y vertical
Grados de libertad	2 grados de libertad (45 y 35 grados)
Tiempo de recorrido movimiento lateral	2.5 segundos
Tiempo recorrido movimiento vertical	2 segundos
Motores	DC 120W
Alimentación principal	110V

Asiento	Fibra de vidrio forrado en tela
Pantalla	LCD de 27 pulgadas
Control	Volante con pedales

Tabla 9: Especificaciones del simulador

***Vehículo:***

Tipo	Truggy AWD a combustión
Tamaño	42 x 43 x 18 cm
Peso	8 lb
Combustible	Nitrometano
Cámara a bordo	HD inalámbrica
Sensor de movimiento	Acelerómetro de 3 ejes y 5V de resolución en cada eje
Control	Teledirigido con módulo programable

Tabla 10: Especificaciones del vehículo

***Costos:***

Tubería y acoples PVC	\$300
Perfiles de estantería	\$40
Motores DC	\$70
Pernos y tornillos	\$20
Cadenillas	\$20
Fuente de poder	\$90
Módulo arduino UNO	\$30

Asiento	\$200
Volante	\$180
	<b><i>Costo del simulador: \$950</i></b>
Pantalla HDTV	\$600
Vehículo R/C a combustión	\$750
Módulos inalámbricos	\$150
Cámara HD y transmisor	\$500
Elementos varios (cables, switches, placas, pintura, etc..)	\$50
	<b><i>Costo Total: \$3000</i></b>

Tabla 11: Costos

## CONCLUSIONES

Durante el presente trabajo realizado se pudo diseñar y construir exitosamente una estructura móvil de dos ejes para simular los movimientos de un vehículo teledirigido tomando en cuenta los objetivos planteados en cuanto al material utilizado, diseño y costos, procurando mantener el esquema dentro del cual el proyecto está basado en ser eficaz y económico.

Se logró establecer una comunicación inalámbrica bidireccional para el control del vehículo y el monitoreo de sus movimientos mediante módulos transmisores reprogramables, los cuales transmiten señales hacia un computador para que puedan ser procesadas y programadas mediante software con interfaz gráfica y a su vez poder ejercer

una simulación en tiempo real en la estructura móvil construida, visualizando la trayectoria del vehículo a través de una cámara inalámbrica.

La estructura del simulador fue diseñada y construida en base a sus factores limitantes debido al material y componentes utilizados, que entre los más importantes fue el peso máximo que puede soportar el chasis ya que la potencia de los motores para mover el simulador llega a su máxima capacidad al mover 160 lb. De igual manera la máxima distancia de comunicación entre los módulos transmisores tiene limitaciones, ya que la cámara inalámbrica montada en el vehículo utiliza un transmisor diferente para el control y monitoreo del mismo, llegando a un máximo de transmisión de imagen de 50 metros a campo abierto, sin embargo, el transmisor del vehículo puede llegar a una comunicación de más de una milla de distancia.

El sistema completo puede ser optimizado en sus limitaciones, mediante la implementación de componentes con mayor potencia, tanto en el aspecto mecánico ó electrónico, como son los motores DC y los módulos de control, así como también los módulos transmisores que podrían utilizar un solo protocolo de comunicación con diferentes canales de transmisión a través de routers utilizados a manera de repetidores de señal, con el fin de poder alcanzar una mayor distancia para el control y monitoreo inalámbrico.

La estructura mecánica del simulador también podría ser utilizada y acoplada en otro tipo de simulaciones como en videos de capacitación para futuros conductores de vehículos o pilotos de aviación, así como también en juegos de videos de diferentes tipos para que puedan ser utilizados como herramienta de entrenamiento en los diferentes campos de la conducción móvil o aérea.

Como conclusión final, hay que señalar que durante todo el proceso de diseño, investigación y construcción del presente trabajo, se pudo alcanzar todos los objetivos



planteados gracias al plan de desarrollo del estudio realizado, en donde se pudo ejercer los conocimientos teóricos y prácticos estudiados durante el transcurso de la carrera universitaria, especialmente en todos los problemas encontrados durante el desarrollo práctico del trabajo, que es en donde realmente se aplica los factores fundamentales e importantes de la ingeniería, como son la imaginación, creatividad e intelecto.

### REFERENCIAS

- [1] <http://es.wikipedia.org/wiki/Simulador>
- [2] <http://definicion.de/simulador>
- [3] <http://definicion.de/wp-content/uploads/2011/07/simulador.jpg>
- [4] <http://www.aguamarket.com/sql/productos/fotos/tuberia%20de%20PVC.jpg>
- [5] [http://www.ferreteriamarti.com/sqlcommerce/ficheros/dk\\_61/productos/50695660-1.jpg](http://www.ferreteriamarti.com/sqlcommerce/ficheros/dk_61/productos/50695660-1.jpg)
- [6] [http://upload.wikimedia.org/wikipedia/commons/thumb/f/f2/H\\_bridge\\_operating.svg/800px-H\\_bridge\\_operating.svg.png](http://upload.wikimedia.org/wikipedia/commons/thumb/f/f2/H_bridge_operating.svg/800px-H_bridge_operating.svg.png)
- [7] [http://es.wikipedia.org/wiki/Fuerza\\_G](http://es.wikipedia.org/wiki/Fuerza_G)
- [8] <http://es.wikipedia.org/wiki/Aceler%C3%B3metro>
- [9] <http://www.ps-3-controller.com/wp-content/uploads/2012/07/wii-remote-controller.jpg>
- [10] [http://arduino.cc/es/uploads/Main/arduino\\_uno\\_test.jpg](http://arduino.cc/es/uploads/Main/arduino_uno_test.jpg)
- [11] [http://es.wikipedia.org/wiki/High-Definition\\_Multimedia\\_Interface](http://es.wikipedia.org/wiki/High-Definition_Multimedia_Interface)
- [12] <https://es.wikipedia.org/wiki/Wi-Fi>
- [13] <http://es.wikipedia.org/wiki/Servomotor>
- [14] [http://www.pyroelectro.com/tutorials/servo\\_motor/parts/hitec\\_servo\\_b.jpg](http://www.pyroelectro.com/tutorials/servo_motor/parts/hitec_servo_b.jpg)
- [15] <http://commons.wikimedia.org/wiki/File:Potentiometer.jpg>

- [16] [http://es.wikipedia.org/wiki/Seis\\_grados\\_de\\_libertad](http://es.wikipedia.org/wiki/Seis_grados_de_libertad)
- [17] [http://en.wikipedia.org/wiki/File:6DOF\\_en.jpg](http://en.wikipedia.org/wiki/File:6DOF_en.jpg)
- [18] <http://i.gzn.jp/img/2009/01/12/joyrider/dreamflyer.jpg>
- [19] <http://i160.photobucket.com/albums/t167/watkykjy1/IMG00044.jpg>
- [20] <http://www.stormerhobbies.com/pics/losb0126.jpg>
- [21] <http://www.hobbyone.com.ar/prodDet-losi-tten.php>
- [22] Manual PDF Losi Truggy Ten T
- [23]  
[http://www.geek.com/xyzimages/stories/reviewimages/norcent\\_27/norcent27\\_001.jpg](http://www.geek.com/xyzimages/stories/reviewimages/norcent_27/norcent27_001.jpg)
- [24] <http://img58.imageshack.us/img58/4491/28000853evoblpu3.jpg>
- [25] [http://2.bp.blogspot.com/--amW-0hV6LU/UDBsVX7IQyI/AAAAAAAAAADE/62Jw-yc7a\\_0/s1600/logitech-momo-wheel.jpg](http://2.bp.blogspot.com/--amW-0hV6LU/UDBsVX7IQyI/AAAAAAAAAADE/62Jw-yc7a_0/s1600/logitech-momo-wheel.jpg)
- [26] [http://www.2carpros.com/images/articles/body/windshield\\_wipers/windshield\\_wiper\\_motor.jpg](http://www.2carpros.com/images/articles/body/windshield_wipers/windshield_wiper_motor.jpg)
- [27] [http://www.alibaba.com/products/499202345/ZD1530\\_wiper\\_motor\\_12V\\_auto\\_wiper.html](http://www.alibaba.com/products/499202345/ZD1530_wiper_motor_12V_auto_wiper.html)
- [28] <http://www.y-intl.com/Item.cfm?ID=2521>
- [29] <http://www.arduino.cc>
- [30] Datasheet PDF VNHBS30
- [31] Datasheet PDF Xbee
- [32] Datasheet PDF MMA7361
- [33] <http://www.dickberryphotography.com/blog/2012/11/01/gopro-hero-3-black-edition/>
- [34] <http://i.ebayimg.com/t/Mini-HDMI-Wireless-1080P-HD-AV-Video-Transmitter-Receiver-Kit-EP-WD3207>
- [35] <http://www.tinydeal.com/edup-mini-hdmi-wireless-1080p-hd-av-transmitter-receiver-kit-p-66846.html>
- [36] <http://www.x-sim.de>
- [37] <http://www.x-simulator.de>

## ANEXOS

- **Manual Losi Truggy Ten T**

Archivo PDF en CD adjunto

- **Datasheet Puente H VNHBS30**

Archivo PDF en CD adjunto

- **Datasheet Xbee**

Archivo PDF en CD adjunto

- **Datasheet MMA7361**

Archivo PDF en CD adjunto