



UNIVERSIDAD UTE

**FACULTAD DE CIENCIAS DE LA INGENIERÍA E
INDUSTRIAS**

CARRERA DE INGENIERÍA AUTOMOTRIZ

**SISTEMA DE ALINEACIÓN PARA VEHÍCULOS LIVIANOS
MEDIANTE ENCODER ÓPTICO.**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO AUTOMOTRIZ**

**AUTOR:
GERMÁN LEONARDO RUIZ PONCE**

**DIRECTOR:
ING. ELVIS PATRICIO GUALOTUÑA QUISHPE, MsC.**

Santo Domingo, enero, 2019

© Universidad UTE. 2019
Reservados todos los derechos de reproducción

FORMULARIO DE REGISTRO BIBLIOGRÁFICO

PROYECTO DE TITULACIÓN

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD:	171978445-4
APELLIDO Y NOMBRES:	Ruiz Ponce German Leonardo
DIRECCIÓN:	Coop.30 De junio Calle Patricia Pilar Y Placer Del Toachi.
EMAIL:	leonard_ruiz@hotmail.com
TELÉFONO FIJO:	3751-115
TELÉFONO MOVIL:	0982627827

DATOS DE LA OBRA	
TÍTULO:	Sistema de alineación para vehículos livianos mediante encoder óptico.
AUTOR O AUTORES:	German Leonardo Ruiz Ponce
FECHA DE ENTREGA DEL PROYECTO DE TITULACIÓN:	09 enero de 2019
DIRECTOR DEL PROYECTO DE TITULACIÓN:	Ing. Elvis Patricio Gualotuña Quisphe, <i>MSc.</i>
PROGRAMA	PREGRADO <input type="checkbox"/> X <input checked="" type="checkbox"/> POSGRADO <input type="checkbox"/>
TÍTULO POR EL QUE OPTA:	Ingeniero Automotriz
RESUMEN:	<p>El presente proyecto de investigación, tiene como finalidad realizar el diagnóstico y comprobación de la alineación de las ruedas, ésta es importante y fundamental para mantener la seguridad de los ocupantes, considerando que los neumáticos son los únicos elementos que están en contacto con la calzada, un mal reglaje de la geometría de las ruedas puede producir perdida de agarre con la calzada en las curvas, así como también perdida de la tracción por el desgaste irregular de los neumáticos.</p> <p>Mediante el diseño e implementación de un sistema de alineación utilizando encoders incrementales, desde una perspectiva nueva y diferente, se disminuye los tiempos del proceso de alineación, costos de operación, teniendo como objetivo un óptimo funcionamiento y fiabilidad para un buen desempeño en los tecnicentros express permitiendo incrementar el porcentaje de vehículos que ingresan diariamente al mantenimiento de la alineación, y brindando un mejor servicio al cliente.</p>
PALABRAS CLAVES:	Sistema de alineación, vehículos livianos, componentes electrónicos, circuitos eléctricos.

Se autoriza la publicación de este Proyecto de Titulación en el Repositorio Digital de la Institución.

A handwritten signature in blue ink, appearing to read 'German Leonardo Ruiz Ponce', written in a cursive style.

GERMAN LEONARDO RUIZ PONCE
C.C. 171978445-4

DECLARACIÓN Y AUTORIZACIÓN

Yo, **GERMÁN LEONARDO RUIZ PONCE**, CI 171978445-4 autor del proyecto titulado: **Sistema De Alineación Para Vehículos Livianos Mediante Encoder Óptico**; previo a la obtención del título de **INGENIERO AUTOMOTRIZ** en la Universidad UTE.

1. Declaro tener pleno conocimiento de la obligación que tienen las Instituciones de Educación Superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.
2. Autorizo a la BIBLIOTECA de la Universidad UTE a tener una copia del referido trabajo de graduación con el propósito de generar un Repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Santo Domingo, 09 de enero de 2019



GERMAN LEONARDO RUIZ PONCE
C.C: 171978445-4

CERTIFICACIÓN DEL TUTOR

En mi calidad de tutor, certifico que el presente trabajo de titulación que lleva por título **Sistema De Alineación Para Vehículos Livianos Mediante Encoder Óptico**, que para aspirar al título de **Ingeniero Automotriz** fue desarrollado por **German Leonardo Ruiz Ponce**, bajo mi dirección y supervisión, en la facultad de Ciencias de la Ingeniería e Industrias; y que dicho trabajo cumple con las condiciones requeridas para ser sometido a las evaluación respectiva de acuerdo a la normativa interna de la Universidad UTE.

A handwritten signature in blue ink, appearing to read 'Elvis Gualotuña Quishpe', written over a horizontal line.

Ing. Elvis Patricio Gualotuña Quishpe Msc.
DIRECTOR DEL TRABAJO
C. I: 1716876600

DEDICATORIA

A Dios por ser mi pilar, mi fuerza, mi fortaleza, por bendecirme en cada uno de las decisiones que he tomado, una de las más importantes es continuar con mis estudios y poder cumplir mis metas propuestas. A mis padres quienes me dieron la vida, a mi hermano ya que sin el apoyo de ellos no hubiese continuado con uno de los propósitos más importantes que es ser un profesional. Este logro, esta meta cumplida se la dedico a ellos por estar allí siempre dándome ánimos, fuerzas para seguir adelante con mis estudios por enseñarme que sin ningún esfuerzo o sacrificio no se obtiene nada. Gracias a ellos que me han inculcado buenos valores y principios, como es la responsabilidad y dedicación para alcanzar el éxito profesional.

Germán Leonardo Ruiz Ponce

AGRADECIMIENTO

Agradezco a Dios y a mis padres por tener la dicha de existir, por ser mis guías y derramar sobre mí sus bendiciones llenando mi vida de mucha felicidad. A mi hermano por que pude contar con él en todo momento, porque me dio la seguridad, la confianza en los momentos difíciles que pasé durante mi carrera universitaria por nunca dudar de mí, confiando en mi capacidad e inteligencia. A mis familiares que siempre estuvieron dando positivismo diciéndome no te rindas tú puedes, superarte nada es imposible en esta vida sigue adelante. Me permito agradecer a la Universidad y a los docentes ya que gracias a ellos compartieron sus conocimientos profesionales siendo clave muy importante en mi formación profesional, enseñándome que con perseverancia se llega al éxito.

Germán Leonardo Ruiz Ponce

ÍNDICE DE CONTENIDOS

	PÁGINAS
RESUMEN	1
1. INTRODUCCION	3
1.1. MARCO REFERENCIAL	4
1.1.1. PARÁMETROS DEL SISTEMA DE ALINEACIÓN DEL VEHÍCULO	4
1.1.1.1. Geometría de la dirección.....	4
1.1.1.2. Geometría de giro.....	4
1.1.1.3. Geometría de las ruedas	5
1.1.1.3.1. Angulo de caída de la rueda.....	5
1.1.1.3.2. Ángulo de salida del pivote.....	6
1.1.1.3.3. Ángulo Incluido	7
1.1.1.3.4. Ángulo de avance de la rueda.....	7
1.1.2. CONVERGENCIA	8
1.1.3. DIVERGENCIA	9
1.1.4. EQUIPOS DEL SISTEMA DE ALINEACIÓN	10
1.1.4.1. Alineadora Hunter.....	10
1.1.4.2. Sensores para la alineación.....	10
1.1.4.3. Elevador de cuatro postes para alineacion.....	10
1.1.5. SENSORES	11
1.1.5.1. Encoder	11
1.1.5.1.1. Aplicaciones de los encoders o sensores.....	11
1.1.5.2. Tipos de encoders	12
1.1.5.2.1. Encoder lineal.....	12
1.1.5.2.2. Encoder absoluto y Absoluto multi-vuelta.....	12
1.1.5.2.3. Encoder rotativo.	13
1.1.5.2.4. Encoder optico.....	14
1.1.5.2.5. Mecanismo del encoder optico.....	14
1.1.5.2.6. Encoder incremental.....	15
1.1.6. MÓDULOS PROGRAMABLES	16
1.1.7. SOFTWARE	16
1.1.7.1. Software labview.....	16
1.1.7.2. Software ni-visa	17
1.1.8. HARDWARE	17
1.1.8.1. Hardware arduino	18
2. METODOLOGÍA	19
2.1. DISEÑO DEL SISTEMA DE ALINEACIÓN	19
2.1.1. FUNCIÓN DE PINES DEL ENCODER INCREMENTAL	19
2.1.2. CARACTERÍSTICAS DEL ENCODER INCREMENTAL	20
2.1.2.1. Programación hardware arduino.....	20

2.1.2.2.	Programación de comandos, codificación, conversión de información	21
2.1.2.2.1.	Bloque 1. Asignación de pines	21
2.1.2.3.	Bloque 2. Void setup (inicialización de la comunicación)... ..	22
2.1.2.4.	Bloque 3 void loop (sentencia de repetición continua).....	23
2.1.2.4.1.	Contadores (sensor 1 Y sensor 2).....	23
2.1.2.5.	Bloque 4 caracteres de los contadores.....	24
2.2.	PROGRAMACIÓN DE SOFTWARE LABVIEW.	25
2.2.1.	BUCLE 1 CICLOS DE COMUNICACIÓN, RECEPCIÓN, TRANSFORMACIÓN Y PROYECCIÓN DE DATOS.....	25
2.2.2.	BUCLE 2 (CREACIÓN BASE DE DATOS CLIENTE).	26
2.2.3.	BUCLE 3 (CREACIÓN DE BOTÓN PARA MOSTRAR DATOS ... GUARDADOS DEL CLIENTE).....	27
2.3.	ESQUEMA DE CONEXIÓN DE LOS COMPONENTES DEL SISTEMA DE ALINEACIÓN.....	28
2.3.1.	SIMULACIÓN DE UNA ALINEACIÓN.....	28
3.	RESULTADOS Y DISCUSIÓN	31
3.1.	IMPLEMENTACIÓN DEL SISTEMA DE ALINEACIÓN.....	31
3.1.1.	CONEXIÓN DE LOS SENSORES AL HARDWARE ARDUINO.....	31
3.1.2.	SELECCIÓN DE LOS PLATOS DESLIZANTES.....	32
3.1.3.	MONTAJE DE LOS ENCODERS INCREMENTALES EN LOS PLATOS DESLIZANTES	32
3.1.3.1.	Características de los vehículos a alinear	34
3.2.	PRUEBAS AL SISTEMA	35
3.2.1.	DATOS OBTENIDOS EN LAS PRUEBAS REALIZADAS CON LOS PARÁMETROS ESPECÍFICOS SEGÚN EL FABRICANTE.....	38
3.3.	Análisis y pruebas	40
4.	CONCLUSIONES Y RECOMENDACIONES.....	42
4.1.	CONCLUSIONES.....	42
4.2.	RECOMENDACIONES	43
	BIBLIOGRAFÍA.....	44
	ANEXOS.....	47

ÍNDICE DE TABLAS

	PÁGINA
Tabla 1. Especificaciones técnicas del encoder incremental.....	20
Tabla 2. Asignación de pines.....	21
Tabla 3. Inicialización de comunicación.....	22
Tabla 4. Sentencia de repetición continua.....	23
Tabla 5. Inicialización de los sensores 1 y 2 (encoders incrementales).	23
Tabla 6. Tabla de verdad de los contadores en modo de comunicación.....	24
Tabla 7. Caracteres de los contadores (sensores).	24
Tabla 8. Características de Suzuki Gran Vitara SZ.	34
Tabla 9. Características de Chevrolet Luv D-Max.	34
Tabla 10. Características del Kia Rio Stylus.	35
Tabla 11. Alineación Suzuki Gran Vitara SZ Alineación normal	38
Tabla 12. Alineación Suzuki Gran Vitara SZ (Encoders incrementales).	39
Tabla 13. Alineación Chevrolet Luv D-Max Alineación normal	39
Tabla 14. Chevrolet Luv D-Max 2011 4x2 (Encoder incremental).	39
Tabla 15. Alineación normal Kia Rio Stylus Alineación normal.....	39
Tabla 16. Alineación normal Kia Rio Stylus (Encoder incremental).	39

ÍNDICE DE FIGURAS

	PÁGINA
Figura 1. Geometría de giro.....	4
Figura 2. Ángulo de caída.....	5
Figura 3. Ángulo de caída.....	5
Figura 4. Ángulo de Salida del pivote.	6
Figura 5. Ángulo incluido.	7
Figura 6. Angulo de avance	8
Figura 7. Convergencia, cámbber positivo y negativo.	8
Figura 8. Divergencia ruedas delanteras	9
Figura 9. Alineadora Hunter	10
Figura 10. Encoder de Cuadratura.....	11
Figura 11. Encoder absoluto y Absoluto multi-vuelta.....	13
Figura 12. Encoder rotativo.....	13
Figura 13. Encoder Óptico partes internas	14
Figura 14. Encoder óptico.....	14
Figura 15. Encoder incremental.....	15
Figura 16. Software LabVIEW.	17
Figura 17. Hardware Arduino uno.....	18
Figura 18. Encoder incremental.....	20
Figura 19. Selección del puerto de Arduino a la PC	21
Figura 20. Recepción, transformación y proyección de datos.	25
Figura 21. Creación de base de datos.	26
Figura 22. Creación de la base de datos para mostrar.	27
Figura 23. Esquema de conexión de los componentes del sistema.	28
Figura 24. Simulación de una alineación del vehículo.	28
Figura 25. Alineación del vehículo correcta.	29
Figura 26. Alineación guardada como documento txt.....	29
Figura 27. Revisión datos del cliente de una alineación anterior.	30
Figura 28. Conexión de sensores al Arduino.	31
Figura 29. Selección de los platos deslizantes.	32
Figura 30. Encapsulado de encoders incrementales.	33
Figura 31. Colocación de encoders incrementales en los platos deslizantes.....	33
Figura 32. Empernado de los encoders a los platos deslizantes.	34
Figura 33. El vehículo sube sobre los platos deslizantes.....	35
Figura 34. Encendido del sistema LabVIEW.....	36
Figura 35. Ingreso de datos del cliente.	36
Figura 36. Desviación de rueda derecha 2°.....	37
Figura 37. Corrección de la alineación de la rueda derecha.....	37
Figura 38. Guardar la alineación como documento txt.....	38

ÍNDICE DE ANEXOS

	PÁGINA
Anexo 1. Encuestas.....	47
Anexo 2. Software Arduino 1.8.6 para windows de 32bits.....	48
Anexo 3. Selección todas las opciones en esta pantalla y dar clic en Next.....	48
Anexo 4. Selección de la ruta de instalación recomendación dejar la que está establecida por defecto: c:\program files (x86)\Arduino..	49
Anexo 5. Extracción de los archivos de instalación.....	49
Anexo 6. Permisos necesarios para habilitar los puertos de comunicación entre el software y hardware de Arduino, así como las librerías correspondientes para la conexión por USB, dar clic en instalar.....	50
Anexo 7. Finalización de la instalación de todas las librerías correspondientes el software de Arduino quedara completamente instalado en el computador. dar clic en Close para finalizar la instalación.....	51
Anexo 8. Icono nuevo en el escritorio del computador.....	51
Anexo 9. Permisos acceso a los puertos a través del firewall de Windows.....	52
Anexo 10. El software de arduino en ejecución.....	52
Anexo 11. Software nivisa 1700 full.exe para windows de 32bits.....	53
Anexo 12. Aviso de la extracción de los archivos necesarios para la respectiva instalación.....	53
Anexo 13. Ruta donde se encuentra el archivo a instalar. Dar clic en unzip.....	54
Anexo 14. Archivos satisfactoriamente extraídos.....	54
Anexo 15. Instalación mencionando la introducción del software en cuestión. Dar clic en next.....	54
Anexo 16. Ruta de instalación se recomienda dejar la que viene por defecto.....	55
Anexo 17. Ruta de instalación se recomienda dejar la que viene por defecto.....	55
Anexo 18. Términos de licencia con Microsoft se deben aceptar para continuar con la instalación del mismo.....	56
Anexo 19. Asistente de instalación indica todos los cambios que se van a efectuar al momento de instalar el software ni visa. Dar clic en next.....	56
Anexo 20. Asistente de instalación indica todos los cambios que se van a efectuar al momento de instalar el software ni visa. Dar clic en next.....	57

Anexo 21.	Al reiniciar el computador el firewall de Windows preguntara si se habilita los puertos y establecer los respectivos permisos de red para crear excepciones de seguridad y que los mismos no sean bloqueados al momento de generar su ejecución. dar clic.....	58
Anexo 22.	Software labview (2017lv-wineng.exe) para windows de 32bits.	59
Anexo 23.	Software labview (2017lv-wineng.exe) para windows de 32bits.	59
Anexo 24.	Parámetros de registro.	60
Anexo 25.	Se procede a llenar los parámetros de licenciamiento del software.	60
Anexo 26.	Se procede a llenar los parámetros de licenciamiento del software.	61
Anexo 27.	Contrato de licencia del software.	61
Anexo 28.	Revisión de parámetros a instalar.	62
Anexo 29.	Instalación del VI Package Manager, es importante detallar que para que el proceso de instalación se ejecute con todas sus actualizaciones correspondientes se debe contar con una conexión a internet.	62
Anexo 30.	Se establecen los términos del contrato.	63
Anexo 31.	Se establece la ruta de instalación.	63
Anexo 32.	Instalación de la actualización correspondiente en Labview... ..	64
Anexo 33.	Inicio al software de labview a continuación se presenta la interfaz del software instalado.	64

RESUMEN

El presente proyecto de investigación, tiene como finalidad realizar el diagnóstico y comprobación de la alineación de las ruedas, ésta es importante y fundamental para mantener la seguridad de los ocupantes, considerando que los neumáticos son los únicos elementos que están en contacto con la calzada, un mal reglaje de la geometría de las ruedas puede producir pérdida de agarre con la calzada en las curvas, así como también pérdida de la tracción por el desgaste irregular de los neumáticos.

Mediante el diseño e implementación de un sistema de alineación utilizando encoders incrementales, desde una perspectiva nueva y diferente, se disminuye los tiempos del proceso de alineación, costos de operación, teniendo como objetivo un óptimo funcionamiento y fiabilidad para un buen desempeño en los tecnicentros express permitiendo incrementar el porcentaje de vehículos que ingresan diariamente al mantenimiento de la alineación, y brindando un mejor servicio al cliente. Para el diseño del sistema de alineación es indispensable la aplicación de sistemas electrónicos, informáticos como software y hardware programados por computadora.

Utilizando el hardware Arduino se realiza la programación para dos sensores que permita recibir las señales de entrada captadas por los encoders incrementales estas señales son procesadas, analizadas y enviadas al software LabVIEW que ingresan como caracteres y señales digitales, de manera que se puede observar e interpretar con facilidad dicha información a un sistema operativo Windows de 32 bits.

El sistema de alienación comprueba el ángulo de giro de convergencia y divergencia de las ruedas. Para su comprobación, el vehículo se lo ubica sobre los platos deslizantes, realizando un giro tanto a la derecha como a la izquierda, los valores están dados en un rango de 180 grados, cuando giramos a la derecha aumentamos los grados y cuando giramos a la izquierda disminuimos los grados, para realizar una alineación correctamente las dos ruedas deben estar centradas a 90 grados, es decir que si la rueda izquierda tiene 92 grados, la rueda derecha debe tener 88 grados, indicando una desviación angular hacia el interior formando una convergencia positiva, considerando que los valores deben estar dentro los parámetros permitidos por el fabricante, mínimo 0,1 y máximo de 0.3 grados para las dos ruedas.

El proceso de alineación se realiza en los equipos disponibles en el mercado y en el equipo diseñado con encoders incrementales utilizando tres

tipos de vehículos obteniendo varios resultados los mismos se comparan y analizan según valores establecidos por el fabricante determinado un mínimo margen de error demostrando que este nuevo sistema de alineación cumple con los objetivos propuestos.

1. INTRODUCCIÓN

1. INTRODUCCIÓN.

En la ciudad de Santo Domingo desde agosto del año 2017 se realiza la apertura de un Centro de Revisión Técnica Vehicular, mediante Resolución 005-CNC-2017, emitida por el Consejo Nacional de Competencias, la misma tiene como objetivo garantizar la seguridad mínima de los vehículos, además que cumplan con las normas técnicas y jurídicas para poder circular dentro y fuera de la ciudad los procedimientos técnico normalizados para determinar la aptitud de circulación de los vehículos se encuentra detallados en la norma INEN 2349.

En una vista realizada a la empresa Mapesa S.A la cual presta su servicio de taller express, su gerente propietario Fabricio Mayorga manifiesta el incremento de vehículos que necesitan el mantenimiento y diagnóstico del sistema de alineación, para cumplir con la demanda de los propietarios de vehículos que desean pasar la RTV, se desarrolla un sistema de alineación para vehículos livianos mediante encoders incrementales que permita hacer un diagnóstico de forma rápida y efectiva, sin la necesidad de colocar sensores en las ruedas, el diseño del presente proyecto de investigación, tiene como finalidad determinar los efectos del sistema de alineación mediante encoders incrementales, estos captan y envían señales digitales al software Arduino para la recepción de datos procediendo a la programación al hardware Arduino, una vez programado los valores son enviados al software LabVIEW, el cual recepta caracteres y señales digitales y en donde son transformados en números reales para su lectura y enviadas a un visor Windows donde se observa el estado actual de la alineación obteniendo resultados positivos, como en el consumo de combustible, el desgaste correcto de los neumáticos y de los elementos de la suspensión ya que es necesario e importante, evitando que el cliente tenga gastos innecesarios, y sobre guardando la salud, la integridad de sus ocupantes.

Actualmente los tecnicentros constan con sistemas de alineación con sensores para las cuatro ruedas. El objetivo general es diseñar un Sistema de Alineación para vehículos livianos mediante encoder incremental.

Objetivos específicos:

- Conocer cada uno de los parámetros del sistema de alineación.
- Obtener datos específicos de la alineación de al menos tres vehículos.
- Comparar cada uno de los datos o valores obtenidos en las pruebas realizadas con los parámetros específicos según el fabricante.

1.1. MARCO REFERENCIAL.

1.1.1. PARÁMETROS DEL SISTEMA DE ALINEACIÓN DEL VEHÍCULO.

1.1.1.1. Geometría de la dirección.

El sistema dirección consta de varios parámetros como, cotas y ángulos, que definen los parámetros, la posición de los elementos del sistema de dirección y del eje delantero y trasero, con respecto a la carrocería y al terreno.

Todos estos parámetros tienen unos objetivos prioritarios:

- Estabilidad en carretera.
- Facilidad de conducción.
- Evitar la fatiga mecánica en los elementos de la dirección.
- Disminuir el exceso consumo de combustible.
- Prolongar la vida de las ruedas.

1.1.1.2. Geometría de giro.

Las partes que dirigen la dirección deben actuar relativamente a la eliminación del frotamiento de las ruedas sobre el piso, cuando la trayectoria seguida por ellas no coincide con la impuesta por el sistema de dirección del vehículo. Es importante que todos los neumáticos se orienten en un viraje de manera que describan circunferencias de radios con el mismo centro (centro instantáneo de rotación) figura 1.

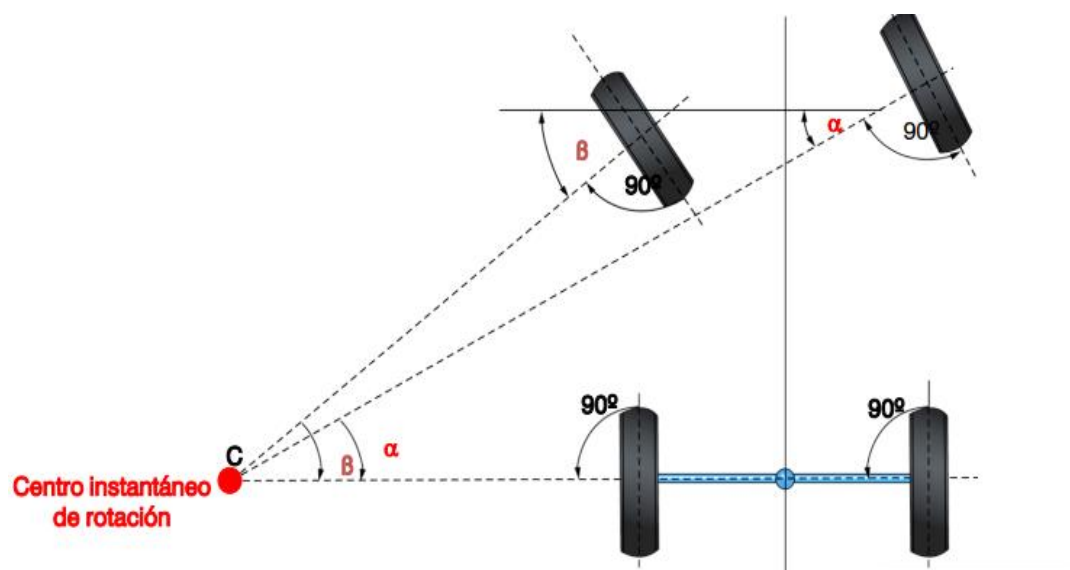


Figura 1. Geometría de giro.
(CarsMorbe 2012)

1.1.1.3. Geometría de las ruedas.

Se refiere a los componentes de los trenes que intervienen para situar las ruedas sobre el piso para lograr un rodamiento estable del vehículo sobre la calzada. La geometría de los trenes de rodaje comprende ciertos ángulos y parámetros, llamados cotas de la dirección:

1.1.1.3.1. Ángulo de caída de la rueda.

Es el ángulo comprendido entre el eje de simetría del neumático y la vertical que pasa por el centro de contacto de la rueda con la calzada figura 2. Llamada inclinación de la rueda o camber. Este un ángulo pequeño está comprendido aproximadamente entre 0° y 2° .

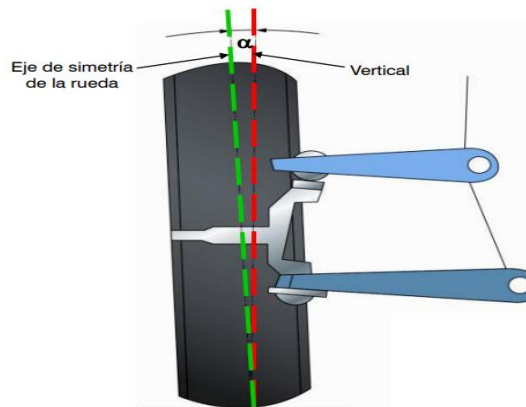


Figura 2. Ángulo de caída.
(CarsMorbe 2012).

Su función es desplazar el peso del vehículo sobre el eje, situado y apoyado en la parte inferior de la mangueta, disminuyendo así el empuje lateral de los cojinetes sobre los que se apoya la rueda (distancia B) figura 3. Evitando el desgaste excesivo de los neumáticos y rodamientos, reduciendo el esfuerzo de giro en la dirección. (Torres 2012).

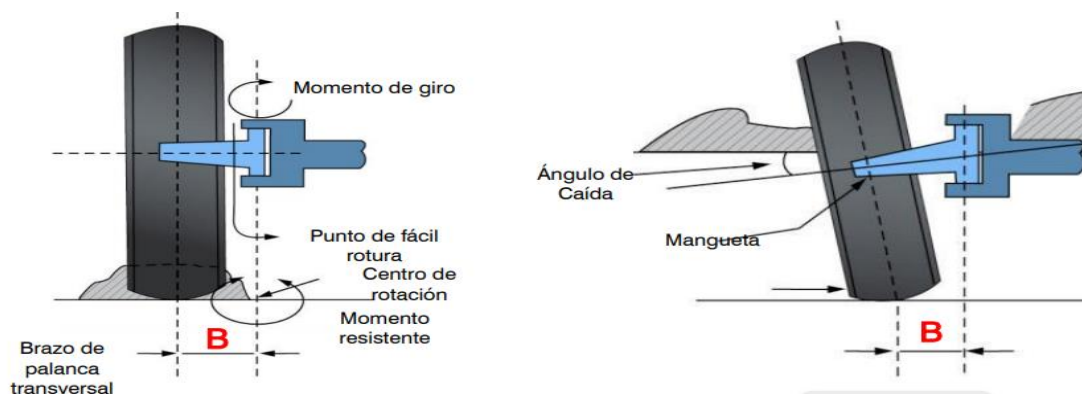


Figura 3. Ángulo de caída.
(CarsMorbe 2012)

Cuando el ángulo de caída está en mal estado, producen un desgaste desigual en la banda de rodadura del neumático y un mal comportamiento del vehículo.

Caída positiva:

- Desgaste más pronunciado en el borde exterior de la rueda.
- Desviación de la dirección hacia el lado que tenga la caída más pronunciada.

Caída negativa:

- Desgaste excesivo en el borde interior de las cubiertas.
- Desviación de la dirección hacia el lado de que tenga la caída menos pronunciada.

1.1.1.3.2. Ángulo de salida del pivote.

Ángulo sobre el cual las ruedas giran para orientarse tal como se muestra en la figura 4.

Tiene como función:

- Reducir el esfuerzo en las ruedas para realizar la orientación.
- Reducir el ángulo de caída del neumático para mejorar el desgaste.
- Ayuda la reversibilidad de la dirección después de realizar un viraje.
- Favorece el autocentrado de la dirección del vehículo.

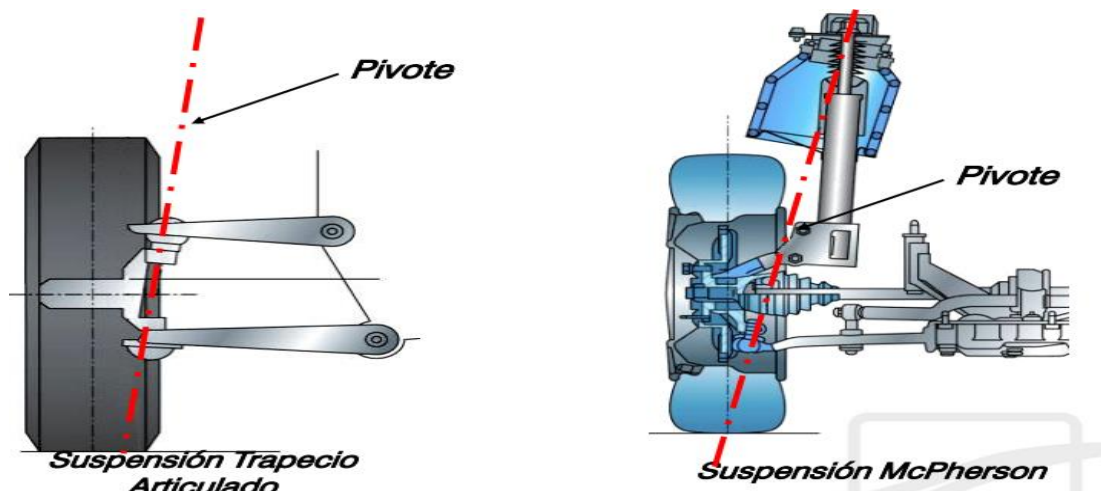


Figura 4. Ángulo de Salida del pivote.
(CarsMorbe 2012).

Cuando el ángulo del pivote está defectuoso estado presenta una variación en el ángulo de salida como:

- Dureza en el sistema de dirección.
- Reversibilidad excesiva.
- Reacciones de dirección provocando inestabilidad.
- Falta de reversibilidad.

1.1.1.3.3. Ángulo Incluido.

Es la suma de los ángulos de Caída y Salida figura 5.

Es decir, que la distancia B comprendida entre el punto de corte con el suelo del eje de rueda y el eje del pivote. Esta distancia se la conoce también como radio de rodadura o radio de pivotamiento.

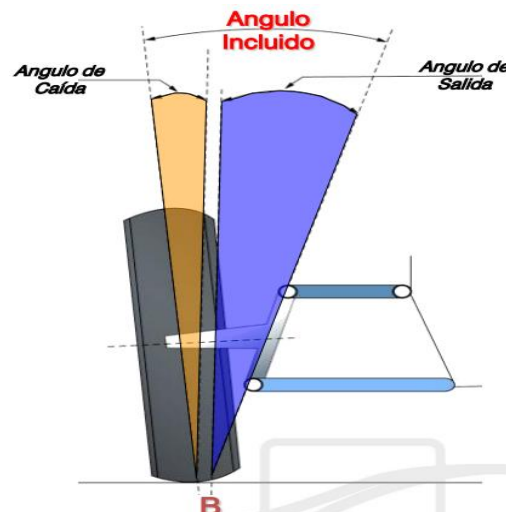


Figura 5. Ángulo incluído.
(CarsMorbe 2012).

1.1.1.3.4. Ángulo de avance de la rueda.

Este ángulo está comprendido entre 5° y 10° grados en vehículos con tracción trasera, y en vehículos de tracción delantera entre 0° y 3° figura 6. (Torres 2012).

Su función es:

- Mantener la dirección estable, con un autocentrado del vehículo.
- Favorece a la reversibilidad para que los neumáticos no tengan ningún inconveniente en regresar a la línea recta después hacer un viraje en una curva.

- Evita las vibraciones en las ruedas y repercusión en la dirección del vehículo.

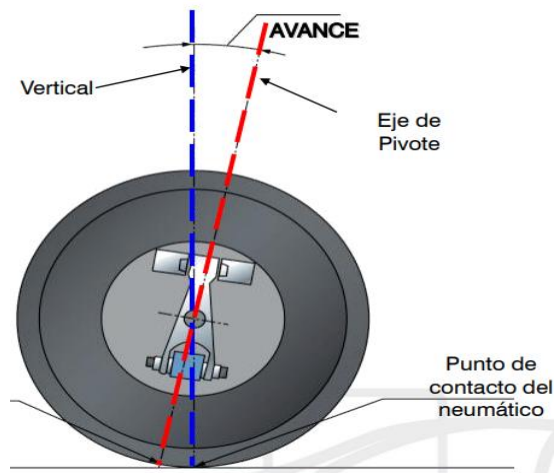


Figura 6. Ángulo de avance
(CarsMorbe 2012).

EL ángulo de Avance en mal estado presenta los siguientes defectos:

- Falta de reversibilidad en la dirección.
- Falta estabilidad de dirección.
- Dirección dura, inestable en virajes.
- Excesiva reversibilidad.
- Hala hacia el lado del ángulo más débil.
- Produce inestabilidad de la dirección.

1.1.2. CONVERGENCIA.

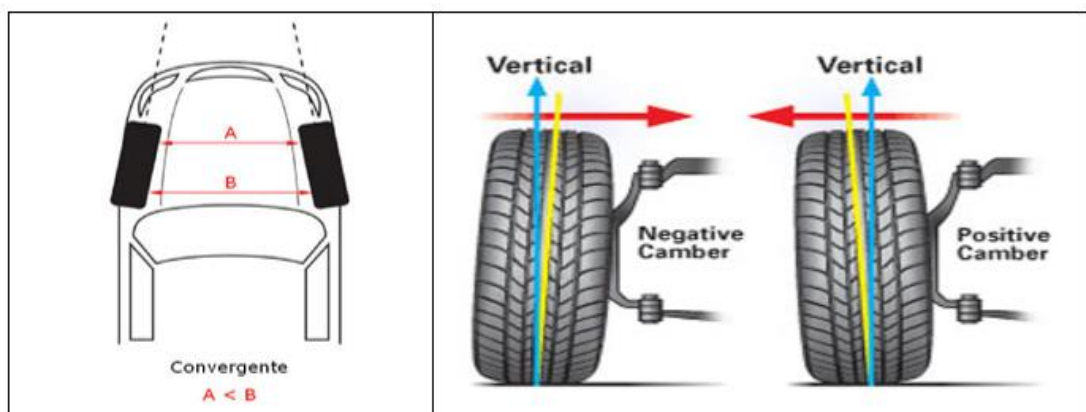


Figura 7. Convergencia, cámbler positivo y negativo.
(Actualidad motor 2012).

La convergencia en las ruedas delanteras tiene como objetivo uniformar y reducir el desgaste, limitando los deslizamientos causados en la superficie

de huella por los ángulos del cámbier (figura 7). En donde la convergencia hace rectilínea y sin deslizamientos la trayectoria del punto de contacto de los neumáticos en la calzada.

Tras varias pruebas que realizan por los fabricantes se establece el mejor valor para la convergencia prolongadas en carretera, a velocidad y cargas medias, en las que se obtiene el ángulo que determina los desgastes más uniformes de los neumáticos.

El valor se obtiene midiendo la distancia entre los bordes de las llantas a la altura de los extremos de los diámetros horizontales de las ruedas y hallando su diferencia. (Sánchez 2012).

1.1.3. DIVERGENCIA.

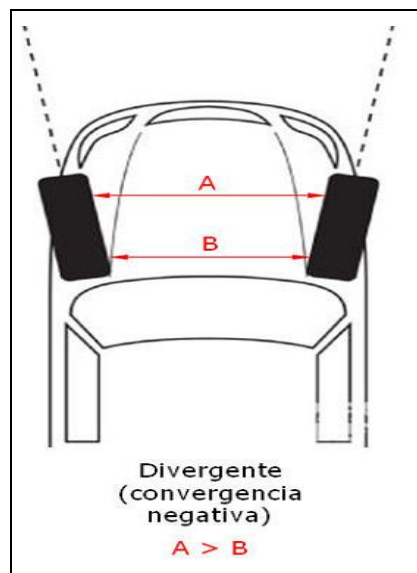


Figura 8. Divergencia ruedas delanteras (Actualidad motor 2012).

En cambio, la divergencia figura 8 son ángulos que existen entre las ruedas de un mismo eje en función de la dirección y del paralelismo entre las ruedas. Las ruedas son divergentes cuando sus líneas medias tienden a separarse, es decir son divergentes, en el sentido del movimiento.

La divergencia evita que la huella del neumático, bajo la acción del efecto de tracción y la flexibilidad de aquél, produzca un par que tiende a hacer converger las huellas de los neumáticos.

En este caso existiría un desgaste excesivo de la banda de rodadura en la parte exterior. En cambio, dando a las ruedas cierta divergencia, se obtiene el paralelismo de las huellas durante la marcha. Además, la divergencia, es

necesaria cuando las ruedas poseen un camber positivo, para reducir los deslizamientos del neumático. (Sánchez 2012).

1.1.4. EQUIPOS DEL SISTEMA DE ALINEACIÓN.

1.1.4.1. Alineadora Hunter.

Este equipo consta de una computadora con procesador Intel COre i3 3.06 Ghz (o mayor), un sistema Operativo Microsoft Windows 7 (64-bit), 4 Gb DDR3 Memoria RAM (o mayor) / Disco Duro de 120 Gb Estado Sólido (o mayor) y Software para alineación Hunter WinAlign Premium figura 9. Por lo que permite tener una mayor precisión toma de los datos obtenidos durante el proceso de la alineación. (Ferri 2016).



Figura 9. Alineadora Hunter
(R-Ferri 2016)

1.1.4.2. Sensores para la alineación.

Los sensores de las alineadoras actuales trabajan con sensores en las ruedas muy diferentes a las alineadoras anteriores estas toman sus datos o valores de manera inmediata por medio las señales infrarrojas sin la necesidad de compensación de los sensores en las cuatro ruedas.

1.1.4.3. Elevador de cuatro postes para alineación.

El elevador de cuatro, este debe estar calibrado y puesto a nivel los cuatro lados para la toma de datos. El vehículo va estar colocado en la rampa sobre los platos deslizantes al momento de la alineación figura 9. (Ferri 2016).

1.1.5. SENSORES.

1.1.5.1. Encoder.

Es un sensor que transforma un movimiento angular en una serie de impulsos digitales, pueden ser utilizados para controlar los desplazamientos de tipo angular o de tipo lineal, cuando son usados en conjunto con dispositivos mecánicos tales como engranajes, ruedas de medición o flechas de motores, para medir movimientos lineales, velocidad y posición. Las señales eléctricas de rotación pueden ser elaboradas mediante controles numéricos, controles lógicos programables (PLC), sistemas de control etc.

A la hora de controlar un sistema, se puede conocer la posición y velocidad de un elemento. Existen variadores que funcionan sin realimentación de posición, pero dichos variadores se utilizan básicamente para control de velocidad o par y un control preciso de la posición. (Jayashree 2016).

1.1.5.1.1. Aplicaciones de los encoders o sensores.

Los Encoders pueden ser utilizados en una gran variedad de aplicaciones. Actúan como transductores de retroalimentación para el control de la velocidad en motores, como sensores para medición, de corte y de posición. También como entrada para velocidad y controles de rango.

En la versión más 'simple' el encoder envía dos trenes de pulsos con un desfase conocido, los pulsos son contados conociendo la posición del motor y examinando qué fase es la primera, podremos saber la dirección, es el típico encoder de cuadratura A/A, B, /B figura 10. Normalmente una señal adicional nos informará de una marca única en el disco que servirá para definir el origen. (Argos 2013).

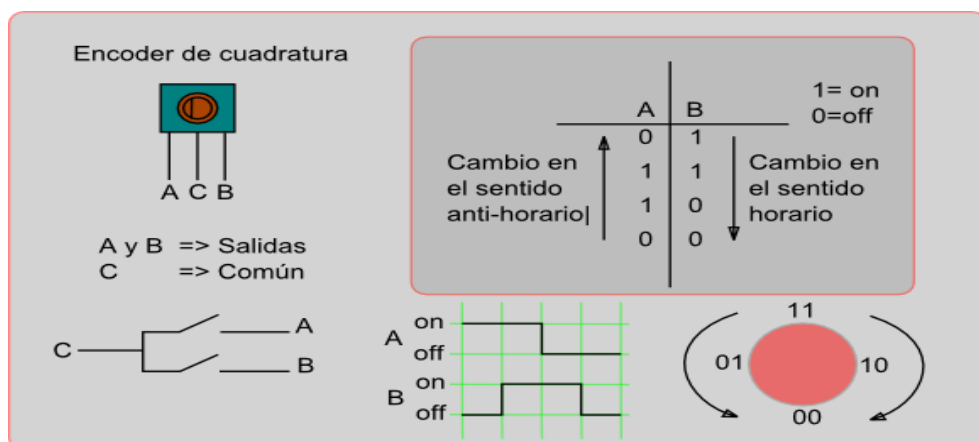


Figura 10. Encoder de Cuadratura.
(Argos 2013)

1.1.5.2. Tipos de encoders.

- Encoder Lineal.
- Encoder Absoluto Y Absoluto Multivuelta.
- Encoder Rotativo.
- Encoder Óptico.
- Encoder incremental.

1.1.5.2.1. Encoder lineal.

Este tipo de encoder o generador de pulsos; desde su existencia se da en los calibres digitales, en donde la medida se muestra en un display digital, detrás de la pieza móvil (corredera) está ubicado un encoder capacitivo que se encarga de leer, por interpolación, la distancia recorrida desde cero. Un encoder lineal se compone de un módulo fijo y otro móvil (el que se une a las partes móviles de la máquina con la cual se hará la interfaz). El módulo fijo contiene el sensor y la electrónica necesarios para detectar y medir el movimiento, y convertirlo en impulsos eléctricos inteligibles por otro circuito digital o analógico. (Nuria 2017).

1.1.5.2.2. Encoder absoluto y Absoluto multi-vuelta.

En el encoder absoluto figura 11, disco que contiene varias bandas dispuestas en forma de coronas circulares concéntricas, dispuestas de tal forma que en sentido radial el rotor queda dividido en sectores, con marcas opacas y transparentes codificadas en código Gray.

Un sistema con encoder absoluto, no es necesario realizar una secuencia, puesto que al poner en marcha la máquina y sin necesidad de movimiento alguno, el encoder del motor informará al controlador su posición.

El encoder absoluto necesita un mecanismo mecánico o electrónico para memorizar su posición. En realidad, se trata de un encoder absoluto-multivuelta, pues en muchas aplicaciones se necesita conocer no sólo la posición absoluta dentro de una vuelta, también en número de vueltas que ha dado el motor, para poder determinar su posición.



Figura 11. Encoder absoluto y Absoluto multi-vuelta.
(Nuria 2017)

Existen dos enfoques para trabajar un encoder absoluto multivuelta, el primero un sistema mecánico de discos y el segundo una batería capaz de mantener la posición durante tiempo determinado. La ventaja de un encoder absoluto no es necesario ningún movimiento para conocer la posición del motor, por lo que se gana tiempo en el arranque de la máquina y se evitan movimientos inconvenientes. En el caso de encoders absolutos multivuelta con batería, se debe ser consciente de que desconectar la batería implicará perder la posición absoluta. (Nuria 2017).

1.1.5.2.3. Encoder rotativo.

Es un sistema analógico figura 12, el voltaje que envía es inducido por dos espiras, que darán como resultado dos señales en lugar de pulsos digitales, examinando la señal podremos conocer la posición, y examinando el desfase la dirección.

Este tipo de encoder es muy sensible desde el punto de vista mecánico porque fuerzas excesivas perpendiculares al eje pueden afectar o dañar el disco lector, pero en sus versiones más avanzadas codifican los datos digitalmente por lo que la detección y corrección de errores de comunicación es posible a nivel de protocolo. (Jayashree 2016).

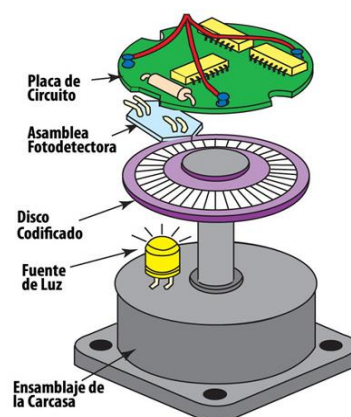


Figura 12. Encoder rotativo
(Encoder 2016).

1.1.5.2.4. Encoder óptico.

Este tipo de encoder normalmente consta de un disco fijado al eje giratorio figura 13, con una serie de marcas o secciones opacas y transparentes y un dispositivo óptico capaz de leer dichas marcas. La combinación de señales digitales permitirá la lectura de dichas marcas, nos dará como resultado señales de pulso que son leídas y esta información será codificada y transmitida para determinar el ángulo exacto del eje. (Rodríguez 2011).

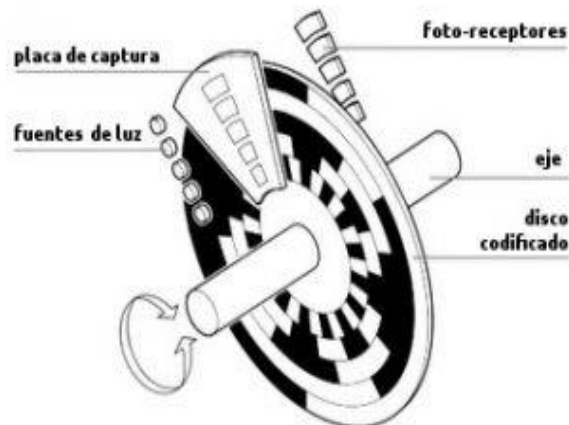


Figura 13. Encoder Óptico partes internas
(Nextia 2016).

1.1.5.2.5. Mecanismo del encoder óptico.

El encoder es un transductor, que mediante una señal eléctrica (normalmente un pulso o una señal senoidal), lo conectáramos mecánicamente con una rueda también permitiría medir distancias lineales. La programación se divide en tres partes claramente diferenciadas:

- Configuración del captador de entrada.
- Manejo de la interrupción.
- Detección de las señales digitales.

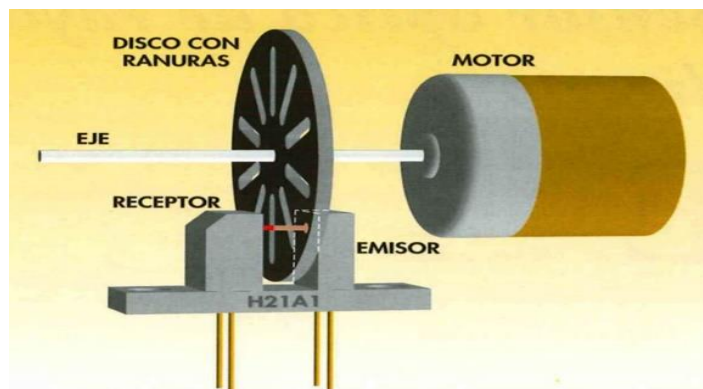


Figura 14. Encoder óptico.
(Maker 2017).

Un encoder figura 14, es un sensor electro-opto-mecánico que, unido a un eje, generan señales digitales en respuesta al movimiento.

Existen dos tipos, uno que responde a la rotación, y otro que responde al movimiento lineal. Cuando son usados en conjunto con dispositivos mecánicos tales como engranajes, ruedas de medición o flechas de motores, estos pueden ser utilizados para medir movimientos lineales, velocidad y posición

1.1.5.2.6. Encoder incremental.

El encoder incremental genera una serie pulsos digitales que codifican la rotación, ya sea en ciclo horario o antihorario, con 20 pulsos por ciclo aproximadamente. A diferencia de un potenciómetro, este tipo encoder no posee topes mecánicos, por lo que puede girar libremente. El encoder puede ser pulsado, de modo que se puede usar para seleccionar algo o medir posiciones. (Rodríguez 2011)

- **Características**

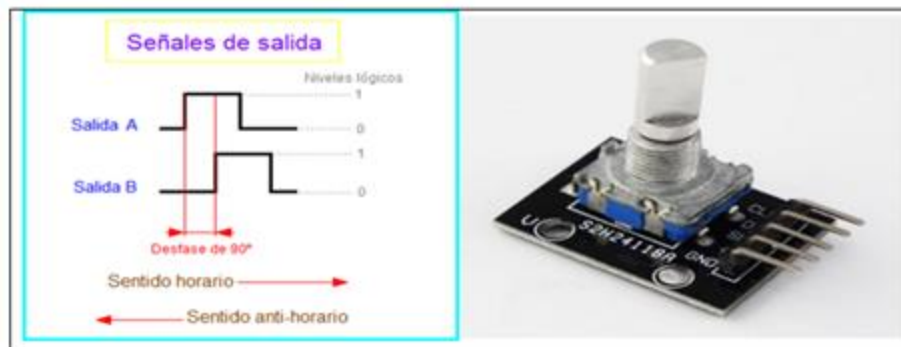


Figura 15. Encoder incremental.
(Tdrobotica 2014).

El encoder incremental figura 15, es un elemento que indica mediante una salida codificada la posición, al girar el eje se generan dos señales cuadradas llamadas canal A y B, si el pin común es puesto a masa y el canal A y B con resistencias de pull-up a Vcc, las señales A y B generadas están desfasadas 90 grados y su secuencia es (A,B):(1,0; 1,1; 01) en un sentido de giro, y (A,B):(0,1; 1,1; 10) en el otro. Esto se conoce como código Grey de 2 bits.

La clave está en detectar el flanco de bajada del canal A. En ese momento el sentido de la rotación viene determinados por el valor del canal B. La información o códigos pueden ser detectados por software y hardware.

1.1.6. MÓDULOS PROGRAMABLES.

Los módulos programables son diseñados para ajustar cualquier tipo parámetros, condición y tiempo de trabajo para el que se requiera.

En su interior no tiene ningún programa registrado, por lo que podemos ingresar los parámetros correspondientes para su inicio de funcionamiento. Con un módulo programable tenemos las siguientes ventajas:

- Se obtiene parámetros de funcionamiento en tiempo real.
- Control desde su inicio hasta el final de su trabajo.

Las configuraciones para estos módulos la hacemos mediante hardware y software para poder ajustar a nuestros requerimientos y poder realizar sin ningún inconveniente el trabajo requerido.

1.1.7. SOFTWARE.

El software es un soporte lógico de un sistema informático, el cual comprende un conjunto de los componentes lógicos necesarios para la realización de tareas específicas, en componentes físicos llamado hardware. La interacción entre el software y el hardware lo hace operativo un ordenador, el Software envía instrucciones para que el Hardware ejecuta, haciendo posible su funcionamiento. Los componentes lógicos incluyen aplicaciones informáticas, tales como procesador de texto, que permite al usuario realizar todas las tareas destinadas a la edición de textos, así como también el software del sistema. El sistema operativo, que básicamente permite al resto de los programas funcionar adecuadamente, facilitando también la interacción entre los componentes físicos y el resto de las aplicaciones, proporcionando una interfaz con el usuario.

Los proyectos de desarrollo de software se diferencian de los otros proyectos de ingeniería tradicional en la naturaleza lógica del producto software, dando como resultado un producto intangible. El software se desarrolla, mas no se fabrica. (Guzmán 2014).

1.1.7.1. Software labview.

LabVIEW figura 16 ofrece un enfoque de programación gráfica que ayuda a visualizar cada aspecto de una aplicación, incluyendo la configuración de hardware, datos de medidas y depuración. Esta visualización hace que sea más fácil integrar hardware de medidas, representar una lógica compleja en el diagrama, desarrollar algoritmos de análisis de datos y diseñar interfaces de usuario personalizadas. (Trujillo 2016).

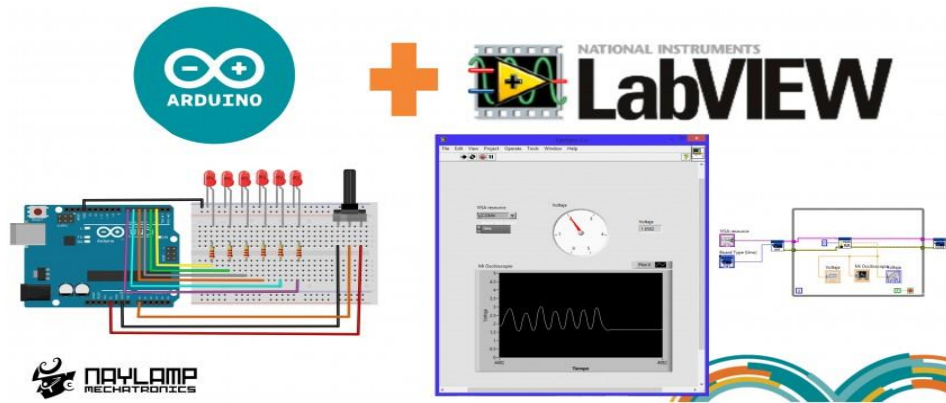


Figura 16. Software LabVIEW.
(Naylamp 2016).

Por medio de este software una vez que se realiza la comunicación de los parámetros correspondientes de los vehículos se podrá observar el estado de la alineación, permitiendo al técnico realizar el trabajo de corrección de convergencia y divergencia del vehículo.

1.1.7.2. Software ni-visa.

NI-VISA es la implementación de National Instruments del estándar de E/S VISA. Proporciona la interfaz de programación entre el hardware y los entornos de desarrollo de aplicaciones como NI LabVIEW, LabWindows.

Las licencias de desarrollo para NI-VISA están incluidas con los productos de National Instruments como entornos de desarrollo y hardware. El software NI-VISA proporciona la interfaz de programación del hardware Arduino, así como el desarrollo de la aplicación en el software LabVIEW. (Trujillo 2016).

1.1.8. HARDWARE.

En informática se conoce como hardware a las partes físicas tangibles de un sistema informático, así como sus componentes eléctricos, electrónicos, electromecánicos y mecánicos. Los cables, así como los gabinetes o cajas, y cualquier otro elemento físico involucrado, componen el hardware o soporte físico. Para el funcionamiento correcto del hardware se requiere de un software específico para que pueda operar adecuadamente al conectarse a una computadora, y le permita ser identificado y reconocido por el sistema operativo y, al contrario, para que un software pueda ejecutarse de manera óptima, requerirá de un hardware lo suficientemente poderoso que le permita ejecutarse con rapidez y eficiencia. (Gaceta 2013).

1.1.8.1. Hardware arduino.

El entorno de desarrollo integrado de Arduino, o del software Arduino (IDE), contiene un editor de texto para el ingreso de códigos, un área de mensajes, una consola de texto, una barra de herramientas para funciones comunes y varios menús. Se conecta al hardware Arduino para cargar programas y comunicarse con ellos. El hardware Arduino es un elemento con el que podemos trabajar libremente ya que tiene aplicación con los softwares ya mencionados cuyos caracteres son compatibles para realizar el proyecto.

El Arduino figura 17 consta de 14 pines digitales de entrada y salida, 6 se pueden ser utilizados como salidas PWM, y 6 entradas analógicas, una conexión USB, un conector de alimentación, un botón de reinicio y más. Contiene todo lo necesario para soportar el microcontrolador; simplemente conectarlo a una computadora con un cable USB. Arduino tiene una plataforma de prototipos electrónica de código abierto.

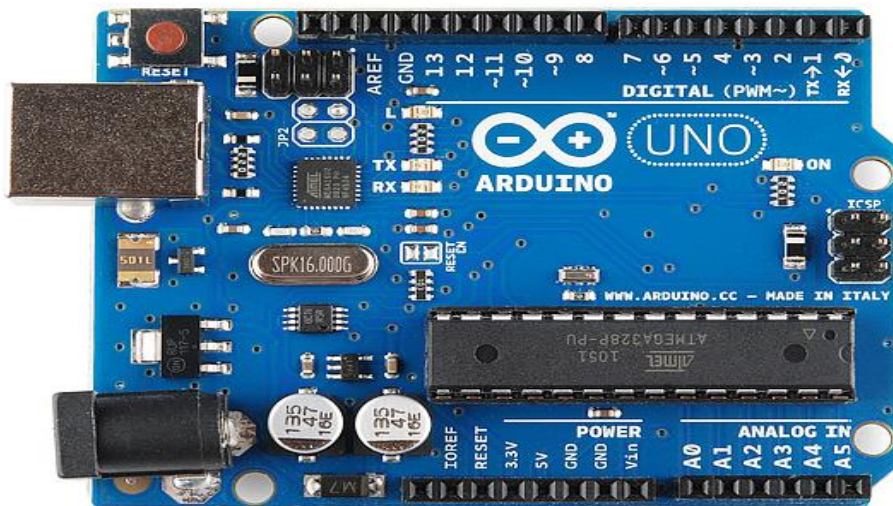


Figura 17. Hardware Arduino uno.
(Arduino 2015).

2. METODOLOGÍA

2. METODOLOGÍA.

Los métodos que se emplean para el presente proyecto es de manera explicativa, mediante la recolección de información se describe cada uno de los fenómenos que intervienen en la alineación, cuantitativo el cual se realiza por medio de encuestas a diez técnicos de diferentes talleres express para conocer el beneficio de este proyecto, analítico mediante este método conocemos los beneficios que recibirán los técnicos, inductivo el cual se realiza interrogando a los propietarios de los diferentes talleres express y conociendo la necesidad de implementar este sistema de alineación, deductivo conociendo los inconvenientes que tienen los talleres express sabremos la importancia de realizar la alineación mediante un encoder incremental ya que su funcionamiento será óptimo y factible para los talleres express y experimental mediante este método se tomara en cuenta el tiempo en que se realiza la alineación del vehículo mediante el método tradicional, así como cada uno de los procesos que realiza el técnico desde que el vehículo es subido a la rampa, para luego proceder a colocar los sensores en las cuatro ruedas, ingresar datos del vehículo, compensar sensores y por ultimo realizar el giro para lo que es convergencia y divergencia para proceder a la alineación del vehículo, tomando en cuenta el tiempo y cada uno de los procesos que realizo el técnico, se tomara así mismo el tiempo y los procesos que tardara el técnico en realizar la alineación con este nuevo sistema de alineación mediante encoder óptico para lo cual el vehículo será subido a la rampa, para luego ingresar datos del vehículo, procediendo a realizar el giro de convergencia y divergencia, procediendo a la alineación del vehículo.

2.1. DISEÑO DEL SISTEMA DE ALINEACIÓN.

El diseño del sistema de alineación está constituido por las siguientes etapas, función de los pines del encoder incremental, programación de (comandos, codificación, conversión de información), ciclos de comunicación, recepción y proyección de información entre la PC y el software LabVIEW, esquema de conexión de los componentes del sistema de alineación, una vez desarrollada cada una de las etapas se realiza una simulación de una alineación a un vehículo.

2.1.1. FUNCIÓN DE PINES DEL ENCODER INCREMENTAL.

Conociendo cada uno de los parámetros importantes sobre el funcionamiento del encoder incremental y su utilización. A continuación, se procede a conocer la función de cada uno de los pines del encoder incremental y sus características figura 18.

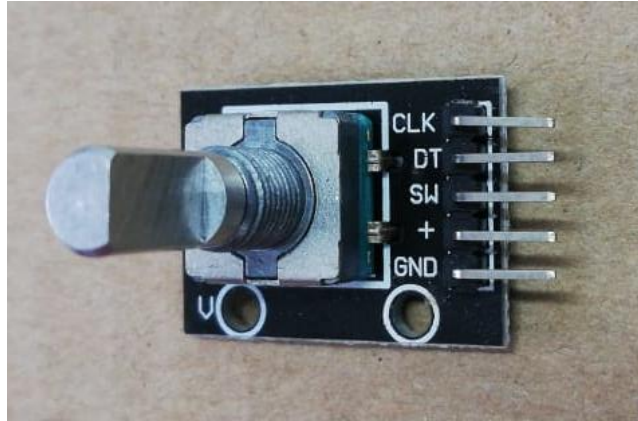


Figura 18. Encoder incremental

El encoder incremental envía señales el cual envía los datos en sentido horario y anti horario. Para identificar cada uno de los pines vamos a tener en cuenta que el pin SW es el pin del interruptor y sube cuando se empuja el codificador giratorio. Los pines CLK (reloj) y DT (datos) son la forma de leer la dirección en que se ha girado el codificador. El pin CLK se baja cuando se ha girado el codificador y el alfiler DT muestra cuál fue su giro, bajo en el sentido de las agujas del reloj, alto en sentido contrario a las agujas del reloj tal como se muestra en la tabla 1.

2.1.2. CARACTERÍSTICAS DEL ENCODER INCREMENTAL.

Tabla 1. Especificaciones técnicas del encoder incremental

Modelo:	KY-040
Tipo:	Encoder incremental
Ciclos por revolución (CPR):	20
voltaje de funcionamiento:	0 - 5V
Material:	PCB + Brass
Peso:	10g
Tamaño:	32 x 19 x 30 mm

(Tdrobotica 2014).

2.1.2.1. Programación hardware arduino.

Para tener comunicación con el sistema operativo, conectamos el Hardware Arduino a la PC, se escoge el puerto o canal con el que vamos a trabajar según lo reconozca en este caso COM 6 figura 19.

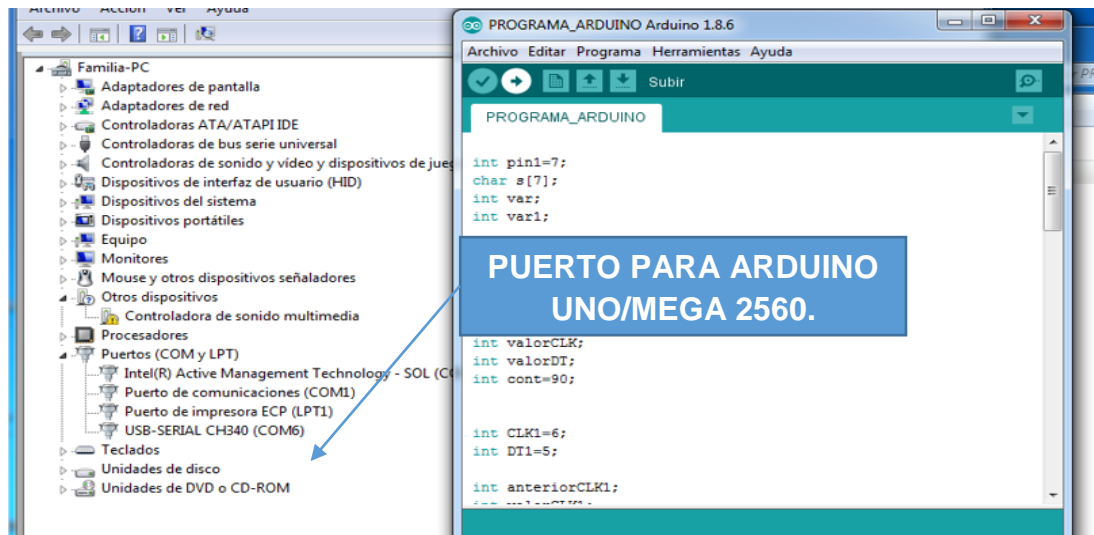


Figura 19. Selección del puerto de Arduino a la PC

2.1.2.2. Programación de comandos, codificación, conversión de información.

La función del software es recopilar datos o señales digitales y enviar para su procesamiento. Para la programación del hardware Arduino se establece bloques para dar la asignación de pines a cada uno de los sensores o encoders incrementales, inicialización de la comunicación, sentencia de repetición y caracteres de los contadores.

2.1.2.2.1. Bloque 1. Asignación de pines

Tabla 1. Asignación de pines

Sensor 1	Sensor 2
char s[7]; int var;	int var1;
int CLK=8; int DT=9; //int led=11;	int CLK1=6; int DT1=5;
int anteriorCLK; int valorCLK; int valorDT; int cont=90;	int anteriorCLK1; int valorCLK1; int valorDT1; int cont1=90;

En el primer bloque tabla 2, se establece un vector o carácter de siete espacios es decir ocho contando el cero. Una (variable Int.) para el sensor 1, y otra (variable Int1) para el sensor 2, estas dos variables son para almacenar datos de información que envían los dos sensores. Un Int. CLK para lo cual se declara la variable CLK 8 para el pin 8 y un Int. DT 9 para el pin 9 de la placa o del Arduino, es decir la señal CLK o información que sale del encoder incremental o sensor 1 va al pin 8 y el DT va al pin 9 de la placa del Arduino.

Luego en el siguiente espacio se declara variables a nombre del primer encoder incremental, es decir para recibir la información que envíe el sensor uno.

El mismo proceso se realiza al segundo encoder incremental, en este caso se va a dar el nombre de Int. CLK 1 que va a ir conectado al pin 6 y el Int. DT1, va a ir conectado al pin 5 de la placa del Arduino y así mismo en el siguiente espacio se declara variables para el segundo sensor, para así recibir toda la información que envíe el sensor dos.

Dentro de este bloque se establece unos contadores tanto para el sensor uno y dos que son iguales a noventa grados, lo cual significa que los sensores están en cero grados siempre que se encienda el sistema.

En programación existen códigos llamados **(ASC)** representados en letras y estas tienen sus valores correspondientes como por ejemplo: A=1, B=13, @=37 es decir números enteros tienen valores de las letras, ya que en códigos **(ASC)** no se trabaja con números enteros directamente y por medio de estos códigos se pueden realizar, dando la información de la posición en la cual se encuentran los sensores uno y dos por esta razón siempre se inicializa en noventa que es igual a cero.

2.1.2.3. Bloque 2. Void setup (inicialización de la comunicación).

Tabla 2. Inicialización de comunicación.

Sensor 1 / Sensor 2 (void setup)
<pre>Serial.begin(9600); //pinMode(pin1,INPUT); pinMode(CLK,INPUT); pinMode(DT,INPUT); anteriorCLK=digitalRead(CLK); pinMode(CLK1,INPUT); pinMode(DT1,INPUT); anteriorCLK1=digitalRead(CLK1);</pre>

Una vez establecida la asignación de pines, en este segundo bloque Tabla 3, se establece una comunicación serial con la PC, en la cual se inicializa los pines de entrada de los dos CLK y los dos DT de cada uno de los sensores (encoders incrementales).

En este bloque void setup de inicialización de comunicación, la velocidad en la que va a trabajar el sistema que es de 9600 baudios, así como la asignación del número de pinmode que va a trabajar como ingreso tanto del (CLK - CLK1), como él (DT- DT 1) con estos parámetros se agrega un anterior tanto para el CLK del sensor 1, como para el CLK1 del sensor 2 en

donde se declara como lectura digital de todos los datos o valores tomados, haciendo como entrada del CLK (sensor 1) Y CLK1 (sensor 2).

2.1.2.4. Bloque 3 void loop (sentencia de repetición continua).

Tabla 3. Sentencia de repetición continua.

Sensor 1 (void loop)	Aensor 2 (void loop)
<pre> valorCLK=digitalRead(CLK); if(valorCLK != anteriorCLK){ valorDT=digitalRead(DT); if(valorDT==valorCLK){//giro derecha cont++; }else{//giro izquierda cont--; } } anteriorCLK=valorCLK; </pre>	<pre> valorCLK1=digitalRead(CLK1); if(valorCLK1 != anteriorCLK1){ valorDT1=digitalRead(DT1); if(valorDT1==valorCLK1){//giro derecha cont1++; }else{//giro izquierda cont1--; } } anteriorCLK1=valorCLK1; </pre>

En este bloque se establece la sentencia de repetición continua Tabla 4, es decir que los valores que envíe cada uno de los sensores o encoders incrementales (CLK y CLK1) serán guardados como una lectura en Arduino. Cuando el valor que envíe CLK o CLK1 es distinto a su valor anterior, el cual ya está estipulado que cero grados es igual a noventa grados, entonces si el valor que ingresa es diferente, el sistema pregunta por el valor tanto de DT como DT1 y hace una lectura de los dos (DT), es decir si el valor de los (DT) son iguales a los valores de los (CLK – CLK1) se suma los grados de desviación caso contrario se le resta tabla 5, para lo cual está dado los contadores a su inicialización en 90 grados.

Tabla 4. Inicialización de los sensores 1 y 2 (encoders incrementales).



2.1.2.4.1. Contadores (sensor 1 Y sensor 2).

Estos son los encargados de conocer la posición de los dos sensores (encoders incrementales). Si estos dos sensores están en la posición cero quiere decir que en el espacio uno da como dato o valor cero.

Cuando los contadores o sensores envían valores iguales quiere decir que el vehículo tiene las ruedas rectas es decir está alineado tabla 6. Pero en el

caso de que cualquiera de los contadores tenga mayor o menor desviación de ángulo en las ruedas quiere decir que el vehículo esta desalineado.

Tabla 5.Tabla de verdad de los contadores en modo de comunicación.

Contador 1 (Sensor 1)	Contador 2 (Sensor 2)	Llanta Derecha	Llanta Izquierda	Estados de Contadores	Represent. de Estados
+	+	OK	OK	1	0
+	-	OK	NO	2	2
-	+	NO	OK	3	3
-	-	OK	OK	1	0

2.1.2.5. Bloque 4 caracteres de los contadores

Tabla 6. Caracteres de los contadores (sensores).

Sensor 1 / Sensor 2 (Caracteres de los contadores)

```
s[0]='0';

if(cont==cont1)
{s[1]='0'; }//llanta D=ok l=ok

if(cont>cont1)
{s[1]='2'; }//llanta D=ok l=no

if(cont<cont1)
{s[1]='3'; }//llanta D=no l=ok

s[2]=char(cont);
s[3]=char(cont1);
//Serial.write(s);
Serial.println(s);
//Serial.println(char(cont1));
//Serial.println(cont1);
delay(10);
```

Por medio de estos caracteres tabla 7, al contador uno y dos, serán forzados a ser caracteres para conocer la información de la rueda del vehículo tanto de la izquierda como la de la derecha, estos datos se imprimen con un salto de línea en la cual va a tener una variable que es (S), la cual está dada con un vector de siete posiciones en total serian ocho posiciones ya que se incluye el cero y todos estos valores son enviados hacia el puerto serial y todos los valores que se obtiene de los sensores uno y dos son enviados a la PC.

Al final de la programación se inserta un (**Delay**) que es un espacio de tiempo de diez milisegundos, para que la comunicación serial haga una pausa y se establezca ya que el sistema está trabajando a una velocidad de 9600 baudios, en el caso de no hacer esta pausa tan importante todos los datos que entrarían al sistema tienden a chocarse, saturándolo y por ende se perdería la comunicación.

2.2. PROGRAMACIÓN DE SOFTWARE LABVIEW.

Para la programación de LabVIEW se lleva a cabo los siguientes procesos, bucle de ciclos de (comunicación, recepción y proyección de datos entre la PC y el software LabVIEW), bucle de base de datos del cliente, bucle de creación del botón para mostrar datos guardados del cliente.

2.2.1. BUCLE 1 CICLOS DE COMUNICACIÓN, RECEPCIÓN, TRANSFORMACIÓN Y PROYECCIÓN DE DATOS

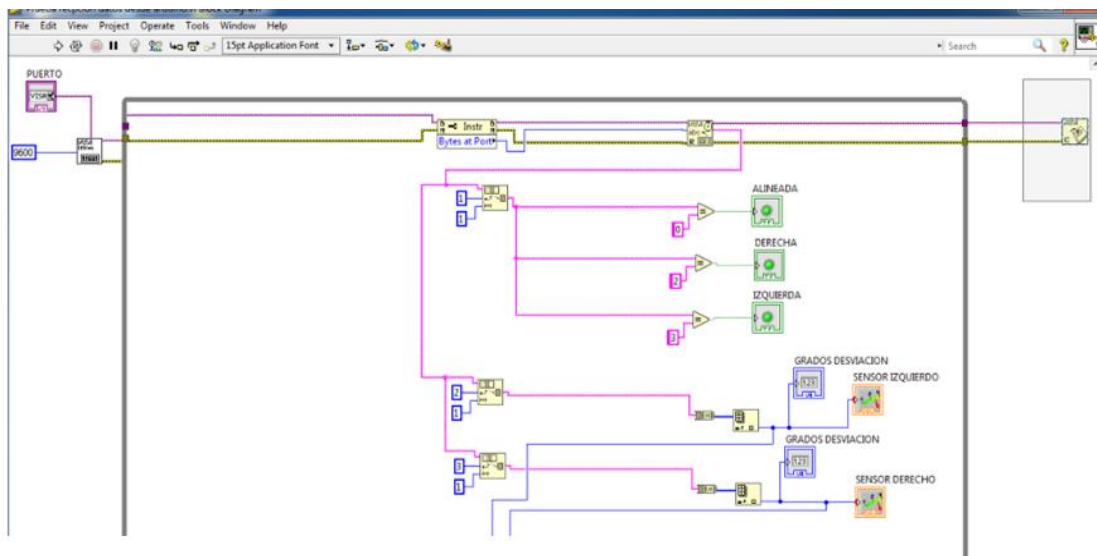


Figura 20. Recepción, transformación y proyección de datos.

Se desarrolla la programación figura 20, se crea un bucle de verdadero o falso siendo el principal, un bloque serial para para que haya una comunicación serial, en bloque serial se crea también un selector de puertos y un selector de velocidades que va hacer de 9600 baudios la misma que la del software Arduino y están ubicadas fuera del bucle principal, estos elementos afectan a tres cosas importantes dentro de todo el sistema que son:

- La velocidad de procesamiento de la PC.
- La velocidad de procesamiento del Arduino.
- El reloj de la PC y del Arduino.

Los ciclos de programación son distintos, para que no haya ninguna interferencia en cada uno de los procesos se inserta al final del bucle principal un **delay** para producir una pausa y estabilizar las líneas de comunicación. El bloque serial inicia la comunicación, se crea un detector de puertos (bytes at port) encargado de leer los bytes de entrada, seguido a un bloque visa de lectura su conexión termina en un bloque visa (close) encargado de cerrar la comunicación.

Siguiendo la línea de comunicación, un bloque string subset lee los datos que envía el hardware Arduino, envía una serie de cadena o caracteres (S), LabVIEW recoge la información y pregunta en el espacio (1) del carácter (S) o espacio de inicialización del (bits) 1 string subset, en donde labview solo envía datos por sentencia de bits que son (7) y en un byte envía una cadena de caracteres (S), la cadena de caracteres es (S0-S1-S2-S3-S4-S5-S6-S7).

Entonces en el espacio uno (Bloque-String Subset) de byte 1 pregunta, si es igual a cero compara los datos receptados son correctos enciende una luz verde que significa que el vehículo está alineado, este proceso lo realiza tanto para la rueda derecha como para la izquierda, estos valores llegan como string carácter, a la misma línea de bytes donde el contador uno envía datos y pregunta, en el espacio dos (Bloque-String Subset) del byte (1) que valor tengo pero como no se puede visualizar ya que son valores string, es decir no llegan números, estos valores son transformados en bytes por medio del bloque (String To Bytes Array), donde son acoplados o convertidos por medio del bloque (Index Array) a la cantidad indicada de números enteros, se imprime o proyecta en un visor Windows los grados de desviación de las ruedas del vehículo, esto se puede visualizar con un indicador de datos reales y un medidor de números tanto para el sensor uno como el sensor dos. Lo mismo es en el espacio tres (Bloque-String Subset) del byte (1) el contador dos envía los datos o valores, se realiza el mismo proceso que en el segundo espacio del byte (1), y en cada uno de los procesos se realizan individualmente por cada uno de los sensores y observando que valor está enviando cada uno de los sensores.

2.2.2. BUCLE 2 (CREACION BASE DE DATOS CLIENTE).

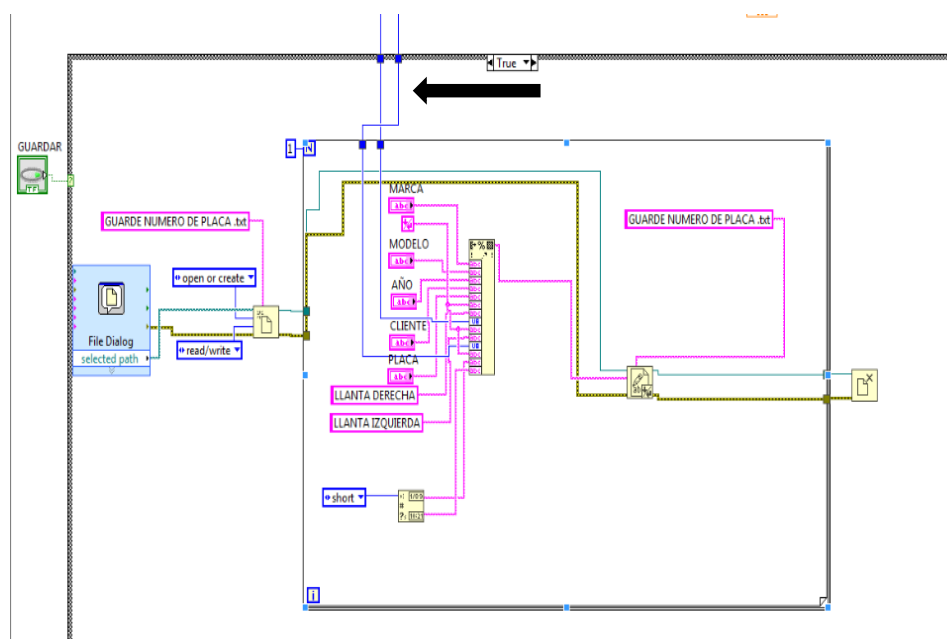


Figura 21. Creación de base de datos.

Se inserta el botón (**Guardar**) para las alineaciones realizadas de los vehículos y conectado al bucle principal, se agrega un segundo bucle de verdadero o falso, dos líneas de color azul figura 21, están conectadas en a la entrada de la carpeta (**Form Intro String**) como (**UB**) que es un dato entero el cual ya está dado en el bucle anterior, una carpeta llamada también generador de archivos en la cual tiene la función de abrir o crear, de leer o escribir, que es donde va a guardar los números de las placas de los diferentes vehículos, trabaja conjuntamente con una carpeta en la que se forma un dialogo (**File Dialog**), es importante crear el generador de archivos si no el sistema no va a saber que crear, que leer, que inicializar, esto es lo principal para la actualización de la base de datos. A partir de la inicialización se agrega una carpeta (**Form Intro String**), la cual tiene la función de realizar una repetición solo una vez para cada acción realizada. Dentro de la carpeta (**Form Intro String**), la cual es una caja de contextos se inserta las ventanas para agregar la marca, modelo, año, cliente, placa, del vehículo esto se lo realiza manualmente para guardar, en cuanto a los valores que den los encoders incrementales o sensores uno y dos de la llanta izquierda y llanta derecha (posición en grados de desviación) será guardado automáticamente y todo los datos se guardan en bus de datos y se agrega un bloque en la parte inferior de la misma carpeta, el cual está asignado solo para guardar la fecha y la hora de que fue realizado el trabajo.

A la salida de la caja de contextos creo una carpeta (write to text file) en esta se va guardar las alineaciones realizadas a los vehículos con el número de placa, como archivo (**TXT**). Para finalizar se agrega una carpeta (**close file**) está carpeta tiene la función de informar si hay algún error en el sistema, no se escribió nada o algún otro error.

2.2.3. BUCLE 3 (CREACION DE BOTON PARA MOSTRAR DATOS GUARDADOS DEL CLIENTE)

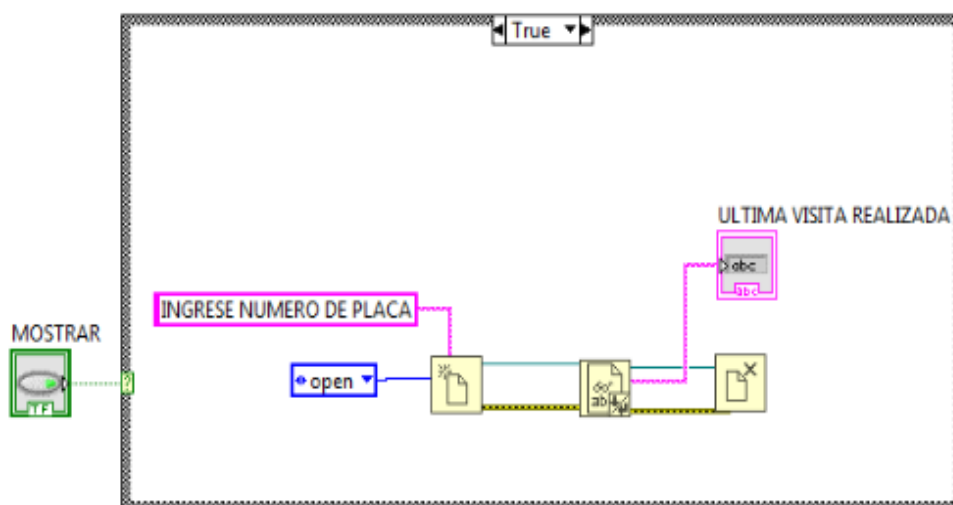


Figura 22. Creación de la base de datos para mostrar.

El tercer bucle figura 22, es independiente no lleva ninguna línea del circuito, mostrara solo los datos guardados del cliente en el bus de datos, se agrega el botón (**Mostrar**), conectado a la línea del bucle de verdadero o falso en el interna del bucle se agrega una carpeta o generador de archivos (**Open/write**) con el nombre (**Ingrese Número De Placa**) en este caso solo para abrir el archivo guardado de las alineaciones de los vehículos que se han realizado y se buscaran solo con el número de placa, guardando el archivo como documento (**TXT**) por ende no es necesario crear un dialogo, a partir de allí se crea un bloque de lectura (**Read Form Tex File**) y tendrá la función de leer el número de placa ingresado en el bus de datos y proyectarla en una ventana (**ABC**), la cual está como (**Última Visita Realizada**).

2.3. ESQUEMA DE CONEXIÓN DE LOS COMPONENTES DEL SISTEMA DE ALINEACIÓN.

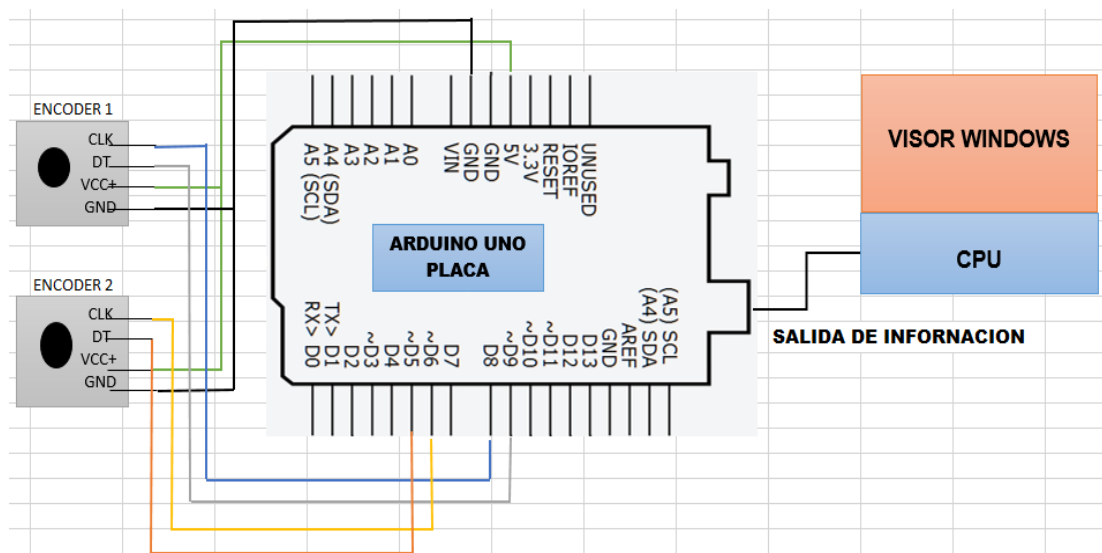


Figura 23. Esquema de conexión de los componentes del sistema.

2.3.1. SIMULACIÓN DE UNA ALINEACIÓN.



Figura 24. Simulación de una alineación del vehículo.

Se enciende el sistema y escogiendo el puerto del Arduino con el que se va a trabajar puede ser el puerto (**COM 3, COM4 O COM 6**), una vez que encendido el sistema se ingresa los datos del cliente, dando inicio a la alineación del vehículo en grados de desviación de convergencia y divergencia una vez que se enciende los indicadores se va a graduar en 90 grados y se sobreentiende que está en 0 grados figura 24, como por ejemplo, se realiza una simulación de la alineación en un vehículo, Chevrolet Vitara año 2002, la rueda izquierda está en 90 grados quiere decir que la rueda esta recta, en el caso de la rueda derecha este en 88 grados figura 25, se sobre entiende que tiene una desviación angular hacia el interior de 2 grados, procediendo a realizar las respectivas correcciones en la alineación, una vez ya corregida la desviación de la rueda derecha se observa que se prende una luz verde que nos indica que ya está alineado el vehículo figura 26.



Figura 25. Alineación del vehículo correcta.

Se procede guardar la alineación del vehículo con los datos ya registrados del cliente tal y como se muestra en la figura 26 guardando como archivo txt.



Figura 26. Alineación guardada como documento txt.

En el caso de que el cliente regrese y necesite conocer la alineación se hace clic en el botón mostrar, se busca por el número de placa como archivo o documento (txt), guardada en una carpeta con la placa de vehículo dentro de esta carpeta se guarda el mes y el año en que se realizó la alineación, la cual se observa en la ventana izquierda inferior los valores que se obtuvieron en la corrección de la alineación en la figura 27.

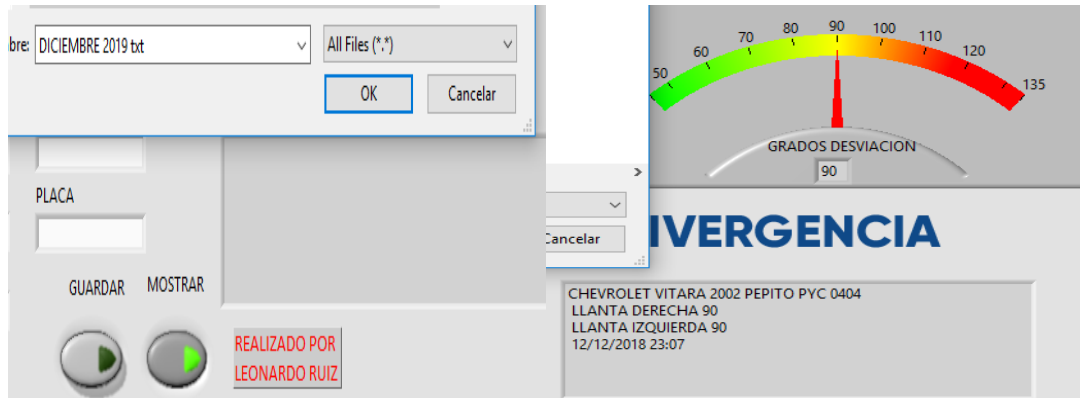


Figura 27. Revisión datos del cliente de una alineación anterior.

3. RESULTADOS Y DISCUSIÓN

3. RESULTADOS Y DISCUSIÓN

3.1. IMPLEMENTACION DEL SISTEMA DE ALINEACIÓN.

Para la implementación de este nuevo sistema de alineación mediante encoder incremental se tomaron en cuenta los siguientes parámetros como la conexión de los sensores o encoders incrementales al hardware Arduino, la selección de los platos deslizantes, montaje de los encoders incrementales en los platos deslizantes, características de los vehículos a alinear, pruebas al sistema, Comparar cada uno de los datos o valores obtenidos en las pruebas realizadas con los parámetros específicos según el fabricante.

3.1.1. CONEXIÓN DE LOS SENSORES AL HARDWARE ARDUINO

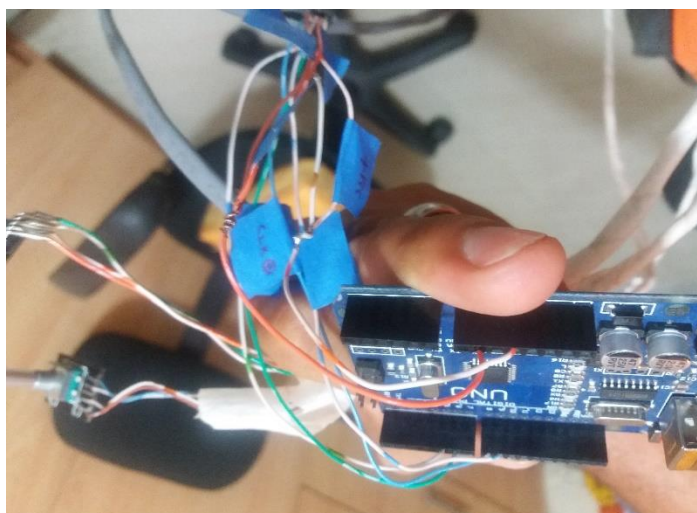


Figura 28. Conexión de sensores al Arduino.

La conexión de se realiza de acuerdo a la programación con sus respectivos pines figura 28, en donde GND va hacer tierra, VCC va hacer el voltaje de alimentación que es de 5 voltios, DT 5 al pin 5, CLK 6 al pin 6, CLK 8 al pin 8, DT 9 al pin 9 y la salida de la información o señales digitales a la PC donde están instaladas los softwares.

Para la ejecución del programa o sistema de alineación se trabaja en las siguientes partes:

- Sensor 1 y 2 (Encoders Incrementales).
- Software Arduino.
- Software LabVIEW.
- Pc

Dentro de este sistema los sensores 1 y 2 (encoders incrementales) estos tomaran datos y enviaran la comunicación al software Arduino este los codifica, transforma esa comunicación y la envía a la PC, en donde LabVIEW la recopila y como tal te muestra en un visor de Windows PC. LabVIEW receipta la información y produce un envío que solamente trabaja entre la PC y el visor de Windows en donde se genera un ciclo de información.

3.1.2. SELECCIÓN DE LOS PLATOS DESLIZANTES.

La factibilidad de utilizar los platos deslizantes para el diseño e implantación de este sistema de alineación figura 29, Estos elementos facilitarán la alineación del vehículo, constan de una base metálica en su interior, tiene una cantidad determinada de rulimanes que le permiten girar sin ningún inconveniente, por ende, costa de unos pasadores para mantenerlos fijos y en los cuales en la parte inferior se ubicara los encoders incrementales encargados de tomar datos de la convergencia y divergencia del vehículo.



Figura 29. Selección de los platos deslizantes.

El encoder incremental instalado en cada uno de los platos deslizantes figura 29, tendrá la función de tomar datos de las tres gamas de vehículos livianos escogidos para el presente proyecto que son los siguientes: Suzuki Gran Vitara Sz año (2005 - 2014) -(4x2 - 4x4), Chevrolet Luv D-Max. (2003 – 2015)- (4x2 - 4x4), Kia Rio Stylus (2005-2011).

3.1.3. MONTAJE DE LOS ENCODERS INCREMENTALES EN LOS PLATOS DESLIZANTES.

Antes de instalar los encoder incrementales en los platos deslizantes procedemos a encapsular para proteger de algún golpe o filtración de

humedad, ya encapsulado se ensambla a una base la cual va a estar fija figura 30.

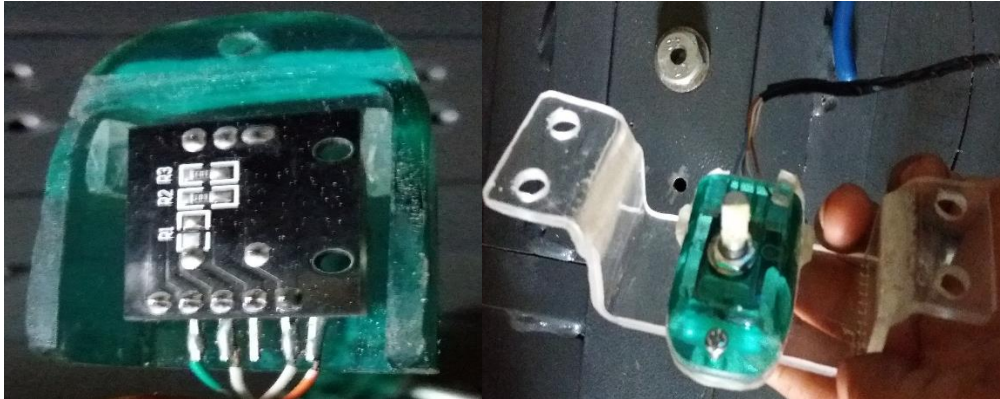


Figura 30. Encapsulado de encoders incrementales.

Luego de encapsular los encoders incrementales, se procede en el eje central (perno guía) figura, se procede a perforar a una medida de 6mm, en donde el eje del encoder entra con una presión moderada para no afectar la función de los encoders incrementales figura 31.



Figura 31. Colocación de encoders incrementales en los platos deslizantes.

También se toma en cuenta la altura que tiene el encoder al momento de instalar en los platos deslizantes por lo que se tuvo que soldar unas bases metálicas de 3cm de altura para que los encoders incrementales al momento de colocar los platos deslizantes correctamente no vayan a topar en el piso y por ende a producirse algún daño.



Figura 32. Empernado de los encoders a los platos deslizantes.

En este proceso se toma en cuenta los parámetros necesarios para que los sensores o encoders no tenga ninguna dificultad al momento de realizar el giro tanto a la izquierda como a la derecha. En la figura 32 se observa dónde van a ir ubicados los sensores o encoders incrementales.

3.1.3.1. Características de los vehículos a alinear.

Las características de cada uno de estos vehículos son primordiales para la hora de realizar una alineación ya que sus parámetros están dados por el fabricante con sus debidas tolerancias para la convergencia de las dos ruedas dando un mínimo y un máximo de tolerancia para que la alineación sea la correcta como se observa en la siguiente tabla 8.

Tabla 7. Características de Suzuki Gran Vitara SZ.

Suzuki Gran Vitara SZ año (2005 - 2014) -(4x2 - 4x4).	
Convergencia (tolerancia 2 ruedas).	Divergencia (tolerancia 2 ruedas).
De 0.1 hasta + 0.03 grados	- 0.00 grados
Tolerancias erróneas de la alineación.	
Convergencia	Divergencia
De 0.04 hasta + 0.45 grados	De - 0.01 hasta - 0.45 grados

(HUNTER 2013).

Tabla 8. Características de Chevrolet Luv D-Max.

Chevrolet Luv D-Max (2003 – 2015) -(4x2 - 4x4).	
Convergencia (tolerancia 2 ruedas).	Divergencia (tolerancia 2 ruedas).
De 0.1 hasta + 0.03 grados	- 0.00 grados
Tolerancias erróneas de la alineación.	
Convergencia	Divergencia
De 0.04 hasta + 0.45 grados	De - 0.01 hasta - 0.45 grados

(HUNTER 2013).

Tabla 9. Características del Kia Rio Stylus.

Kia Rio Stylus (2005-2011)	
Convergencia (tolerancia 2 ruedas).	Divergencia (tolerancia 2 ruedas).
De 0.1 hasta + 0.03 grados	- 0.00 grados
Tolerancias erróneas de la alineación.	
Convergencia	Divergencia
De 0.04 hasta + 0.45 grados	De - 0.01 hasta - 0.45 grados

(HUNTER 2013).

Los parámetros de la convergencia y divergencia de los vehículos no siempre tienen que dar cero en este caso noventa grados para que una alineación quede bien realizada y haya una conducción recta, ya que en los vehículos intervienen otros parámetros en el momento de alinear, el técnico con la experiencia ya adquirida hará las respectivas correcciones, pero lo que es importante es sus grados de tolerancia tienen que tener las mismas tolerancias en las dos ruedas.

3.2. PRUEBAS AL SISTEMA.

Una vez encendido el sistema de alineación y conociendo los parámetros y características de los vehículos se procede a realizar las pruebas pertinentes para lo cual subimos el vehículo en los platos deslizantes, para realizar la alineación correspondiente. La alineación se procede a realizar en el vehículo de marca Chevrolet Gran Vitara SZ figura 33.



Figura 33. El vehículo sube sobre los platos deslizantes.

Una vez que el vehículo sobre los platos deslizantes se conectan los sensores o encoders incrementales al Arduino uno y este al CPU, se enciende el sistema y refrescamos la opción de los puertos, escogiendo el canal en este caso en canal 3 figura 34.

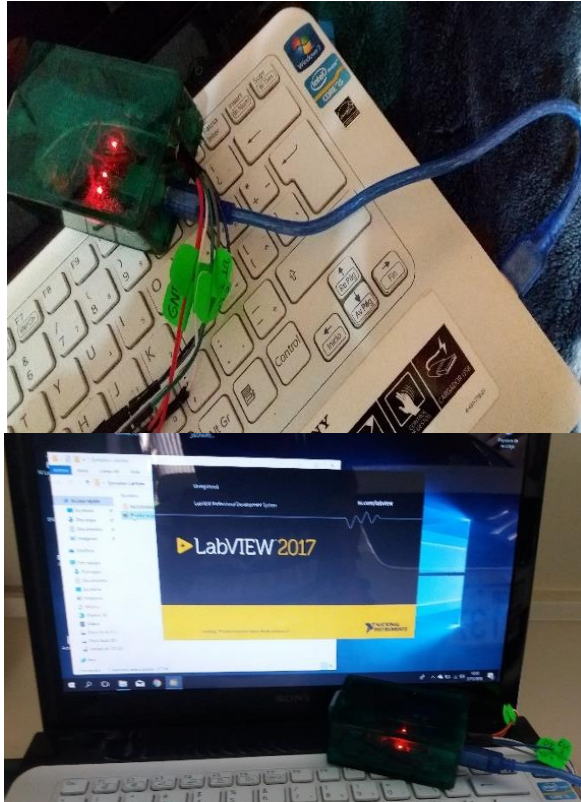


Figura 34. Encendido del sistema LabVIEW.

Ya una vez encendido el sistema de alineación el indicador va a dar en 90 grados que se sobre entiende que esta en cero grados figura 35. Se procede a ingresar los datos del cliente y del vehículo para luego guardar la alineación y tener como respaldo del trabajo realizado.



Figura 35. Ingreso de datos del cliente.

Se realiza un giro al volante del vehículo tanto a la izquierda como a la derecha en un rango de 45 grados aproximadamente y se observa en la pantalla de Windows como se encuentra el estado de la alineación del vehículo figura 36.

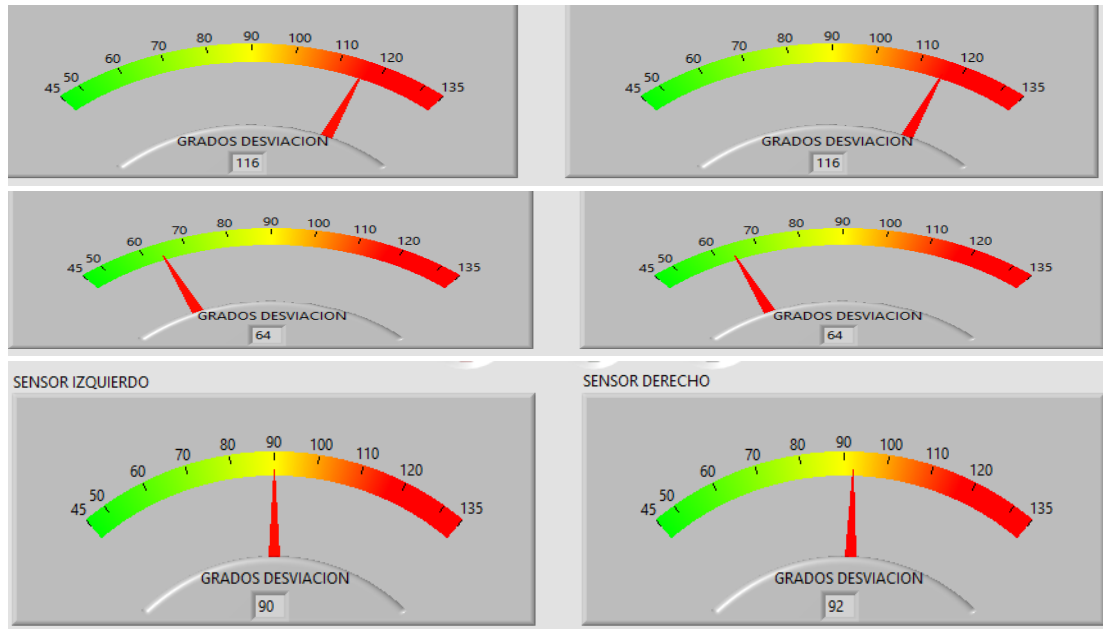


Figura 36. Desviación de rueda derecha 2°.

Se visualiza la de desviación de los ángulos en grados de la convergencia y divergencia de las ruedas y procediendo a realizar los cambios respectivos para la correcta alineación del vehículo ya que esta tiene una desviación de un grado en la rueda derecha figura 37.



Figura 37. Corrección de la alineación de la rueda derecha.

Una vez realizado los respectivos cambios o correcciones se observa que la alineación es correcta dando como resultado 90 grados que es igual a 0 grados, se guarda la alineación como documento (txt) figura 38, solo con el número de placa de vehículo teniendo como respaldo la alineación en caso el cliente regrese o también para futuras alineaciones y conozca como estuvo la alineación de su vehículo en la anterior alineación.

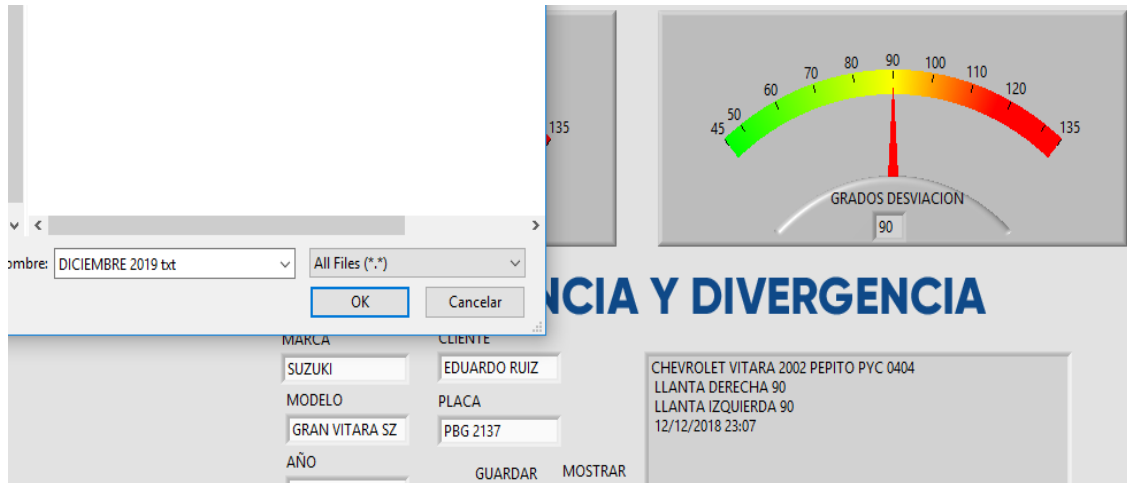


Figura 38. Guardar la alineación como documento txt.

3.2.1. DATOS OBTENIDOS EN LAS PRUEBAS REALIZADAS CON LOS PARAMETROS ESPECIFICOS SEGÚN EL FABRICANTE.

Para realizar la alineación correctamente, las ruedas deben estar centradas es decir 90 grados las dos ruedas, esto dependiendo de los demás elementos de la suspensión, es decir si la rueda izquierda tiene 92 grados, la rueda derecha debe tener 88 grados indicando una desviación angular hacia el interior formando una convergencia positiva, considerando que los valores están dentro los parámetros permitidos mínimo 0,1 y máximo de 0.3 grados para las dos ruedas.

En las siguientes tablas observamos la alineación realizada a los vehículos Suzuki Gran Vitara SZ, Chevrolet Luv D-max. y Kia Rio Stylus.

Tabla 10. Alineación Suzuki Gran Vitara SZ Alineación normal

Suzuki Gran Vitara SZ (2005 – 2014)	
Valores obtenidos antes de la corrección en la alineación	
Convergencia	Divergencia
0°03 grados	0°01
Valores obtenidos después de la alineación.	
Convergencia derecha corregida.	Convergencia izquierda corregida
0°00	0°00
Tiempo aproximado de alineación: 45 minutos	

Fuente: Alineadora Hunter

Tabla 11. Alineación Suzuki Gran Vitara SZ (Encoders incrementales)

Suzuki Gran Vitara SZ (2005 – 2014)	
Valores obtenidos antes de la corrección en la alineación	
Convergencia	Divergencia
0°03 grados	0°01
Valores obtenidos después de la alineación.	
Convergencia derecha corregida.	Convergencia izquierda corregida
0°00	0°00
Tiempo aproximado de alineación: 15 minutos	

Tabla 12. Alineación Chevrolet Luv D-Max Alineación normal

Chevrolet Luv D-Max (2003 – 2015) -(4x2 - 4x4).	
Valores obtenidos antes de la corrección en la alineación	
Convergencia	Divergencia
0°04 grados	0°01
Valores obtenidos después de la alineación.	
Convergencia derecha corregida.	Convergencia izquierda corregida
0°02	0°02
Tiempo aproximado de alineación: 45 minutos (HUNTER 2013).	

Tabla 13. Chevrolet Luv D-Max 2011 4x2 (Encoder incremental).

Chevrolet Luv D-Max 2007 - 4x4.	
Valores obtenidos antes de la corrección en la alineación	
Convergencia	Divergencia
0°04	0°01
Valores obtenidos después de la alineación.	
Convergencia derecha corregida.	Convergencia izquierda corregida
0°02	0°02
Tiempo aproximado de alineación: 15 minutos	

Tabla 14. Alineación normal Kia Rio Stylus Alineación normal.

Kia Rio Stylus (2005-2011)	
Valores obtenidos antes de la corrección en la alineación	
Convergencia	Divergencia
0°03	0°2
Valores obtenidos después de la alineación.	
Convergencia derecha corregida.	Convergencia izquierda corregida
0°01	0°01
Tiempo aproximado de alineación: 35 minutos (HUNTER 2013).	

Tabla 15. Alineación normal Kia Rio Stylus (Encoder incremental).

Kia Rio Stylus (2005-2011)	
Valores obtenidos antes de la corrección en la alineación	
Convergencia	Divergencia
0°03	0°2
Valores obtenidos después de la alineación.	
Convergencia derecha corregida.	Convergencia izquierda corregida
0°01	0°01
Tiempo aproximado de alineación: 15 minutos	

Una vez aplicados los métodos anteriores se analizó tanto la viabilidad como la factibilidad de este sistema de alineación para vehículos livianos mediante

3.3. ANÁLISIS Y PRUEBAS.

En los análisis y pruebas realizadas a los vehículos en los cuales se aplicó el método experimental, comparando la alineación normal con la alineación mediante encoders incrementales, con lo cual se constató cada uno de los procesos y tiempos que se lleva en realizar la alineación.

En el caso de alineación por el método tradicional se lleva a cabo el siguiente proceso, el vehículo Suzuki Gran Vitara SZ es puesto en el elevador, se instalan los sensores en las cuatro ruedas, se ingresan los respectivos datos del cliente y del vehículo, se ingresa al sistema para escoger la marca, modelo y año del vehículo, para luego proceder a la compensación de los sensores en las ruedas, giro de las ruedas a la izquierda y a la derecha, teniendo un tiempo determinado de 25 minutos, solo para conocer el estado de desviación de los ángulos de las ruedas, en la cual se observó una desviación en la rueda izquierda de $0^{\circ}03$ grados y en la rueda derecha $0^{\circ}01$ grados, procediendo a su corrección tanto en la izquierda como en la derecha de $0^{\circ}00$ grados, obteniendo un tiempo determinado de 20 minutos que en total fueron 45 minutos, tomando demasiado tiempo para conocer el estado actual de la alineación del vehículo.

Por otra parte, la alineación realizada al mismo vehículo mediante el nuevo sistema de encoders incrementales, tomó un tiempo determinado de 15 minutos en total para realizar la corrección de la alineación, es decir 5 minutos en colocar el vehículo, ingresar datos del cliente y del vehículo, los giros de las ruedas tanto a la izquierda como a la derecha obteniendo los mismos valores que en la alineación normal: rueda izquierda $0^{\circ}03$, rueda derecha $0^{\circ}01$ y 10 minutos en realizar su corrección, obteniendo tanto en la rueda izquierda como en la derecha $0^{\circ}00$ grados.

En otras dos pruebas realizadas en vehículos de marca Chevrolet Luv d-Max. Y Kia Rio Xcite, el tiempo aproximado fue de 45 minutos con la alineación normal y 15 minutos con la alineación mediante encoders incrementales, en realizar la alineación total siendo igual que en la primera prueba que se realizó, al vehículo de marca Suzuki Gran Vitara SZ.

En el vehículo de marca Chevrolet Luv d-Max, los valores obtenidos antes de su corrección con la alineación normal, fue en la rueda izquierda de $0^{\circ}04$ y en la rueda derecha $0^{\circ}01$, los mismos valores obtenidos con la alineación mediante encoders incrementales, ya corregida la alineación realizada, mediante los dos sistemas de alineación, se obtuvieron valores tanto en la rueda izquierda de $0^{\circ}02$ como en la derecha de $0^{\circ}02$ grados.

El mismo procedimiento con el vehículo de marca Kia Rio Stylus, los valores obtenidos por los dos sistemas de alineación antes de su corrección fueron de $0^{\circ}03$ en la rueda izquierda y $0^{\circ}02$ grados en la rueda derecha y los valores obtenidos, después de las correcciones pertinentes fueron de $0^{\circ}01$ grados para las dos ruedas.

Obteniendo los resultados esperados en donde se pudo comprobar los tiempos en cada uno de los procesos con un margen de error mínimo que están dentro de los parámetros. Así como también las encuestas realizadas a los técnicos de los diferentes talleres express dando como resultado la necesidad de este sistema de alineación en los talleres express.

4. CONCLUSIONES Y RECOMENDACIONES

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES.

- Al momento de programar Arduino y LabVIEW es algo complejo, pero no imposible de hacerlo, en la programación de Arduino cada proceso fue separado por bloques para conocer la función de cada bloque e identificarlos.
- Cuando realizamos la repetición de los ciclos se debe hacer que el sistema o el bloque de repetición capte solo una vez para que no se interrumpa la comunicación en el sistema.
- En la programación de Arduino para que el sistema trabaje sin ningún inconveniente este debe trabajar a una velocidad de 9600 baudios esto hará que el software tenga una buena comunicación con los bloques ya creados.
- Para la programación del software LabVIEW en este sistema su funcionamiento está separado por bucles de verdadero o falso, los cuales intervienen en el funcionamiento del sistema al contrario que en Arduino está dado el nombre de bloques para poder identificar el proceso de cada funcionamiento.
- LabVIEW al igual que Arduino debe trabajar a una velocidad de 9600 baudios, el objetivo principal es que tengan una buena comunicación entre los bucles y el software Arduino.
- Las alineaciones realizadas a los vehículos dieron como resultado positivo ya que se pudo realizar sin ningún contratiempo y obteniendo los resultados esperados.
- El tiempo límite que se requirió para realizar las alineaciones con este nuevo sistema de alineación mediante encoders incrementales fue aproximadamente de 15 minutos logrando los objetivos propuestos en el proyecto.
- Durante el desarrollo del presente proyecto se pudo determinar que este sistema es fácil de utilizarlo responde rápidamente a los movimientos angulares de convergencia y divergencia del vehículo logrando obtener los valores de la alineación de cada uno de los vehículos escogidos.

- Los tiempos realizados en cada una de las alineaciones demuestran la factibilidad de este nuevo sistema consiguiendo los mismos resultados que en una alineación normal.

4.2. RECOMENDACIONES.

- Se debe tomar en cuenta que para programar el software Arduino, Ni - VISA y LabVIEW estos tres softwares deben ser descargados para que trabajen conjuntamente, es decir deben ser compatibles para los cuales deben ser descargados para trabajar con Windows de 32bits, así como también al momento de programar es importante hacerlo por bloques ya que cada bloque tiene su función específica.
- Es importante agregar los ciclos de repetición de todos los datos o valores que ingresan al sistema, para captar solo una vez y observar el ángulo de desviación de la rueda, así el sistema no se satura.
- En Arduino la velocidad de comunicación es importante que trabaje dentro de los parámetros siempre teniendo en cuenta la velocidad del CPU y la velocidad del software LabVIEW se recomienda que sea de 9600 baudios para que no se sature la comunicación.
- Es importante crear los bucles de verdadero o falso para conocer qué es lo que está pasando dentro del bucle, que datos o valores están ingresando al sistema.
- Hay que tener en cuenta que la velocidad de comunicación del software LabVIEW debe ser la misma del software Arduino la cual es de 9600 baudios.
- Es importante que al momento subir los vehículos los platos deslizantes deben tener los seguros puestos para que no se muevan y poder realizar la alineación en el tiempo requerido.
- Al momento de realizar las pruebas de alineación con este sistema siempre que se encienda se recomienda refrescar el puerto para que lo reconozca y poder realizar la alineación sin ningún inconveniente ya que este sistema es fácil de utilizar.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

- A, Escalera. (2015). *Vision por computadora, fundamentos y metodos*. Madrid- España: Person Edision.
- Aguirre Molina, H. d. (2015). *Control de velocidad de un motor de C.D. basado en redes neuronales artificiales*. Mexico: Instituto Politecnico Nacional. Obtenido de <http://tesis.ipn.mx/handle/123456789/21321>
- Alcívar Cevallos, V. I. (2015). *Diseño e implementación de una plataforma intercal de dos grados de libertad y evaluación del modelado matemático a través de matlab/simulink*. Guayaquil: Diseño e implementación de una plataforma intercal de dos grados de libertad y evaluación del modelado matemático a través de matlab/simulink. Obtenido de <https://dspace.ups.edu.ec/handle/123456789/10426>
- Altamirando, V. (2011). *Tecnicas de investigacion*. Lima- Peru.
- Calderón Pérez, F. M. (2015). *Obtención de datos en un tablero digital para un vehículo convencional a través de la plataforma de Arduino*. Quito: Universidad Internacional SEK. Obtenido de <http://7labs.uisek.edu.ec/handle/123456789/1567>
- Control, M. (23 de febrero de 2016). *Control de movimientos y accionamientos electricos*. Obtenido de Motion Control: http://Codificador_rotatorio
- Córdova Cox, E. A. (2016). *Implementación de un banco de prueba para el control automático de posición, temperatura y velocidad, en el laboratorio de control y manipulación automática de la Facultad de Mecánica*. Chimborazo: Escuela Superior Politécnica de Chimborazo. Obtenido de <http://dspace.esPOCH.edu.ec/handle/123456789/5807>
- Córtez Alay, A. B. (2014). *Plan de negocios para la creacion de una empresa de servicio de Automotriz en la ciudad de Jutiapa*. Mexico: Plan de Negocios.
- Creus Solé, A. (2012). *Neumática e Hidráulica*. Mexico: 2ªed.
- Daneri, P. (2016). *Automatización y Control Industrial*. Valencia-España: ASA.
- Díaz Rodríguez, B. (2017). *Plataforma robótica basada en la estructura de un coche de*. Madrid: Universidad de Alcalá. Obtenido de <https://ebuah.uah.es/dspace/handle/10017/30593>
- Enrique, P. M. (2013). *Sistema Electronico Digitales*. Mexico: Marcombo S.A. Obtenido de <https://encoders opticos funciones especificas handle>
- Fernando, R. C. (2013). *Robotica*. Mexico: Alfaomega Grupo Editor. Obtenido de <https://Encoder optico funcionamiento>
- García Álvarez, D. P. (2014). *Construcción de un dispositivo para almacenamiento de variables durante la conducción*. Cuenca: Ingeniería Mecánica Automotriz CUE - Tesis Pregrado. Obtenido de <https://dspace.ups.edu.ec/handle/123456789/7396>

- Guillermo. (2016). *Sancando al maximo provecho a capacidad de instalada y manejo de rendimientos*. Mexico.
- Herrera, P. (2013). *Correspondencia esteroscopica en imagenes obtenidas con proyeccion omnideracional para entornos forestales (Tesis de grado)*. Mexico.
- Jesus, M. C. (2014). *Sensores Y Actuadores*. Mexico: Patria, S.A De C.V. Obtenido de <https://encoder optico características>
- Llamas, L. (10 de Junio de 2016). *Manual De Aplicacion De Encoders*. Obtenido de <https://www.amee.com.mx/ENCODERS.pdf>
- Luque Alcántara, M. P. (2017). *Diseño, construcción y control de un aerogenerador de equipo de prácticas de bajo coste*. Sevilla: Universidad de Sevilla. Departamento de Ingeniería de Sistemas y Automática. Obtenido de <https://idus.us.es/xmlui/handle/11441/65373>
- Mendieta Mejía, J. L. (2013). *Manejo de un decodificador óptico (encoder) en aplicación*. Guayaquil: Escuela Superior Politecnica del Litoral. Obtenido de <https://www.dspace.espol.edu.ec/handle/123456789/6572>
- Mendieta, M. (2013). *Manejo de un decodificador óptico (encoder) en aplicación con un dsPIC*. Guayaquil: ESCUELA SUPERIOR POLITECNICA DEL LITORAL. Obtenido de <http://www.dspace.espol.edu.ec/xmlui/handle/123456789/37186>
- OMRON ELECTRONICS EUROPE B.V. (2015). Acerca de Motion Control. *Motion Control*, 45-46.
- Paico, P. (2013). *Análisis y diseño para el funcionamiento del brazo robot CRSA255 del laboratorio de Mecatrónica de la universidad nacional de Piura*. Piura: UNIVERSIDAD DE PIURA. Obtenido de <http://repositorio.unp.edu.pe/handle/UNP/681>
- Ricardo, S. A. (2016). *VIII Congreso Ibérico de Agroingeniería*. España: Universiad Miguel Hernandez De Elche. Obtenido de <https://Encoders /tipos utilizacion handle>
- Roldan, J. (2013). *Motores Eléctricos y Automatismos de Control*. Mexico: Paraninfo.
- Rosaleny, R. T. (abril de 2013). *Encoder Optico*. Obtenido de http://www.infopl.net/files/documentacion/instrumentacion_deteccion/infoPLC_net_ENCODERS_OPTICOS.pdf
- Soriano, P. R. (2012). *Proceso de investigacion cientifica*. Mexico.
- Spano, B. (2013). Diseño e implementación de tarjeta de conversión de señal de comunicación serial de sensores de flujo de gasolina a pulsos para la entrada a tarjeta de control de suministro de combustible de Intelcon System C.A. Venezuela: Tesis de Grado de Ingeniería Electrónica. Obtenido de <http://www.miunespace.une.edu.ve/jspui/handle/123456789/1244>

- Trujillo, V. (Septiembre de 2016.). *Identificar los tipos de sensores utilizados en la robótica*. Mexico: Centro universitario UAEM Zumpango. Obtenido de <http://ri.uaemex.mx/bitstream/handle>
- Villar, C. (Junio 2013). *Estudio sobre instalación, montaje y funcionamiento de un codificador para determinar posición y velocidad de un vehículo eléctrico*. España: Universidad de Almeira. Obtenido de <http://repositorio.ual.es/handle/10835/2462>

ANEXOS

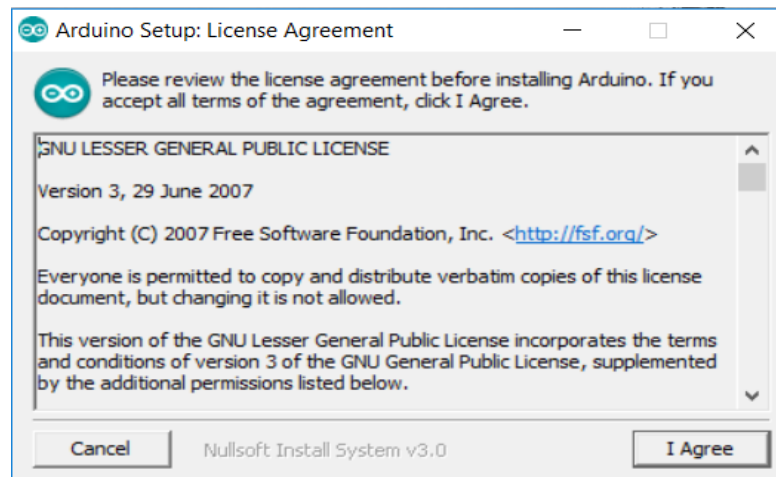
ANEXOS

Anexo 1. Encuestas

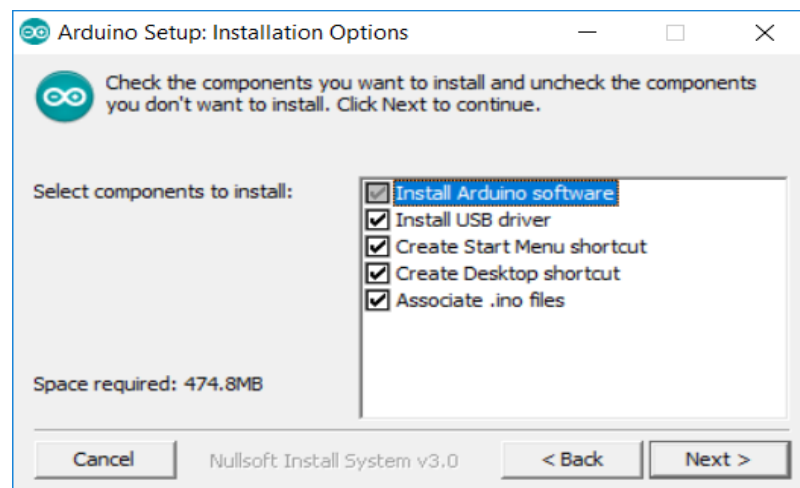
CONTESTE EL SIGUIENTE CAMPO DE PREGUNTAS.

1. ¿TIENE USTED CONOCIMIENTO SOBRE LO QUE ES LA ALINEACIÓN DE UN VEHÍCULO?
SI NO
2. ¿TIENE USTED CONOCIMIENTO SOBRE CONVERGENCIA Y DIVERGENCIA?
SI NO
3. ¿CREE USTED QUE ES IMPORTANTE LA ALINEACIÓN DEL VEHÍCULO PARA EVITAR EL DESGASTE EXCESIVO DE LOS NEUMATICOS Y OTROS ELEMENTOS?
SI NO
4. ¿CHEQUEA USTED EL VEHÍCULO ANTES DE EMPEZAR LA ALINEACIÓN?
SI NO
5. ¿INFORMA USTED AL CLIENTE DE LOS POSIBLES DAÑOS ENCONTRADOS EN EL VEHÍCULO ANTES DE PROCEDER A REALIZAR LA ALINEACIÓN?
SI NO
6. ¿PROCEDERÍA USTED A REALIZAR LA ALINEACIÓN DEL VEHÍCULO SI ALGUN ELEMENTO SE ENCUENTRA EN MALAS CONDICIONES?
SI NO
7. ¿UNA DE LAS DESVENTAJAS DE ALINEAR UN VEHICULO CON SENSORES EN LAS CUATRO RUEDAS ES EL TIEMPO QUE CONLLEVA REALIZAR EL TRABAJO?
SI NO
8. ¿CREE USTED QUE EL SISTEMA DE ALINEACIÓN MEDIANTE SENSORES SI NO SE ACTUALIZA Y SE CALIBRA PRODUCE ERRORES EN LA TOMA DE DATOS?
SI NO
9. ¿USTED COMO TECNICO ESPECIALIZADO EN ALINEACIONES CREE QUE ES NECESARIO UN SISTEMA DE ALINEACION DISTINTO AL SISTEMA ACTUAL?
SI NO
10. ¿CREE USTED QUE ES UN BENEFICIO PARA EL TECNICO UN SISTEMA DE ALINEACIÓN, PARA REALIZAR EL TRABAJO DE 15 MINUTOS A 30 MINUTOS?
SI NO

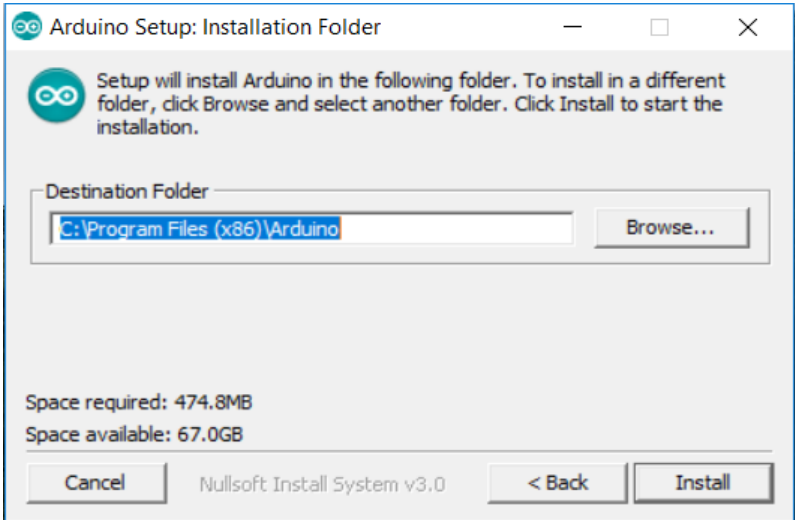
Anexo 2. Software Arduino 1.8.6 para windows de 32bits.



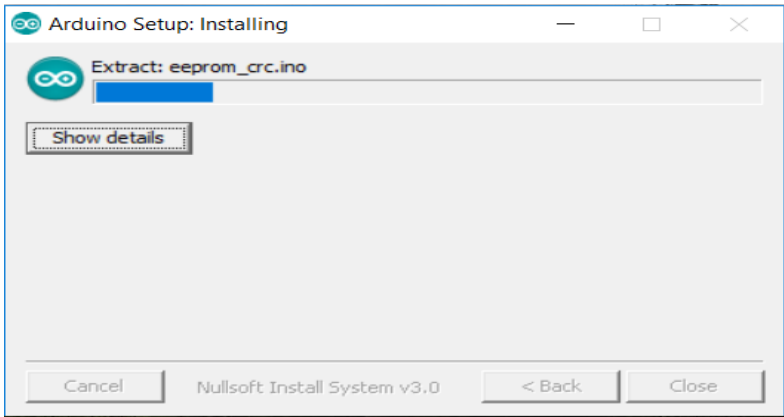
Anexo 3. Selección todas las opciones en esta pantalla y dar clic en next.



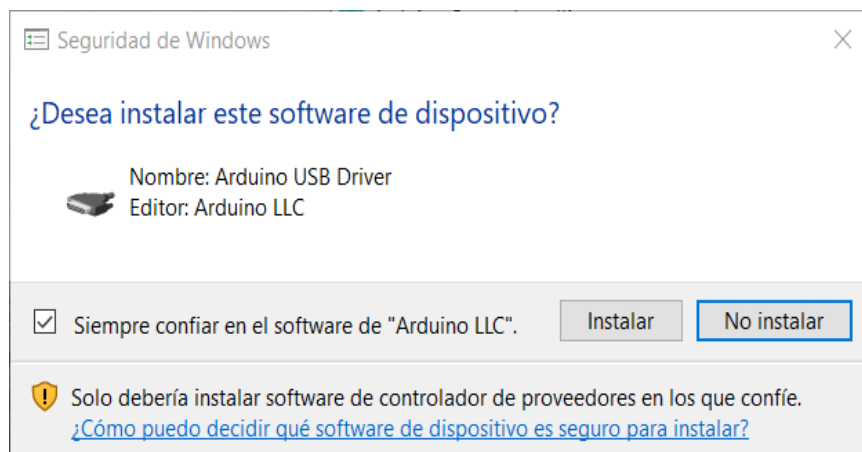
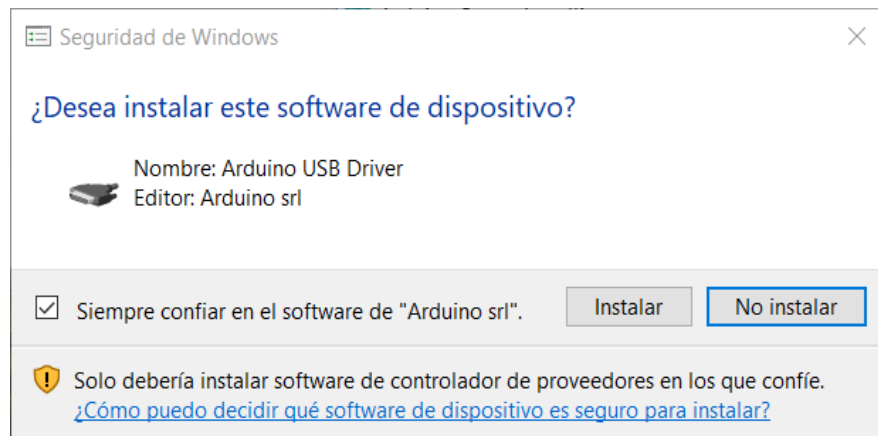
Anexo 4. Selección de la ruta de instalación recomendación dejar la que está establecida por defecto: c:\program files (x86)\arduino.



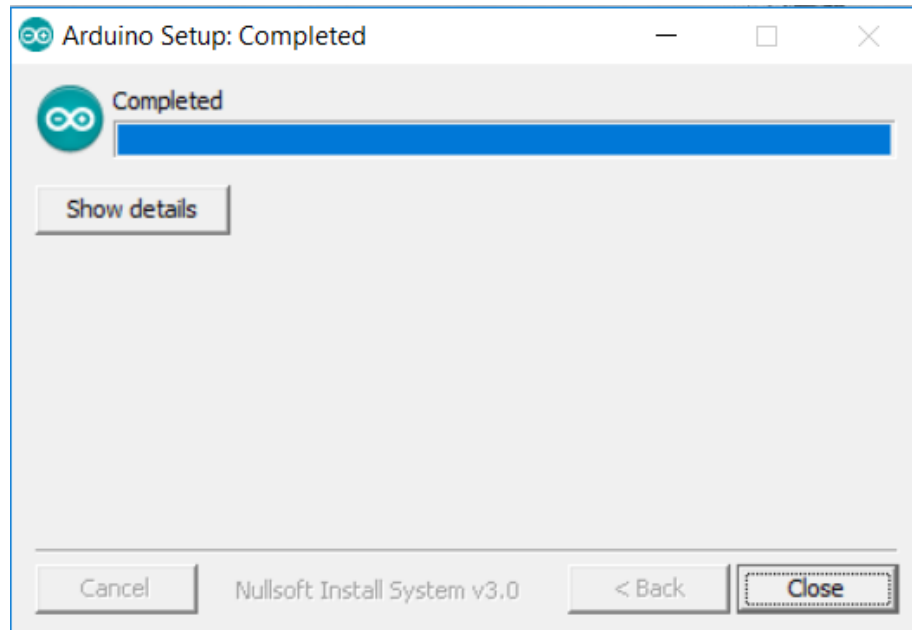
Anexo 5. Extracción de los archivos de instalación.



Anexo 6. Permisos necesarios para habilitar los puertos de comunicación entre el software y hardware de Arduino, así como las librerías correspondientes para la conexión por USB, dar clic en instalar.



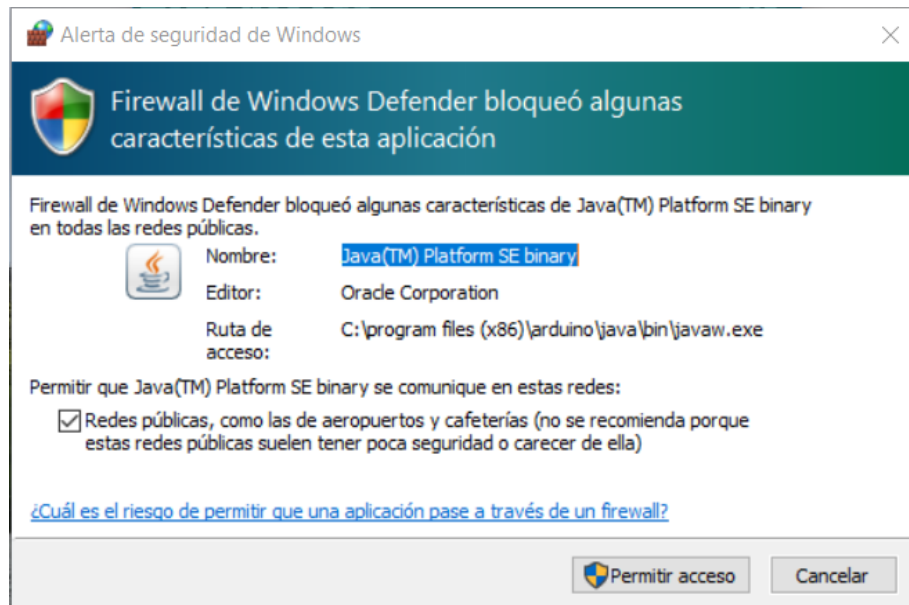
Anexo 7. Finalización de la instalación de todas las librerías correspondientes el software de Arduino quedara completamente instalado en el computador. Dar clic en close para finalizar la instalación.



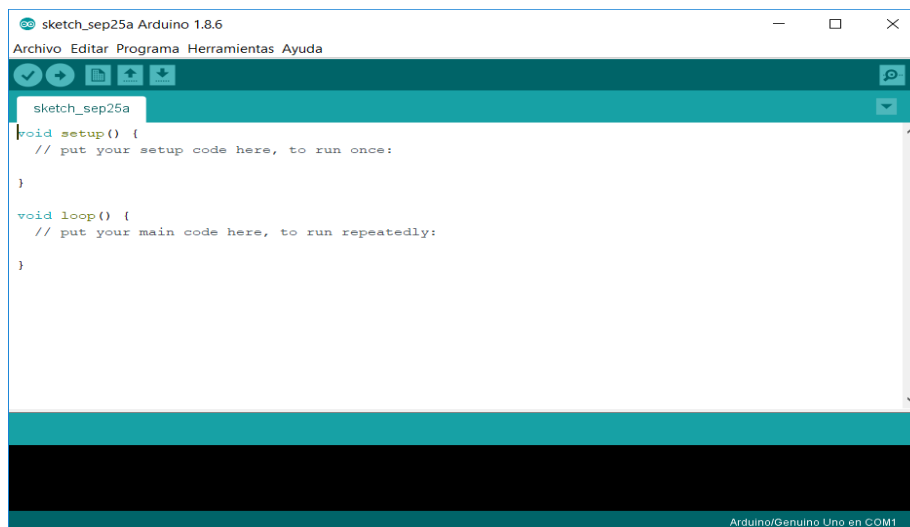
Anexo 8. Icono nuevo en el escritorio del computador.



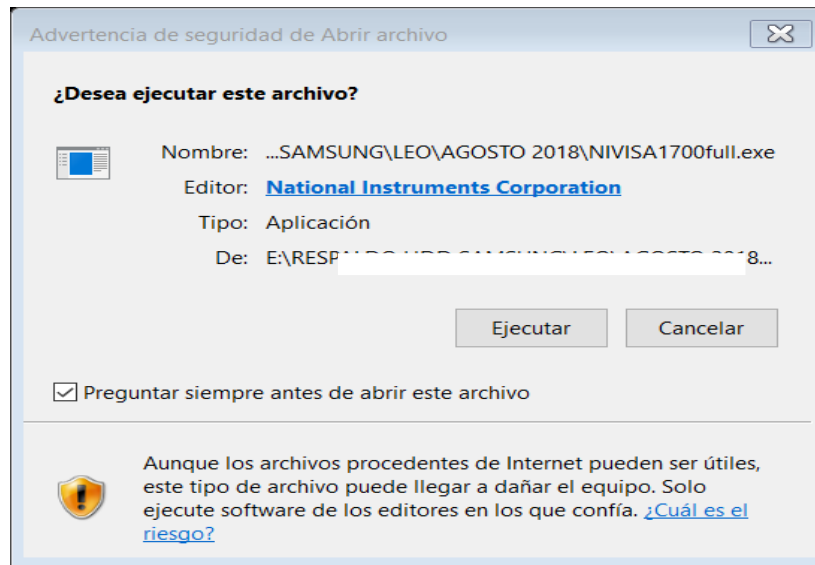
Anexo 9. Permisos acceso a los puertos a través del firewall de Windows.



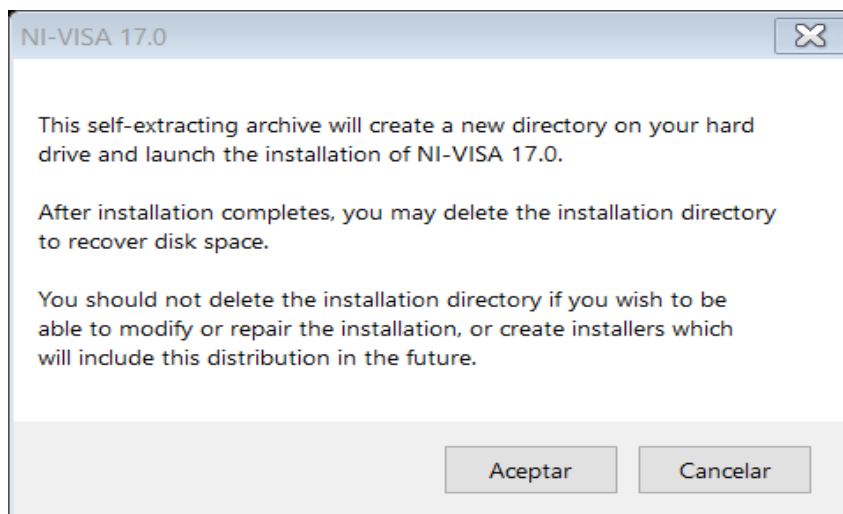
Anexo 10. El software de arduino en ejecución.



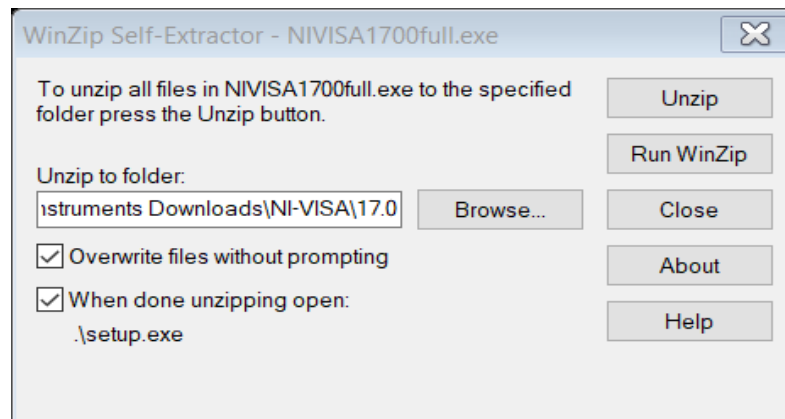
Anexo 11. Software ni-visa 1700 full.exe para windows de 32bits.



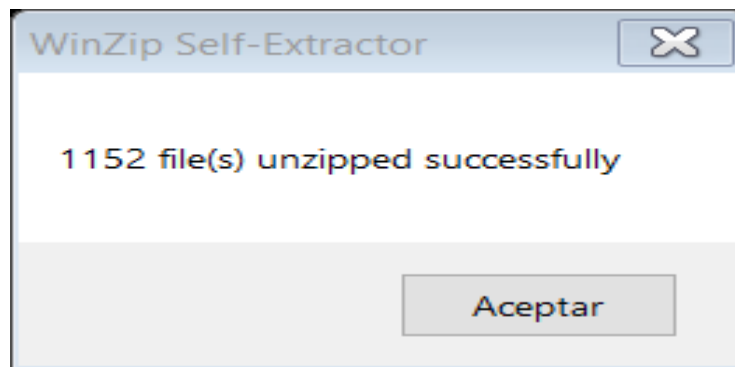
Anexo 12. Aviso de la extracción de los archivos necesarios para la respectiva instalación.



Anexo 13. Ruta donde se encuentra el archivo a instalar. Dar clic en unzip.



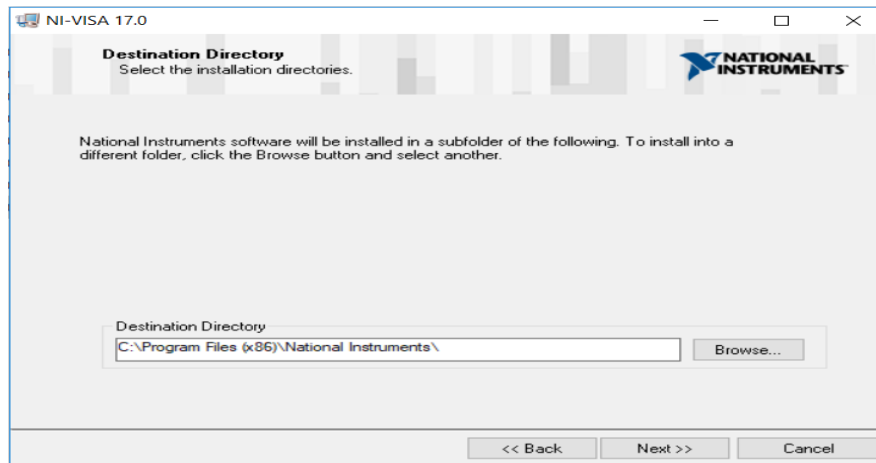
Anexo 14. Archivos satisfactoriamente extraídos.



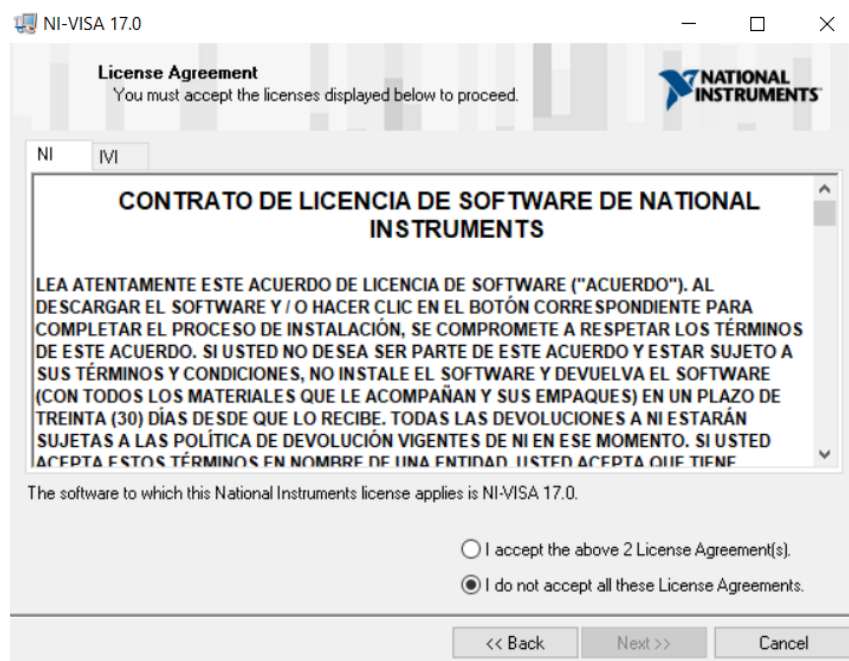
Anexo 15. Instalación mencionando la introducción del software en cuestión. Dar clic en next.



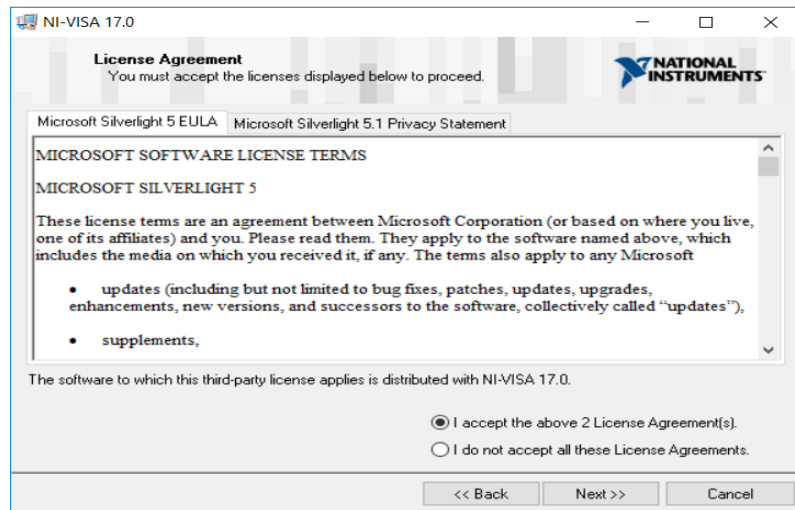
Anexo 16. Ruta de instalación se recomienda dejar la que viene por defecto.



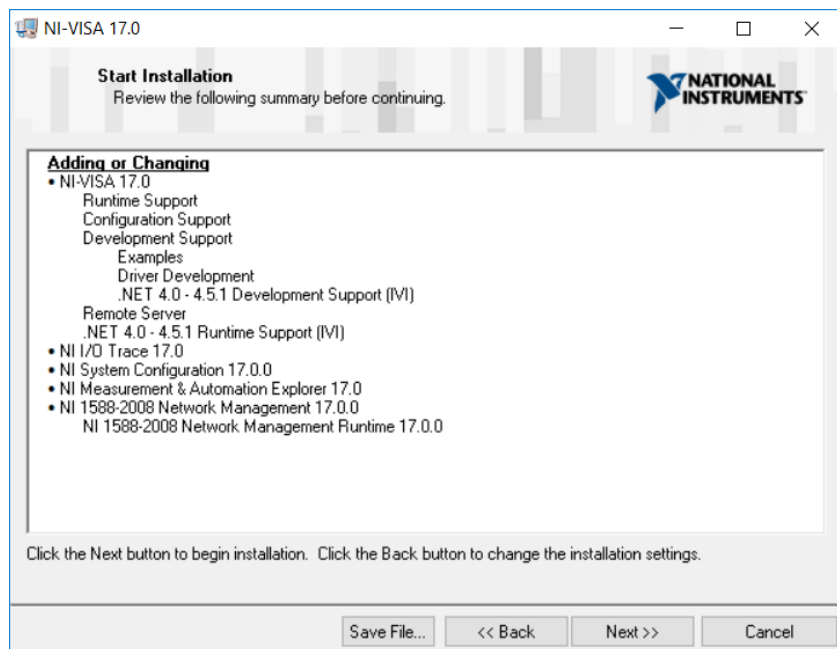
Anexo 17. Ruta de instalación se recomienda dejar la que viene por defecto.



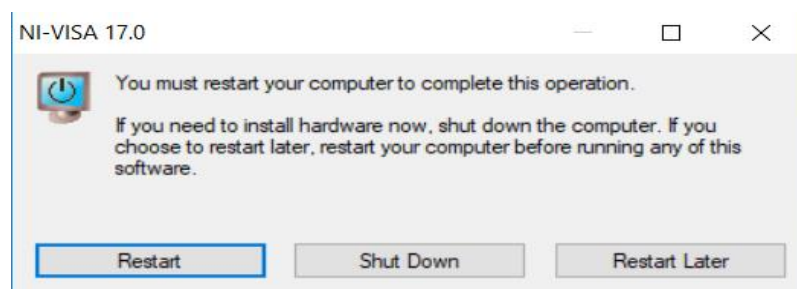
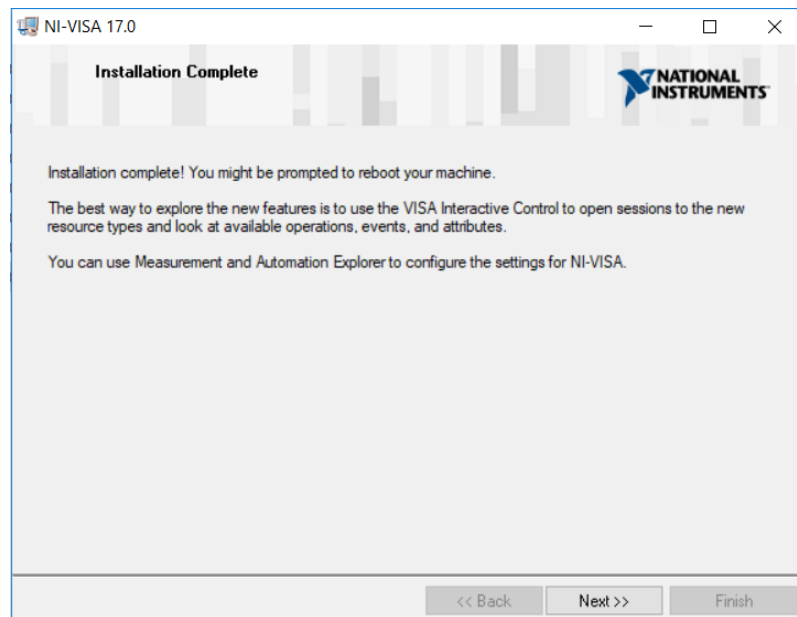
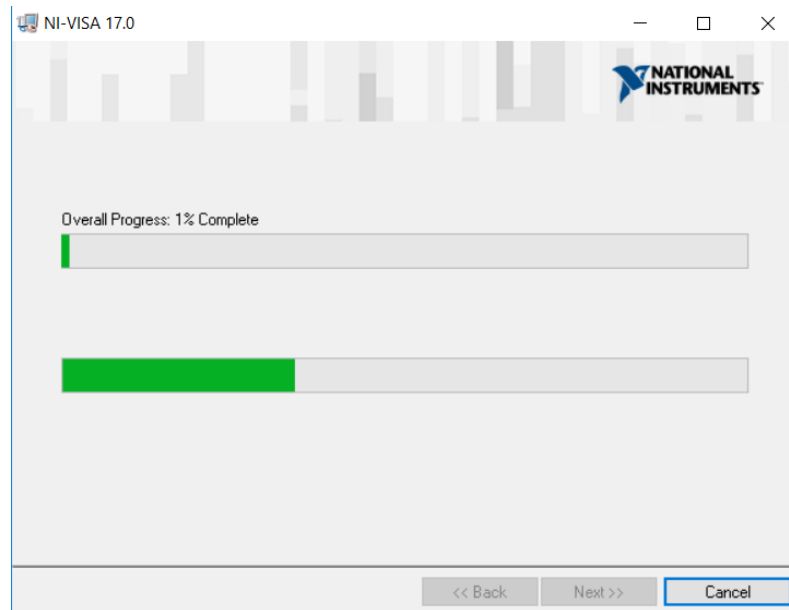
Anexo 18. Términos de licencia con Microsoft se deben aceptar para continuar con la instalación del mismo.



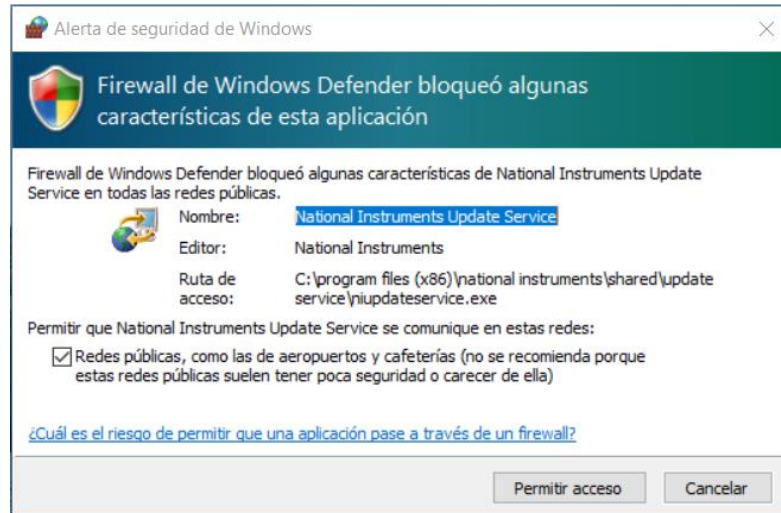
Anexo 19. Asistente de instalación indica todos los cambios que se van a efectuar al momento de instalar el software ni visa. Dar clic en next.



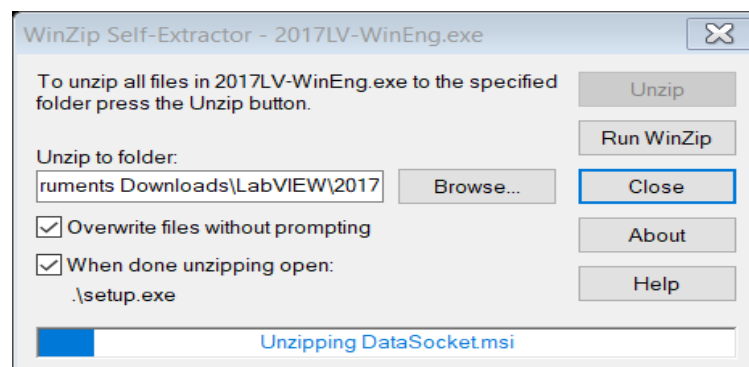
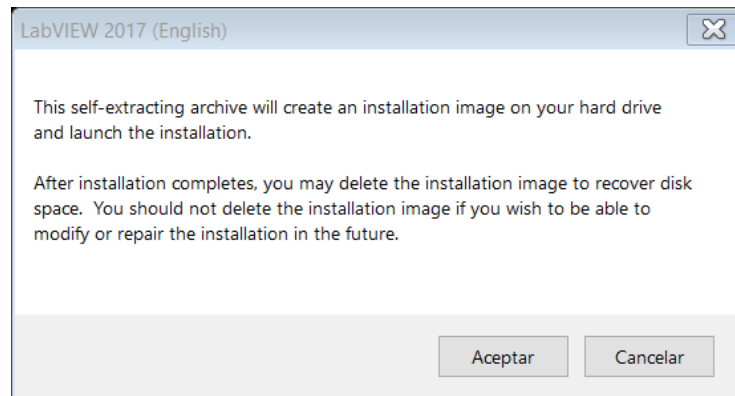
Anexo 20. Asistente de instalación indica todos los cambios que se van a efectuar al momento de instalar el software ni visa. Dar clic en next.



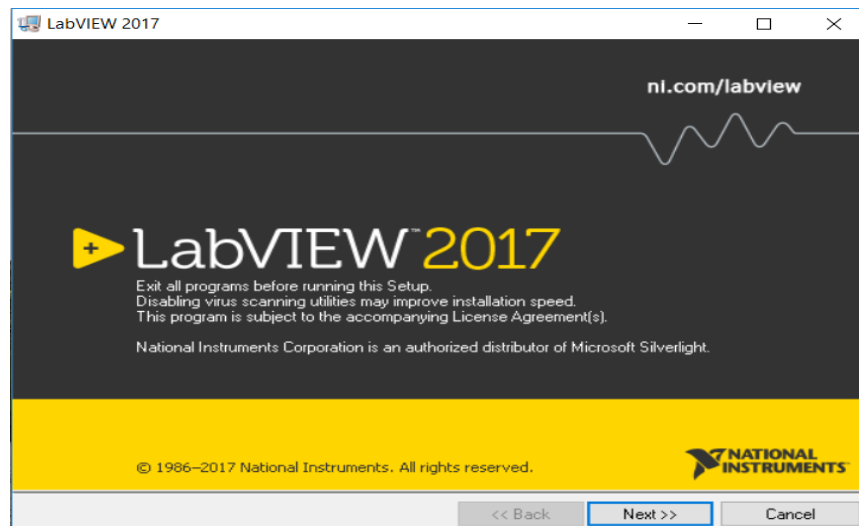
Anexo 21. Al reiniciar el computador el firewall de Windows preguntara si se habilita los puertos y establecer los respectivos permisos de red para crear excepciones de seguridad y que los mismos no sean bloqueados al momento de generar su ejecución. Dar clic en permitir acceso.



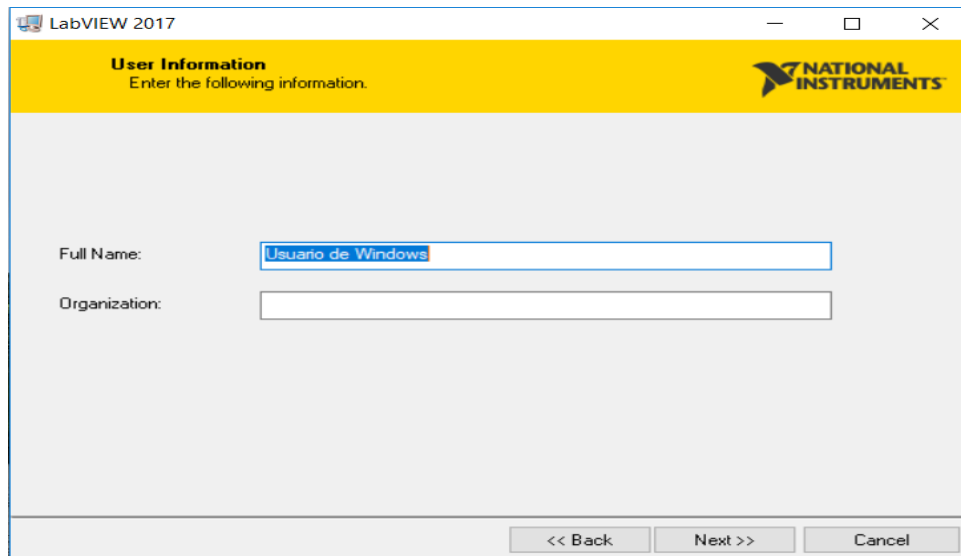
Anexo 22. Software labview (2017lv-wineng.exe) para windows de 32bits.



Anexo 23. Software labview (2017lv-wineng.exe) para windows de 32bits.

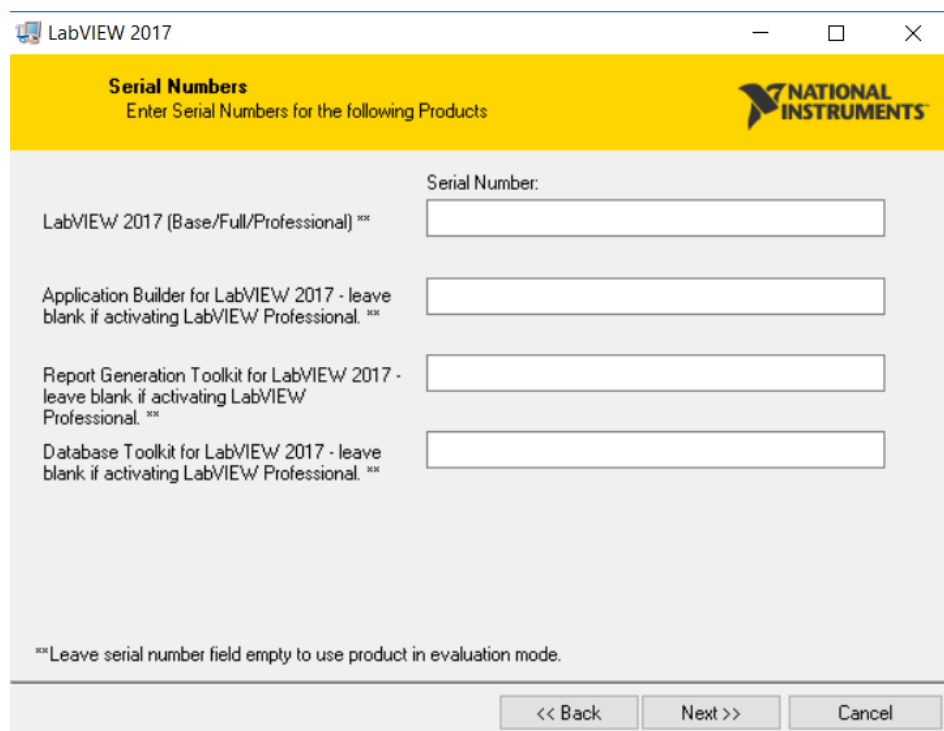


Anexo 24. Parámetros de registro.



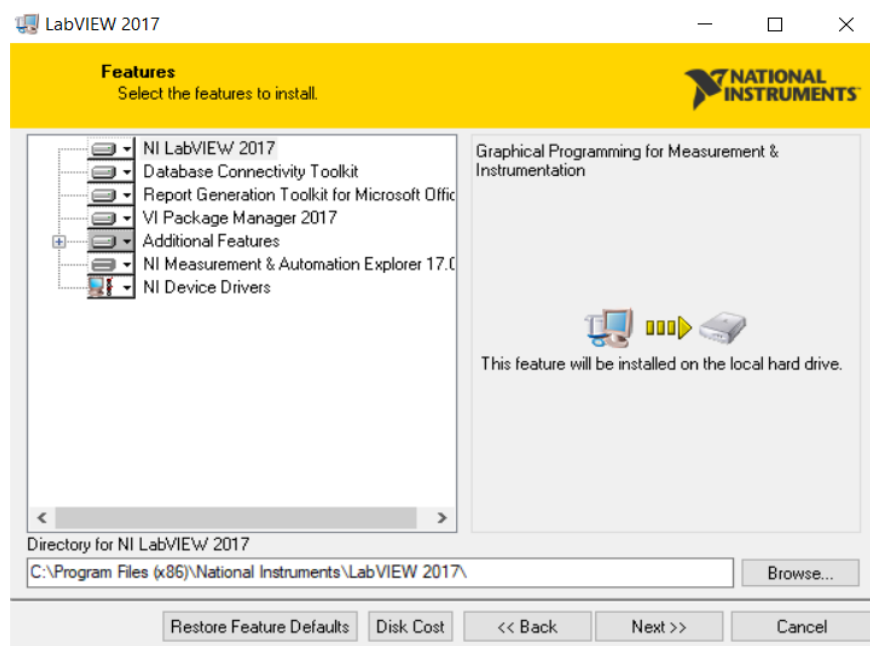
The screenshot shows the 'User Information' registration window in LabVIEW 2017. The window title is 'LabVIEW 2017'. The header is yellow and contains the text 'User Information' and 'Enter the following information.' along with the National Instruments logo. The main area is light gray and contains two text input fields: 'Full Name:' with the value 'Usuario de Windows' and 'Organization:'. At the bottom, there are three buttons: '<< Back', 'Next >>', and 'Cancel'.

Anexo 25. Se procede a llenar los parámetros de licenciamiento del software.

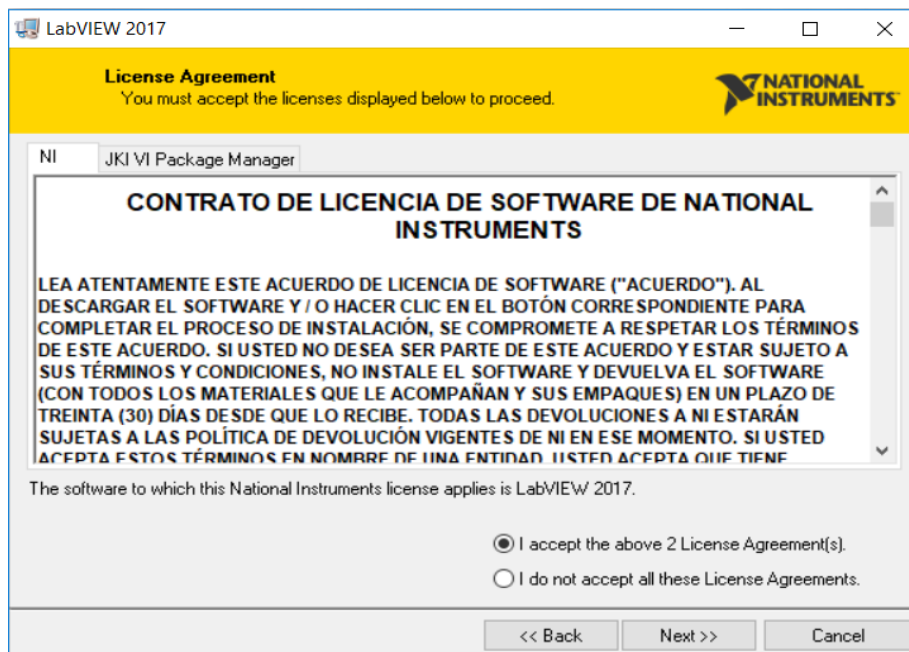


The screenshot shows the 'Serial Numbers' registration window in LabVIEW 2017. The window title is 'LabVIEW 2017'. The header is yellow and contains the text 'Serial Numbers' and 'Enter Serial Numbers for the following Products' along with the National Instruments logo. The main area is light gray and contains four rows of text and input fields. The first row is 'LabVIEW 2017 (Base/Full/Professional) **' with a 'Serial Number:' label and an empty input field. The second row is 'Application Builder for LabVIEW 2017 - leave blank if activating LabVIEW Professional. **' with an empty input field. The third row is 'Report Generation Toolkit for LabVIEW 2017 - leave blank if activating LabVIEW Professional. **' with an empty input field. The fourth row is 'Database Toolkit for LabVIEW 2017 - leave blank if activating LabVIEW Professional. **' with an empty input field. At the bottom, there is a note: '**Leave serial number field empty to use product in evaluation mode.' and three buttons: '<< Back', 'Next >>', and 'Cancel'.

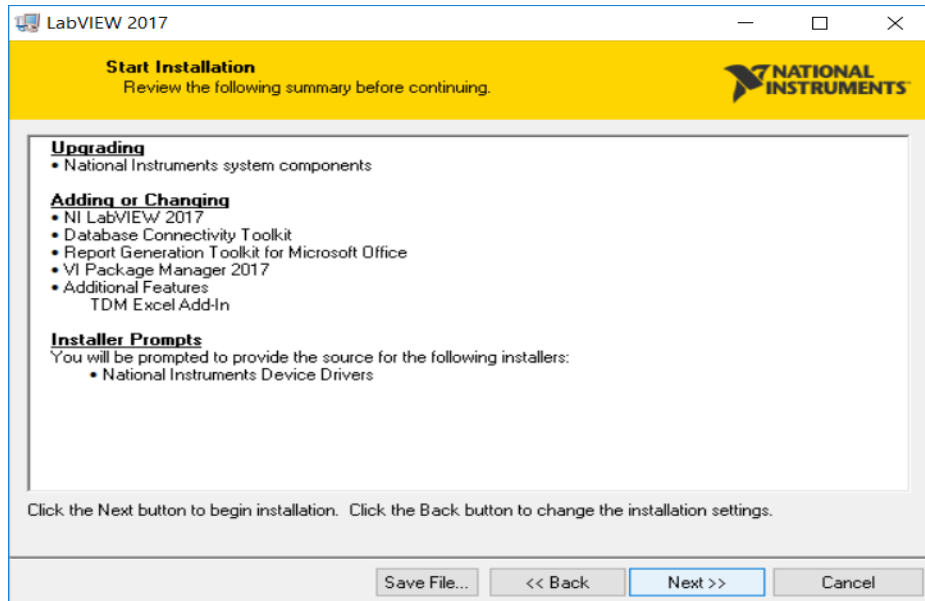
Anexo 26. Se procede a llenar los parámetros de licenciamiento del software.



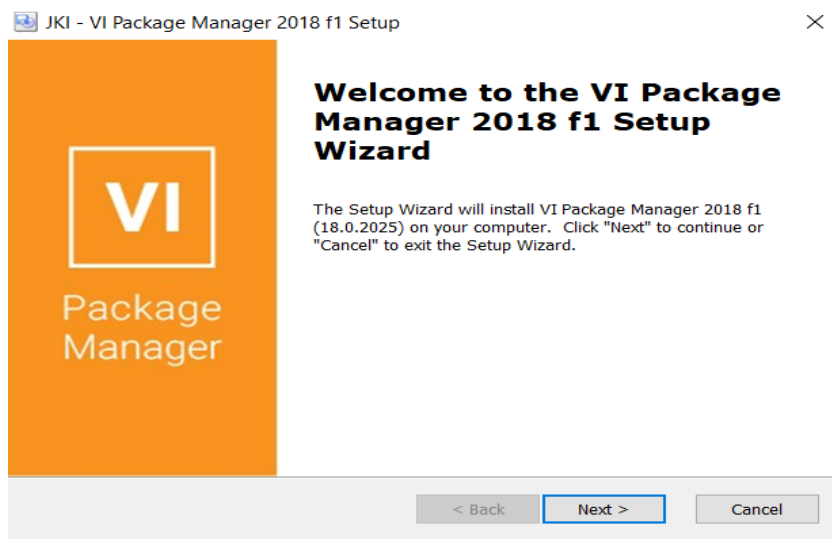
Anexo 27. Contrato de licencia del software.



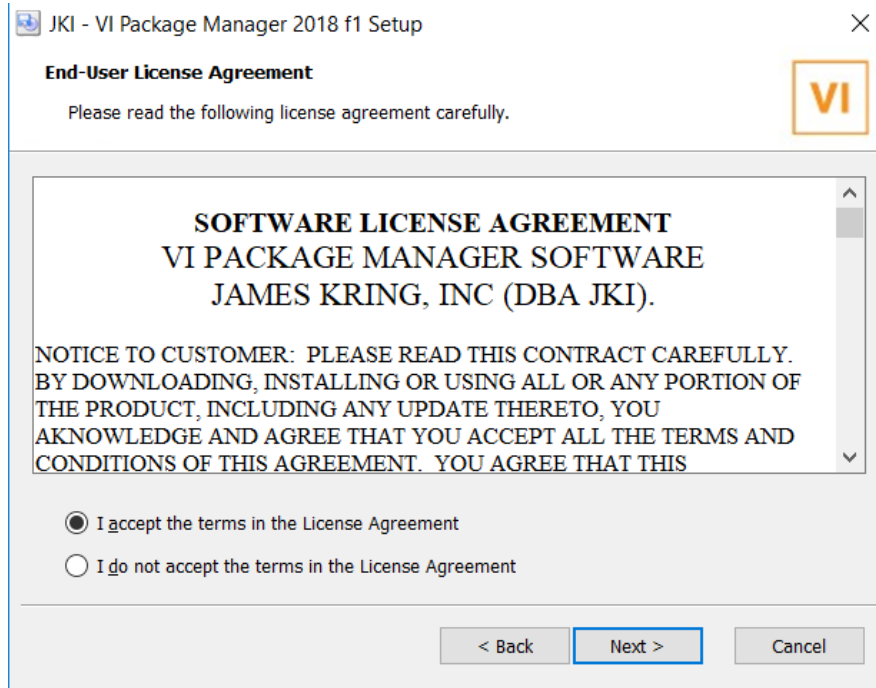
Anexo 28. Revisión de parámetros a instalar.



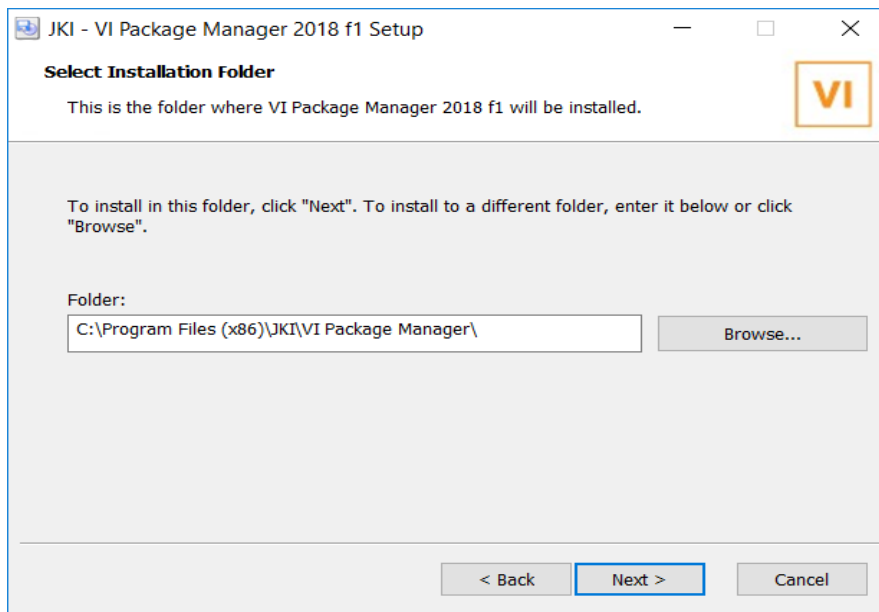
Anexo 29. Instalación del VI Package Manager, es importante detallar que para que el proceso de instalación se ejecute con todas sus actualizaciones correspondientes se debe contar con una conexión a internet.



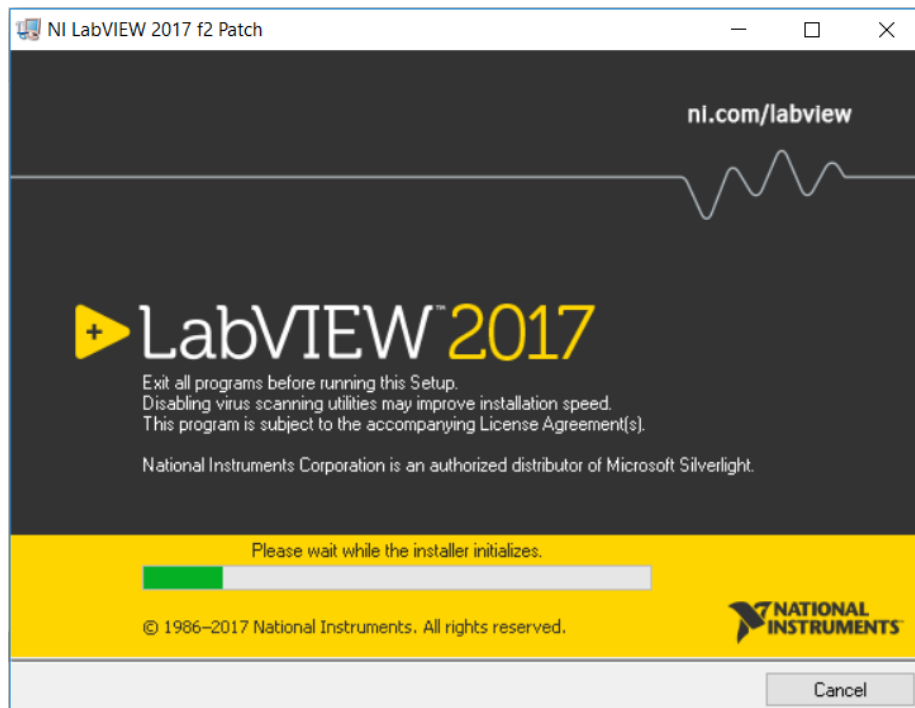
Anexo 30. Se establecen los términos del contrato.



Anexo 31. Se establece la ruta de instalación.



Anexo 32. Instalación de la actualización correspondiente en Labview.



Anexo 33. Inicio al software de labview a continuación se presenta la interfaz del software instalado.



