



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**SISTEMA ELECTRÓNICO PARA EL CONTROL
DE UNA DIRECCIÓN DIFERENCIAL
CON MOTORES BRUSHLESS.**

TESIS

**QUE PARA OBTENER EL TÍTULO DE:
INGENIERO MECATRÓNICO.**

PRESENTA:

LUIS ALBERTO LÓPEZ DE JESÚS.

Director de tesis:

M.I. Serafín Castañeda Cedeño.



CIUDAD UNIVERSITARIA, MÉXICO D.F.

AGOSTO DEL 2012

A dios, por darle salud a mis padres y por alcanzar una meta más en mi vida.

A mis padres:

Fernando López Matías simple y sencillamente sin su apoyo, esfuerzo, compromiso y trabajo, nada de esto sería posible. Gracias por todo tu apoyo y por todos los sacrificios que hizo por mí. Gracias por todas sus enseñanzas, porque a pesar de que no me podías enseñar académicamente, me enseñaste a afrontar la vida, siempre me guiaste por el camino correcto.

Gracias por toda su paciencia, y que a pesar de que caí muchas veces, usted me levanto, me guio y me apoyo para seguir por el buen camino. Gracias por enseñarme el valor del trabajo, la responsabilidad. A tener fortaleza ante los problemas de la vida. Toda mi vida te he admirado con respeto y orgullo, te admiro porque a pesar de no estar conmigo, siempre estuviste al pendiente de nosotros, porque todos los riesgos y carencias que vives han sido por nosotros, por todo ese amor que nos tienes. Gracias Papá

Rosalía de Jesús Martínez, “bochitos” gracias por dedicar su vida, su trabajo, su cariño y hasta su sueño a mi cuidado, muchas gracias por estar conmigo a lo largo de este camino, muchas gracias porque nunca me dejaste que faltara a la escuela, ni el primer día de primaria. Gracias por todas sus enseñanzas a lo largo de mi vida, porque siempre estuvo al pendiente de mi, para levantarme y corregirme en el camino. Gracias por formarme como persona de bien, por enseñarme el valor de las cosas y la responsabilidad del trabajo. Porque me enseñaste a ser independiente en el conocimiento, a buscar la manera de aprender.

A mis hermanos, Ricardo López de Jesús y Araceli López de Jesús.

A mis familiares, muchas gracias por su apoyo y su comprensión a lo largo de este camino llamado ingeniería. En especial a mis abuelitos, tanto maternos como paternos.

A la UNAM por haberse convertido en mi casa durante 8 años de mi vida, primero en el bachillerato y luego en la carrera.

A la Facultad de Ingeniería de la UNAM, por permitirme cursar mis estudios de licenciatura. Por darme la oportunidad de estudiar en la mejor facultad y en la mejor facultad de ingeniería, me llevo un pedazo de cada rincón de la facultad.

A Serafín Castañeda Cedeño, “Inge” por todas sus enseñanzas académicas y personales, por su apoyo a lo largo de las materias cursadas con usted y a lo largo de este proyecto. Gracias también por permitirme ingresar a culminar mi servicio social en la sala de proyectos, por haberme apoyado y haber confiado en mí para el proyecto del BIMO, así como todas sus enseñanzas y experiencias. Gracias también por todos esos momentos extra-académicos, gracias por convertirse en un amigo para mí.

A mis sinodales, por haberse tomado el tiempo para revisar y mejorar este documento. Por aportar de su importante conocimiento, para enriquecer esta tesis. En especial al M.I. Ulises

Peñuelas, por haberme dado la oportunidad de llevar circuitos digitales con usted, y haberme enseñado lo maravilloso del PIC. Gran parte de mi conocimiento es gracias a usted.

A mis amigos del CCH, gracias por su amistad todos estos años. En especial a Patricia Vargas Fuentes, por todo este tiempo compartido, por tantos y tan bellos recuerdos, y a pesar de todo lo que ha pasado siempre cuento contigo, en las buenas y en las malas. Muchas gracias por darme ánimos en tiempos difíciles y por haber aguantado tantos “Berrinches” todos estos años. Gracias por tu amistad y tu sabes que eres parte importante de este proyecto llamado vida, y también gracias por tu apoyo para este proyecto del VET, por siempre tener algo con que hacerme reír y con que alegrar la vida. Gracias Paty!!!!

A mis Querido s amigos de la facultad, divididos en 2 Grandes Grupos: “los Inges” y “los Chidos” a “los inges” gracias por el tiempo que los he conocido, por tanta diversión a lo largo de la carrera, por todas esas retas de los viernes y porque esta amistad perdure por muchos años. Gracias amigos mechanic@s y mecatronic@s, Porque ahora que nos hemos convertido en unas papas para el fut, seguimos disfrutando de las reuniones.

A “los chidos” gracias por hacer la carrera más amena, ya que en la gran mayoría recorrimos el camino juntos desde el primer día de clases hace 6 años, por todas esas convivencias en la cueva del abuelo y por que sigamos juntos muchísimos años más.

Sistema electrónico para el control de una dirección diferencial con motores brushless.

Índice

Capitulo 1. Introducción	4
Capitulo 2. Objetivos	16
Capitulo 3. Sistema electrónico de control.	18
Capitulo 4. Programación del control.	25
Capitulo 5. Diseño del sistema electrónico del control	48
Capitulo 6. Pruebas	63
Conclusiones	70
Referencias	73
Anexo A. Código fuente.	75
Anexo B. Diagramas eléctricos del SEC.	82
Anexo C. Diseño del PCB.	90

Capítulo 1

Introducción

PLANTEAMIENTO DEL PROBLEMA.

Hoy en día los autos, han dejado de ser un artículo de lujo y se han convertido en una necesidad para esta ciudad. Es por esa necesidad que muchas veces no hacemos conciencia de lo que ocasionamos al medio ambiente y a nosotros mismos.

La contaminación atmosférica es la alteración por causas naturales o por el hombre de la composición del aire que respiramos (78% nitrógeno, 21% de oxígeno, 0.093% de argón) [1]. En gran parte esta alteración es causada por los automóviles que despiden monóxido de carbono, hidrocarburos, óxidos de nitrógeno, etc. Actualmente la mayoría de los autos cuenta con convertidor catalítico, esto ha ayudado a bajar los índices de plomo hasta 10 veces, pero ha contribuido al incremento del ozono troposférico.

El incremento en el número de vehículos circulando, ha influido en el incremento de la contaminación atmosférica, según estadísticas de vehículos registrados en México, en el año 2000 se contaba con 15,611,916 vehículos, en 2005 ya eran 22,138,478 vehículos y para 2010 ya eran 32,338,820 vehículos. Además de estos vehículos, el 14 % lo utilizan para viajar solo, el 32.3% con un acompañante y el 25.7% con dos acompañantes. [6]

Viendo estas cifras no resulta complicado entender los niveles de contaminación, tanto en México como a nivel mundial. Las tendencias dados los problemas de cambio climático, han sido las de optar por formas de energía “verde”, es decir que no dañen el medio ambiente. Además dada la situación actual, con el alza del precio del petróleo y el problema ambiental, se ha iniciado con el desarrollo de tecnología que logre solucionar en la medida de lo posible estos problemas. Los constructores de vehículos eléctricos han sabido sacar provecho de esta situación, aferrándose a los adelantos tecnológicos, ofreciendo no sólo vehículos que no contaminan (al menos no directamente) y que permiten la independencia de los combustibles fósiles, sino modelos que además poseen prestaciones que le permiten competir hasta con deportivos de la talla de Porsche o Ferrari.

Los motores eléctricos poseen la cualidad de transformar la energía eléctrica en energía mecánica. En otras palabras obtienen su capacidad de movimiento a partir de la energía eléctrica liberada por una batería o por una célula de combustible de hidrógeno.

Un motor eléctrico posee una menor cantidad de piezas en comparación a un motor de combustión interna, razón por la cual está menos propenso a fallos. Hay que agregar también que en el motor de combustión. Sólo alrededor del 18% de la energía del combustible es utilizada para mover el vehículo, el resto sirve para accionar el motor, mientras que en el vehículo eléctrico el 46% de la energía liberada por las baterías sirve para mover el vehículo, por lo que el tiempo de reacción, del vehículo a motor eléctrico, es menor, pudiendo pasar del reposo al funcionamiento máximo en lapsos mucho menores.

Otras ventajas de los autos eléctricos:

1. Utilizan una energía alternativa.
2. Son más eficientes que los motores de combustión interna.
3. Tienen costos de mantenimiento menores.
4. Son más confiables que los motores de combustión interna.
5. Tienen menos sistemas que los de combustión interna.
6. Los autos eléctricos son hasta 98% menos contaminantes.
7. Son menos ruidosos.

Basados en esta problemática y en la búsqueda de nuevas opciones de transporte, es que se decidió la realización (diseño y construcción) de un vehículo eléctrico. Se busca con esto fomentar el uso de energías alternas, ya que contamos con recursos naturales favorables para la obtención de energía eléctrica. Actualmente el costo de combustibles derivados del petróleo va en aumento, mientras que la obtención de energía eléctrica se puede dar de varias formas naturales, ya sea aprovechando el caudal de un río (presa), a través de molinos de viento (energía eólica) o inclusive utilizar la energía solar (celdas solares).

1.1 ANTECEDENTES.

1.1.1 VEHÍCULO ELÉCTRICO.

Se define como un vehículo de combustible alternativo impulsado por uno o más motores eléctricos.

A diferencia de un motor de combustión interna que está diseñado para funcionar quemando combustible, un vehículo eléctrico obtiene la tracción de los motores eléctricos, pero la energía puede ser suministrada de varios modos, como pueden ser:

- Alimentación externa del vehículo durante todo su recorrido.
- Energía proporcionada en forma de producto químico, que mediante una reacción química producida a bordo, genere energía eléctrica.
- Energía generada a bordo mediante celdas solares.
- Energía química almacenada en baterías.

1.1.2 CONFIGURACIONES TÍPICAS. [2]

Existen diferentes configuraciones cinemáticas para los vehículos eléctricos, éstas dependen principalmente de la aplicación del vehículo. No obstante, de manera general se tienen las siguientes configuraciones:

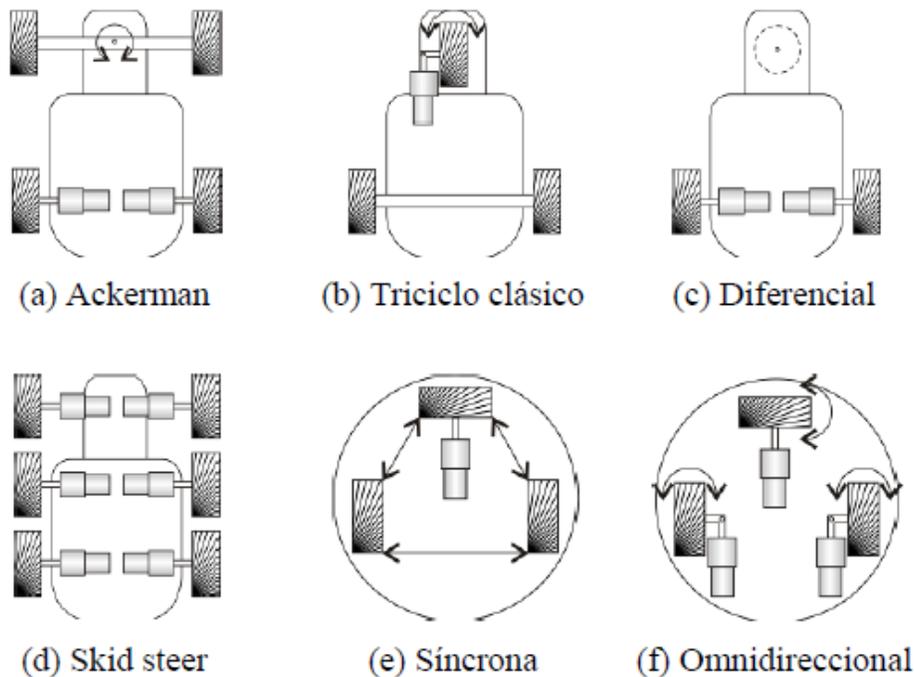


Figura 1.1 configuraciones clásicas de los vehículos móviles

a) CONFIGURACIÓN ACKERMAN.

Complejo y estable. Es el sistema más ampliamente utilizado en automoción. Se basa en un diseño con cuatro ruedas, dos traseras y dos delanteras. Los diseños más extendidos son:

- a) con las ruedas traseras motrices y las delanteras directrices.
- b) con las ruedas delanteras motrices y directrices.
- c) ambos pares de ruedas directrices y tracción delantera, trasera o 4x4

Existe la complejidad de la unión entre ambas ruedas directrices. En este sistema es frecuente el uso de suspensiones para mayor estabilidad y evitar daños por desperfectos del terreno, sin embargo se pueden usar éstas en cualquier diseño.

Este diseño, al igual que el de triciclo, tiene la desventaja de que su orientación está ligada a la posición, es decir, no puede orientarse en una determinada dirección sin avanzar o retroceder, es decir, sin variar su posición, lo que supone un inconveniente en algunas situaciones

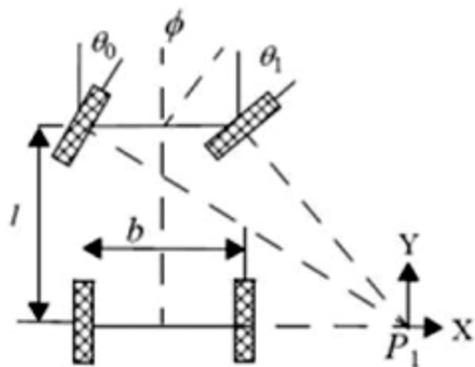


Figura 1.2: Configuración ackerman.

b) CONFIGURACIÓN TRICICLO.

Consta de tres ruedas. Dos ruedas traseras pasivas que no están acopladas a ningún motor, sirven únicamente de soporte. La dirección y tracción del vehículo son proporcionadas por la rueda delantera. En este tipo de configuración no se pueden realizar giros de $\pm 90^\circ$. El radio de giro instantáneo, siempre se encuentra a lo largo del eje que cruza las dos ruedas traseras. Un problema asociado a la configuración triciclo, es que el centro de gravedad del vehículo tiende a moverse lejos de la rueda delantera en terrenos inclinados, cuando el vehículo se encuentra de subida, causando pérdida de tracción.

c) CONFIGURACIÓN DIFERENCIAL.

Consta de dos ruedas colocadas en el eje perpendicular a la dirección del vehículo. Cada rueda es controlada por un motor, de tal forma que el giro del vehículo queda determinado por la diferencia de la velocidad de las ruedas. Así, para girar a la izquierda, hay que darle una mayor velocidad a la rueda derecha. Para girar a la derecha hay que darle una velocidad mayor a la rueda izquierda.

Uno de los problemas que tiene la configuración diferencial es mantener el equilibrio del vehículo, ya que consta de dos ruedas, por lo cual se agregan ruedas de libre giro. Estas ruedas sirven para mantener horizontal el vehículo, por lo que giran libremente según su movimiento

El problema debido a más de tres apoyos del vehículo es la pérdida de tracción y errores en los cálculos de odometría cuando el vehículo se encuentra en terrenos irregulares. La falta de tracción es provocada cuando existe más de una rueda de libre giro, como se puede observar en la figura 1.3.

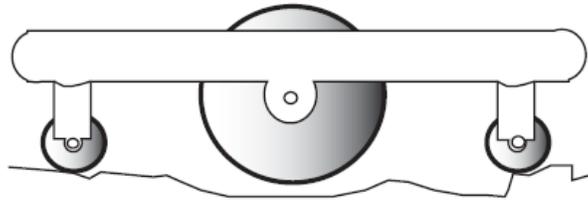


Figura 1.3 Falta de tracción debido a las ruedas de libre giro.

d) CONFIGURACIÓN SKID STEER

La estructura *Skid steer*, es la misma que la configuración diferencial, pero con la diferencia que tiene más de una rueda en cada lado. Las ruedas del mismo lado son manejadas por un mismo motor.

La configuración *Skid steer* puede ser encontrada en una amplia gama de vehículos todo terreno, como por ejemplo montacargas, vehículos de minería y de guerra. Adicionalmente son muy utilizados en aplicaciones relacionadas con la exploración.

Aunque existen varios beneficios mecánicos en la estructura de un *skid steer*, el control dinámico de su movimiento es complejo, debido a que las ruedas deben deslizar lateralmente para lograr un cambio en la dirección del vehículo.

e) CONFIGURACIÓN SÍNCRONA.

Este tipo de configuración, está conformado por tres o más ruedas, todas ellas acopladas mecánicamente y dotadas de tracción, de tal forma que todas rotan en la misma dirección y a la misma velocidad. La orientación de los ejes de rotación se puede controlar.

Este sistema mejora la odometría, reduciendo el deslizamiento de las ruedas, ya que todas las ruedas generan fuerzas con vectores de igual módulo y paralelos en todo momento.

El sistema sincronizado posee dos motores, uno para el movimiento y otro para la dirección, así el vehículo no tiene la necesidad de girar para cambiarla dirección, ya que sólo proporcionan tracción la ruedas traseras.

f) CONFIGURACIÓN OMNIDIRECCIONAL.

Este tipo de configuración está provista de ruedas omnidireccionales, lo que hace que los cálculos de odometría sean más complicados; pero el vehículo podrá moverse en cualquier dirección. La rueda omnidireccional, se define como una rueda estándar, a la cual se le ha dotado de una corona de rodillos. Los ejes de giro de los rodillos, son perpendiculares a la dirección normal de avance. De este modo, al aplicarle una fuerza lateral, los rodillos giran sobre sí mismo y permite que la componente de la velocidad en el eje x no sea nula.

1.2 ESTADO DEL ARTE. [7]

El vehículo eléctrico fue uno de los primeros automóviles que se desarrollaron, al punto que existieron pequeños vehículos eléctricos anteriores al motor de cuatro tiempos sobre el que Diesel y Benz, basaron los diseños actuales. Entre 1832 y 1839 el escocés Robert Anderson, inventó el primer vehículo eléctrico puro, con pila de energía no recargable.

El “*Jamais Contente*” fue el primer vehículo eléctrico en superar los 100 km/h. Es un coche eléctrico con una carrocería de aleación ligera con forma de torpedo (Figura 1.4).

Los automóviles eléctricos, producidos en los Estados Unidos por Anthony Electric, Baker, Detroit y Edison (figura: 1.5) tuvieron cierto éxito comercial a principios del siglo XX. Ocasionado por las limitaciones tecnológicas, la velocidad máxima de estos primeros vehículos se limitaba a 32 km/h.



Figura 1.4: Dibujo de la “*Jamais Contente*”.

A finales de 1930, la industria del automóvil eléctrico desapareció por completo, quedando relegada a algunas aplicaciones industriales muy concretas, como montacargas o más recientemente como carros de golf eléctricos.



Figura 1.5: Coche eléctrico fabricado por Thomas Edison en 1913. Museo nacional de historia americana

Desechando prototipos, modelos anecdóticos y vehículos de uso industrial, fue en 1990, cuando *General Motors* presenta el *impact*. Fue el precursor de vehículo eléctrico más famoso de la historia el *General Motors Experimental Vehicle 1 (EV-1)*.

En la década de los 90's debido a iniciativas legislativas como la "*Zero Emission Mandate*", los fabricantes de autos comenzaron a sacar coches de emisiones cero para cumplir con la ley.

Algunos otros ejemplos son: Chevrolet S-10 (100 km, recarga 7 h), Solectria Geo Metro (80 km, recarga 8 h), Ford Ecostar (112 km/h, hasta 160 km) o Ford Ranger (120 km/h, 105 km). Pero el que fue un éxito con diferencia fue el EV-1.



Figura 1.6: General Motors Experimental Vehicle 1 (EV-1).

Actualmente se desarrollan vehículos eléctricos con muchas más capacidades que sus antecesores, la mejora en las baterías y diseños futuristas hacen cada vez más llamativo un coche eléctrico sobre uno de combustión.

1.3 VEHÍCULOS ELÉCTRICOS E HÍBRIDOS DE 3 RUEDAS.

Hoy en día el desarrollo de vehículos eléctricos está en aumento, es así como en el mercado han aparecido prototipos de tres o cuatro ruedas. A continuación se muestran algunos prototipos de vehículos eléctricos de 3 ruedas:

1.3.1 TWIKE. [8]

El Twike diseñado por la empresa alemana Mobile es un vehículo híbrido que funciona a pedales y electricidad. La velocidad máxima que puede alcanzar usando solamente la batería es de 85 km/h, con una autonomía de 120 Km. Cuando se acaba la carga puede continuar su camino, gracias a los pedales, además con esto se va recargando la batería.

El interior tiene espacio para dos pasajeros, el vehículo es controlado por un "joystick" que incorpora controles para acelerar y frenar.

En caso de que no se quiera o no se pueda pedalear, el Twike tiene la opción de funcionar con gasolina.



Figura 1.7: Vehículo eléctrico Twike.

1.3.2 MYERS NMG. [9]

Myers Motors desarrolló el Nmg (por sus siglas en inglés: *No more gas*), un auto eléctrico para una sola persona con un diseño futurista, este pequeño automóvil puede alcanzar velocidades hasta de 120 km/h, con una autonomía hasta de 48 km.

Su batería se recarga en un 100 % con tan sólo 3 horas de carga conectado a cualquier toma corriente común.



Figura 1.8: Vehículo Myers Nmg.

1.3.3 SAM. [10]

Vehículo eléctrico desarrollado en Suiza con capacidad para dos personas, cuenta con tres ruedas, dos delanteras y una posterior.

Con poco más de 500 Kg de peso, puede alcanzar velocidades de hasta 85 km/h, logrando acelerar de 0 a 50 km/h en 7 segundos.

La carga completa de sus baterías toma aproximadamente seis horas y es suficiente para darle autonomía de unos 50 a 70 km.



Figura 1.9: Vehículo eléctrico SAM

1.3.4 APTERA. [11]

La empresa estadounidense Aptera desarrolló el vehículo eléctrico de mismo nombre. El aptera contiene un par de motores, un motor diesel y un motor eléctrico. Con poco más de 400 kg de peso el Aptera puede adquirir velocidades hasta de 120 km/h.

Con un consumo de 1 litro de diesel logra tener una autonomía hasta de 100 km con una velocidad media de 89 km/h. el vehículo se puede recargar con un toma corriente común y su tiempo de carga es de aproximadamente 6 horas.



Figura 1.10: Vehículo eléctrico Aptera.

1.4 VEHÍCULOS ELÉCTRICOS EN MÉXICO.

1.4.1 ZILENT AUTOS ELÉCTRICOS. [12]

La marca mexicana de automóviles Zilent actualmente comercializa 4 modelos de vehículos eléctricos, todos ellos con configuración ackerman. Se tienen básicamente 3 modelos familiares y uno para transporte de mercancía.

El lanzamiento del primer modelo comercial de esta empresa fue en marzo de 2009. Las características de los 4 modelos son las siguientes:

1.4.2 COURANT 2010.

Vehículo eléctrico de 5 puertas, con un peso aproximado de 1100 kg logra alcanzar una velocidad máxima de 90 km/h. funciona a con un motor eléctrico alimentado por 10 baterías de ácido-plomo.

Tiene una autonomía de entre 60 y 120 km, logrando transportar hasta 5 pasajeros.



Figura 1.11: Courant 2010

1.4.3 DREAM.

Es un vehículo eléctrico de 5 puertas, con capacidad para cuatro pasajeros. Con un peso aproximado de 900 kg logra alcanzar velocidades de hasta 80 km/h con una autonomía de entre 60 y 120 km.

Su motor se alimenta con 10 baterías de ácido-plomo de ciclo profundo recargables.



Figura 1.12: Dream.

1.4.4 Nissan. [13]

La empresa automotriz Japonesa Nissan en asociación con el gobierno de la Ciudad de México distribuirá vehículos eléctricos a final de 2011 en la Ciudad de México. Con la intención de disminuir la dependencia de hidrocarburos.

El Nissan LEAF, un sedan compacto para cinco pasajeros con velocidad de hasta 160 kilómetros por hora. El auto tiene una carga de ocho horas o una rápida de treinta minutos.



Figura 1.13: LEAF de Nissan.

1.4.5 VEHÍCULO ELÉCTRICO DE TURISMO (VET).

Actualmente en la Universidad Nacional Autónoma de México, Facultad de Ingeniería, en el Centro de Diseño Mecánico e Innovación Tecnológica (CDMIT), se está desarrollando un vehículo totalmente eléctrico de uso turístico.

Entre los requisitos de diseño del VET, se solicitó que éste debería ser innovador en el mercado actual, haciéndolo atractivo para los clientes y competitivo con vehículos de su misma clase, por lo cual, el equipo de diseño decidió dotarlo, entre otros aspectos que hacen del VET un vehículo completamente innovador, de un volante extraíble con controles al mando, característica que actualmente sólo se encuentra en autos de carreras de F1 o deportivos comerciales con adaptaciones especiales. [3]

Por otra parte el cliente solicitó que el vehículo eléctrico fuera de 3 ruedas, para cuatro personas, para uso turístico y de bajo costo, que alcanzara una velocidad cruceo de 40 km/h y una velocidad máxima de 60 km/h. De este modo para el diseño del sistema motriz se tienen los siguientes requerimientos: [4]

- Tiene que competir a la velocidad promedio de los vehículos en la ciudad.
- Periodos largos de trabajo.
- Fácil mantenimiento.
- Tiempos cortos de manufactura.
- Bajos costos.

Hasta este momento se ha desarrollado el chasis del vehículo así como los sistemas motriz, de mando, luces traseras y delanteras. Dentro del sistema motriz se encuentran ya instalados los dos

motores que darán tracción al vehículo, y al término de este trabajo se contará con el sistema electrónico de control.

El sistema electrónico de control se encargará de regular la velocidad de los motores para dar la dirección al vehículo solicitada por el usuario mediante el acelerador y el volante, además también se encargará de administrar los periféricos con los que se encuentra equipado el VET.



Figura 1.14: Propuesta para el VET.

Capítulo 2

Objetivos

- Regular la velocidad de los motores *Brushless* a través del ángulo de giro del volante y el acelerador del vehículo.
- Poder controlar la dirección del vehículo, por medio de la diferencia de velocidades en los motores delanteros del vehículo.
- Diseñar la tarjeta electrónica para poder controlar los motores, así como los demás periféricos que tiene el vehículo.
- Realizar pruebas de la tarjeta de control.

2.1 ALCANCES.

Desarrollar el algoritmo de control para la operación de los motores *Brushless*.

Diseñar la electrónica para el acondicionamiento de señales, adecuándolas a los requerimientos del microcontrolador.

Diseñar la tarjeta impresa (PCB) para la electrónica de control y acondicionamiento de las señales de la tarjeta.

Supervisar la fabricación y ensamblaje de la tarjeta de control.

2.2 RESEÑA GENERAL.

Actualmente en el Centro de Diseño Mecánico e Innovación Tecnológica (CDMIT) de la Facultad de ingeniería, se está desarrollando un vehículo eléctrico turístico (VET). Con base en la necesidad de ser de 3 ruedas se seleccionó una configuración de tracción diferencial para el vehículo. Utilizando dos motores, que proporcionan la tracción en las llantas delanteras, y baterías eléctricas para las funciones del vehículo.

El reto de tener una configuración diferencial se encuentra en la regulación de velocidad en las llantas delanteras. Es por eso que este trabajo está dedicado a presentar una de las múltiples opciones para regular la velocidad de los motores. Mediante una tarjeta electrónica basada en un microcontrolador. Esta tarjeta hará el cálculo para la regulación de velocidad de las llantas delanteras para poder dar la orientación al vehículo, así como controlar los periféricos que contiene el VET.

Es indispensable comenzar por explicar los avances que se han tenido en cuestión de vehículos eléctricos. Es por esto que en el capítulo 2 se hace referencia a los modelos desarrollados a nivel internacional. En esta parte de la investigación, se trató de hacer énfasis en los vehículos que cuentan con una configuración similar a la del VET, aunque también se mencionan algunos con configuraciones distintas. Finalmente se da una introducción a lo que es el proyecto VET, los requerimientos y las expectativas de éste.

En el capítulo 3 se presentan las características que tiene el sistema electrónico de control (SEC) los actuadores que se utilizaron para el VET así como los sensores y sus principales características. En este capítulo también se da una visión de las principales funciones del SEC, incluida la comunicación con el volante del vehículo.

En el capítulo 4 se hace referencia a los algoritmos utilizados para la resolución de los diversos problemas que se presentaron en el SEC. Se dividió el problema en varios sub-problemas y se resolvieron por separado. El método fue plantear el problema, definir los alcances de la posible solución y posteriormente desarrollar el algoritmo. Una vez que se tuvieron las soluciones parciales para los sub-problemas se integró en una solución global. Y se plasmó en lenguaje que fuese entendido por el microcontrolador, para desarrollar el programa del SEC.

El desarrollo de todo el acondicionamiento de las señales, se especifica en el capítulo 5, donde se tiene de manera detallada la selección de los componentes electrónicos que integran al SEC, y cuál es su función dentro del mismo. Además se explica cómo fue el desarrollo de la tarjeta PCB del SEC, a partir del diseño del esquema eléctrico hasta su fabricación, algunos puntos del diseño de la tarjeta que fueron importantes y algunos detalles como el diseño para que la PCB del SEC se pudiera acoplar a la tarjeta de desarrollo del microcontrolador seleccionado.

El probar las cosas que se desarrollan es parte del proceso de diseño y éstas se explican en el capítulo 6, donde se muestran algunas pruebas que se realizaron al SEC, sin carga y con carga. Es decir, en las primeras pruebas se verificaron las salidas y las entradas, que todas funcionaran adecuadamente a través del osciloscopio y apoyados de fuentes de alimentación. Posteriormente, se acoplaron los demás elementos al SEC (actuadores y sensores) para finalmente validar el sistema electrónico de control.

Capítulo 3

Sistema electrónico de control.

3. CARACTERÍSTICAS DEL SISTEMA ELECTRÓNICO DE CONTROL. (SEC)

El sistema electrónico de control (SEC) hará las funciones de “cerebro” para el vehículo eléctrico de turismo (VET). El SEC recibirá información de los distintos sensores con los que cuenta el VET, para ser procesada y posteriormente ser enviada hacia el volante del vehículo.

Además el SEC detectará el estado de los pedales (acelerador y freno) para determinar las acciones que ejecutará los actuadores (motores) del vehículo.

Por su parte, el volante le envía información al SEC para indicarle el “estado requerido por el usuario” de los periféricos. Es decir, qué periféricos está solicitando el usuario (luces internas, intermitentes, direccionales, etc.), el SEC procesa la información y manda las señales para que se activen los periféricos solicitados.

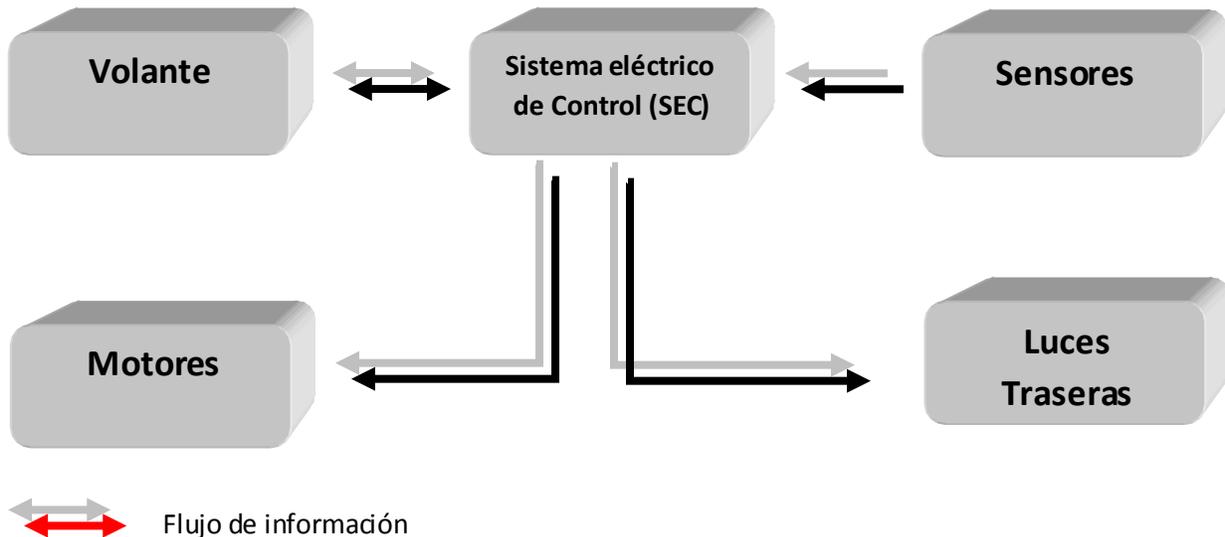


Figura 3.1: Diagrama de flujo de información.

3.1 VOLANTE.

El volante es la parte del sistema de dirección que se emplea a modo de interfaz entre el usuario y el vehículo, con éste se logra que el vehículo tome la trayectoria deseada por el usuario. En el VET esto se logra a partir de una señal analógica, representando la posición angular del volante, que es procesada por el SEC para poder determinar la velocidad angular que debe tener cada llanta, para así poder seguir la trayectoria solicitada.

La Interfaz de Usuario (IU) es la parte del sistema electrónico del módulo electrónico de gestión del volante (MEGV) que tiene interacción con el usuario y tiene dos funciones principales. Una de las funciones es informar el estado del vehículo a través de un display, donde se muestra la velocidad del vehículo, nivel de batería y estimación del recorrido restante, las actualizaciones se realizan al ocurrir cambios. La otra función es comunicar el estado del MEGV al SEC debido a la modificación de ciertas funcionalidades del automóvil, tales como: luces, daxon, direccionales, intermitentes y limpia parabrisas [3].



Figura 3.2: Volante del vehículo.

El volante cuenta con botones de mando, los cuales son utilizados por el usuario para dar la orden de encender algún periférico del vehículo, como se puede apreciar en la figura 3.2. Estos botones son:

- Direccional derecha.
- Direccional izquierda.
- Claxon.
- Luces frontales.
- Luces interiores.
- Reversa.
- Indicación de parada.
- Limpiaparabrisas.

Además se encuentra equipado con un display en su centro, donde muestra de manera gráfica el estado de la batería (nivel de carga), la velocidad del vehículo, el estado del parabrisas y las luces, tanto direccionales como frontales (tabla 3.2).

	Direccional izquierda	Al presionarse una vez prenderá la direccional izquierda y apagará la derecha en caso de estar prendida. Al presionarla otra vez se apagará.
	Direccional derecha	Al presionarse una vez prenderá la direccional derecha y apagará la izquierda en caso de estar prendida. Al presionarse otra vez se apagará.
	Luces de emergencia	Al presionarse una vez se prenderán las luces de emergencia, si alguna direccional está prendida se apagará. Al presionarse otra vez se apagará.
	Luces exteriores	Al presionarse una vez se encenderán las luces exteriores, al presionarse otra vez se apagarán. (Ver uso del contraste).
	Luces interiores	Al presionarse una vez se encenderán las luces interiores, al presionarse otra vez se apagarán. (Ver uso del contraste).
	Reversa	Al presionarse una vez la reversa se encenderá, al presionarse otra vez la reversa se apagará.
	Limpia parabrisas	Al presionarse una vez se activa el limpiaparabrisas en velocidad baja, si se presiona otra vez se activa en velocidad media, si se presiona por tercera vez se activa la velocidad alta, si se presiona por cuarta vez se apaga.
	Claxon	Al presionarse suena el claxon, al soltarse se apaga.

Tabla 3.2: botones del teclado, de interfaz de usuario.

Cuenta con comunicación RS-485 Half-Dúplex a una velocidad de 115,200 [bps], por donde se enviará y se recibirá información del SEC.

La comunicación se realiza de la siguiente manera:

1. La tarjeta del volante (esclavo) espera información del procesador central SEC (maestro), si ésta no llega se reinicia el *WatchDog Timer (WDT)*.
2. Cuando el esclavo recibe la trama de información la almacena para su posterior procesamiento.
3. Después de almacenar y procesar los datos, el esclavo envía información al maestro.
4. Se reinicia el proceso.

El volante se comunica a través de 8 bits en paralelo con el sistema de luces traseras (SLT) para indicarle el estado de los comandos en el volante, y así el SLT ejecuta la acción indicada. (Intermitentes, direccional, cuartos, etc.).

3.2 MOTORES. [14]

Actualmente, los motores *brushless* de corriente directa (BLDC) se emplean en sectores industriales tales como: automóvil, aeroespacial, equipos de automatización etc. Los motores BLDC tienen la característica de que no emplean escobillas en la conmutación para la transferencia de energía; en este caso, la conmutación se realiza electrónicamente. Esta propiedad elimina el enorme problema que tienen los motores eléctricos convencionales con escobillas, los cuales producen rozamiento, disminuyen el rendimiento, desprenden calor, son ruidosos y requieren una sustitución periódica y, por lo tanto un mayor mantenimiento.

Los motores BLDC tienen muchas ventajas frente a los motores DC con escobillas. Algunas de estas ventajas son:

- ✓ Mejor relación velocidad-par motor
- ✓ Mayor respuesta dinámica.
- ✓ Mayor eficiencia.
- ✓ Mayor vida útil.
- ✓ Menor ruido.
- ✓ Mayor rango de velocidad.
- ✓ Menor peso para la misma potencia
- ✓ Mejor disipación de calor.

El VET utiliza dos motores *brushless* de la compañía *Golden Motor* (figura 3.3) con capacidad de 72 volts y 5Kw, son dos motores *brushless* (uno en cada rueda delantera del vehículo) que estarán alimentados con 6 baterías de 12 volts de corriente directa.



Figura 3.3: Motor Modelo Golden motor HPM500B

Se utilizó un controlador modelo HPC300A (figura 3.4) de la marca Golden Motor. Dicho controlador se alimenta con 72 volts de corriente directa. Y se necesita indicarle, al controlador, la velocidad para los motores a través de una señal analógica de 0 a 5 volts (de 0 a 5000 rpms). con esta señal es con la que se hace la regulación de velocidad, al poder controlar la señal analógica de 0 a 5 volts, se pueden regular las rpms para cada motor, Además de esto, también provee la corriente necesaria para los motores.



Figura 3.4: Controlador HPC300A.

3.3 SENSORES.

Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Los sensores detectaran las variaciones en las magnitudes (posición angular, velocidad angular) y las transforman en magnitudes eléctricas que pueda interpretar el sistema eléctrico de control.

Dadas las características del VET se seleccionaron los sensores que más se adecuaran a sus necesidades. A continuación se muestran los sensores que se utilizaron.

3.3.1 PEDAL ACELERADOR Y FRENO.

El pedal tiene un potenciómetro, que convierte la posición angular en una señal eléctrica cuando se alimenta al pedal del acelerador con 5 volts DC, proporciona una señal analógica de 0 a 5 volts, que será procesada por el sistema electrónico de control para con esto poder regular la velocidad de los motores *brushless*.



Figura 3.5: Pedal acelerador. Cortesía Golden motor.

El potenciómetro proporciona un divisor del voltaje, al alimentar en sus extremos una diferencia de potencial de 5V, la salida del divisor de voltaje varía dependiendo de la posición del eje o flecha, como se puede observar en la figura 3.6(A). Esto debido a que la resistencia del potenciómetro incrementa o decremента. Así por ejemplo, cuando el potenciómetro se encuentre completamente a la izquierda (Pedal totalmente presionado) mandará una señal de 5 volts, y cuando se encuentre completamente a la derecha (pedal sin presionar) mandara una señal de 0 volts, como se puede observar en la figura 3.6 (B)

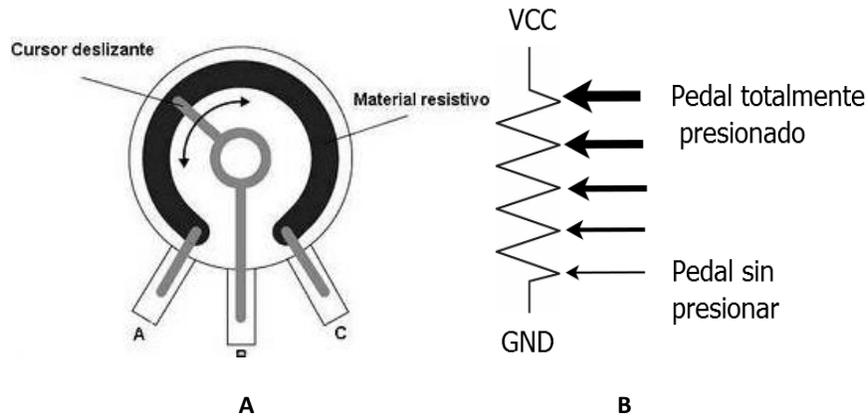


Figura 3.6: Configuración del potenciómetro en el pedal acelerador..

3.3.2 POSICIÓN DEL VOLANTE.

Para poder registrar las diferentes posiciones del volante también se utilizó un potenciómetro, colocado en su centro de giro. Este potenciómetro se encuentra en una posición natural (es decir cuando el vehículo se mueve en línea recta) en un voltaje de 2.5 [V] dependiendo el sentido en el que se mueva, incrementa o decremента el valor del voltaje. Así la terminal variable en el potenciómetro cambiará de posición proporcionalmente a la orientación del volante (figura 3.7 B). Así por ejemplo cuando el potenciómetro se encuentre completamente a la izquierda ($\theta=45^\circ$)

mandará una señal de 5 volts, y cuando se encuentre completamente a la derecha ($\theta = -45^\circ$) mandará una señal de 0 volts, como se puede observar en la figura 3.7 (B)

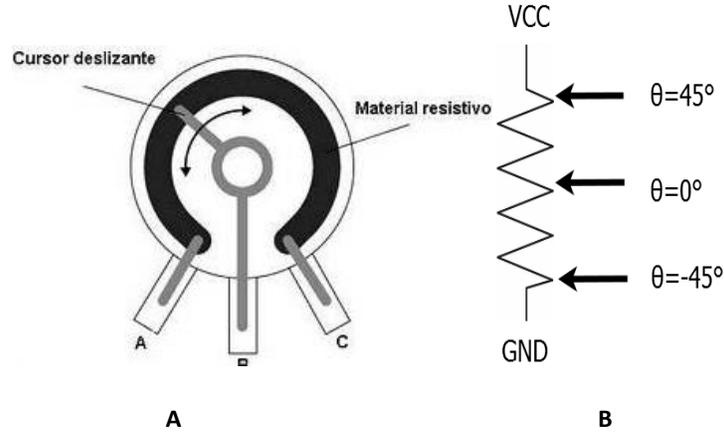


Figura 3.7: Configuración del potenciómetro en el volante.

3.3.3 VELOCIDAD ANGULAR DE LAS LLANTAS.

Para poder medir la velocidad de las ruedas del vehículo, se utilizaron sensores de efecto Hall 103SR13A-1 que mandan una señal cada que localizan el campo magnético alrededor de la rueda. En forma de un pulso que es contabilizado por la tarjeta SEC. Una vez teniendo las revoluciones de las llantas, se puede estimar la velocidad media que lleva el vehículo. Esta información nos sirve para poder visualizar la velocidad en el display del volante y para poder cerrar el lazo de control en la regulación de velocidad.

Se alimenta con un voltaje de corriente directa de entre 4.5 y 20 volts, con 152 mm de largo y un cable trenzado, tiene una salida de voltaje del 40% del voltaje de alimentación en corriente directa. [15].



Figura 3.8: Sensor 103SR13A-1.

Capítulo 4

Programación del control

INTRODUCCIÓN

4.1 ALGORITMOS [5]

Un algoritmo es una serie *finita* de pasos para resolver un problema. Se tiene que hacer énfasis en dos aspectos para que el algoritmo exista:

1. El número de pasos debe ser finito. De esta manera el algoritmo debe terminar en un tiempo finito con la solución del problema.
2. El algoritmo debe ser capaz de determinar la solución del problema.

Las características de un algoritmo son:

1. Entrada: definir lo que necesita el algoritmo.
2. Salida: definir lo que produce.
3. No ambiguo: explícito, siempre sabe que comando ejecutar.
4. Finito: el algoritmo termina en un número finito de pasos.
5. Correcto. Hace lo que se supone debe hacer. La solución es correcta.
6. Efectividad: cada instrucción se completa en tiempo finito. Cada instrucción debe ser lo suficientemente básica como para que en un principio pueda ser ejecutada por cualquier persona usando papel y lápiz.
7. General. Debe ser lo suficientemente general para contemplar todos los casos de entrada.

Cuando se tiene un problema para el cual debemos especificar un algoritmo solución, se debe tener en cuenta varios puntos.

- ◆ Si no se conoce un método para solucionar el problema en cuestión, debemos hacer un análisis del mismo para llegar a una solución, después de evaluar alternativas, exigencias y excepciones.
- ◆ Si se conoce un “buen” método de solución al problema, se debe especificar exacta y completamente el método o procedimiento de solución en un lenguaje que se pueda interpretar fácilmente.

Al definir con exactitud un método de solución para un problema, éste debe estar en capacidad de encontrarla si existe, en caso de que no exista, el método debe reconocer esta situación y suspender cualquier acción.

El análisis de un problema tiene dos etapas claramente definidas y relacionadas.

- Formulación o planteamiento del problema.
- Resolución del problema.

A su vez la formulación la podemos descomponer en tres etapas.

- Definición del problema.
- Supuestos: aserciones y limitaciones.
- Resultados esperados.

La fase de planteamiento del problema lo que pretende definir un algoritmo es sintetizar de alguna forma una tarea, cálculo o mecanismo antes de ser transcrito en el código de un programa. Los pasos que hay que seguir son los siguientes:

- Análisis previo del problema.
- Primera visión del método de resolución.
- Descomposición en módulos.
- Programación estructurada.
- Búsqueda de soluciones parciales.
- Ensamblaje de soluciones finales.

La fase de resolución del problema se puede descomponer en tres etapas.

- Análisis de alternativas y selección de la solución.
- Especificación detallada del procedimiento de solución.
- Utilización de una herramienta para su implementación.

En otras palabras, no es una computadora o un microcontrolador el que soluciona los problemas, somos nosotros quienes lo hacemos y de alguna manera le contamos cómo es la cosa para que ellos, con su velocidad y exactitud, trabajen con grandes volúmenes de datos.

La solución de los problemas se describe mediante el diseño de procedimientos llamados algoritmos, los cuales posteriormente se implementan como programas.

4.2 ESTRATEGIAS PARA EL DISEÑO DE ALGORITMOS.

A lo largo del tiempo, se han identificado diversas técnicas generales que a menudo producen algoritmos exitosos, eficientes para la resolución de muchos problemas. A continuación se enumeran un par de ellas.

I) RECURSIÓN

La recursividad es una técnica fundamental en el diseño de algoritmos eficientes, que está basada en la solución de versiones más pequeñas del problema, para obtener la solución general del mismo. Una instancia del problema se soluciona según la solución de una o más instancias diferentes y más pequeñas que ella.

La recursividad es una alternativa a la iteración o repetición, y aunque en tiempo de ejecución y espacio de memoria la solución recursiva es menos eficiente que la solución iterativa, existen numerosas situaciones en las que la recursividad es una solución simple y natural a un problema que pudiese ser difícil de resolver.

La característica importante de la recursividad es que siempre existe un medio de salir de la definición, mediante la cual se termina el proceso recursivo.

Ventajas:

- Puede resolver problemas complejos.
- Solución más natural.

Desventajas:

- Se puede llegar a un ciclo infinito.
- Puede ser difícil de programar.

a) Propiedades para un Procedimiento Recursivo

- Debe de existir cierto criterio (criterio base) en donde el procedimiento no se llama a sí mismo. Debe existir al menos una solución no recursiva.
- En cada llamada se debe de acercarse al criterio base (reducir rango). El problema debe reducirse en tamaño, expresando el nuevo problema en términos del propio problema, pero más pequeño.

Los algoritmos de divide y vencerás pueden ser procedimientos recursivos.

b) Propiedades para una Función Recursiva

- Debe de haber ciertos argumentos, llamados valores base para los que la función no se refiera a sí misma.
- Cada vez que la función se refiera a sí misma el argumento de la función debe de acercarse más al valor base.

Recursividad directa o simple, es cuando un subprograma se llama así mismo una o más veces directamente.

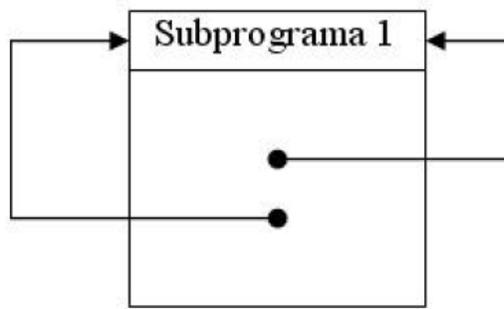


Figura 4.1: Representación de recursividad directa.

Recursividad Indirecta o mutua: un subprograma "1" llama a otro subprograma "2" y éste a su vez llama al subprograma "1".

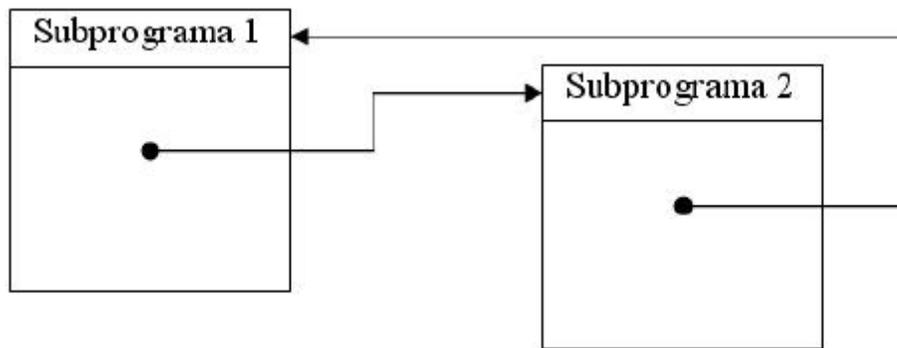


Figura 4.2: Representación de recursividad indirecta.

II) DIVIDIR PARA CONQUISTAR

Muchos algoritmos útiles tienen una estructura recursiva, de modo que para resolver un problema se llaman recursivamente a sí mismos una o más veces para solucionar sub-problemas muy similares.

Esta estructura obedece a una estrategia dividir y conquistar, en que se ejecutan tres pasos en cada nivel de la recursión:

- **Dividir:** Dividen el problema en varios sub-problemas similares al problema original pero de menor tamaño.
- **Conquistar:** Resuelven recursivamente los sub-problemas si los tamaños de los sub-problemas son suficientemente pequeños, entonces resuelven los sub-problemas de manera directa; y luego...
- **Combinar:** Combinan estas soluciones para crear una solución al problema original.

La técnica Dividir para Conquistar (o Divide y Vencerás) consiste en descomponer el caso que hay que resolver en sub-casos más pequeños, resolver independientemente los sub-casos y por último, combinar las soluciones de los sub-casos para obtener la solución del caso original.

Para aplicar la estrategia Divide y Vencerás es necesario que se cumplan tres condiciones:

- La decisión de utilizar el sub-algoritmo básico en lugar de hacer llamadas recursivas debe tomarse cuidadosamente.
- Tiene que ser posible descomponer el caso en sub-casos y recomponer las soluciones parciales de forma eficiente.
- Los sub-casos deben ser, en lo posible, aproximadamente del mismo tamaño.

4.3. ALGORITMO DEL SEC.

Primeramente se plantearon los problemas a resolver en el control del VET, para resolver cada una de esas problemáticas por separado y posteriormente integrarla en una solución global para el SEC.

1. **Comunicación del SEC con el volante:** dado que el volante se comunica mediante el módulo de comunicación serial USART utilizando el estándar o norma de comunicación RS-485, el SEC se debe manejar este estándar. Además el SEC debe recibir los datos que envíe el volante procesarlos y enviar una respuesta de regreso al volante.
2. **Encendido:** como medida de seguridad, se determinó que debe existir una secuencia de encendido. Oprimiendo una combinación de los botones de mando en el volante.
3. **Ciclo de trabajo PWM para cada motor:** Determinar la diferencia de velocidades que tendrá cada llanta, para ajustar la orientación del vehículo indicada por el usuario mediante el volante.
4. **Determinar la velocidad lineal de cada llanta:** como parte de la información que se manda del SEC hacia el volante, se incluye la velocidad media del vehículo, lo cual implica de manera directa conocer la velocidad que tiene cada llanta, para poder estimar la velocidad.
5. **Determinación del error en la velocidad:** con base en la determinación de la velocidad en las llantas, también se puede estimar una velocidad media deseada por el usuario (por medio del pedal acelerador y la dirección) esto nos lleva a poder establecer un error de velocidad que hay que corregir en la velocidad de cada llanta.

4.3.1 COMUNICACIÓN DEL SEC CON EL VOLANTE.

Este debe ser el primer problema a resolverse, dado que el funcionamiento de los demás sistemas depende directa o indirectamente de la información obtenida entre la comunicación del SEC y el volante. Así por ejemplo la activación de cualquier periférico del vehículo (luces delanteras, claxon, intermitentes, etc.) depende directamente del valor que envíe el volante hacia el SEC.

Se definió un protocolo de comunicación entre el SEC y el volante, para poder establecer una comunicación bilateral entre estos dos elementos del vehículo. Los protocolos se muestran en las tablas 4.1 y 4.2. Así en la cadena enviada del esclavo al maestro (volante al SEC) se puede observar que se inicia con un encabezado o ID Slave (8 bits), que hace la función de referencia de quien envía la cadena. Posteriormente viene el ID Master (8bits), enseguida se envía información de los botones pulsados en el volante (*Data0*), en 8 bits representados por un “cero” o estado lógico bajo y por “uno” o estado lógico alto. Enseguida en un *Data1* se envía la información del freno y de los limpia-parabrisas, de igual manera que en *Data0* representados por estados lógicos.

Finalmente se envía un código de detección de errores, para identificar cambios accidentales en los datos enviados, llamado CRC (por sus siglas en inglés *cyclic redundancy check*). Que se construye, mediante la función booleana XOR entre los datos enviados. Así el CRC enviado en esta trama es: $Data0 \text{ XOR } Data1$.

De igual manera, la cadena de se envía del maestro al esclavo (SEC del volante), contiene sus dos encabezados al inicio, ID MASTER e ID SLAVE. Posteriormente se manda el *Data0*, que contiene la información de la velocidad con un valor entero de 0 a 60 (representando la velocidad en km/hr del vehículo). El *Data1* contiene en un entero de 0 a 99 la información del nivel de carga de las baterías del vehículo. El *Data2* envía en un entero de 0 a 99 una estimación del recorrido que puede hacer el vehículo con el nivel de carga que se cuente. Posteriormente se envía información del freno en un solo bit. Y por último se conforma el CRC a enviar, que en este caso es $Data0 \text{ XOR } Data1 \text{ XOR } Data2$

ALGORITMO.

Como sólo se trata de comunicación, por el momento sólo se hace énfasis en “leer” y “enviar” información. Posteriormente se manipulará la información recibida. Así se procede a describir el algoritmo. (Figura 4.3)

1. Configuración de variables. Es la parte donde se configuran las variables que se utilizarán, en este caso 2 arreglos uno de tamaño 7 (transmisión de datos del SEC al volante) y otro de tamaño 5 (recepción de datos del Volante al SEC).
2. Asignación de valores a la cadena a enviar. Se debe de “leer” el valor de los sensores (acelerador, batería) y estimar el número de kilómetros que se puede recorrer con el nivel de carga que se tiene. Así se asignan los valores en la cadena a enviar. Recordando que los primeros 2 datos son los ID MASTER e ID SLAVE.

INFORMACIÓN DE ESCLAVO A MAESTRO (VOLANTE A SEC)						
ID_SLAVE	ID_MASTER	DATA 0				
CC	AA	VACÍO	CLAXON	REVERSA	LUCES_INT	LUCES_EXT
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3
		Vacío	0:apagado	0:apagado	0:apagado	0:apagado
		vacío	1:encendido	1:encendido	1:encendido	1:encendido

INFORMACIÓN DE ESCLAVO A MAESTRO (VOLANTE A SEC)						
			DATA 1			CRC
INTER	DRI_DER	DIR_IZQ	FRENO	LIMPIADORES		DATA0 XOR DATA1
Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
0:apagado	0:apagado	0:apagado	0:suelto	00 apagado	01 vel1	
1:encendido	1:encendido	1:encendido	1:presionado	10 vel2	11 vel3	

Tabla 4.1: Protocolo de datos a enviar del volante al SEC.

INFORMACIÓN DE MAESTRO A ESCLAVO (SEC A VOLANTE)						
ID_MASTER	ID_SLAVE	DATA0	DATA1	DATA2	FRENO	CRC
AA	CC	VELOCIDAD	BATERÍA	RECORRIDO	Bit	DATA0 XOR DATA1 XOR DATA2
		Int	Int	Int	0:suelto	
		0-60	0-99	0-99	1:presionado	

Tabla 4.2: Protocolo de datos a enviar del SEC al volante

3. Enviar la cadena. Una vez que se cuenta con la cadena, se envían los datos a través del puerto serial en una secuencia de uno por uno, para que el volante pueda reconocerlos y almacenarlos de manera correcta.
4. Esperar a que el volante termine de recibir los datos y de almacenarlos.
5. Leer los valores que mande el volante
6. Almacenar en el arreglo previamente definido los valores enviados por el volante y en el orden indicado.
7. Volver a asignar valores a las variables a enviar (actualizar información).



Figura 4.3: Algoritmo para la comunicación entre el volante y el SEC.

4.3.2 ENCENDIDO.

Una vez que se tiene la comunicación con el volante, se procede a manipular los datos enviados del volante al SEC. Para que el SEC active las salidas digitales que está requiriendo el usuario desde el volante.

Mientras el vehículo no haya arrancado, podrá mantener comunicación del volante con el SEC, para poder monitorear el nivel de batería y recorrido. También se manipulan los periféricos con acceso desde el volante, sin ningún problema. Una vez que se haya encendido el vehículo, el usuario podrá manipular todas sus funciones.

Se eligió que para poder arrancar el vehículo (poder energizar los controladores a través del acelerador) se tenga que apretar 3 veces el botón del claxon, ubicado en el volante. De esta manera el SEC tendrá que detectar cuando el claxon se ha oprimido por tercera ocasión y mandar la señal hacia los controladores y estos hacia los motores.

ALGORITMO

El algoritmo de encendido utiliza el algoritmo “comunicación con el volante”, ya que se leen los valores guardados en el arreglo (que se nombró “recep”). Así en el arreglo de dimensión 5 el valor que nos interesa conocer será el tercero (figura 4.4) . Cabe mencionar que para llegar a este punto, previamente se debió de hacer corroborado los encabezados de la cadena, como se explicará más adelante en el algoritmo.

Recep [0] ID_SLAVE	Recep[1] ID_MASTER	Recep[2] Datos de los botones	Recep[3] Datos de limpiadores.	Recep[4] CRC
-----------------------	-----------------------	----------------------------------	-----------------------------------	-----------------

Figura 4.4: configuración del arreglo “recep”.

El desarrollo del algoritmo para el encendido del vehículo, se basa en leer el claxon primordialmente, ya que como se mencionó se debe de apretar 3 veces el claxon para poder arrancar el vehículo. Para saber el estado de los botones del volante, se lee la variable almacenada en el arreglo “recep[2]” y se hace una comparación para saber qué botón se está oprimiendo.

A continuación se procede a explicar el algoritmo mostrado en la figura 4.5

1. El algoritmo comienza asignando una variable “contador (cont)” que es la que se utiliza para contar el número de veces que se ha oprimido el botón del claxon.
2. Se verifica si los valores de RECEP[0] y RECEP[1] son iguales a 204 y 170 respectivamente. Estos valores corresponden a sus representaciones en sistema hexadecimal y son los ID_SLAVE = CC = 204 e ID_MASTER = AA = 170.
Si la respuesta es afirmativa entonces se procede al siguiente nivel
Si la respuesta es negativa, el programa no realiza tarea alguna y vuelve a preguntar lo mismo.
3. Si la variable contador es igual o mayor que 3, es decir si el claxon se ha oprimido en más de 3 ocasiones, entonces se manda a la rutina para accionar las salidas digitales y cerrar los contactores para arrancar los motores del vehículo. Una vez que se han accionado los contactores se mantienen encendidos y continúa con el siguiente nivel en el algoritmo.
Si la variable *cont* es menor que 3, el programa no ejecuta ninguna acción (Inclusive el claxon no sonará) y se continúa con el siguiente nivel.
Si el valor que está almacenado en RECEP[2] es igual al byte 64, 64 en su representación binaria 01000000, que hace referencia a que el botón del claxon está oprimido en el volante. El SEC activa la salida digital para encender el claxon y además se incrementa la variable *cont* en 1. Al finalizar avanza al siguiente nivel en el programa.

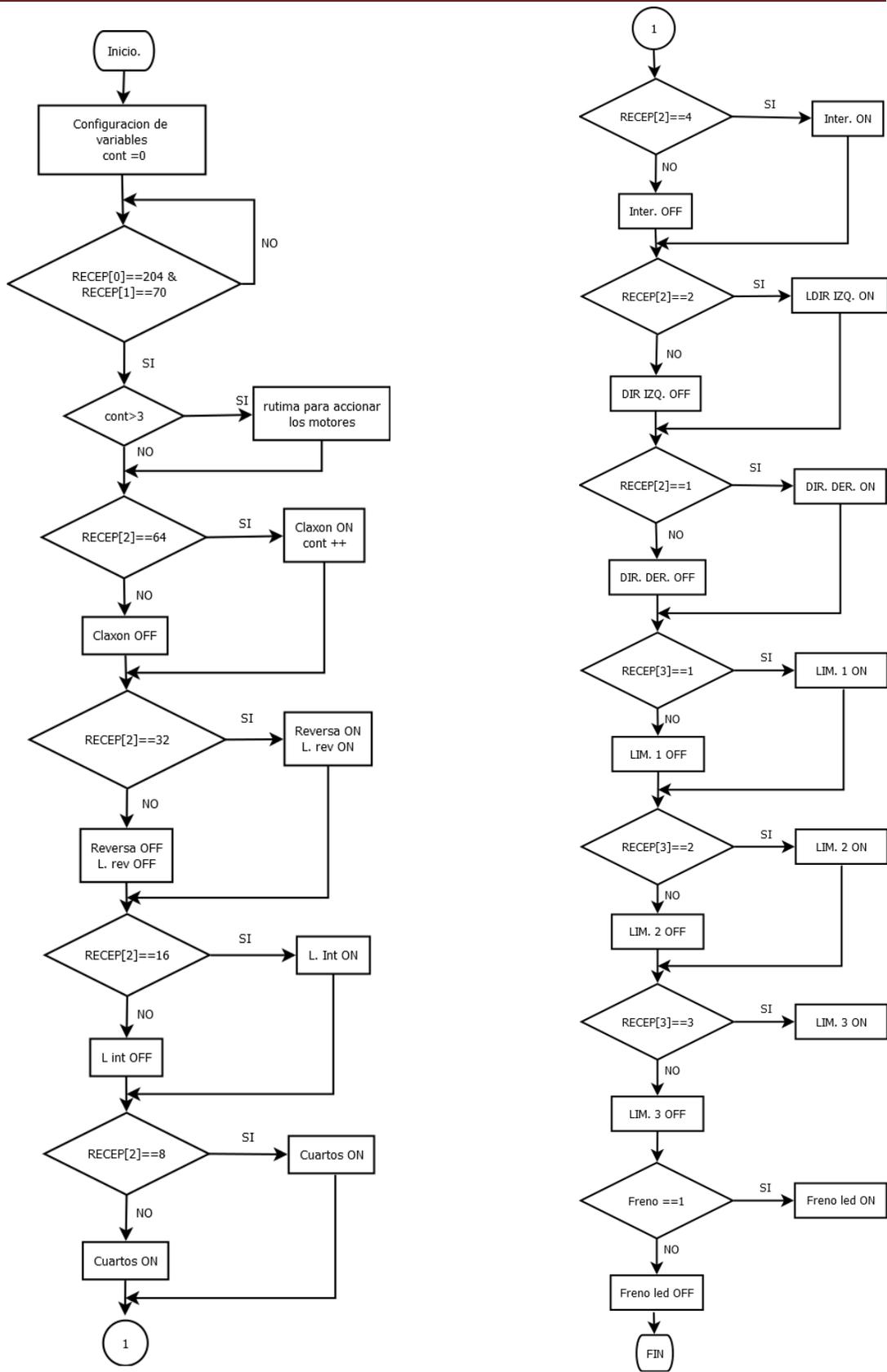


Figura 4.5: Algoritmo para el encendido del vehículo.

Si es distinto que el byte 64, se mantiene apagada la salida digital del claxon y se avanza al siguiente nivel en el programa.

4. Así se irán recorriendo cada uno de los niveles preguntando por el estado de la variable RECEP[2] y comparando con diferentes valores, como se muestra en la tabla 4.3.

Si la respuesta es afirmativa, el SEC activará la salida digital correspondiente a la acción solicitada y avanzará al siguiente nivel.

En caso de ser negativa, mantiene apagada la salida digital y avanza al siguiente nivel en el programa.

Valor a comparar	Representación binaria	Señal a activar.
32	00100000	Reversa y luces traseras.
16	00010000	Luces internas.
8	00001000	Luces externas y cuartos.
4	00000100	Luces intermitentes.
2	00000010	Direccional izquierda.
1	00000001	Direccional derecha.

Tabla 4-3: Relación de valores a comparar y salidas a activar.

5. Posteriormente se procede a comparar la variable almacenada recep[3] para conocer qué salidas se deben activar en este caso del limpia parabrisas.
6. Por último se pregunta sobre el estado del freno, si está presionado el pedal del freno se manda la señal al SEC para que éste la envíe al sistema de luces traseras y activar las luces correspondientes.

En caso de ser negativa la respuesta, el SEC mantiene la orden de mantener apagadas las luces y regresa al comienzo del programa para continuar.

4.3.3 CICLO DE TRABAJO PWM. (MOTORES)

Una vez que se ha arrancado el vehículo, se necesita conocer la velocidad que se debe de tener cada motor, para poder responder a las órdenes dadas por el usuario, tanto de dirección (volante) y de velocidad (acelerador).

Como se mencionó el vehículo cuenta con una configuración diferencial, esto es que la dirección del vehículo depende directamente de la diferencia de velocidades existente entre sus llantas traseras. Para poder adaptarse a la orientación requerida, el SEC incrementa (motor 1) y decrementa (motor 2) mutuamente la velocidad de los motores para dirigir el vehículo hacia un sentido (por ejemplo vuelta a la izquierda) y decrementa (motor 1) e incrementa (motor 2) para dirigirse en sentido contrario (por ejemplo vuelta a la derecha).

EL valor de la velocidad deseada en los motores, se le proporciona en forma de ciclo de trabajo de una señal PWM, que al acoplarse a la salida analógica del SEC, se convierte en una señal analógica para así poder indicarle al controlador del motor, la velocidad deseada.

ALGORITMO.

Este algoritmo se desarrolla para calcular el ciclo de trabajo PWM que se debe de tener en cada motor, para poder dar la dirección y velocidad requeridas por el usuario. A continuación se explica el algoritmo mostrado en la figura 4.6

1. Configuración de las variables a utilizar, básicamente son las variables donde se guardará el valor leído de los sensores (acelerador y dirección del volante). Así como las variables donde se guardará el valor de PWM para cada motor.
También se tendrán variables para hacer el cálculo del ciclo de trabajo PWM.
2. Se lee el valor de los sensores del acelerador y de la dirección del volante, ya que estos datos son indispensables para poder calcular el ciclo de trabajo de los motores.
Estos se mapearon en un nuevo rango de 0 a 99 para mayor practicidad en la realización de los cálculos. Después de ser guardados los valores de los sensores, se procede al siguiente nivel.

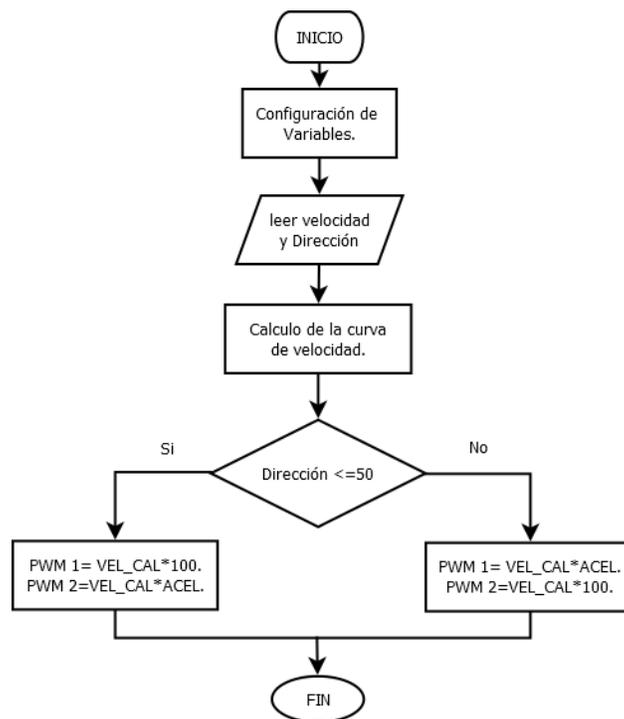


Figura 4.6: Algoritmo para el cálculo de valores PWM.

3. Posteriormente se realiza el cálculo del ciclo de trabajo. Originalmente se tenía una propuesta de un par de rectas intersectantes en dirección = 50 (figura 4.7), donde el comportamiento es algo similar a lo esperado, excepto en el punto donde la dirección es igual a 50 (vehículo avanza en línea recta)
En este punto el vehículo tendría ciclo de trabajo 255 en ambos motores, si el acelerador estuviera a fondo. Conforme se mueve la dirección hacia la izquierda o a la derecha, un motor debe aumentar el ciclo de trabajo y el otro lo debe de disminuir, lo cual no puede cumplir este control.

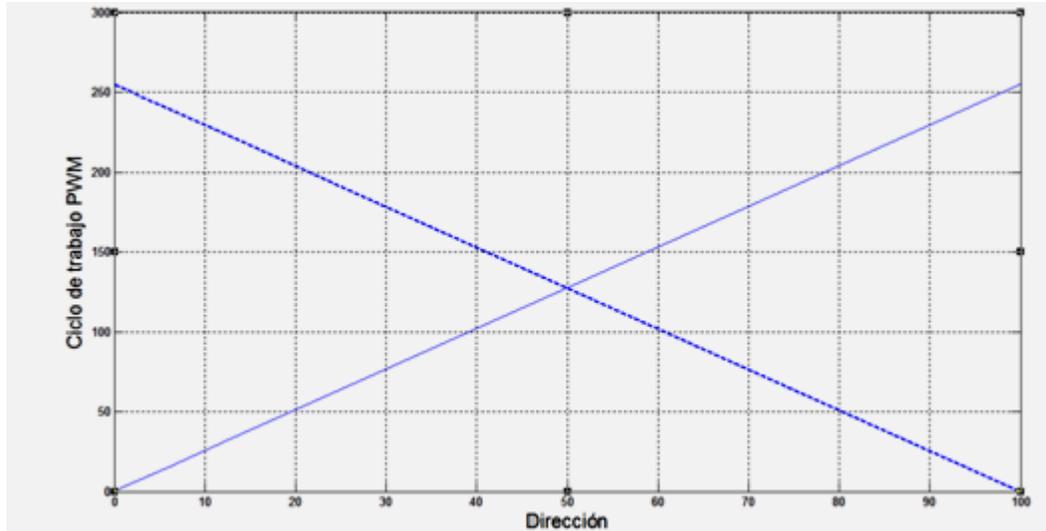


Figura 4.7: Propuesta de señales PWM.

Es por eso que se descartó esta idea y se buscó una función que cumpliera las necesidades del vehículo en todos los puntos, tanto de la dirección y del acelerador. La ecuación obtenida es:

$$Direccion2 = 100 * \left(-\frac{(Direccion - 50)^2}{2500} + 1 \right)$$

Y se propone la función cuadrática graficada en la figura 4.8. Donde se observa el comportamiento del ciclo de trabajo de los motores, cabe mencionar que este comportamiento sólo es válido para el motor 1 en un rango en la dirección de [0,50] y para el motor 2 de [50,100]. Más adelante en la generación de PWM se explica qué sucede con el resto del rango para los motores.

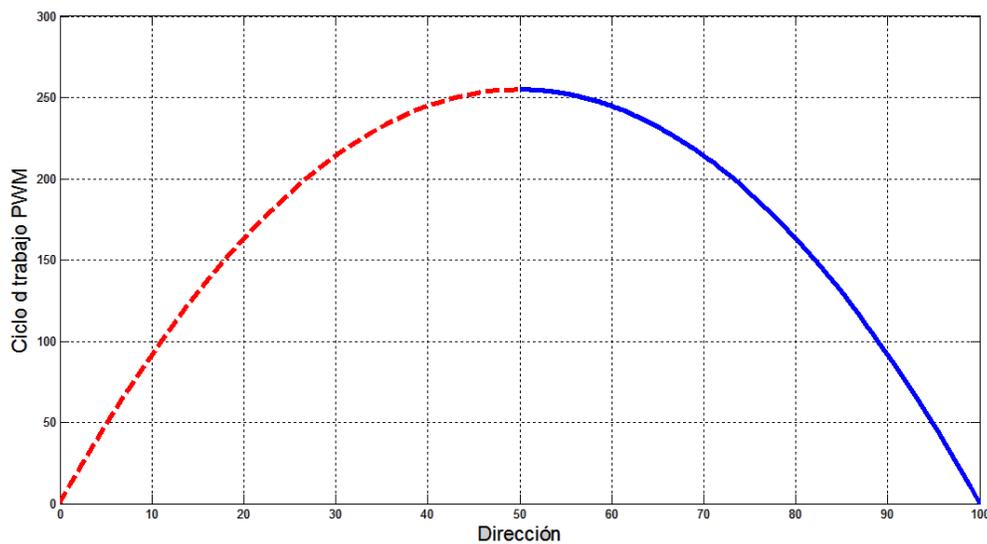


Figura 4.8: Gráfica de la función propuesta para la generación de las señales PWM.

4. Posteriormente se pregunta si la dirección del vehículo es menor o igual que 50. De ser afirmativa, quiere decir que el vehículo debe dar vuelta a la izquierda. De ser negativa la respuesta, el vehículo debe dar vuelta a la derecha.

Ahora que se tiene calculada la nueva dirección por medio de la función cuadrática, se procede a normalizarla, así las ecuaciones para el cálculo del ciclo de trabajo de cada motor quedan de la siguiente manera:

Cuando se quiere dar vuelta a la izquierda, se tienen las siguientes funciones:

$$PWM1 = direccion2 * Acelerador$$

$$PWM2 = Acelerador * 100$$

Y cuando se quiere dar vuelta a la derecha se tienen las siguientes funciones:

$$PWM1 = Acelerador * 100$$

$$PWM2 = direccion2 * Acelerador$$

De este modo la curva representativa de ciclo de trabajo del PWM depende directamente del acelerador y de la dirección del volante, como se puede observar en la figura 4.9, las distintas funciones dependiendo del valor del acelerador. Ahora el comportamiento de ambos motores se puede apreciar en la figura 4.10 con líneas punteadas las curvas del motor 1 con incrementos de 10 en el acelerador, y con líneas continuas las curvas del motor 2 con los mismos incrementos.

Una vez calculado el valor de PWM para cada motor, el SEC envía los valores correspondientes hacia las salidas analógicas para ser recibido por el controlador de los motores y accionar los motores.

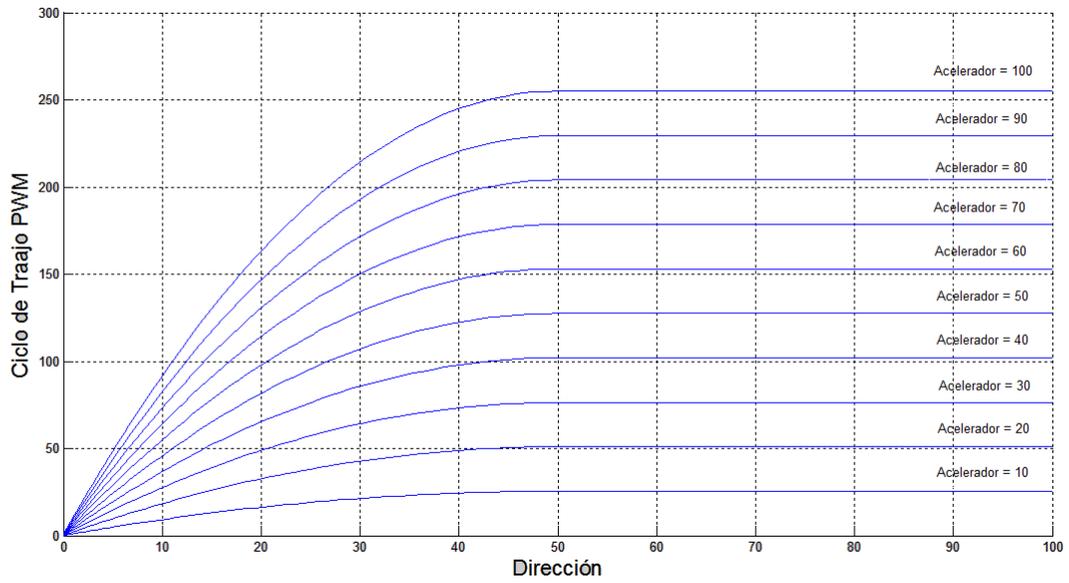


Figura 4.9: Familia de curvas para el motor derecho.

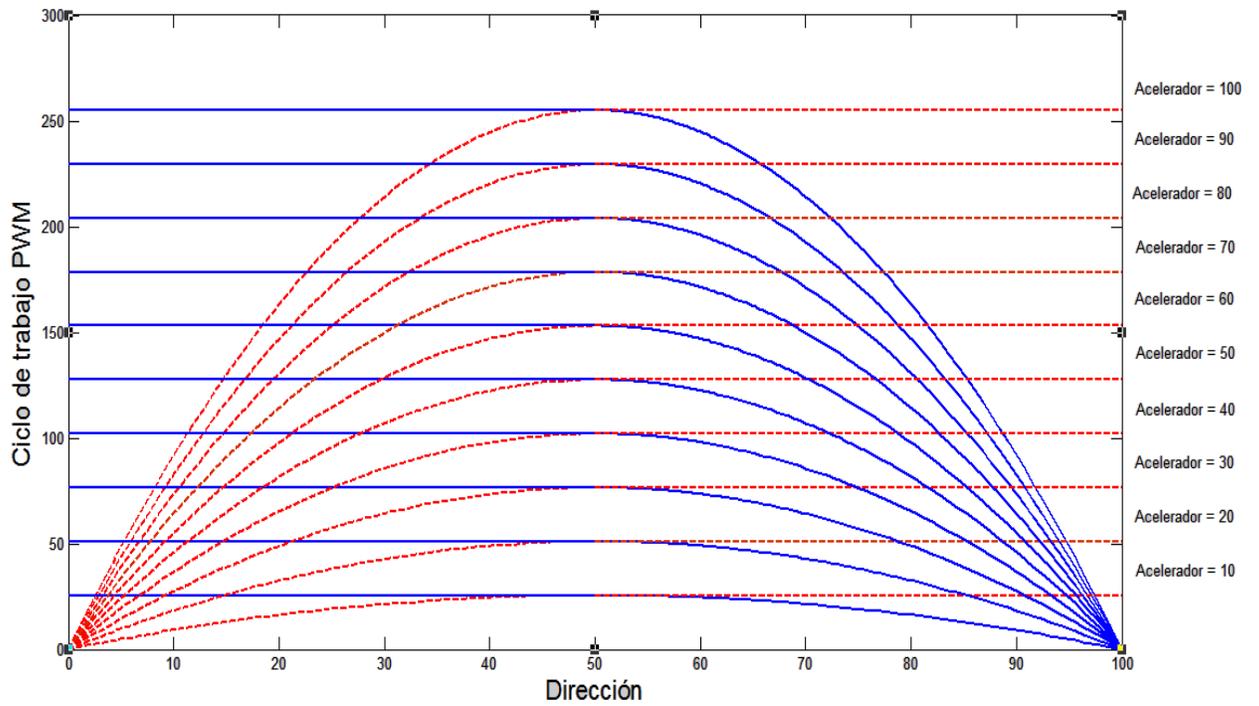


Figura 4.10: Familia de curvas para ambos motores, a diferentes valores del acelerador..

4.3.4 DETERMINACIÓN DE LA VELOCIDAD LINEAL EN CADA LLANTA.

Calcular la velocidad de las ruedas es necesario para poder compensar los errores que puedan existir en diferencia de velocidades, entre la ideal (la que se manda desde el acelerador y contando la dirección del vehículo) y la real (velocidad en las llantas). Esto principalmente debido a que los motores no responden de la misma manera (no cuentan con características idénticas) aunque sean del mismo modelo y marca.

Para poder estimar la velocidad se utilizó la ecuación 4.1, a partir de conocer las revoluciones por minuto se puede conocer la velocidad de la rueda, ya que se conoce el diámetro de la llanta.

$$v \left[\frac{km}{h} \right] = rpms \left[\frac{rev}{min} \right] * \pi * d[m] * \frac{60 min}{1 h} * \frac{1km}{1000 m} \quad (4.1)$$

La velocidad máxima que puede alcanzar el vehículo es de 60 km/h, a continuación se muestran en la tabla 4.4 valores de revoluciones que debe tener la llanta para poder alcanzar ciertas velocidades en terreno plano.

Velocidad (km/h)	RPM
0	0
1	9.30728485
5	46.5364242
10	93.0728485
20	186.145697
30	279.218545
40	372.291394
50	465.364242
60	558.437091

Tabla 4.4. Relación de velocidad del vehículo y RPM de las llantas.

Debido a que el microcontrolador debe actualizar la velocidad del vehículo hacia el display del volante, se muestrea la información del sensor cada 200 [ms] por lo cual se obtiene la relación de revoluciones cada 200 [ms]. Quedando los siguientes valores.

Velocidad (km/h)	RPM	Revoluciones cada 200 [ms]
0	0	0
1	9,30728485	0,03102428
5	46,5364242	0,15512141
10	93,0728485	0,31024283
20	186,145697	0,62048566
30	279,218545	0,93072848
40	372,291394	1,24097131
50	465,364242	1,55121414
60	558,437091	1,86145697

Tabla 4.5: Relación de velocidad del vehículo y revoluciones en las llantas.

Como se puede observar en la tabla 4.5 el SEC deberá de contar 1.86 pulsos cada 200 [ms], cuando el vehículo vaya a una velocidad de 60 km/h. y tendrá que contar 0.03 pulsos cuando el vehículo vaya a 1 km/h. Debido a que el microcontrolador no puede medir fracciones de pulsos es por esto que se colocaron 10 imanes a través de la circunferencia de la polea para poder ampliar el rango de lectura de pulsos del SEC.

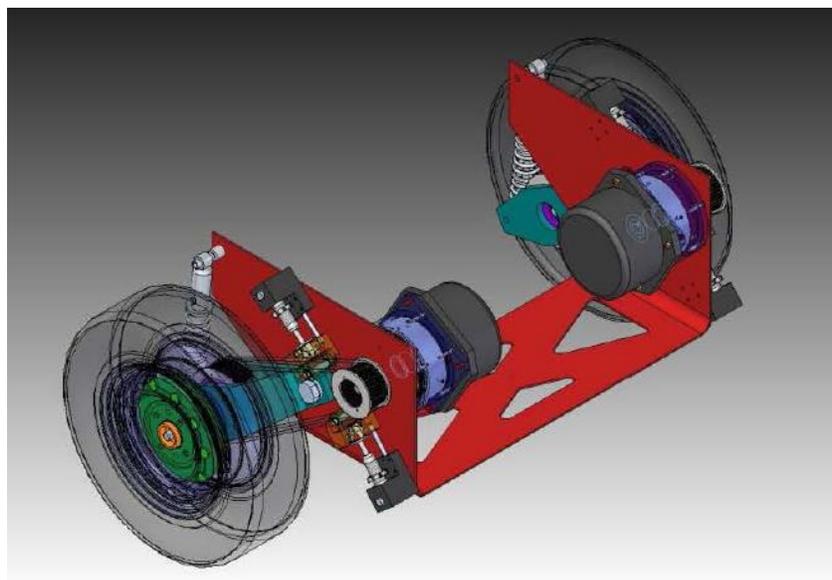


Figura 4.11: Configuración de la transmisión del VET.

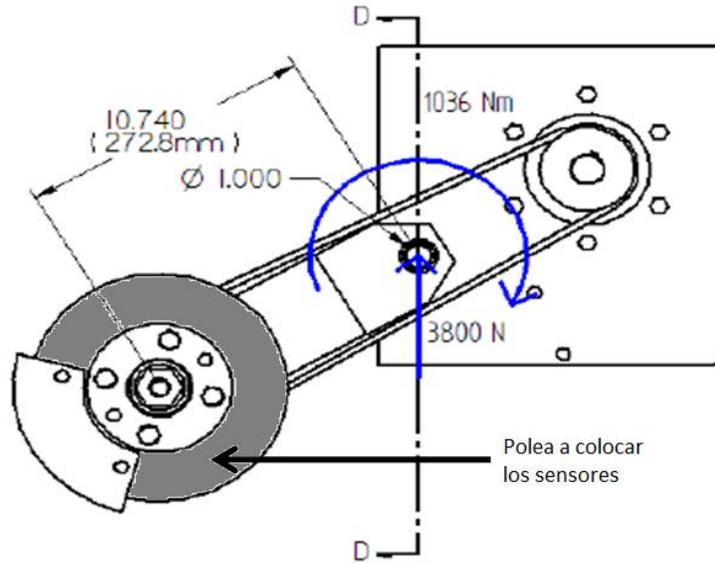


Figura 4.12: Polea a colocar los imanes.

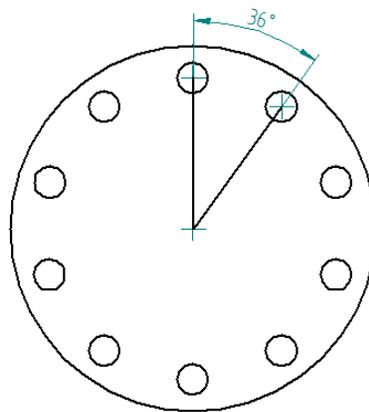


Figura 4.13: Distribución de los imanes.

Con esta configuración, se obtienen los siguientes valores de pulsos, que tendrá que procesar el microcontrolador. Como se puede apreciar en la tabla 4.6 colocando 10 imanes a lo largo de la circunferencia de la polea, se obtiene un rango de pulsos más grande para poder ser procesados por el microcontrolador. Con esto, cada que el SEC cuente 10 pulsos sabrá que la rueda ha dado una vuelta y se podrá calcular la velocidad del vehículo.

Velocidad (km/h)	Pulsos cada 200 [ms] con 10 imanes
0	0
1	0,31024283
5	1,55121414
10	3,10242828
20	6,20485657
30	9,30728485
40	12,4097131
50	15,5121414
60	18,6145697

Tabla 4.6: Relación de velocidad del vehículo y revoluciones en las llantas con 10 imanes.

ALGORITMO

Para el desarrollo de este algoritmo, se tienen 3 algoritmos. El algoritmo del *Timer* (200 [ms]), el algoritmo contador de pulsos y el algoritmo que suma y promedia los valores de pulsos.

Inicialmente el programa principal configura el tiempo del *Timer* y después lo inicializa, para así empezar a ejecutar los demás programas dependientes del *Timer*.

1. Se configuran las variables a utilizar, que en este caso es el tiempo predefinido de 200 [ms], tiempo de muestreo.
2. Posteriormente se pregunta si el tiempo es igual a 200ms. Si la respuesta es afirmativa se activa una bandera, que es la señal para el algoritmo del contador de pulsos. En caso de ser negativa la respuesta, el programa sigue incrementando en tiempo.

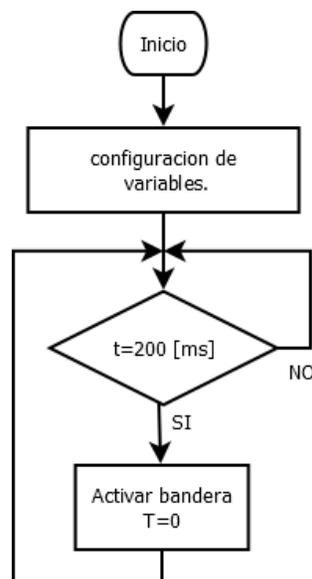


Figura 4.14: Algoritmo para el *Timer*.

El algoritmo para contar los pulsos, depende de la señal del *Timer* para contar el número de pulsos provenientes de los sensores colocados en las llantas.

1. Configurar las variables, crear una variable que lleve el registro de pulsos contados.
2. Se pregunta si el *Timer* ha mandado la señal de activar la bandera. en caso de ser afirmativa la respuesta, se envía el valor de pulsos contados al algoritmo de “captura, suma y promedio”, se reinicia la variable de conteo y se vuelve a preguntar sobre la bandera del *Timer*. En caso de ser negativa la respuesta el algoritmo incrementa en uno el número de pulsos y vuelve a preguntar.

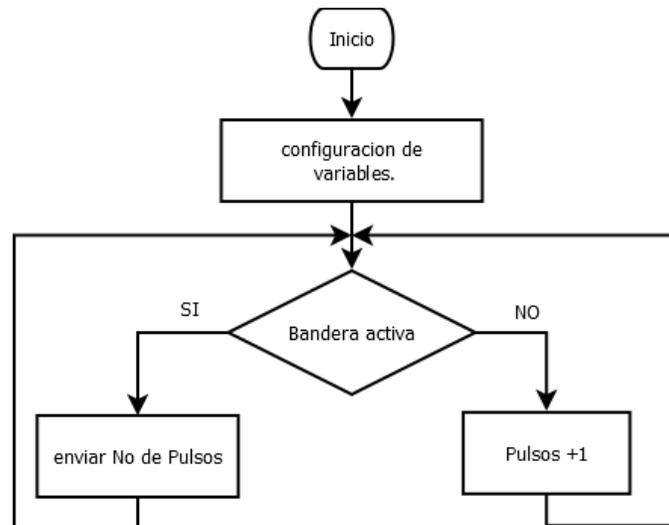


Figura 4.15: Algoritmo para conteo de pulsos.

El algoritmo de captura, suma y promedio, es el que dará el resultado final del número de pulsos contados, Almacena en un arreglo de tamaño 5, 5 valores recibidos por el algoritmo de pulsos para ser sumados y promediados.

1. Se configuran las variables, en este caso los arreglos `dato1[]` y `dato2[]` que son donde se almacenan los valores de pulsos contados. Además se tienen otras 2 variables, `suma1` y `suma2` que son las que van llevando la suma de los valores guardados en `dato1[]` y `dato2[]`. Dos variables adicionales “*i*” y “*N*”, sirven como contadores.
2. Primero se pregunta si `i=5` es decir si se han guardado ya 5 valores en el arreglo `dato1[]` y `dato2[]`. Si la respuesta es afirmativa se avanza al siguiente nivel. Si la respuesta es negativa, se manda señal para indicar que las sumas de los datos son cero y se avanza al siguiente nivel.
3. En el caso de la respuesta afirmativa, se pasa a un ciclo FOR donde se suman los 5 valores guardados. Al momento de sumar los 5 valores, se manda entonces el promedio de los valores. Y se indica que la variable `i=0` para que se vuelva a almacenar 5 datos en `dato1[]` y `dato2[]`.
4. En caso de ser negativa la respuesta y después de inicializar las variables de suma a cero, se guarda el valor obtenido de pulsos proveniente del algoritmo “pulsos” en el espacio del

arreglo correspondiente. Una vez que se guardó, se reinicia la variable pulsos para que el algoritmo “pulsos” empiece a contar el número de pulsos recibidos desde cero y no los acumule.

5. Estimar la velocidad media del vehículo.

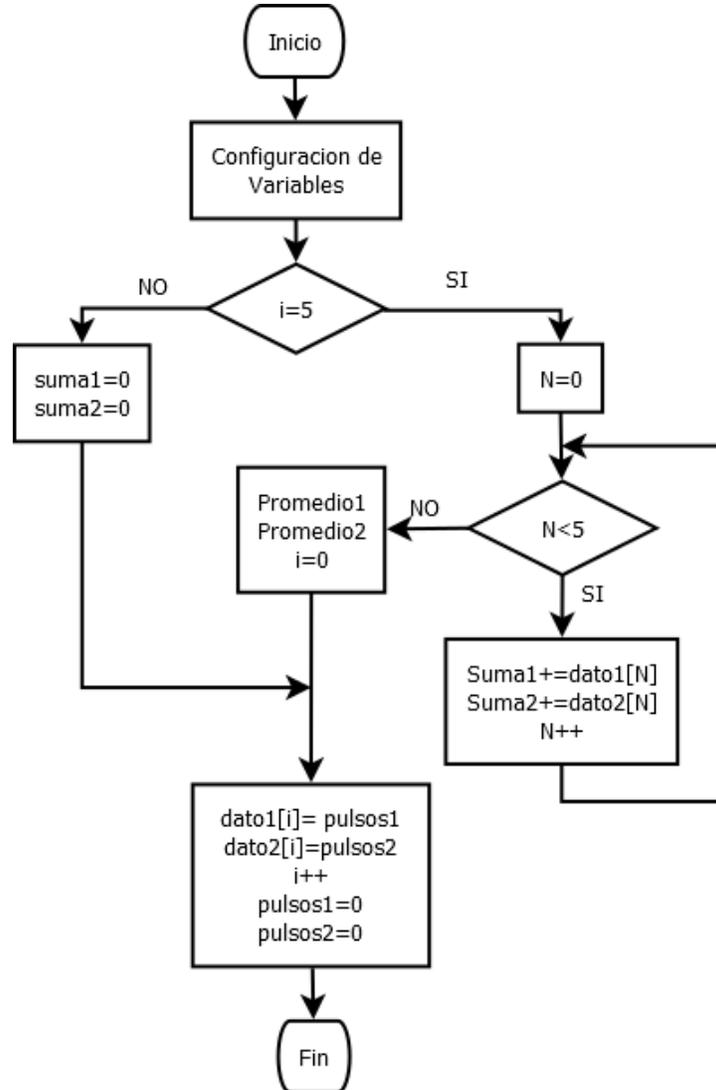


Figura 4.16: Algoritmo para obtener el promedio de pulsos.

4.3.5 MANEJO DEL ERROR.

Como parte de la regulación de velocidades de las llantas del vehículo, se debe tomar en cuenta que aunque los motores son de las mismas características, no se comportan de manera exactamente igual. Es decir, no responden de manera exactamente igual a excitaciones iguales.

Esto es que se genera un error de velocidades. Para el cálculo del error se tiene la siguiente ecuación.

$$error = velocidad_{estimada} - velocidad_{real}$$

La $velocidad_{real}$ la conocemos por medio de los sensores colocados en las llantas del vehículo, y la $velocidad_{estimada}$ se puede determinar a partir del acelerador del vehículo y las ecuaciones de generación de ciclo de trabajo de PWM.

Teniendo el error, solamente se retroalimenta al sistema para poder generar un nuevo valor de ciclo de PWM y enviarlo a las salidas analógicas del SEC.

Algoritmo.

Como ya se había comentado el algoritmo del error está ligado al algoritmo para calcular la velocidad media del vehículo. Además se encuentra relacionado con el algoritmo para generar el ciclo de trabajo de los motores. Así el algoritmo del error hará una corrección en el ciclo de trabajo del PWM en caso de que exista error. A continuación se explica el algoritmo de la figura 4.17.

1. Se hace el cálculo tanto de la velocidad real como de la velocidad ideal, para así poder obtener el error.

La velocidad real ya se conoce del algoritmo que calcula la velocidad en las llantas del vehículo, pero esta velocidad se muestra en revoluciones por minuto, y para poder corregir el ciclo de trabajo PWM se deben de manejar las mismas "unidades". Es por eso que se puede asociar la velocidad que lleva el vehículo con un ciclo de trabajo PWM, es decir cuando el vehículo vaya a su máxima velocidad (60 km/h) se puede establecer que el ciclo de trabajo en los motores es del 100%

Así para poder saber qué porcentaje del ciclo de trabajo le corresponde a distintas velocidades se establece la siguiente ecuación.

$$\%CT = \frac{Promedio * 100\%}{18.6}$$

De igual manera para poder establecer la posición del acelerador asociado a un porcentaje de ciclo de trabajo se tiene la siguiente ecuación.

$$\%CT = \frac{Voltaje * 100\%}{5}$$

Dado que el voltaje máximo que puede proporcionar el acelerador es 5 volts cuando se encuentra totalmente presionado, en ese momento el porcentaje de ciclo de trabajo es de 100%.

2. En este punto se tienen las velocidades asociadas a un porcentaje en ciclo de trabajo del PWM, y se puede hacer el cálculo del error, que también estará en función de un porcentaje del ciclo de trabajo del PWM.

Si el error es positivo quiere decir que el vehículo va más despacio de lo que se le está ordenando con el pedal acelerador y hay que acelerar más los motores.

Si el error es negativo quiere decir que el vehículo va más rápido de lo que se le está ordenando con el pedal acelerador y hay que disminuir la velocidad.

3. Ahora se hace la corrección del ciclo de trabajo mandado por el algoritmo “ciclo de trabajo PWM”. Se le suma el error a los ciclos de trabajo obtenidos, para hacer la corrección, es decir aumentar o disminuir la velocidad dependiendo de cómo sea el error obtenido.
4. Dado que el algoritmo “ciclo de trabajo PWM” entrega un valor entero entre 0 y 100% y además sumado el error puede llegar a sobrepasar el 100% o ser menor que 0%, hay que delimitar el ciclo de trabajo para que el ciclo de trabajo siempre esté dentro del rango y evitar problemas con la programación.
5. Por último se envía el nuevo valor de ciclo de trabajo hacia las salidas analógicas para corregir la velocidad del vehículo.

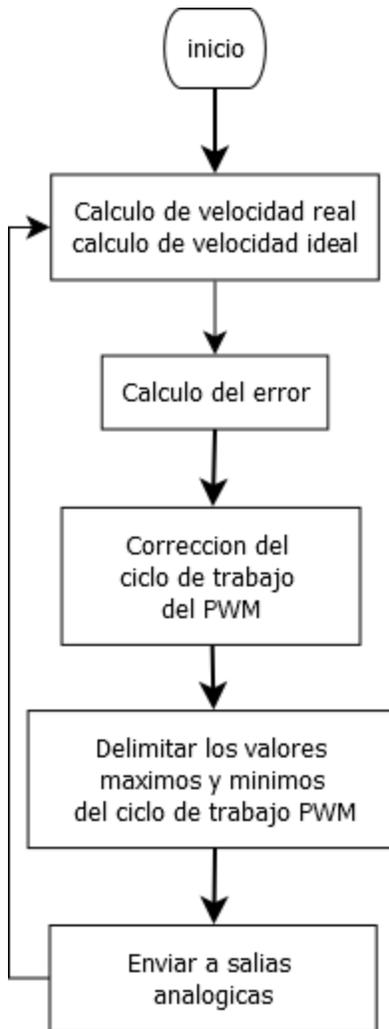


Figura 4.17: Algoritmo para el error.

Diseño del sistema electrónico de control SEC.

Introducción

El sistema eléctrico de control (SEC) se basa en el uso de una tarjeta de desarrollo Arduino Mega, la cual se seleccionó por cumplir con el número de entradas y salidas que se necesitan para controlar los motores, sensores y demás periféricos del vehículo.

Dentro de las funciones del SEC se encuentran:

- Procesar la señal de voltaje correspondiente a la posición del volante.
- Interpretar el valor de la posición del acelerador, freno, y nivel de carga de las baterías.
- Generar las señales de control hacia el sistema de tracción (*Motores Brushless*).
- Detectar y procesar las señales de la velocidad de las ruedas izquierda y derecha.
- Mandar información hacia el volante (velocidad, daxon, luces, etc.) a través del protocolo RS-485.
- Control de periféricos. (limpia parabrisas, luces frontales, claxon, etc.)



Figura 5.1: Diagrama de entradas y salidas.

La tarjeta se diseñó tomando en cuenta las diferentes salidas y entradas. Dependiendo el tipo de señal que vaya a entrar o a salir del SEC será el tipo de acondicionamiento que se coloque en el diseño de la tarjeta impresa PCB. En la siguiente tabla se describen los diferentes tipos de señales.

Alias	Entrada/Salida	Tipo
Claxon	Salida	Digital
Posición del volante	Entrada	Analógica
Dirección motores	Salida	Digital
Luces frontales	Salida	Digital
Comunicación con el volante	Bidireccional	RS-485
Freno	Entrada	Analógica
Velocidad de los motores	Salida	Analógica
Velocidad de las ruedas	Entrada	Digital

Tabla 5.1: Entradas y salidas.

5.1 Microcontrolador [16].

El Arduino mega es una placa basada en un microcontrolador Atmega 2560, contiene 54 entradas/salidas digitales. De las cuales 14 también pueden ser usadas como salidas PWM, 16 entradas analógicas, 4 UARTS (Universal Asynchronous Receiver-Transmitter), un cristal oscilador de 16 MHz, conexión USB.



Figura 5.2: Arduino mega

Dentro de las características más importantes de Arduino mega se encuentran:

Microcontroladores	ATmega1280
Voltaje de funcionamiento	5V
Voltaje de entrada	5-12V
Terminales E/S digitales	54 (14 proporcionan salida PWM)
Terminales de entrada analógica	16
Corriente por pin	40 mA
Corriente en pin 3.3V	50 mA
Memoria Flash	128 KB de las cuales 4 KB las usa el gestor de arranque(bootloader)
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz

Tabla 5.2: Características de la tarjeta Arduino mega.

5.2 Acondicionamiento de señales.

Difícilmente en el diseño de un circuito eléctrico, en el cual se emplee un sensor o transductor, se conectará éste directamente ya que la señal que envía el sensor es muy débil o contiene ruido y componentes que no deseamos, por eso se realizan etapas de acondicionamiento de señales. Tales como pueden ser amplificación, filtrado, acoplamiento de impedancias y modulación o demodulación. Recordado que el circuito de acondicionamiento puede proporcionar una impedancia de entrada adecuada para no demandar mucha corriente del sensor.



Figura 5.3: Manejo de señales eléctricas.

Los microcontroladores necesitan entradas digitales, aunque la mayoría ya incluye un convertidor analógico digital, por ello, cuando un sensor produce una salida analógica es conveniente convertirla a una señal digital. Sin embargo, muchos sensores sólo producen señales muy pequeñas, a veces de unos cuantos milivolts. Este tipo de señales es insuficiente para que la detecten los puertos del microcontroladores, para ello habrá que amplificarla.

También se deben de considerar las salidas de los microcontroladores, quizás se requiera algún actuador. Aquí también es necesario acondicionar la señal, ya sea amplificar la señal en voltaje y corriente, o incluso como protección para evitar que se regresen las señales por los puertos de salida.

Es por esto que se diseñó en el SEC un acondicionamiento para cada tipo de señal, y así garantizar tener una lectura adecuada de las señales y proporcionar una señal clara (sin ruido) y apta para ser detectada por la tarjeta Arduino Mega. A continuación se describen los tipos de acondicionamiento que se diseñaron para los sensores utilizados en el SEC.

5. 2.1 Alimentación.

El sistema eléctrico de control está alimentado por una batería de 12 volts, dado que si alimentamos a la tarjeta Arduino con 12 volts estaremos en su límite superior de voltaje de alimentación, se utilizó un regulador 7805 para suministrar voltaje al Arduino con 5 volts.

Para la alimentación de voltaje (8 volts) del volante se utilizó un regulador 7808, además se tiene una salida de 12 V para alimentar desde el sistema eléctrico de control (SEC) el sistema de luces traseras.

A la entrada de voltaje de cada regulador se tiene un capacitor de 1uF para amortiguar los picos de voltaje que pudiese haber, de igual forma se tienen capacitores a la salida de los reguladores.

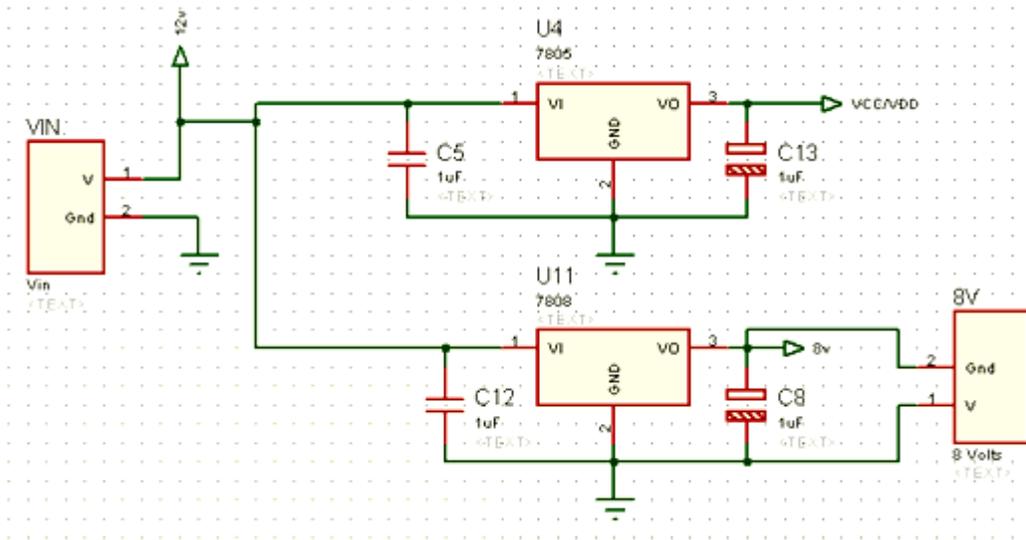


Figura 5.4: Circuito para alimentación de SEC.

5.2.2. Entradas digitales

Para poder acondicionar las señales digitales, se diseñó un circuito que garantizará que a la entrada del microcontrolador le llegarán los valores Transistor-Transistor Logic (TTL) dentro del límite de Arduino.

Se utilizó un circuito integrado 7414, que son disparadores invertidos Schmitt-Trigger. Estos circuitos se utilizaron para acondicionar las entradas por interrupciones, que serán proporcionadas por los sensores de efecto Hall, para garantizar que a la entrada de la tarjeta de desarrollo Arduino se tengan señales de onda cuadrada y así evitar el ruido en la señal. Por otro lado se mantienen los voltajes TTL dentro del rango de 2.4 – 5 volts para una señal en estado alto y de 0.2-0.8 para una señal en estado bajo.

Se utilizaron dos de estos circuitos, para acoplar las señales a la entrada de la tarjeta Arduino Mega y mantener entradas con lógica verdadera.

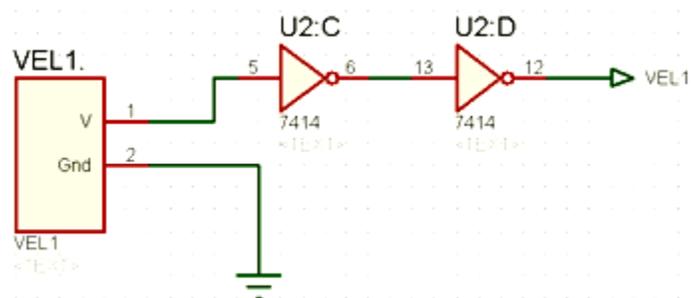


Figura 5.5: Acondicionamiento para entrada digital.

Esta configuración se utilizó para las entradas de las velocidades de la llantas, para poder estimar la velocidad del vehículo y para el uso del freno.

4.2.3 Entradas analógicas

En las entradas analógicas se utilizaron amplificadores operacionales LM324 en una configuración de seguidor de voltaje o amplificador de ganancia unitaria.

El voltaje de entrada se aplica directamente a la entrada (+) del amplificador operacional. Ya que el voltaje entre las terminales (+) y (-) del amplificador puede considerarse 0, esto origina que el voltaje de salida sea igual al voltaje de entrada.

Debido a que la impedancia de entrada del seguidor de voltaje es muy alta, hace que se demande poca corriente de los sensores.

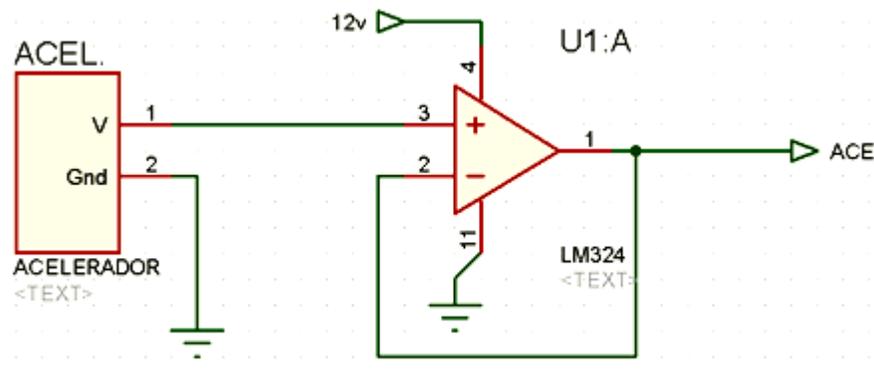


Figura 5.7: Acondicionamiento para entrada analógica.

La alimentación de amplificador operacional se tiene a 12 volts, aunque el máximo voltaje de entrada a la tarjeta Arduino es de 5 volts dados por el acelerador, freno y la dirección del volante.

5.2.4 Configuración para salidas digitales

La salida digital comprende básicamente 2 etapas: eliminación de ruido y saturación del transistor y activación del relevador.

En la primera etapa cuando el microcontrolador tiene una salida “alta” o un uno lógico, señal pasa a través de un par de disparadores invertidos *Schmitt-Trigger* que eliminan el ruido que pueda tener la señal. (Figura 5.8). Es decir, el disparador invertido *Schmitt-Trigger* hace que sus valores de conmutación varíen dependiendo del último valor leído, para hacer aumentar, así, la estabilidad del circuito

En un circuito electrónico digital, las entradas de las compuertas tienen unos valores límite (en los circuitos con tecnología TTL de 0 a 0.8 [V] será un '0' lógico y de 2 [V] a 5 [V] será un '1'). Si la señal llega a una entrada con 2.1 [V] esa lo considera un '1' lógico. Pero si esa señal baja un poco hasta los 1.9 [V] sería un '0' lógico.

El disparador invertido *Schmitt-Trigger* lo que hace es, si le llega un señal de 2.1 [V], lo reconoce como un '0' y pone el nivel del '1' a 3 [V]. Ahora, aunque la señal suba a 2.5 [V] no lo reconocerá como un '1'. Si le llega una señal menor de 2V, lo reconocerá como un '0'.

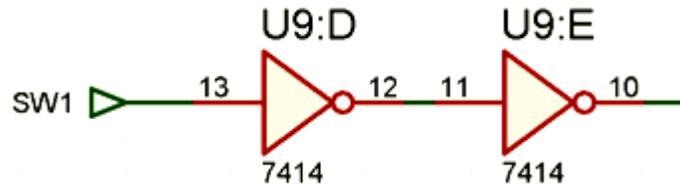


Figura 5.8: Acoplamiento de la señal digital

La segunda etapa comprende la saturación del transistor NPN BC 547, para establecer la corriente necesaria para activar el relevador en el circuito del colector. Sin entrada en la base del transistor, tanto la corriente de base. La corriente de colector y la corriente de la bobina son prácticamente de 0 [A], y el relevador permanecerá en estado inactivo. Sin embargo cuando se aplica un pulso positivo en la base, el transistor se enciende, haciendo que circule la corriente por la bobina para activar el relevador.

Cuando deje de llegar la señal a la base del transistor y deje de activar la bobina del relevador, pueden surgir problemas. Idealmente la corriente a través de la bobina y del transistor caerá rápidamente a cero, el relevador se desactivará y permanecerá así hasta la siguiente señal. Sin embargo se sabe que la corriente a través de la bobina no puede cambiar instantáneamente y mientras más rápido cambie, mayor será el voltaje inducido a través de la bobina. En este caso la corriente que cambia rápidamente a través de la bobina, desarrollará un voltaje mayor a través de la bobina, como se muestra en la figura 5.9.

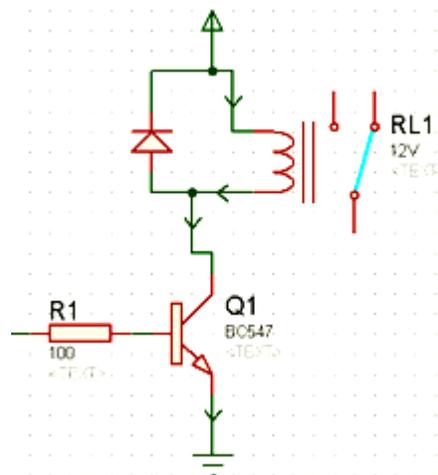


Figura 5.99: Manejo del relevador.

Es probable que su magnitud exceda los límites nominales de voltaje máximos del transistor, y probablemente quede dañado. El voltaje en la bobina no permanecerá en su nivel más alto, si no que oscilará, hasta que su voltaje caiga hasta cero a medida que el sistema se asiente.

Para evitar esto, se colocó un diodo en paralelo con la bobina. Durante el encendido del transistor, el diodo se encuentra en polarización inversa, permaneciendo como un circuito abierto sin afectar. Sin embargo cuando el transistor se apaga, el voltaje en la bobina se invertirá y polarizará de forma directa al diodo, encendiéndolo. La corriente en la bobina establecida durante el encendido del transistor podrá continuar fluyendo a través del diodo, eliminando el cambio acelerado en el nivel de corriente. Y el sistema se asentará, evitando daños al transistor.

5.2.5 Salidas analógicas

Dado que el controlador de los motores *brushless* necesita un voltaje analógico para poder indicarle la velocidad a la que deben girar los motores, se utilizó un circuito RC para generar una salida analógica de voltaje. Esto debido a que la tarjeta de desarrollo Arduino no cuenta con salidas analógicas de voltaje, es por esta razón que se utilizó una salida PWM y un circuito RC.

Primero se pusieron dos disparadores invertidos Schmitt-Trigger (figura 5.10), para eliminar el ruido que pueda provenir de la tarjeta de desarrollo Arduino, y así garantizar que lleguen señales de onda cuadrada al circuito RC.

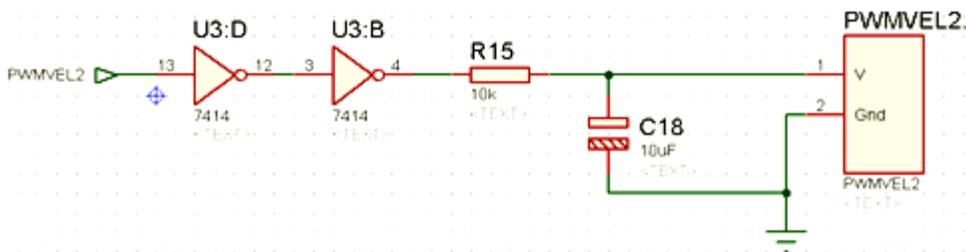


Figura 5.10: Configuración para salidas analógicas.

Dada la señal PWM de entrada al circuito RC (figura 5.11), durante un solo periodo $[0, T]$. El voltaje de entrada al circuito tiene dos partes claramente diferenciadas, una es la parte de la “carga” del capacitor $[0, T_1]$. En este intervalo el capacitor se carga por la fuente de alimentación. La segunda parte es la “descarga” $[T_1, T]$, durante este intervalo el voltaje aplicado es cero, por lo cual el capacitor se descarga a través de la resistencia R de $10k\Omega$.

Durante el intervalo de carga del capacitor, podemos pensar que al circuito RC se le aplica una señal de entrada escalón de V [volts]. Si se asume que el capacitor tienen un voltaje inicial de V_0 al comienzo del intervalo de carga (tiempo $t = 0$), entonces la respuesta del circuito está dada por la ecuación (4.4) para todo $t \in [0, T_1]$.

$$v_c(t) = \left(V_0 e^{-\frac{t}{RC}} + V \left(1 - e^{-\frac{t}{RC}} \right) \right) u(t) \quad (4.4)$$

Durante la descarga, no hay ningún voltaje externo aplicado al circuito RC, esto significa que la respuesta del sistema se debe únicamente al voltaje del capacitor que se presente en el momento T_1 después del intervalo de carga. Como resultado de esto, el voltaje del capacitor en el intervalo de tiempo $[T_1, T]$ es simplemente la respuesta natural del circuito RC y el voltaje del capacitor está dado por la ecuación (4.5) para $t \in [T_1, T]$, donde el voltaje V_1 es la tensión en el condensador cuando $t = T_1$.

$$v_c(t) = V_1 e^{-\frac{t}{RC}} \quad (4.5)$$

La figura 5.11 muestra la señal resultante después del circuito RC, se puede observar como cuando la señal de PWM se mantiene en alto, la curva de voltaje del capacitor se incrementa y decae cuando la señal de PWM es baja

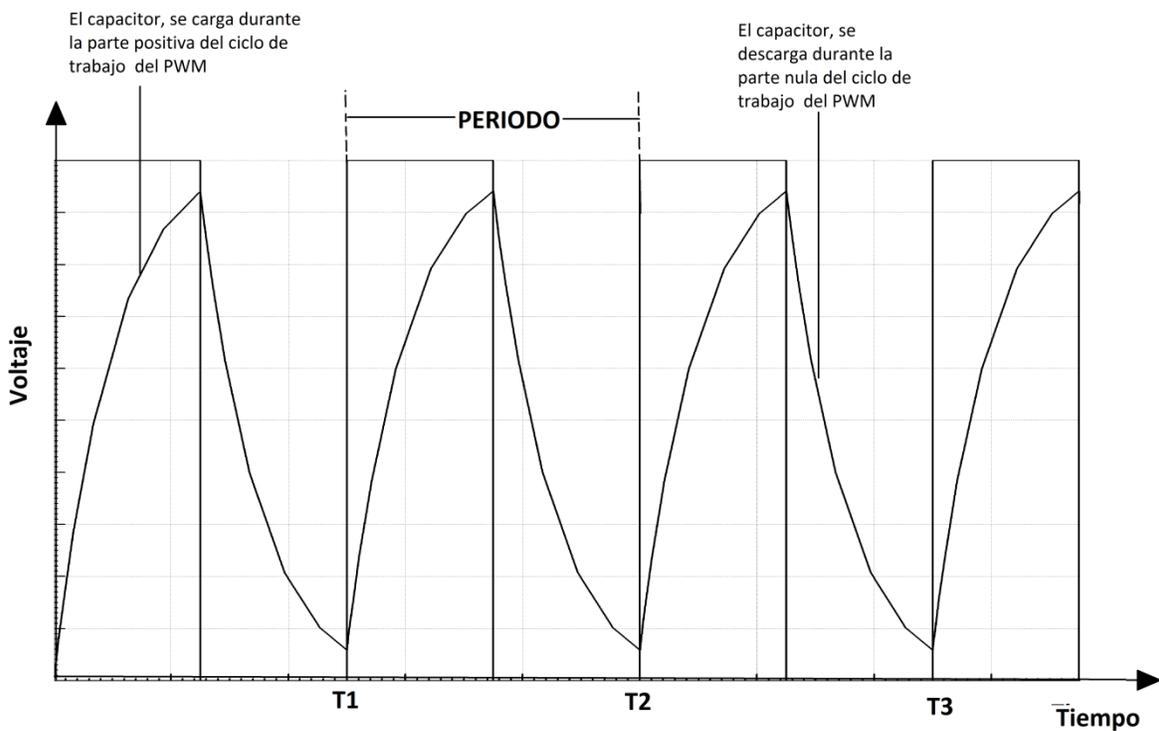


Figura 5.11: Respuesta del circuito RC a la señal PWM.

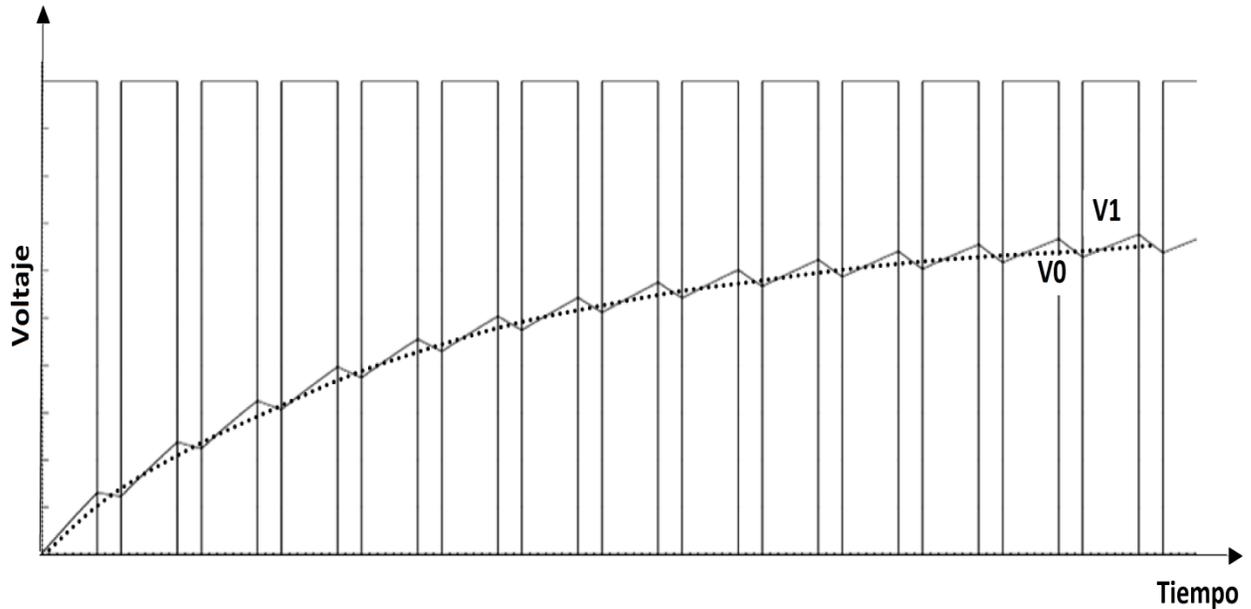


Figura 5.12: Respuesta del circuito RC a la señal PWM.

A medida que el capacitor se va cargando y descargando, el voltaje del capacitor sigue una trayectoria de diente de sierra que eventualmente llega a un estado estacionario. En esta región de estado estacionario el voltaje del condensador oscila entre V_0 y V_1 volts. Por lo cual hace que se mantenga un voltaje a lo largo de todo el periodo de la señal PWM y generando a la salida una señal analógica de voltaje. Figura 5.12.

5.2.6 Comunicación SEC-Volante

La comunicación del SEC con el volante se da a través de una comunicación serial bajo la interfaz RS-485, mediante este estándar de comunicación el volante recibe la información que se despliega en la pantalla y a la vez manda información hacia el SEC sobre el estado de los controles integrados en el volante.

Interfaz RS-485.

Está definido como un sistema en bus de transmisión multipunto diferencial, es ideal para transmitir a altas velocidades sobre largas distancias (35 Mbps hasta 10 metros y 100 Kbps en 1.200 metros) y a través de canales ruidosos, ya que reduce los ruidos que aparecen en los voltajes producidos en la línea de transmisión. El medio físico de transmisión es un par entrelazado que admite hasta 32 estaciones en 1 solo hilo, con una longitud máxima de 1.200 metros operando entre 300 y 19.200 bps y la comunicación half-duplex(semiduplex). Soporta 32 transmisiones y 32 receptores. La transmisión diferencial permite múltiples drivers dando la posibilidad de una configuración multipunto. Al tratarse de un estándar bastante abierto permite muchas y muy diferentes configuraciones y utilidades. Dentro de sus principales características se encuentran:

- Interfaz diferencial
- Conexión multipunto
- Alimentación única de +5V
- Hasta 32 estaciones (ya existen interfaces que permiten conectar 256 estaciones)
- Velocidad máxima de 10 Mbps (a 12 metros)
- Longitud máxima de alcance de 1.200 metros (a 100 Kbps)
- Rango de bus de -7V a +12V

Configuraciones.

Dentro de las más importantes, se encuentran:

- Half-dúplex: permite la comunicación entre un maestro y varios esclavos o entre un esclavo el maestro, pero por turnos (no simultáneamente)
- Full-dúplex: permite la comunicación entre maestro y esclavos y entre un esclavo y el maestro, ambas a la vez. Sólo un esclavo puede comunicarse a la vez, de ahí que el maestro arbitre el canal.

Half-dúplex

Para esta tipología es necesario un par de cables para transmitir una tensión diferencial positiva y negativa. Suele existir un tercer cable para poner las tierras en común.

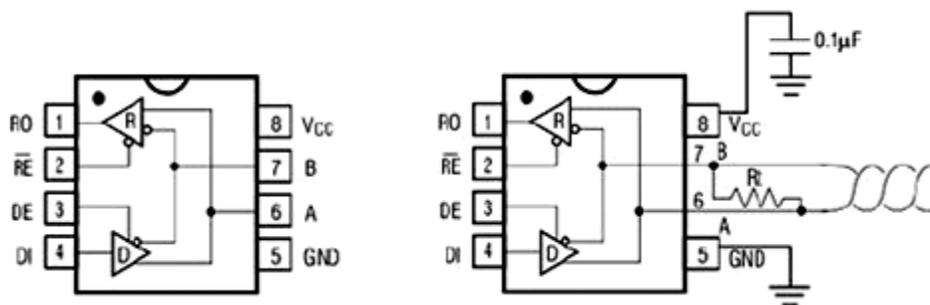


Figura 5.13: Configuración half-duplex.

Como se puede ver en la figura 5.13 la configuración en half-dúplex requiere controlar 2 líneas más: RE y DE. Éstas permiten activar un driver u otro para conmutar entre el modo TX y el modo RX.

Circuito.

Debido a que el microcontroladores Arduino cuentan únicamente con puerto de comunicación UARTS y éste maneja valores TTL, se requiere un conversor de nivel, en este caso el MAX 485.

De igual manera el volante contiene un microcontrolador COP8SBR9N (National Semiconductors™), dicho microcontrolador incorpora un periférico de comunicación (entre otros) con un protocolo RS232. Y de igual manera necesita del conversor de voltaje MAX. Esto se puede apreciar en la figura 5.14

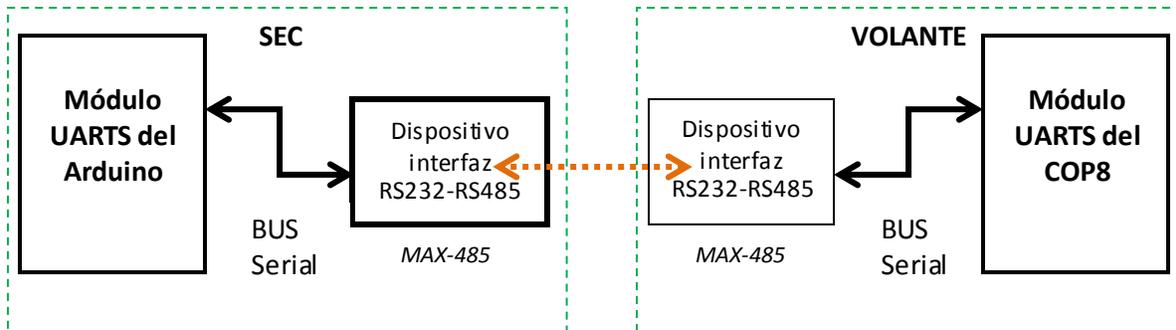


Figura 5.14: Estructura de transmisión de datos.

En la Figura 5.15 se observa, cómo se ha implementado el circuito electrónico (configuración half-dúplex), se utiliza una resistencia que sirve para balancear la carga entre las dos líneas, ya que el protocolo RS485 utiliza la diferencia de voltajes de dos líneas para lograr la comunicación, esto permite ser menos vulnerable al ruido y a las pérdidas debidas a las largas distancias de comunicación (1[Km] como máximo sin repetidores).

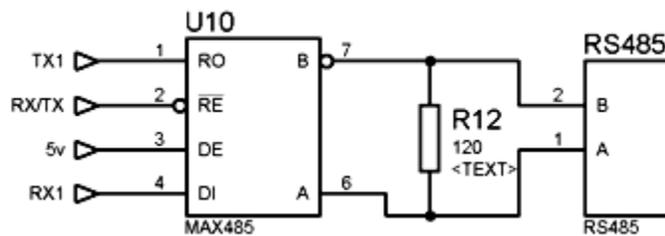


Figura 5.15: Circuito para comunicación RS-485

5.3 Diseño de la tarjeta PCB

Un circuito impreso, tarjeta de circuito impreso o PCB (por sus siglas en inglés *printed circuit board*), es una superficie constituida por caminos o pistas de material conductor laminadas sobre un sustrato no conductor. El circuito impreso se utiliza para conectar por medio del sustrato, un conjunto de componentes electrónicos. Las pistas son generalmente de cobre mientras que el sustrato se fabrica de resinas de fibra de vidrio reforzada, cerámica, plástico, teflón o polímeros como la baquelita [17].

La implementación del esquemático del circuito eléctrico, se realizó en el programa de desarrollo *Isis Proteus*. Donde se elaboró todo el circuito eléctrico que llevará el SEC, desde la alimentación de la tarjeta, hasta el circuito de comunicación con el volante. Como se puede apreciar en la figura 5.16

Si bien las librerías del programa *Isis Proteus* no permitieron que se pudiera simular el 100% de los componentes utilizados, si se pudo realizar una simulación de los circuitos de salidas analógicas y digitales, así como las entradas analógicas.

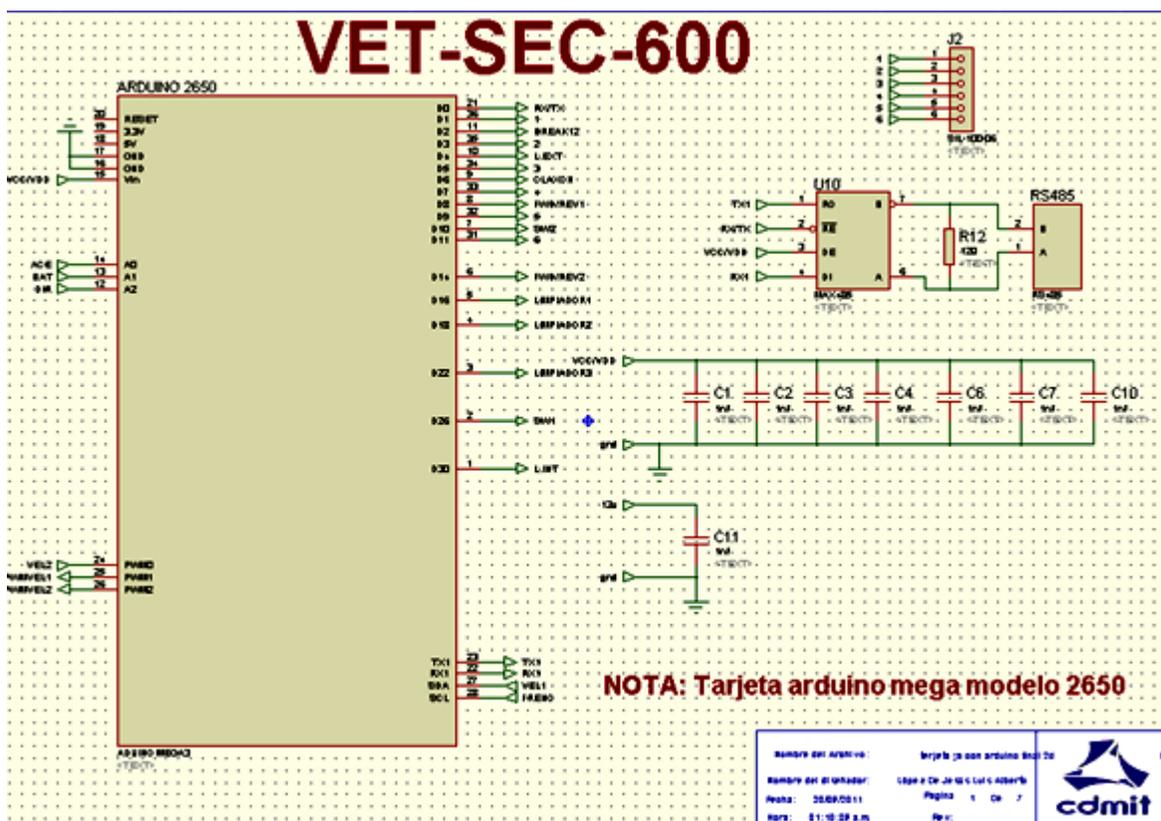


Figura 5.16: Diagrama esquemático de la tarjeta VET-SEC-600.

Ya que la “tarjeta Arduino Mega” es una placa desarrollada, por la empresa Arduino, usando como base el microcontrolador ATMEGA 2650, no se cuenta con librerías dentro del programa *Isis Proteus* que puedan simular el esquemático o el PCB de dicha tarjeta. Es de esta manera que se desarrolló un esquemático y su diagrama PCB dentro del programa Arduino. El desarrollar el PCB de la tarjeta Arduino favoreció mucho en el diseño de la tarjeta VET-SEC-600, ya que se omitieron barrenos de la tarjeta Arduino, que no se necesitaban, para poder hacer el trazado de las pistas más simple.

Tamaño y forma del circuito.

Antes de empezar la distribución de los componentes que forman el circuito, se definió el tamaño y la forma que finalmente tendrá la tarjeta VET-SEC-600.

La tarjeta VET-SEC-600 se diseñó para que se pueda acoplar físicamente con la tarjeta Arduino Mega, además de esto se buscó que fuera lo más compacta posible en sus dimensiones. La forma más común y la que se utilizó en el diseño de la tarjeta VET-SEC-600 es rectangular.

Colocación de componentes.

En cualquier diseño de un circuito eléctrico impreso la colocación de los componentes en la placa debe de seguir un criterio lógico, pero sobre todo que sea ordenado y que la longitud de las pistas en general sea lo más corta posible. La siguiente tabla muestra algunos criterios utilizados al momento de colocar los componentes.

Posición similar al esquemático	La posición de los componentes en la placa será similar a la disposición de los mismos en el esquemático. Esto es sólo recomendable con circuitos sencillos.
En función de la frecuencia de operación	En circuitos cuya frecuencia de operación sea elevada, hay que tener mucha precaución en las longitudes y las formas de las pistas, para evitar problemas de temporización y de reflexiones de señales. Minimizar siempre la longitud de las pistas, no emplear nunca pistas con ángulos rectos.
En función del espacio disponible	En este caso será necesario adaptarse al espacio y la forma del circuito, intentando a su vez respetar las reglas anteriores.

Tabla 5.3: Criterio para la colocación de componentes en circuitos impresos.

En el diseño de la VET-SEC-600 se buscó que fuera lo más compacta posible en dimensiones, basado en esto es como se fueron colocando los componentes a través de la tarjeta.

Definición de los barrenos de sujeción.

Nunca hay que olvidarse a la hora de hacer un circuito impreso que éste siempre va a estar dentro de algún aparato o chasis y va a necesitar soportes de sujeción. Lo más común es emplear tornillos que sujeten la placa al chasis. Para colocar estos tornillos necesitamos hacer barrenos en la placa del diámetro un poco mayor del que tenga el tornillo.

Es muy importante que la separación mínima permitida a cualquier otro elemento metálico sea suficiente. No se debe de olvidar que la cabeza del tornillo tiene un diámetro mayor y que si al apretarlo llega a tocar alguna pista pondremos en corto circuito la pista con el chasis, pudiendo hacer funcionar de manera inadecuada el circuito o incluso poner en peligro a los usuarios.

En el diseño de la tarjeta VET-SEC-600 se consideraron 4 barrenos para sujeción con tornillo de 1/8 de pulgada, colocados en cada una de las 4 esquinas de la tarjeta.

Trazado de pistas

Se debe de tratar de hacer el trazado de las pistas en una sola cara de la tarjeta, así se reduce el costo de la elaboración. Se deben de colocar los componentes de manera apropiada y si es necesario emplear puentes (para pasar las pistas por encima de otras sin provocar corto).

Se recomienda emplear un trazado de pistas en 45°, ya que:

- En circuitos de alta velocidad presenta menor cambio de impedancia que el trazado de pistas a 90° teniendo menores problemas de onda reflejada.
- La esquina sufre menor tensión que en 90°, al haber más superficie de metal pegada al circuito, por lo que facilita su fabricación.

En cuanto a los grosores de las pistas, en general han de ser mayores conforme mayor corriente pase por ellas. Para las alimentaciones y tierras emplear grosores entre 20 y 40 mil, según el circuito, para minimizar la impedancia de las pistas, y para señales entre 10 y 20 mil.

Para el trazado de las pistas de la tarjeta VET-SEC-600 se utilizaron segmentos a 45° para poder cambiar el sentido del trazado, además de esto, se colocaron "Thermal Relief" aliviadores térmicos, que son la forma con la que los pines conectados a planos metálicos se unen a ellos. Su función es *aliviar térmicamente* o bloquear el paso del calor del pin al resto de los elementos, evitando que se puedan desoldar o calentar en exceso al soldar el pin.

En vez de unir directamente el pin al plano, la unión sólo se realiza en unos puntos, garantizando la conductividad del pin y aumentando la resistencia térmica hacia el resto de los elementos. Se pueden apreciar en la figura. 5.17 el trazado de pista a 45° y los thermal relief.

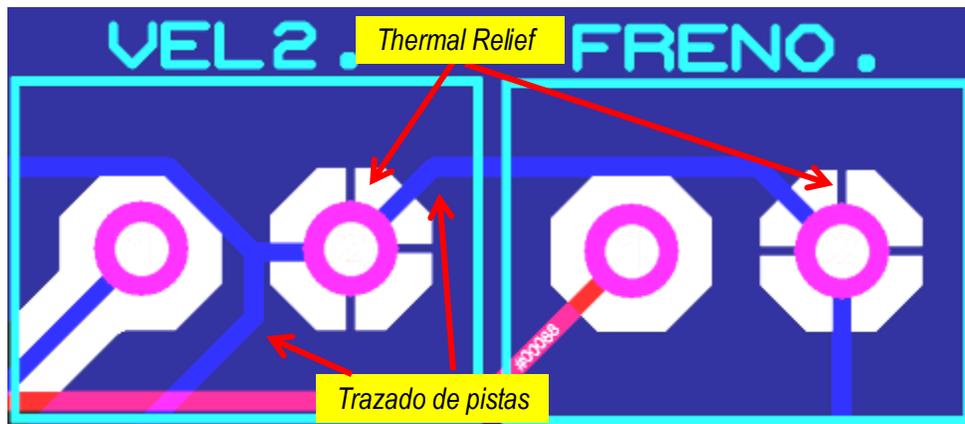


Figura 4.17: Trazado de pistas y thermal Relief

Dado que el circuito de la tarjeta VET-SEC-600 es algo extenso, no se podía desarrollar en una sola cara, es por esto que se diseñó a dos caras, bottom layer (Apéndice C) y top layer (Apéndice C), para facilitar el trazado de las pistas y así disminuir el tamaño de la tarjeta.

Posteriormente se colocaron planos de tierra (top layer) y masa (bottom layer) en la tarjeta VET-SEC-600 (anexo C)

Se colocaron todas las etiquetas para los componentes, para saber la posición en la que van colocados, así como simbología de referencia, para conducir con el diseño de la tarjeta PCB.

Pruebas

FABRICACIÓN

La etapa final del proceso de diseño, es la fabricación de un prototipo funcional de la tarjeta VET-SEC-600. Antes de este punto se ha verificado el correcto funcionamiento y desempeño de los elementos de la simulación virtual, el cual fue desarrollado en la fase de diseño, que ha revisado los planos de detalle y que se han seleccionado los componentes electrónicos para la realización del diseño final (que los componentes electrónicos sean comerciales).

Una vez que se tuvo desarrollado el diseño de la tarjeta VET-SEC en el programa de desarrollo *Ares de Isis Proteus* (Anexo B), se generaron los archivos Gerber para su fabricación, en la figura 6.1 se puede observar la tarjeta fabricada.

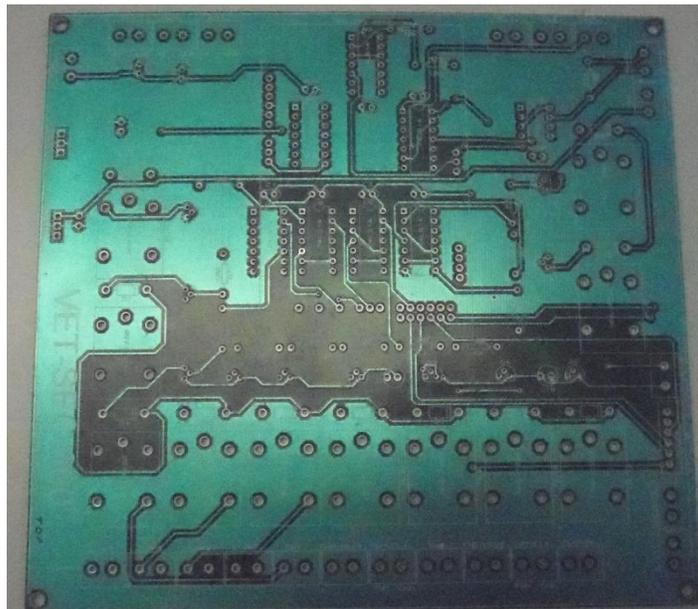


Figura 6.1: Tarjeta PCB VET-SEC-600.

Una vez que se tuvo fabricada la tarjeta, se procedió a soldar los componentes electrónicos (Anexo B), conforme las especificaciones del circuito. En la figura 6.2 se muestra la tarjeta al acabar de soldar los componentes.



Figura 6.2: Montaje de los componentes electrónicos

6.1 PRUEBAS

Para poder establecer el adecuado funcionamiento del Sistema electrónico de control, se han decidido hacer básicamente 2 pruebas.

- Pruebas en frío.
 - Verificar protocolo de comunicación
 - Encendido del vehículo.
 - Verificar entradas y salidas del SEC.
- Pruebas en Caliente.
 - Conexión del vehículo.
 - Comportamiento de los motores.

6.1.1 PRUEBAS EN FRIO

Estas pruebas se desarrollarán en el laboratorio (figura 6.3), se tiene una fuente de alimentación a 12 [volts] para proveer de energía al SEC, un generador de funciones que hace la vez del sensor de efecto hall en las ruedas del vehículo, es decir, con éste se controla la velocidad del VET. Un osciloscopio para poder ver de manera gráfica las señales de salida de voltaje.

Se hizo una evaluación del algoritmo de control implementado en la plataforma de desarrollo *Arduino* (Anexo A), los puntos principales de esta prueba se enumeran a continuación:

1. **Comunicación de SEC con el volante:** donde se verificará que el protocolo de comunicación sea el adecuado, que tanto el volante como el SEC se comunican de manera correcta. Es decir que se envíen y reciban los datos del protocolo de comunicación, esto se puede visualizar en el LCD del volante, que muestra la velocidad media del vehículo, nivel de carga y recorrido restante estimado del vehículo, el correcto funcionamiento de la comunicación se puede observar en la figura 6.4. donde se varía la

velocidad media del vehículo y se observa que el volante cambia la velocidad, conforme a lo calculado.

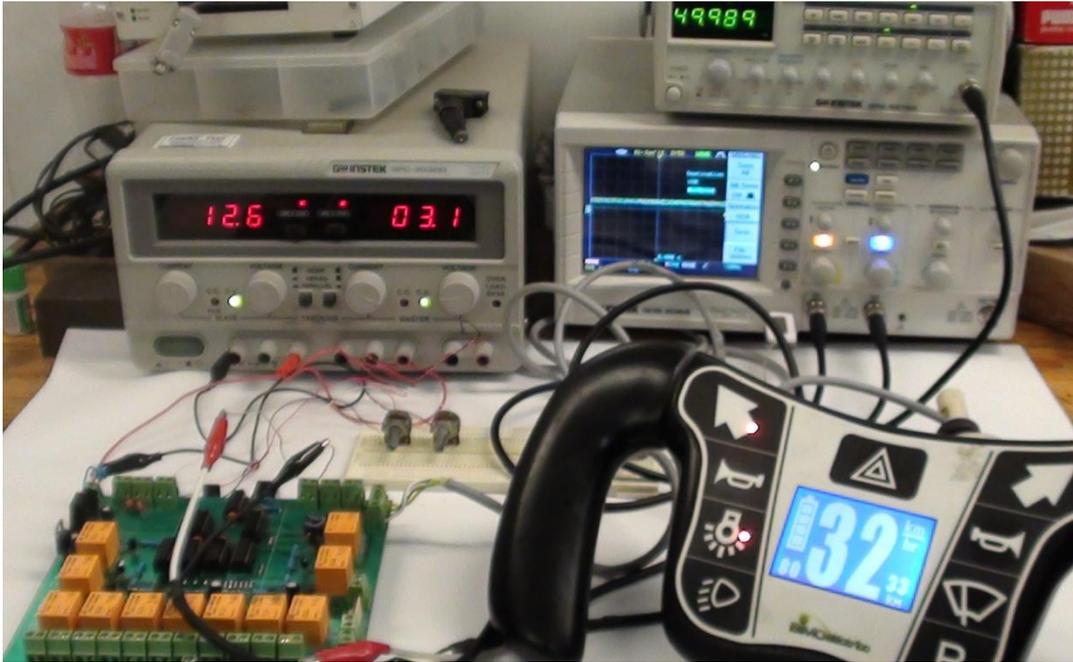


Figura 6.3: montaje para pruebas de laboratorio.

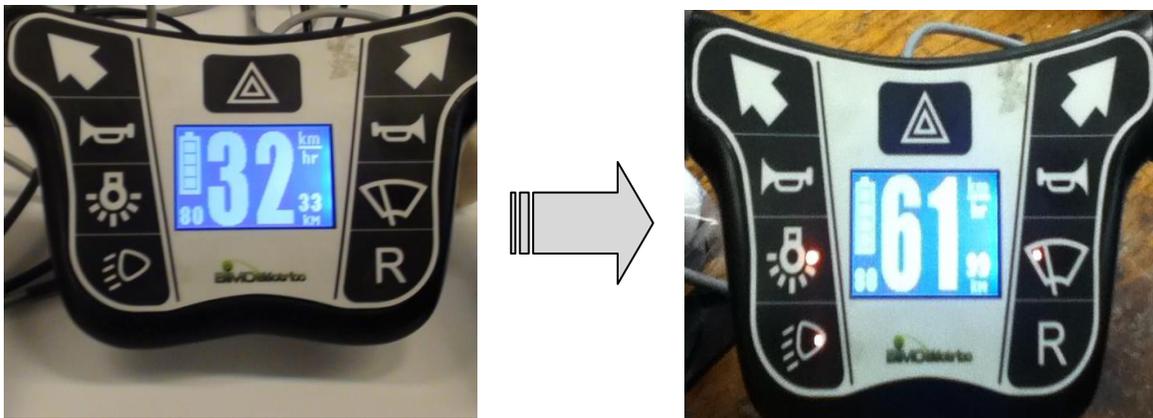


Figura 6.4: Prueba de comunicación, cambio en la velocidad.

2. **Entradas del SEC:** se verificaron las entradas del SEC, entre las cuales se encuentran entradas analógicas y entradas digitales (interrupciones). Para poder verificar las entradas analógicas se alimentó al SEC con una entrada analógica a través de un potenciómetro.

Para las entradas digitales, las interrupciones que se utilizaron para poder obtener los pulsos provenientes de los sensores de las llantas del SEC, se probaron utilizando un generador de funciones y se corroboró su correcto funcionamiento al poder observar la

velocidad en el display LCD del volante conforme a lo esperado por los cálculos realizados en la parte “determinación de la velocidad lineal en cada llanta” dentro del capítulo 4 de este trabajo. Ambas entradas trabajaron de manera correcta.

3. **Salidas digitales:** para poder probar las salidas digitales del SEC, se presionaron los botones de mando en el volante, a sabiendas que el correcto funcionamiento en la comunicación y en el programa del SEC harían que se activaran los relevadores correspondientes a cada botón del volante.

Las 11 salidas digitales funcionan de manera correcta, es decir, se activan de manera adecuada. Haciendo énfasis en la salida digital del claxon que se activa por primera vez al pulsar 3 veces el botón de claxon. Esto corresponde al requerimiento del encendido del vehículo.

4. **Salidas analógicas:** para poder probar las salidas analógicas, se utilizó el osciloscopio para poder visualizar las señales. En la figura 6.5 se pueden observar las señales de salida analógica del SEC. Con esto se pudo verificar que los voltajes de salida fueran de 0 a 5 [volts].

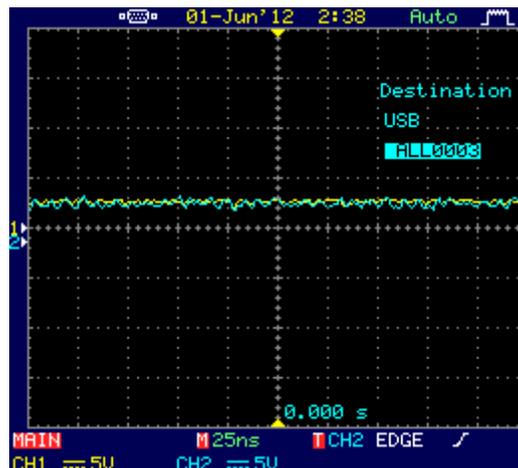


Figura 6.5: Salidas analógicas del SEC.

El programa del SEC funcionó de manera correcta, primeramente la comunicación con el volante funcionó de manera adecuada enviando y recibiendo los datos, se activan los relevadores correspondientes a los botones en el volante. Por otro lado los relevadores para activar los contactores del vehículo, no se activan hasta que se aprieta por tercera vez el botón del claxon.

Una vez encendidos los contactores, la regulación de velocidad en las salidas analógicas es la esperada por las ecuaciones establecidas. También el algoritmo del “error” está funcionando de manera adecuada al compensar el error en cuanto se nota una variación en las velocidades.

6.1.2 PRUEBAS EN CALIENTE

Para poder realizar estas pruebas, se procedió a montar el SEC en el vehículo. En el cual ya se tenían los demás elementos para el funcionamiento del vehículo (motores, controlador de los motores, baterías, contactores, etc.)

Se verificó la correcta instalación de la tarjeta SEC, así como el alambrado de conexiones entre los componentes conforme a la imagen 6.6. Se montó la SEC sobre el brazo del volante, para mantenerla fija y de ahí hacer las conexiones. Como se puede observar en la imagen 6.7 las conexiones fueron realizadas con facilidad gracias a las kulkas que tiene integrada la SEC.

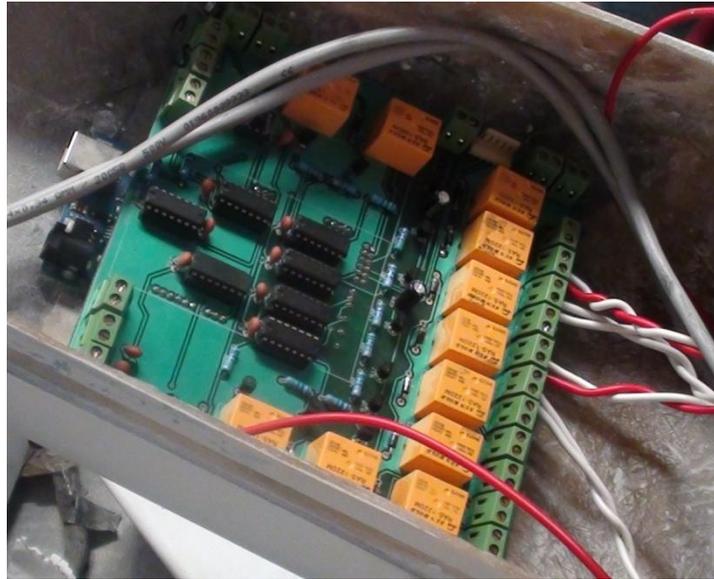


Figura 6.7: Salidas analógicas del SEC.

En esta prueba se verificó el funcionamiento del SEC al acoplarse con los demás sistemas que conforman el VET e interactúan con la tarjeta SEC. Se verificaron los siguientes puntos:

- Se comprobó que los motores del vehículo respondieran de manera adecuada a la dirección solicitada por el usuario. Es decir que el SEC regulara de manera correcta las velocidades de los motores tomando en cuenta el acelerador del vehículo y la dirección del volante.
- Que las señales llegaran de manera correcta hacia la tarjeta SEC, provenientes de los sensores (potenciómetros de dirección y acelerador).
- Las salidas digitales del SEC, desde el encendido del vehículo por medio de la combinación de los controles de mando del vehículo, así como las salidas que activan los de más periféricos del vehículo funcionan de manera adecuada.

Se puede concluir que tanto la tarjeta PCB SEC y el algoritmo de regulación de velocidad, cumplen de manera satisfactoria los requerimientos y necesidades. En términos de operación, el vehículo se comporta de acuerdo a lo estipulado.

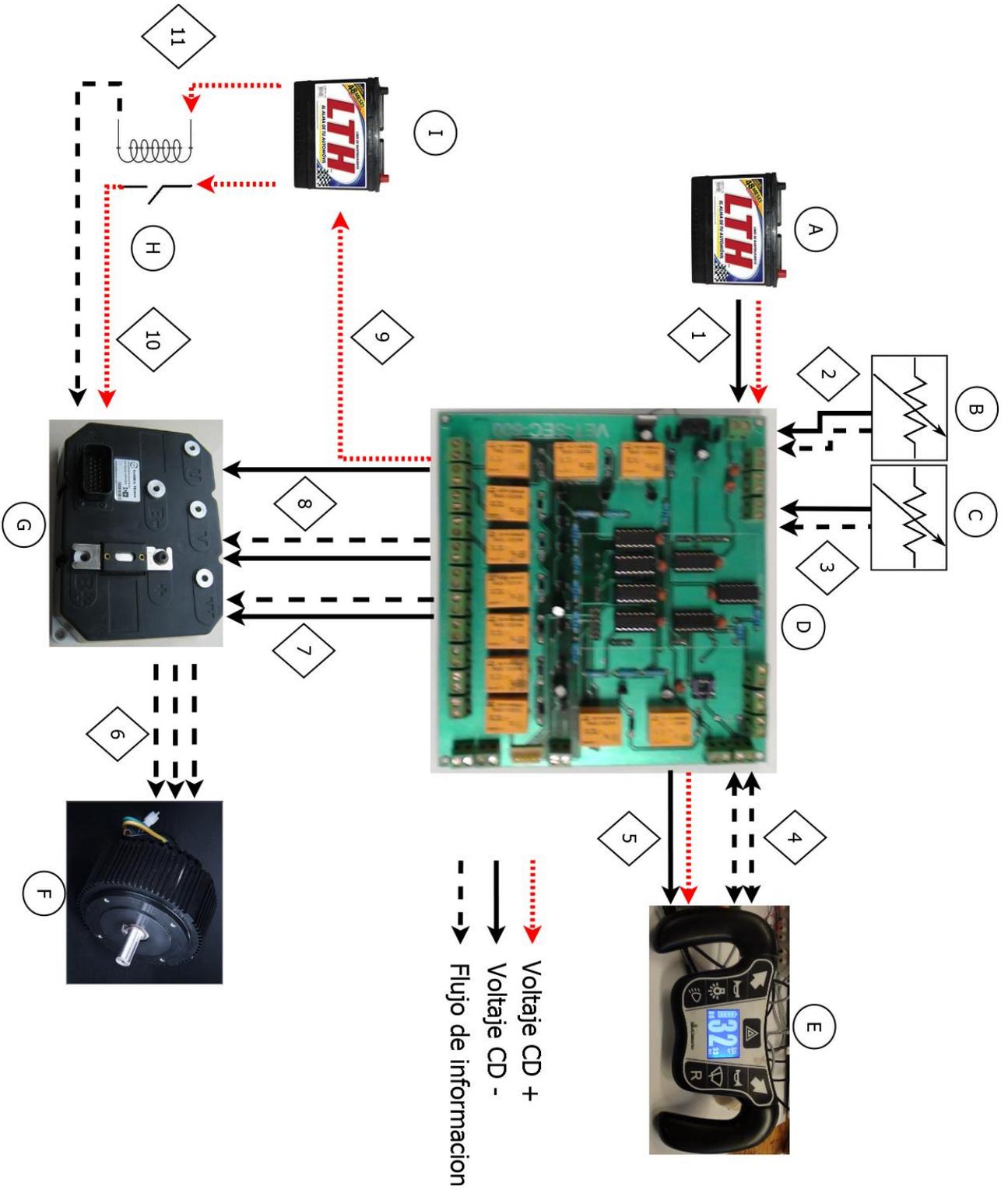


Figura 6.6: Conexiones para el SEC

ELEMENTO	DESCRIPCIÓN	CONEXIÓN	DESCRIPCIÓN
A	Batería de alimentación 12[V] de DC para la tarjeta SEC.	1	Dos líneas de conexión V+ y V-
B	Pedal acelerador, para indicar la velocidad del vehículo	2	Dos líneas donde se envía el voltaje analógico de 0 a 5 [V]. V+ y GND
C	Potenciómetro para indicar la dirección del vehículo.	3	Dos líneas donde se envía el voltaje analógico de 0 a 5 [V]. V+ y GND
D	Tarjeta PCB del sistema electrónico de control.		
E	Volante del vehículo.	5	Dos líneas de alimentación de voltaje al volante 8 [V]. V+ y GND
		4	Dos líneas de comunicación serial RS-485. A y B
F	Motor	6	Conexiones del controlador al motor por 3 líneas.
G	Controlador	7	Velocidad PWM, el SEC indica a través de un voltaje analógico la velocidad que debe de tener el motor. V+ y GND
		8	Reversa, el SEC indica al controlador a través de una señal digital la orden de reversa.
		9	Conexión de encendido/apagado del contactor.
H	Contactor para el encendido de los motores	10	Alimentación de voltaje hacia el controlador.
		11	Alimentación de la bobina para activar el contactor.

Conclusiones

Con base a las necesidades del usuario y de las especificaciones presentadas para este proyecto, se pudo establecer una solución viable y demostró, a lo largo del presente trabajo ser una buena solución para el problema de la regulación de velocidad de los motores a partir de la posición del acelerador y la dirección indicada por el volante.

En la parte del diseño electrónico, se seleccionaron los componentes adecuados para el acondicionamiento de las señales, tanto de entrada como de salida. Se adecuaron de manera satisfactoria para que no existieran problemas entre los diferentes rangos que maneja la tarjeta de desarrollo Arduino Mega y los diferentes componentes que se utilizaron (sensores, pedal acelerador, pedal freno, volante, sistema de luces traseras.) Así mismo se buscó que la utilización de componentes electrónicos garantizaran el correcto funcionamiento del SEC en todo momento, protegiendo al SEC, volante y sistema de luces traseras con reguladores de voltaje para su correcta alimentación de energía.

Siguiendo con la estructura que tiene el presente documento, los algoritmos se desarrollaron a partir de buscar soluciones simples a los problemas. A partir de estas soluciones aisladas se agrupó en un algoritmo que da solución al problema global. De igual manera se buscó que el algoritmo fuera lo más simple y confiable. Se realizaron varias pruebas previas en la programación para determinar cuál sería una solución confiable, como fue el caso de las ecuaciones para establecer la regulación de velocidad de ambos motores. Como se explicó previamente se utilizó la idea de usar dos rectas intersectantes en el punto medio del ángulo de giro, pero se observó que esta solución sólo era parcial, ya que al momento de girar el volante hacia un lado o hacia el otro, la regulación de velocidad de los motores no mostraban los comportamientos deseados.

Al contar el vehículo con sensores en las ruedas, se pudo establecer una retroalimentación de información para poder controlar mejor el vehículo y no dejarlo en un lazo abierto. Lo cual favoreció poder corregir en caso de que un motor esté respondiendo de diferente manera a lo esperado.

El diseño de la tarjeta PCB fue un nuevo reto en mi formación académica, ya que previamente contaba con muy pocos conocimientos acerca del diseño y fabricación de circuitos impresos. Después de que se condujo la etapa de diseño del esquema electrónico, se procedió a hacer el diseño de la tarjeta PCB teniendo que buscar variantes en las salidas y entradas de la tarjeta de desarrollo Arduino Mega, esto con el fin de minimizar el número de pistas. Es por este motivo que la distribución de entradas y salidas en la tarjeta Arduino no sigue un orden consecutivo.

A partir de que se concluyó el diseño de la tarjeta PCB, se tuvieron que generar los archivos GERBER para poder mandarla a fabricar, en este sentido tampoco se contaba con experiencia, debido al poco trabajo que se había elaborado previamente a este proyecto.

En la etapa de pruebas, primeramente se contemplaron las pruebas de laboratorio en las cuales se pudo apreciar de manera satisfactoria que tanto la tarjeta SEC como los algoritmos funcionaran de manera adecuada. Que la comunicación entre el SEC y el volante no tuviera problemas y de igual manera no existiera alguna falla entre el SEC y el sistema de luces traseras. Al final de este proceso se puede concluir que las pruebas en laboratorio al SEC fueron exitosas.

Posteriormente se colocó el SEC en el vehículo, donde se comprobó su funcionamiento de manera eficiente acoplado a los sistemas definitivos que tendría el VET. Estas pruebas también se concluyeron de manera exitosa, al responder de manera satisfactoria los dos motores a la regulación de velocidades que estaba ordenando el SEC. La comunicación con el volante no tuvo ninguna falla y el vehículo respondió correctamente a las instrucciones del usuario. No reportando al término de estas pruebas ninguna falla.

Por último de manera general al término de este documento se puede establecer que, se cumplieron los objetivos que se plantearon al inicio del mismo, pudiendo establecer las siguientes justificaciones:

- ✓ *Regular la velocidad de los motores brushless a través del ángulo de giro del volante y el acelerador del vehículo.* Cumplido con el algoritmo de regulación de velocidades explicado en el capítulo 4 del presente trabajo y comprobado en la etapa de pruebas.
- ✓ *Poder controlar la dirección del vehículo, por medio de la diferencial de velocidades en los motores delanteros del vehículo.* Cumplido al poder establecer las ecuaciones para las velocidades de los motores 1 y 2 (izquierdo y derecho) expuesto en el capítulo 4 del presente trabajo y corroborado en la etapa de pruebas.
- ✓ *Diseñar la tarjeta electrónica para poder controlar los motores, así como los demás periféricos que tiene el vehículo.* Cumplido, como se expuso en el capítulo 4 donde se detallan los componentes que se utilizaron para el diseño del SEC. Y que se puede corroborar en las imágenes del capítulo 6, donde se hicieron las pruebas a dicha tarjeta.
- ✓ *Realizar pruebas de la tarjeta de control.* Cumplido, dedicando un capítulo del presente trabajo, a la realización de pruebas del SEC.

Además de los objetivos, también se lograron cumplir los alcances planteados al inicio de este proyecto, los cuales incluyen:

- ✓ Desarrollar el algoritmo de control para la operación de los motores *brushless*.
- ✓ Diseñar la electrónica para el acondicionamiento de señales, adecuándolas a los requerimientos del microcontrolador.
- ✓ Diseñar la tarjeta impresa (PCB) para la electrónica de control y acondicionamiento de las señales de la tarjeta.
- ✓ Supervisar la fabricación y ensamble de la tarjeta de control.

Todos estos alcances se cumplieron y se especifican a lo largo del presente trabajo.

TRABAJO A FUTURO.

Hasta la culminación de este trabajo, los sensores para leer la velocidad de las ruedas de los motores, no habían sido colocados los sensores ni los imanes. Por tal motivo las pruebas del vehículo se realizaron sin considerar estas entradas.

Realizar el análisis cinemático del vehículo, para establecer cuál sería el radio de curvatura del VET, es decir la distancia que necesita el vehículo para cambiar de dirección completamente. En este caso se denomina “diámetro de giro” (el doble del radio de giro) y determina si el vehículo va a ser capaz de dar la vuelta en una calzada sin hacer maniobra.



Figura 7.1: Radio de curvatura de un vehículo [18]

Efectuar pruebas del vehículo con carga, en una pendiente para poder establecer si el vehículo cumple con lo requerido. De ser así se estarían comprobando los cálculos efectuados para par y torque del motor, establecidos en la tesis “Diseño de sistema de tracción para vehículo eléctrico.” [4]

Por otro lado, ahora con más conocimientos sobre el diseño y elaboración de circuitos impresos PCB, se podría mejorar el diseño de la misma y corregir algunos detalles. Como el grosor de las pistas y el trazado de algunas de ellas. Con el fin de reducir el tamaño de la tarjeta PCB SEC así como el “*layer silk*” para la identificación de los componentes, ya que algunos quedaron cubiertos por los componentes, como en el caso de las kulkas.

Referencias

BIBLIOGRAFIA

- [1] ROJAS Bracho, Leonora. Tercer almanaque de datos y tendencias de la calidad del aire en nueve ciudades mexicanas. Secretaria del medio ambiente y recursos naturales: 2007
- [2] CARDENAS Arboleda, Oscar Santiago. Técnicas de control difuso aplicadas a la navegación de un robot móvil. Escuela Politécnica Nacional. Quito, Ecuador: 2006.
- [3] MOEDANO Hernández, David. Volante extraíble con controles al mando para vehículo eléctrico. Universidad Nacional Autónoma de México: 2012
- [4] SÁNCHEZ Medina, Emilio Ataulfo. Diseño de sistema de tracción para vehículo eléctrico. Universidad Nacional Autónoma de México. 2012.
- [5] VALENZUELA Ruz, Víctor. Manual de análisis y diseño de algoritmos Versión 1.0. Instituto Nacional de Capacitación Copiapó, Chile: 2003

MESOGRAFIA

- [6] Vehículos de motor registrados por entidad federativa, 2000 a 2009. Instituto nacional de estadística y geografía. <<http://www.inegi.org.mx/est/contenidos/Proyectos/registros/economicas/vehiculos/default.aspx>>, [Consultada el 9 de abril de 2012].
- [7] COSTAS Torres, Javier. Historia de los coches eléctricos, <<http://www.motorpasion.com/coches-hibridos-alternativos/historia-de-los-coches-electricos>> [consultada el 4 de octubre de 2011]
- [8] TWIKE information, < <http://www.twike.com/en/home/home.html>>, [Consultada el 7 de noviembre de 2011].
- [9] MYERS Motors , < <http://myersmotors.com/> >, [Consultada el 7 de noviembre de 2011].
- [10] CREE company, < <http://www.friends-of-sam.com/> >, [Consultada el 7 de noviembre de 2011].
- [11] APTERA Motors < <http://www.aptera.com/> >, [Consultada el 12 de noviembre de 2011].
- [12] Zilent de México < <http://zilentmexico.com/>>, [Consultada el 12 de noviembre de 2011].
- [13] Nissan LEAF® < <http://nissanleaf.mx/?gclid=CPOHrabF3q8CFeJgTAodNgKdAw>>, [Consultada el 21 de noviembre de 2011].

[14] Golden Motor, company info. *Brushless* DC motors and controllers
<<http://www.goldenmotor.com/>>, [Consultada el 23 de abril de 2012].

[15] Productos Honeywell, información de productos.
<http://sensing.honeywell.com/product%20page?pr_id=36074>, [Consultada el 9 de noviembre de 2011].

[16] Arduino <<http://www.arduino.cc/>>, [Consultada el 24 de abril de 2012].

[17] The History of PCBs <<http://www.4pcb.com/pcb/>>, [Consultada el 19 de agosto de 2011].

[18] Radio de giro de los coches <<http://blogdemotor.com/diseño-vehículos/el-radio-de-giro-de-los-coches/>> [Consultada el 05 de Agosto de 2012].

Código del algoritmo de control

```
/******* DECLARACIÓN DE VARIABLES*****//
```

```
#include <FlexiTimer2.h> //encabezado para induir la librería del timer.
float PWM1_VAL; //Variable que indica la velocidad ideal del motor 1
float PWM2_VAL; //Variable que indica la velocidad ideal del motor 2
float direccion; //variable que guarda el valor de la dirección
float velocidad; //variable que guarda el valor de la velocidad media del vehículo
float acelerador; //variable que guarda el valor del acelerador lectura del canal analógico.
float acelerador2; //variable que guarda el valor del acelerador mapeado a un nuevo rango.
float eva_dir; //variable que guarda el valor del cálculo de ciclo de trabajo PWM
float bateria=80; //variable que guarda el valor del nivel de la batería
float recorrido=0; //variable que guarda el valor del recorrido
float prom1 = 0; //variable que guarda el valor de los pulsos promedio del motor 1
float prom2 = 0; //variable que guarda el valor de los pulsos promedio del motor 2
int CRC=0; //variable que guarda el valor del CRC calculado.
int i=0,j=0,a=0,r=0,s=0; //variables auxiliares para conteo
int cont=0; //variable para contar las pulsaciones del botón "claxon"
int PWM1=0; //variable para enviar el Ciclo de trabajo PWM motor 1
int PWM2=0; //variable para enviar el Ciclo de trabajo PWM motor 2
int error1=0; //variable para el cálculo del error del motor 1
int error2=0; //variable para el cálculo del error del motor 2
int pulsos1 = 0; //variable para contar los pulsos del sensor 1
int pulsos2 = 0; //variable para contar los pulsos del sensor 2
int suma1 = 0; //variable para sumar los totales pulsos del sensor 1
int suma2 = 0; //variable para sumar los totales pulsos del sensor 1
int freno=0; //variable que guarda el valor del freno.

int freno=21; //Señal del "freno" por salida digital 21 de la tarjeta Arduino Mega
int claxon=28; //Señal del "claxon" por salida digital 28 de la tarjeta Arduino Mega
int reversa=30; //Señal de "reversa" por salida digital 30 de la tarjeta Arduino Mega
int lext=26; //Señal de "luces exteriores" por salida digital 26 de la tarjeta Arduino Mega
int lint=52; //Señal de "luces interiores" por salida digital 52 de la tarjeta Arduino Mega
int limpia1=38; //Señal de "limpiadores1" por salida digital 36 de la tarjeta Arduino Mega
int limpia2=40; //Señal de "limpiadores2" por salida digital 40 de la tarjeta Arduino Mega
```

```

int limpia3=44; //Señal de "limpiadores3" por salida digital 44 de la tarjeta Arduino Mega
int SW1=48; //Señal de "encendido motor 1" por salida digital 48 de la tarjeta
Arduino Mega
int SW2=32; //Señal de "encendido motor 2" por salida digital 32 de la tarjeta Arduino Mega
int frenoled = 33; //Señal de "luces del freno" por salida digital 33 de la tarjeta Arduino Mega
int reversaleds= 31; //Señal de "reversa" por salida digital 31 de la tarjeta Arduino Mega
int inter= 29; //Señal de "intermitentes" por salida digital 29 de la tarjeta Arduino Mega
int cuartos= 27; //Señal de "cuartos" por salida digital 27 de la tarjeta Arduino Mega
int dir_izq= 25; //Señal de "direccionales izquierda" por salida digital 25 de la tarjeta Arduino
Mega
int dir_der=23; //Señal de "direccionales derecha" por salida digital 23 de la tarjeta Arduino Mega
int PWM11=3; //señal de motor 1 por salida analogica (PWM) 3 de la tarjeta Arduino Mega.
int PWM22=4; //señal de motor 2 por salida analógica (PWM) 4 de la tarjeta Arduino Mega.

int dato1[5]; //arreglo para almacenar 5 valores de pulsos contados motor 1
int dato2[5]; //arreglo para almacenar 5 valores de pulsos contados motor 2
int trans[7]; //arreglo de 7 valores a transmitir al volante
int recep[5]; //arreglo de 5 valores a recibir del volante

void setup(){
  pinMode(claxon,OUTPUT); //configuración de salidas
  pinMode(reversa,OUTPUT);
  pinMode(lext,OUTPUT);
  pinMode(lint,OUTPUT);
  pinMode(limpia1,OUTPUT);
  pinMode(limpia2,OUTPUT);
  pinMode(limpia3,OUTPUT);
  pinMode(frenoled,OUTPUT);
  pinMode(reversaleds,OUTPUT);
  pinMode(inter,OUTPUT);
  pinMode(cuartos,OUTPUT);
  pinMode(dir_izq,OUTPUT);
  pinMode(dir_der,OUTPUT);
  pinMode(freno,INPUT);
  pinMode(PWM11,OUTPUT);
  pinMode(PWM22,OUTPUT);

  pinMode(freno,INPUT); //configuración de entradas

  Serial1.begin(115200); //inicio de comunicación serial 1 a 115200
  attachInterrupt(3,cuenta1,FALLING); //configuración de interrupción 3

```

```

attachInterrupt(0,cuenta2,FALLING); //configuración de interrupción 0
FlexiTimer2::set(200,tiempo); //configuración del timer cada 200 [ms]
FlexiTimer2::start(); //inicio del timer

trans[0]=170; //ID MASTER [0xAA]
trans[1]=204; // ID SLAVE [0xCC]
delay(3000);
}

//***** RUTINA PRICIPAL*****//
void loop(){

acelerador=(map(analogRead(0),0,1023,0,99)); //leer el valor analógico del acelerador
velocidad= ((prom1+prom2)/2)*3.1416*0.57*1.8; //cálculo de la velocidad media del vehiculo
recorrido=(map(analogRead(2),0,1023,0,99)); //lectura del valor de la dirección (volante)
direccion=recorrido;
frenoval=digitalRead(freno); //lectura del estado del freno.
eva_dir=100*((-direccion-50)*(direccion-50)/2500)+1);

//*****RUTINA PARA EL CÁLCULO DEL PWM DE LA DIRECCIÓN DIFERENCIAL*****//

if (direccion<=50) //giro del volante a la izquierda
{
PWM1_VAL=acelerador*eva_dir; //calculo de ciclos de trabajo M2 y M2
PWM2_VAL=acelerador*100;
}
else //giro del volante a la derecha
{
PWM2_VAL=acelerador*eva_dir; //cálculo de ciclos de trabajo M2 y M2
PWM1_VAL=acelerador*100;
}

//*****RUTINA PARA CORRECCIÓN DEL ERROR*****//
acelerador2=(map(analogRead(0),0,1023,0,255)); //mapeo de la lectura del acelerador a
ciclo de trabajo PWM
error1=acelerador2-((255*prom1)/1.86); //cálculo del error en M1
error2=acelerador2-((255*prom1)/1.86); //cálculo del error el M2

PWM1=map(PWM1_VAL,0,10000,0,255); //mapeo del ciclo de trabajo de 0-10000 a 0-255 en
M1
PWM2=map(PWM2_VAL,0,10000,0,255); //mapeo del ciclo de trabajo de 0-10000 a 0-255 en
M2

```

```

if (PWM1>255)                //limites de ciclo de trabajo
PWM1=255;                    //para no sobre pasar el rango de
if (PWM1<0)
PWM1=0;                      // 0-255 en motor 1

if (PWM2>255)                //limites de ciclo de trabajo
PWM2=255;                    //para no sobre pasar el rango de
if (PWM2<0)
PWM2=0;                      // 0-255 en motor 1

analogWrite(PWM11,PWM1); //enviar el ciclo de trabajo a salida analógica de M1
analogWrite(PWM22,PWM2); //enviar el ciclo de trabajo a salida analógica de M1

trans[2]=int(velocidad);     //asignación de valores al arreglo a transmitir
trans[3]=int(bateria);
trans[4]=int(recorrido);
trans[5]=freno;
CRC=trans[2]^trans[3]^trans[4]^trans[5]; //cálculo del CRC
trans[6]=CRC;

//*****RUTINA PARA TRANSMITIR Y RECIBIR INFORMACIÓN *****//
delay(200);
Serial1.write(trans[0]); //Envío de información hacia el volante
Serial1.write(trans[1]);
Serial1.write(trans[2]);
Serial1.write(trans[3]);
Serial1.write(trans[4]);
Serial1.write(trans[5]);
Serial1.write(trans[6]);
delay(10);

recep[0]=Serial1.read(); //recepción de información desde el volante
recep[1]=Serial1.read();
recep[2]=Serial1.read();
recep[3]=Serial1.read();
recep[4]=Serial1.read();

//*****RUTINA PARA EL ARRANQUE Y ASIGNACIÓN DE SALIDAS DIGITALES *****//

```

```

if (recep[0]==204&&recep[1]==170){ //verificación de los ID_SLAVE e ID_MASTER

if(byte(recep[2]&64)==byte(64)){ //si botón del claxon se presionó
  if(cont<3){ //¿se presionó menos de 3 veces?
    cont++; // Si, incrementar el contador
    digitalWrite(claxon,LOW); // mantener apagado el daxon
  }
  else {
    digitalWrite(claxon,HIGH); // No, prender el claxon
    Arranque(); //invocar a la rutina de arranque del vehículo
    cont=4; //variable cont, se queda fija en 4.
  }
}
else //No se presionó el botón del daxon
  digitalWrite(claxon,LOW); //mantiene apagado el claxon
if(byte(recep[2]&32)==byte(32))
{ // botón de la reversa se oprimió
  digitalWrite(reversa,HIGH); //encender la salida de reversa
  digitalWrite(reversaeds,HIGH); //encender las luces traseras (reversa)
}
else //no se oprimió la reversa
{
  digitalWrite(reversa,LOW); //se mantiene apagada la señal de reversa
  digitalWrite(reversaeds,LOW); //se mantienen apagadas las luces traseras (reversa)
}

if(byte(recep[2]&16)==byte(16))
  digitalWrite(lint,HIGH); //se oprimió el botón de luces internas, entonces manda la
señal de control
  else
    digitalWrite(lint,LOW); //No se oprimió el botón de luces internas, se mantiene
apagado la señal de control

if(byte(recep[2]&8)==byte(8))
{ //se presionó el botón de luces externas
  digitalWrite(lext,HIGH); //enciende la señal de luces externas
  digitalWrite(cuartos,HIGH); //enciende las luces traseras (cuartos)
}
else
{ //no se presionó el botón de luces externas
  digitalWrite(lext,LOW); //se mantiene apagada la señal de luces externas
  digitalWrite(cuartos,LOW); //se mantienen apagadas las luces traseras (cuartos)
}

```

```

}

if(byte(recep[2]&4)==byte(4)) //se presionó el botón de luces intermitentes
digitalWrite(inter,HIGH); //enciende la señal de luces intermitentes
else //no se presionó el botón de luces intermitentes
digitalWrite(inter,LOW); //se mantienen apagadas las luces intermitentes

if(byte(recep[2]&2)==byte(2)) //se presionó el botón de direccional izquierda
digitalWrite(dir_izq,HIGH); //enciende la señal de luces direccional izquierda
else //no se presionó el botón de luces direccional izquierda
digitalWrite(dir_izq,LOW); //se mantienen apagadas las luces direccional izquierda

if(byte(recep[2]&1)==byte(1)) //se presionó el botón de direccional derecha
digitalWrite(dir_der,HIGH); //enciende la señal de luces direccional derecha
else //no se presionó el botón de luces direccional derecha
digitalWrite(dir_der,LOW); //se mantienen apagadas las luces direccional derecha

if(byte(recep[3]&1)==byte(1)) //se presionó el botón de limpiadores
digitalWrite(limpia1,HIGH); //enciende la señal de limpiadores
else //no se presionó el botón de limpiadores
digitalWrite(limpia1,LOW); //se mantienen apagadas los limpiadores

if(byte(recep[3]&2)==byte(2)) //se presionó el botón de limpiadores
digitalWrite(limpia2,HIGH); //enciende la señal de limpiadores
else //no se presionó el botón de limpiadores
digitalWrite(limpia2,LOW); //se mantienen apagadas los limpiadores

if(byte(recep[3]&3)==byte(3)) //se presionó el botón de limpiadores
digitalWrite(limpia3,HIGH); //enciende la señal de limpiadores
else //no se presionó el botón de limpiadores
digitalWrite(limpia3,LOW); //se mantienen apagadas los limpiadores

if(frenoal==1) //se presionó el botón de luces traseras (freno)
digitalWrite(frenoled,HIGH); //enciende la señal de luces traseras (freno)
else //no se presionó el botón de luces traseras (freno)
digitalWrite(frenoled,LOW); //se mantienen apagadas las luces traseras (freno)

}
}

//*****RUTINA PARA EL CONTEO DE PULSOS*****//
void cuenta1(){

```

```

    pulsos1++;                //incrementa cada que llega un pulso en M1
}
void cuenta2(){              //incrementa cada que llega un pulso en M2
    pulsos2++;
}

void tiempo()
{
    if(r==5){
        for(s=0;s<5;s++){
            suma1+=dato1[s];    //suma de pulsos guardados del motor 1
            suma2+=dato2[s];    //suma de pulsos guardados del motor 2
        }
        prom1=suma1/5.0;        //Calcula el promedio de pulsos cada 200 [ms] para el motor 1
        prom2=suma2/5.0;        //Calcula el promedio de pulsos cada 200 [ms] para el motor 1
        r=0;
    }
    else {
        suma1=0;                //reinicio de la variable suma 1
        suma2=0;                //reinicio de la variable suma 2
    }
    dato1[r]=float(pulsos1);    //almacena el valor de los pulsos contados
    dato2[r]=float (pulsos2);   //almacena el valor de los pulsos contados
    r++;
    pulsos1=0;                  //reiniciar los pulsos de motor 1
    pulsos2=0;                  //reiniciar los pulsos de motor 2
}

//*****RUTINA DE ARRANQUE *****/
void Arranque(){              //rutina de arranque

digitalWrite(SW1,HIGH);        //encendido de motor 1
digitalWrite(SW2,HIGH);        //encendido de motor 2

}

```

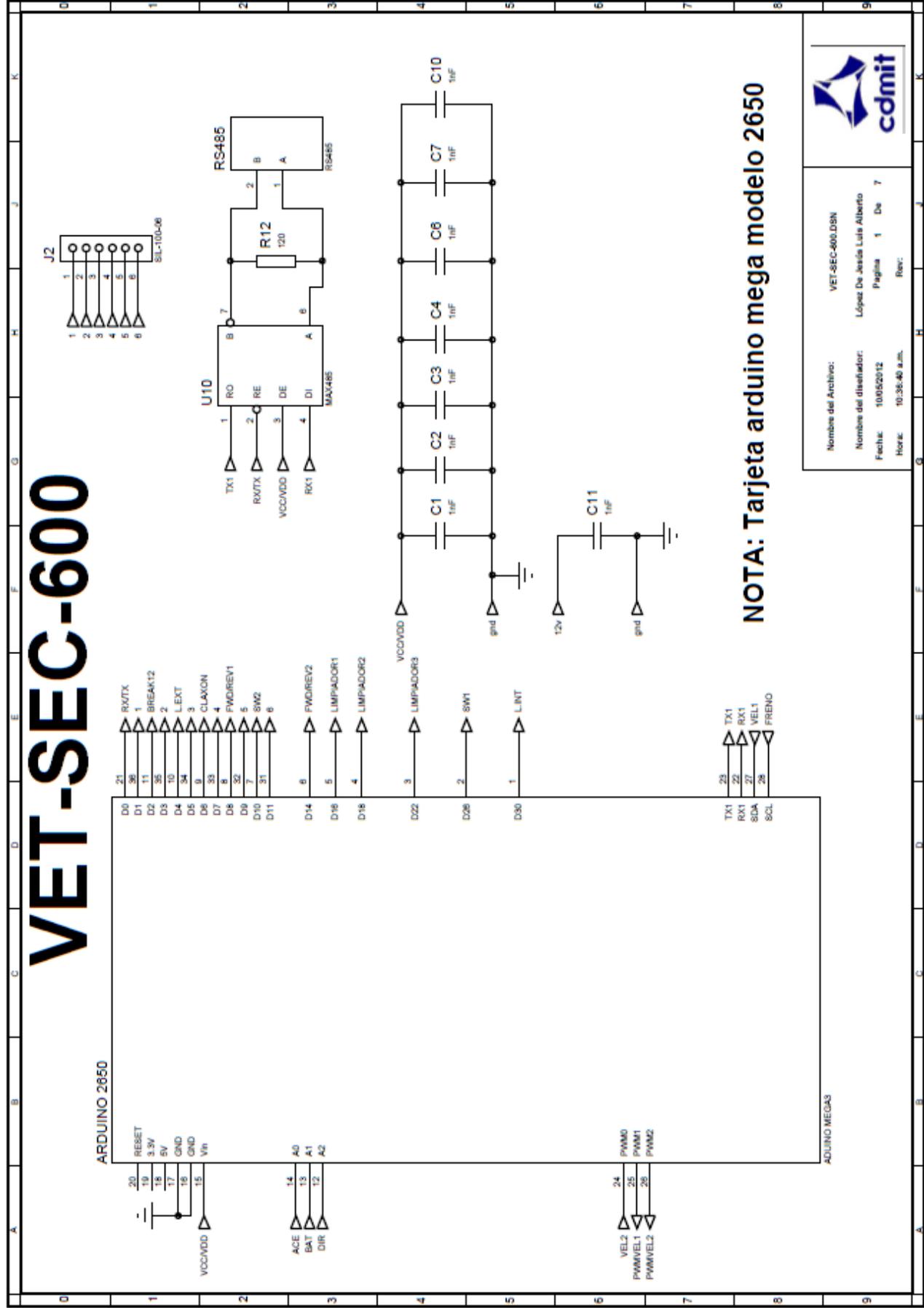
Anexo B

Diagrama eléctrico del SEC.

Como parte del diagrama eléctrico se presenta el BOM de la Tarjeta VET-SEC-600. Donde se hace referencia a los componentes electrónicos y como se encuentran distribuidos en la tarjeta VET-SEC-600.

CANTIDAD	REFERENCIA EN EL DIAGRAMA ELÉCTRICO	VALOR	COSTO UNITARIO (M.N.)	COSTO TOTAL (M.N.)
RESISTENCIAS				
11	R1-R11	100 [Ω]	0.862	8.62
1	R12	120 [Ω]	0.862	0.862
2	R14,R15	10K [Ω]	0.862	1.724
CAPACITORES				
8	C1-C4,C6,C7,C10,C11	1 [nF]	1.724	13.792
4	C5,C8,C12,C13	1 [uf]	1.724	6.896
2	C18,C28	10 [uf]	1.724	3.448
CIRCUITOS INTEGRADOS				
1	U1	LM324	3.448	3.448
6	U2,U3,U6-U9	LS7414	28.448	170.688
1	U4	7805	9.483	9.483
1	U10	MAX485	13.793	13.793
1	U11	7808	7.759	7.759
TRANSISTORES				
11	Q1-Q11	BC547	1.724	18.964
DIODOS				
11	D1-D11	1N4001	0.862	9.482
MISCELÁNEOS				
11	RL1,RL4-RL13	RAS-1220	19	209
1	ARDUINO MEGA	ARDUINO MEGA	950	950
23	CULCA	CULCA 2 TERM.	6	138
1	J8	MOLEX 8	3	3
2	Sensores 103SR	Sensores 103SR	683.4	1,366.8
2	Fabricación de tarjeta PCB		900	900
			Subtotal	3,835.759
			IVA	613.741
			Total	4,449.50

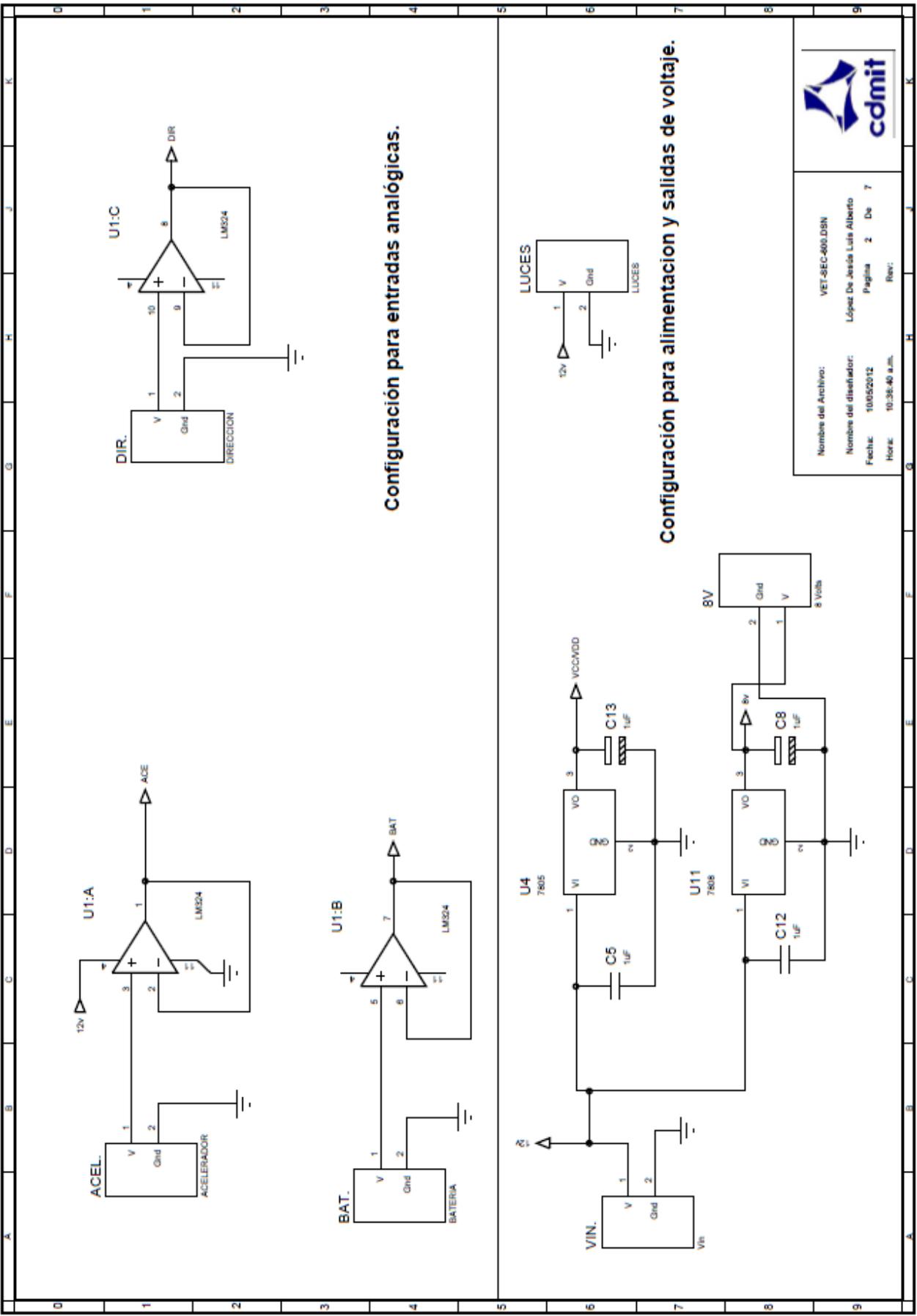
VET-SEC-600



NOTA: Tarjeta arduino mega modelo 2650



Nombre del Archivo:	VET-SEC-600.DSN
Nombre del diseñador:	López De Jesús Luis Alberto
Fecha:	10/05/2012
Hora:	10:38:40 a.m.
Rev:	

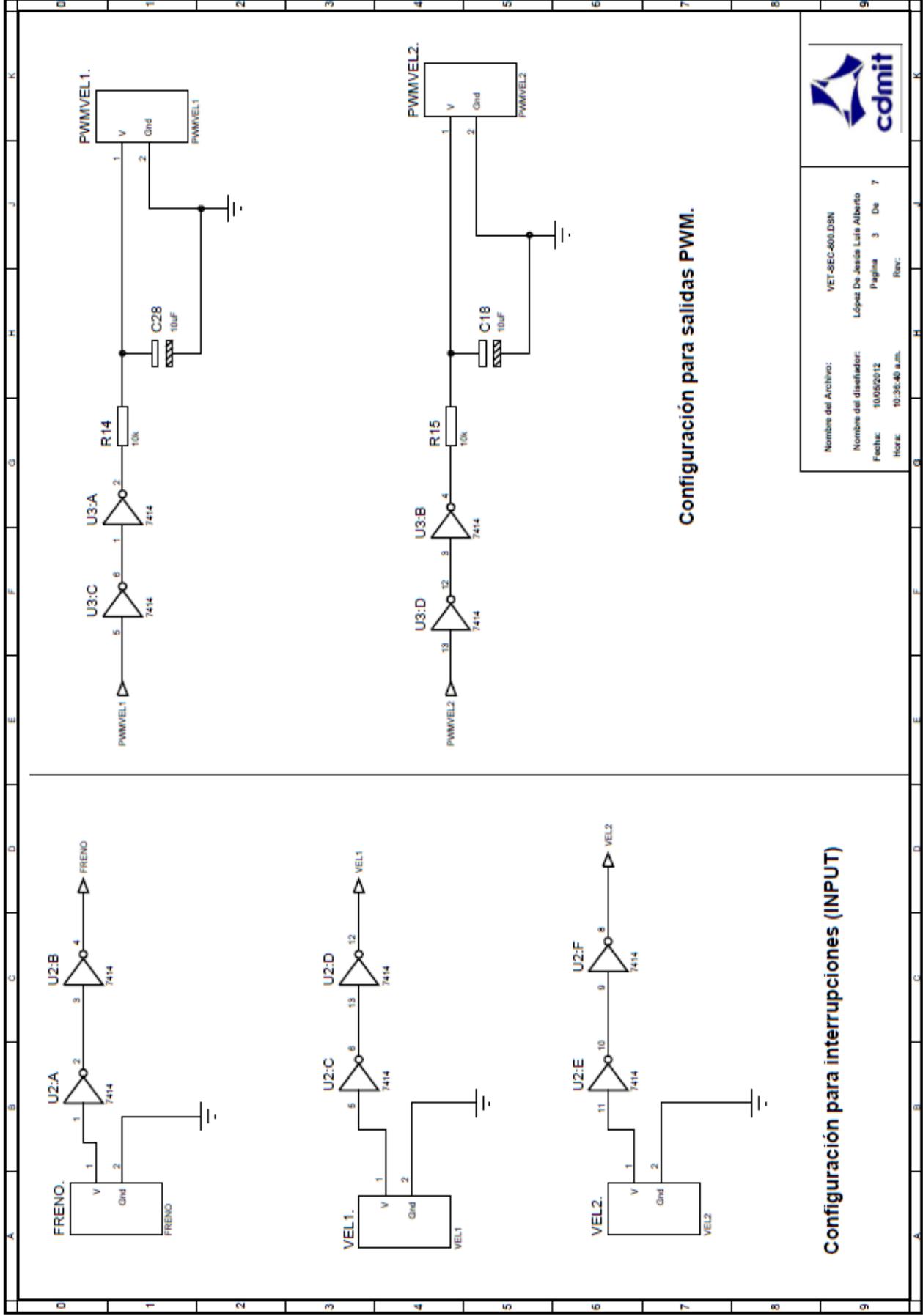


Configuración para entradas analógicas.

Configuración para alimentación y salidas de voltaje.



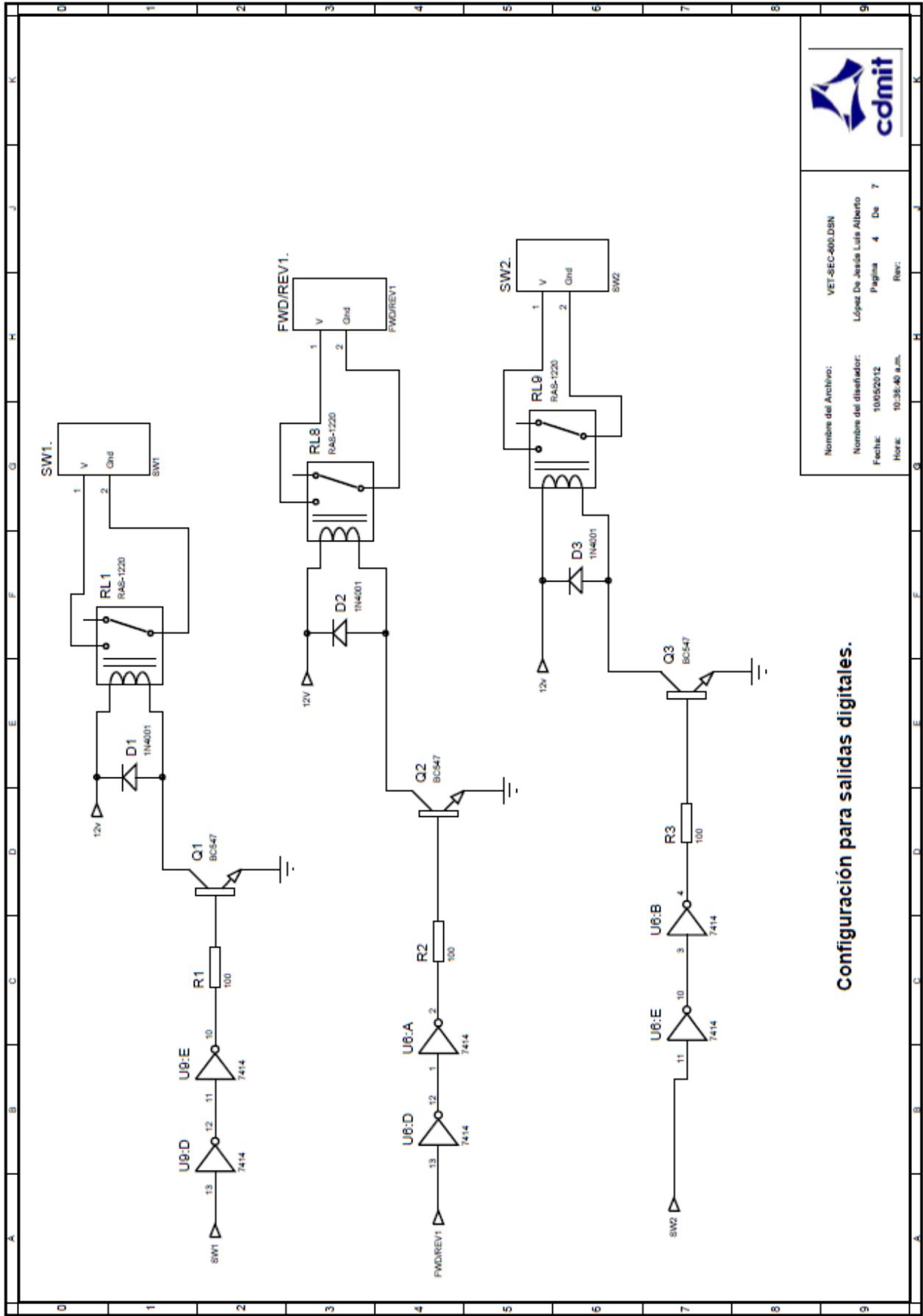
Nombre del Archivo: VET-SEC-600.DSN
 Nombre del diseñador: López De Jesús Luis Alberto
 Fecha: 10/05/2012 Página 2 De 7
 Hora: 10:36:40 a.m. Rev:



Nombre del Archivo: VET-SEC-400.DSN
 Nombre del diseñador: López De Jesús Luis Alberto
 Fecha: 10/05/2012 Página 3 De 7
 Hora: 10:35:49 a.m. Rev:

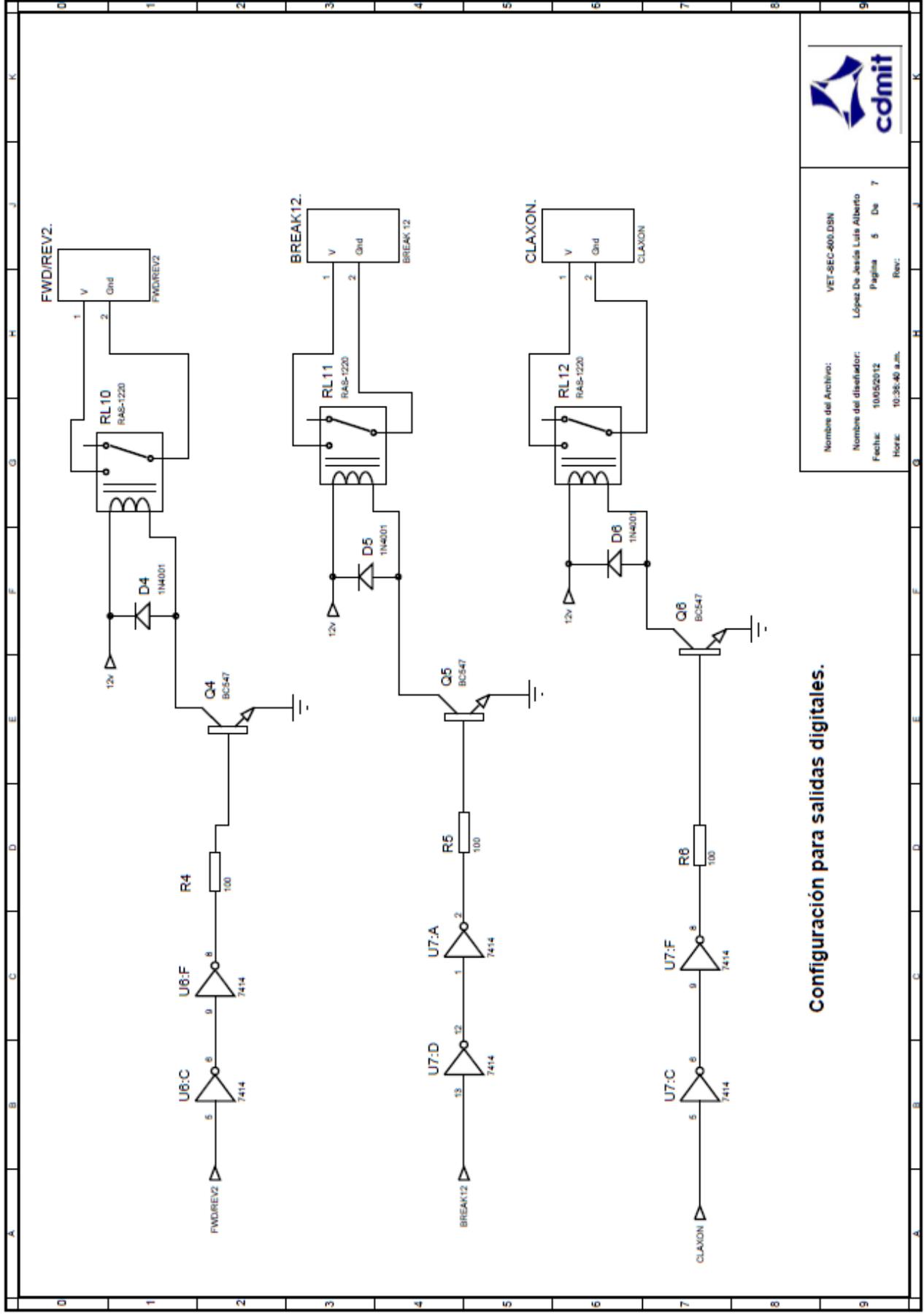
Configuración para salidas PWM.

Configuración para interrupciones (INPUT)



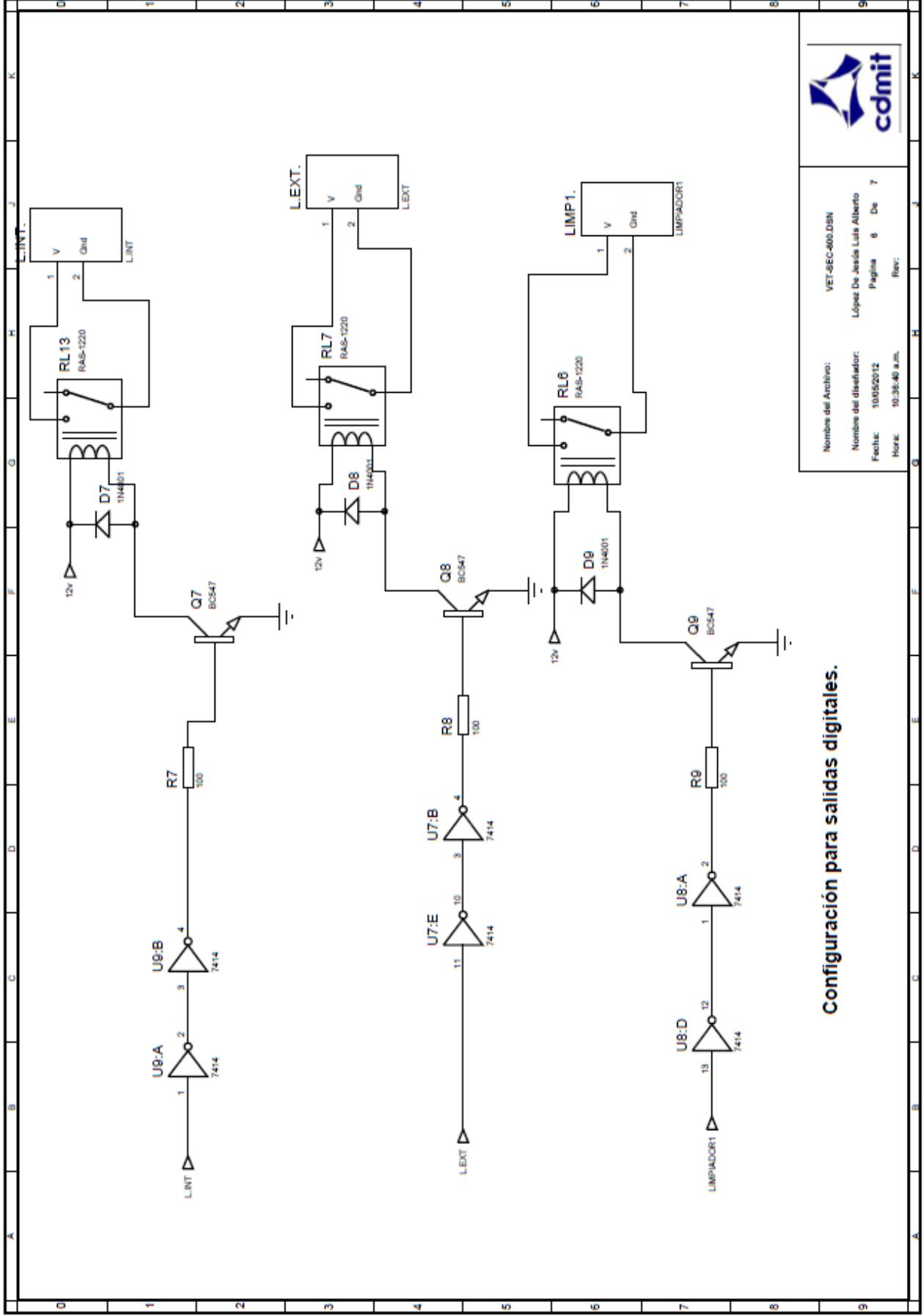
Nombre del Archivo: VET-SEC-600-DSN
 Nombre del diseñador: López De Jesús Luis Alberto
 Fecha: 10/05/2012
 Hora: 10:38:40 a.m.
 Página 4 De 7
 Rev:

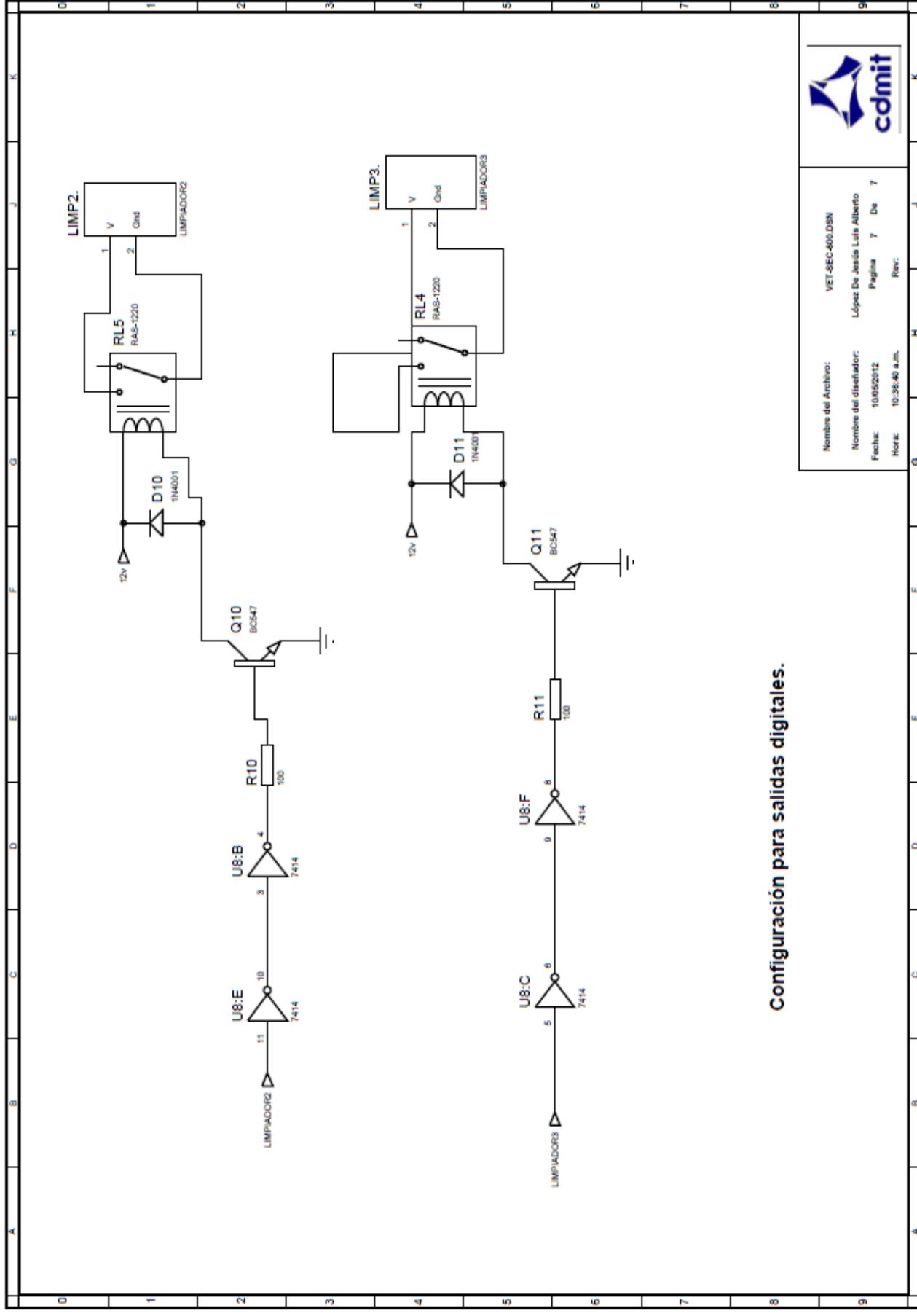
Configuración para salidas digitales.



Nombre del Archivo: VET-SEC-600.DSN
 Nombre del diseñador: López De Jesús Luis Alberto
 Fecha: 10/05/2012 Página 5 De 7
 Hora: 10:36:40 a.m. Rev:

Configuración para salidas digitales.





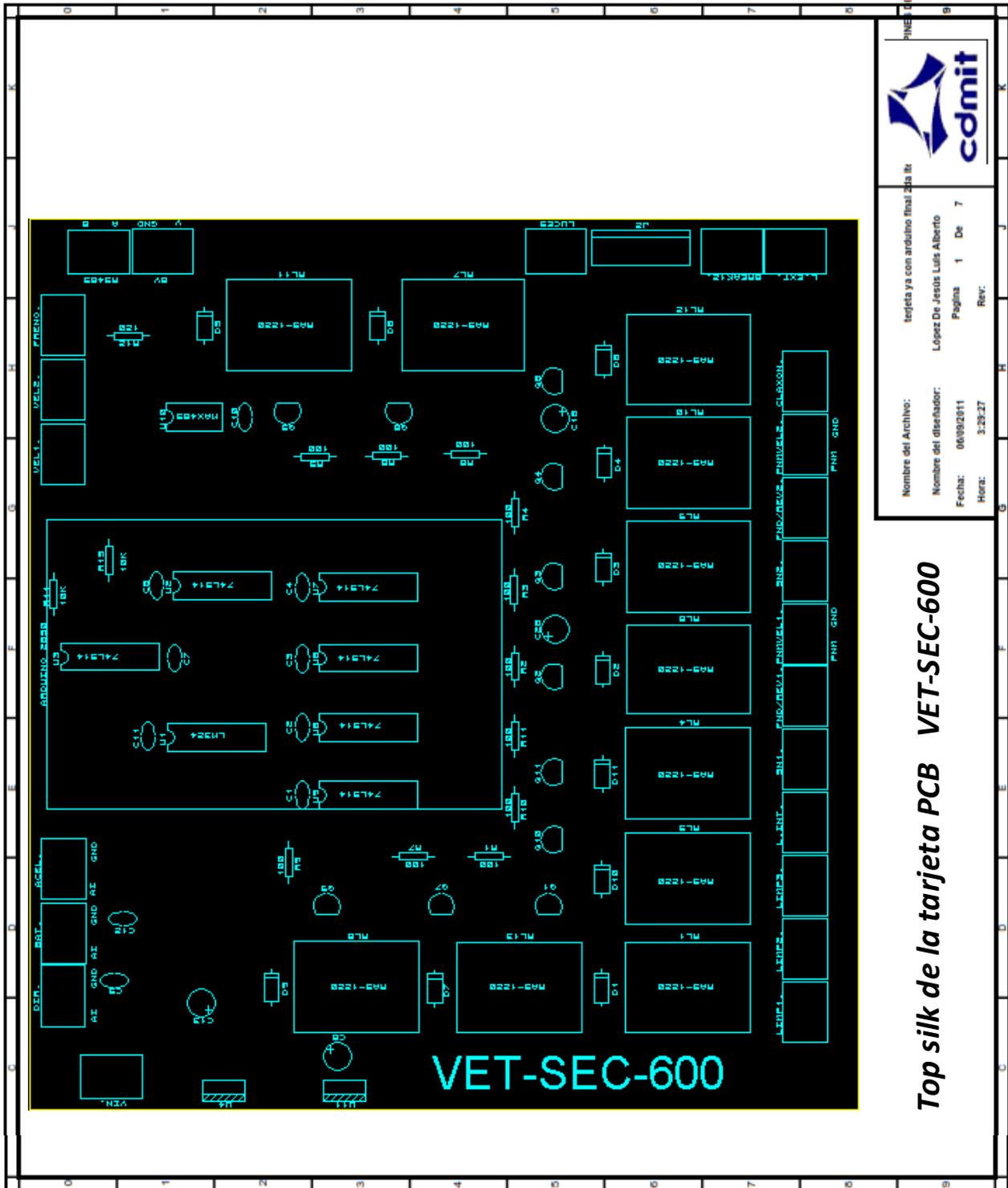
Configuración para salidas digitales.

Nombre del Archivo:	VET-SEC-600.DSN
Nombre del diseñador:	López De Jesús Luis Alberto
Fecha:	10/05/2012
Hora:	10:35:40 a.m.
Rev:	
Page:	7
De:	7



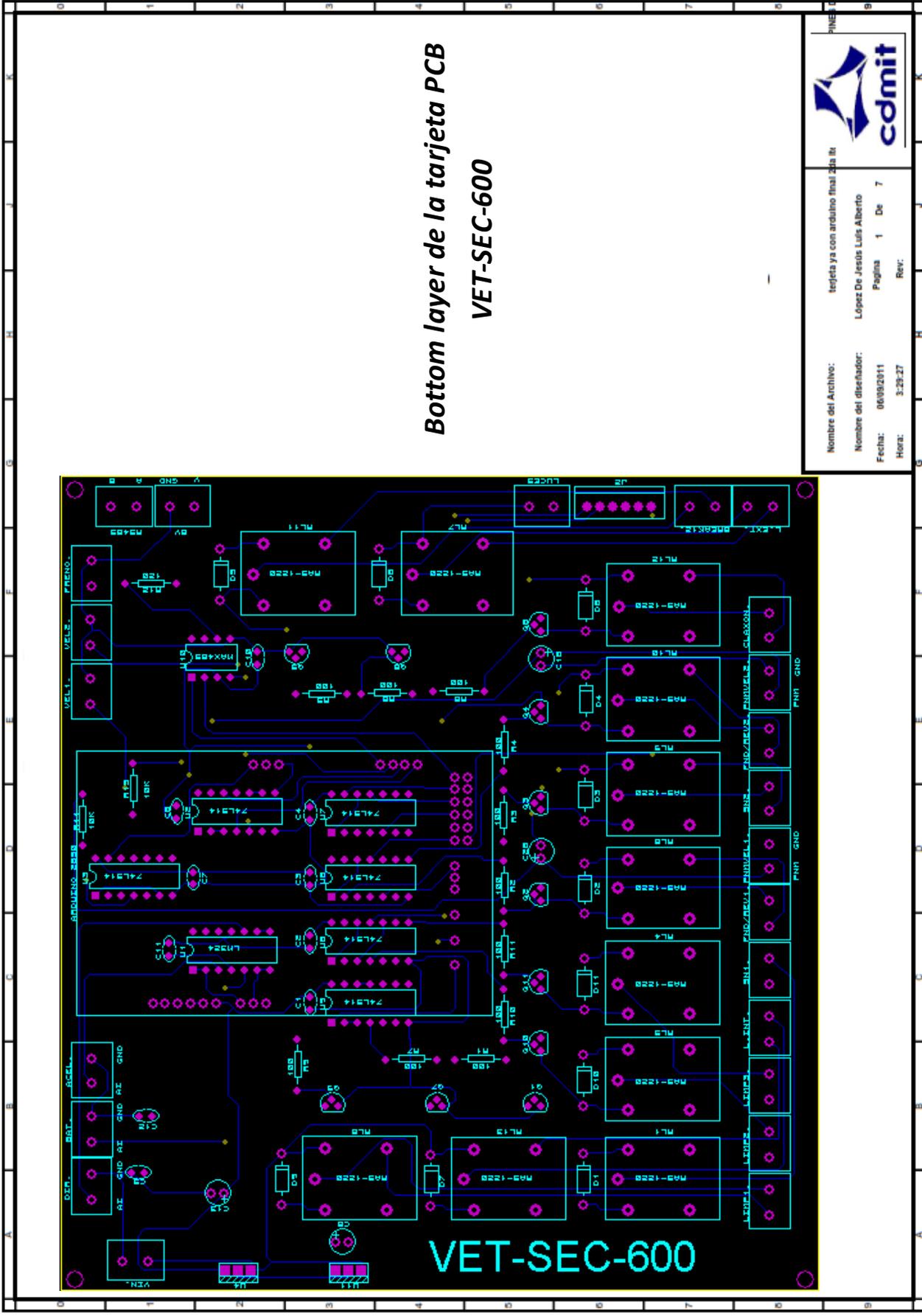
Anexo C

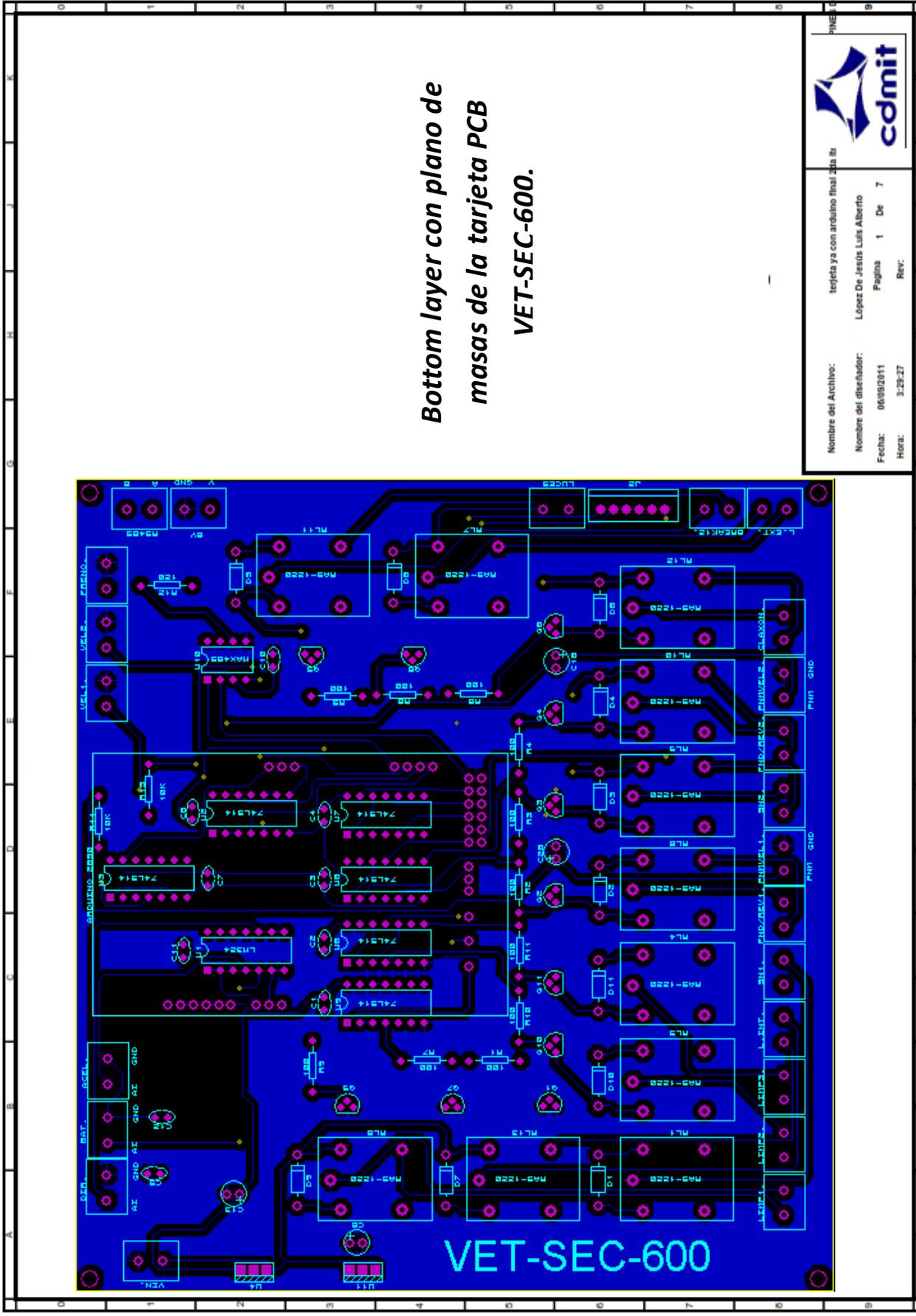
Diseño del PCB VET-SEC 600.



Nombre del Archivo: tarjeta ya con arduino final 2da It
 Nombre del diseñador: López De Jesús Luis Alberto
 Fecha: 06/09/2011
 Hora: 3:29:27
 Página 1 De 7
 Rev:

Top silk de la tarjeta PCB VET-SEC-600

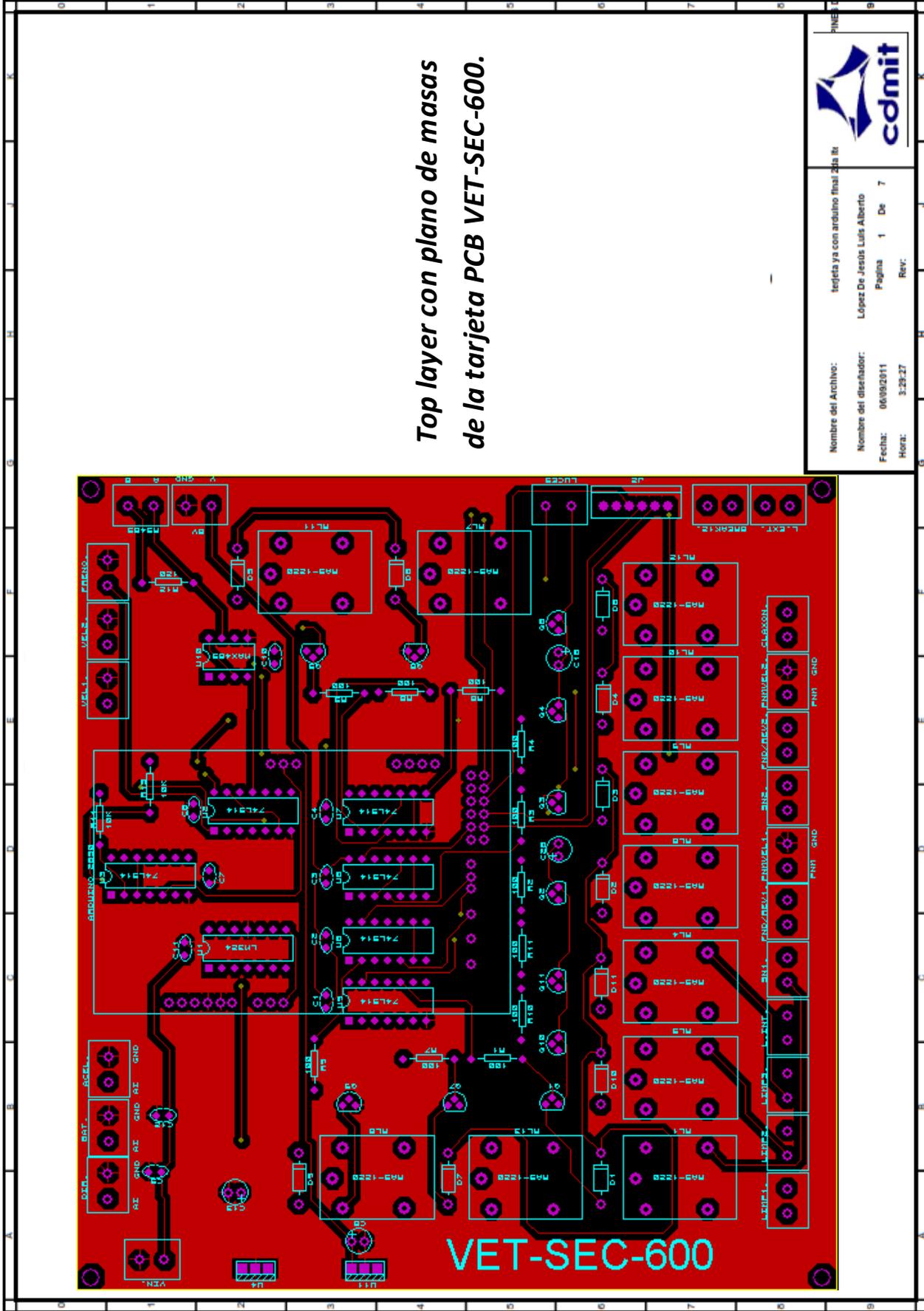




**Bottom layer con plano de
masas de la tarjeta PCB
VET-SEC-600.**



Nombre del Archivo: tarjeta ya con arduino final 2da It
 Nombre del diseñador: López De Jesús Luis Alberto
 Fecha: 06/09/2011
 Hora: 3:29:27
 Página 1 De 7
 Rev:



*Top layer con plano de masas
de la tarjeta PCB VET-SEC-600.*



Nombre del Archivo: tarjeta ya con arduino final 2da it
 Nombre del diseñador: López De Jesús Luis Alberto
 Fecha: 06/09/2011
 Hora: 3:29:27
 Página 1 De 7
 Rev: