



“Responsabilidad con pensamiento positivo”

UNIVERSIDAD TECNOLÓGICA ISRAEL

**TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:
INGENIERO EN ELECTRÓNICA DIGITAL Y
TELECOMUNICACIONES**

TEMA:

**MÓDULO AUTÓNOMO PARA EL APRENDIZAJE DE LOS PRINCIPIOS
BÁSICOS DE LA CINEMÁTICA DE UNA PARTÍCULA EN EL ÁMBITO
UNIVERSITARIO**

AUTOR:

RIVERA GALLARDO MIGUEL ANGEL

TUTOR: ING. MAURO FERNANDO BOLAGAY EGAS, Mg.

QUITO-ECUADOR

2020

DECLARACIÓN

Yo, Miguel Ángel Rivera Gallardo, estudiante de la Universidad Tecnológica Israel de la carrera de Electrónica y Telecomunicaciones, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; que he consultado las referencias bibliográficas que se incluyen en este documento, y que no he plagiado dicha información.

MIGUEL ÁNGEL RIVERA GALLARDO

C.I.: 1002898417-7

UNIVERSIDAD TECNOLÓGICA ISRAEL

CERTIFICACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación **“MÓDULO AUTÓNOMO PARA EL APRENDIZAJE DE LOS PRINCIPIOS BÁSICOS DE LA CINEMÁTICA DE UNA PARTÍCULA EN EL ÁMBITO UNIVERSITARIO”**, presentado por el Sr. Miguel Ángel Rivera Gallardo, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M. Marzo del 2020

TUTOR

.....

Ing. Mauro Fernando Bolagay Egas, Mg.

AGRADECIMIENTO

Mi enorme y profundo agradecimiento a Dios por brindarme la oportunidad de aprender y seguir superándome en mis objetivos.

A mi familia que, con su apoyo en todas las etapas de mi vida, con sus palabras de afecto, amor y su aliento para no desmayar en los momentos difíciles.

También a mi querida Universidad Tecnológica Israel, por brindarme una formación académica y personal de calidad.

“Debemos encontrar tiempo para detenernos y agradecer a las personas que hacen la diferencia en nuestras vidas”. John F. Kennedy

DEDICATORIA

Dedicar este proyecto de grado a mi esposa Maria del Carmen Armas quien con su paciencia, apoyo y amor supo darme aliento para continuar y no desfallecer en los momentos difíciles, a mis hijos para José David y Ariana Gabrielle sigan mis pasos de dedicación esfuerzo en todo ámbito de su vida, a mis padres José Antonio y Rosa Eva que con sus palabras de sabias supieron ser en mí una guía en mi vida y mis Hermanos Paulo Cesar y Maria José por sus palabras de apoyo supieron darme aliento en los momentos difíciles. A todos, mis más sinceros agradecimientos por ayudarme en esta etapa de mi vida

“Hagas lo que hagas, hazlo bien “Abraham Lincoln.

ÍNDICE DE CONTENIDO

DECLARACIÓN	ii
CERTIFICACIÓN DEL TUTOR	iii
AGRADECIMIENTO.....	iv
DEDICATORIA	v
ÍNDICE DE CONTENIDO.....	vi
LISTA DE FIGURAS	x
LISTA DE TABLAS.....	xiv
RESUMEN.....	xv
ABSTRACT.....	xvi
INTRODUCCIÓN	1
Antecedentes.....	1
Planteamiento del problema	2
Objetivos.....	4
Objetivo general.....	4
Objetivos específicos:.....	4
Alcance.....	4
Justificación.....	5
CAPÍTULO 1	7
1 FUNDAMENTACIÓN TEÓRICA.....	7
1.1 Aspectos generales de cinemática.....	7
1.1.1 Velocidad media	8
1.1.2 Movimiento rectilíneo uniforme.....	8
1.1.3 Aceleración media	9
1.1.4 Movimiento rectilíneo uniformemente variado	10
1.2 Equipos de laboratorio para el estudio de movimientos rectilíneos	11
1.2.1 Celda de carga HX711	12
1.2.3 Acelerómetro y giroscopio MPU6050.....	13
Fuente	15
1.2.4 Sensor de fuerza y presión FSR402.....	15

1.2.5 Sensor ultrasónico HCSR 04	16
1.2.6 Módulo bluetooth HC06	17
1.2.7 Sensor infrarrojo TCRT5000	18
1.2.8 Controlador L298N.....	19
1.2.9 Arduino Nano y Mega	20
1.2.10 Display LCD TFT 3.2” Open-Smart	22
1.2.11 Fuente de alimentación	23
1.2.12 Regulador de Voltaje 7805	23
1.2.13 Pantalla LCD 16x2 con I2C.....	24
1.2.14 Software LabVIEW	25
CAPÍTULO 2.....	26
2 MARCO METODOLÓGICO	26
2.1 Métodos de investigación.....	26
2.2 Tipos de investigación.....	27
CAPÍTULO 3.....	28
3. PROPUESTA	28
3.1 Descripción general del proyecto.....	28
3.2 Diagrama de bloques.....	29
3.3 Estructura de plataforma	30
3.4 Módulo autónomo	31
3.5 Fuente de alimentación 12v 30A.....	31
3.6 Microcontrolador y sensores	33
3.6.1 Microcontrolador Arduino Mega 2560 Rev3.....	33
3.6.2 Microcontrolador Arduino Nano.....	35
3.6.3 Pantalla TFT LCD OpenSmart 3.2” táctil.....	36
3.6.4 Módulo Puente H L298n.....	37
3.6.5 Regulador de Voltaje LM7805.....	38
3.6.6 Motor Bipolar.....	38
3.6.7 Sensor Ultrasónico HC-sr04	39
3.6.8 Pantalla LCD 16x2 I2C.....	41
3.6.9 Celda de carga y HX711	42
3.6.10 Sensor MPU6050 Acelerómetro y Giroscopio	43
3.6.11 Sensor TCRT5000 infrarrojo	44

3.6.12 Sensor de fuerza FSR402	45
3.6.13 Bluetooth Hc-06	46
3.7 Diseño de placas.....	46
3.7.1 Placa base de control de torres	46
3.7.2 Placa base de modulo autónomo	48
3.7.3 Diseño de soporte de motores y carcasa de módulo autónomo.	50
3.8 Diseño de interfaz gráfica en LabVIEW	52
3.9 Análisis de costos	55
3.10 Ventajas del Módulo autónomo.	55
CAPITULO 4	57
4. IMPLEMENTACIÓN	57
4.1 Ensamblaje, programación y comprobación de los componentes	57
4.1.1 Control torres.....	57
4.1.2 Instalación, programación de los puentes L298N y motores bipolares	59
4.1.3 Instalación y programación de pantalla OpenSmart TFT 3.2”	62
4.1.4 Regulador de voltaje 7805	63
4.1.5 Sensor HC-sr04 Ultrasonido.....	64
4.1.6 Ensamblaje del control torres.	65
4.1.7 Ensamblaje de estructura.	69
4.1.8 Interfaz gráfica de módulo.....	70
4.2 Ensamblaje, programación y comprobación de módulo autónomo.	75
4.2.1 Programación de módulo autónomo.....	80
4.2.2 Calibración de modulo MPU6050	80
4.2.3 Programación de módulo autónomo.....	81
4.2.4 Aplicativo LabVIEW para la lectura de datos.....	82
4.2.5 Practica MRU	84
4.2.6 Practica MRUV	89
CONCLUSIONES	95
RECOMENDACIONES	97
Referencias bibliográficas.....	98
ANEXOS.....	101
Anexo 1 Manual Usuario.....	101
Anexo 2 Diseño PCB módulo control torres.....	102
Anexo 3 Diseño PCB módulo autónomo	102
Anexo 4 Diseño LabVIEW	103
Anexo 4 Programación módulo autónomo.....	104

Anexo 5 Programación módulo control torres 104

LISTA DE FIGURAS

Figura 1.1 Equipos de laboratorio para el estudio de movimiento rectilíneos.	11
Figura 1.2. Celda de carga y transmisor.....	13
Figura 1.3. Conexión celda de carga.....	13
Figura 1.4. Sensor MPU6050.....	15
Figura 1.5. Comunicación entre sensor MPU 6050 y Arduino.....	15
Figura 1.6. Sensor de fuerza y presión.....	16
Figura 1.7. Sensor ultrasónico HCSR 04.	17
Figura 1.8. Módulo bluetooth HC06.	18
Figura 1.9. Sensor infrarrojo TCRT 5000.....	19
Figura 1.10. Puente H L298N.	20
Figura 1.11. Arduino nano.	21
Figura 1.12. Arduino Mega2650 rev3.....	22
Figura 1.13. Display OpenSmart 3.2" TFT-LCD.	22
Figura 1.14. Fuente de Alimentación 12v 30 A.	23
Figura 1.15. Regulador de voltaje 7805.....	24
Figura 1.16. Pantalla LCD 16x2.	24
Figura 1.17. Plataforma LabVIEW.	25
Figura 3.1 Diagrama de Bloques del módulo autónomo.....	29
Figura 3.2 Estructura de control de inclinación	30
Figura 3.3 Módulo autónomo de simulación de partícula.	31
Figura 3.4 Fuente de alimentación 12v 30A.	32
Figura 3.5 Arduino MEGA 2560R3.....	34
Figura 3.6 Arduino Nano.	35
Figura 3.7 Pantalla TFT OpenSmart 3.2"	36
Figura 3.8 Conexión L298N	37
Figura 3.9 Conexión de regulador de voltaje 7805.....	38
Figura 3.10 Conexión motor Bipolar	39
Figura 3.11 Funcionamiento de Motor.....	39
Figura 3.12 Conexión Sensor ultrasónico HCSR04.....	40
Figura 3.13 Conexión Pantalla LCD 16x2 I2C	41
Figura 3.14 Conexión celda de carga y HX711	42
Figura 3.15 Conexión de MPU6050.	43

Figura 3.16 Conexión de Sensor Infrarrojo.....	44
Figura 3.17 Conexión de FSR.....	45
Figura 3.18 Conexión HC-06.....	46
Figura 3.19 Diseño de Placa base Control torres	47
Figura 3.20 Diseño de PCB placa base Control Torres	47
Figura 3.21 Diseño de módulo autónomo.....	48
Figura 3.22 Diseño PCB módulo autónomo.	49
Figura 3.23 Contenedor de Motor bipolar y sensor ultrasónico.....	50
Figura 3.24 Diseño de módulo autónomo.....	51
Figura 3.25 Diseño de torres y Rampa.....	52
Figura 3.26 Programación de interfaz en LabVIEW	53
Figura 3.27 Interfaz de captura de datos	54
Figura 4.1 Distribución de componentes de la placa base	58
Figura 4.2 Diseño PCB placa base.....	59
Figura 4.3 Programación para los L298N.....	60
Figura 4.4 Montaje del L298N en la placa base control	61
Figura 4.5 Instalacion de motor Bipolar	61
Figura 4.6 Programación de pantalla OpenSmart TFT 3.2"	62
Figura 4.7 Montaje de OpenSmart con Arduino Mega2560r3	63
Figura 4.8 Regulador de voltaje 5v DC	63
Figura 4.9 Programación de HC-sr04	64
Figura 4.10 Instalación de HC-sr04.....	65
Figura 4.11 Instalación ejes de torres.....	66
Figura 4.12 Bases de las torres.....	66
Figura 4.13 Instalacion de parte superior de torres	67
Figura 4.14 Soportes laterales	67
Figura 4.15 Instalación de rampa	68
Figura 4.16 Marcas métricas y Cableado.....	68
Figura 4.17 Instalación de Fuente de alimentación y Control Torres	69
Figura 4.18 Estructura terminada.....	69
Figura 4.19 Estructura terminada.....	70
Figura 4.20 Pantalla de inicio de módulo.....	71
Figura 4.21 Pantalla principal del módulo del control.....	71
Figura 4.22 Pantalla MRU nivelación de la rampa.....	72

Figura 4.23 Pantalla lista para iniciar la práctica	72
Figura 4.24 Inicio de la práctica.....	73
Figura 4.25 Ingreso a configuración MRUV	73
Figura 4.26 Pantalla de configuración de MRUV	74
Figura 4.27 Diseño de la estructura del módulo autónomo y placa base PCB.	75
Figura 4.28 Estructura externa del módulo autónomo	75
Figura 4.29 Placa base módulo autónomo	76
Figura 4.30 Pantalla LCD 16x2	76
Figura 4.31 Celda de carga y transductor.....	77
Figura 4.32 Instalación de Sensor TCRT5000.....	77
Figura 4.33 Sensor MPU6050.....	78
Figura 4.34 Sensor Fsr402	78
Figura 4.35 Colocación de Arduino Nano y Hc-06 Bluetooth.....	79
Figura 4.36 Funcionamiento de todos los componentes módulo autónomo.....	79
Figura 4.37 Módulo autónomo.....	80
Figura 4.38 Calibración de MPU6050	81
Figura 4.39 Programación de módulo autónomo.....	81
Figura 4.40 Configuración de conexión serie	82
Figura 4.41 Conexión y datos obtenidos.....	82
Figura 4.42 Pantalla del aplicativo.....	83
Figura 4.43 Aplicativo LabVIEW de captura de datos	83
Figura 4.44 Toma de tiempo del movimiento.....	84
Figura 4.45 Movimiento del Módulo autónomo	85
Figura 4.46 Botón MRU	86
Figura 4.47 Colocación de módulo autónomo en rampa.	86
Figura 4.48 Encendido de módulo autónomo.	87
Figura 4.49 Conexión serial.	87
Figura 4.50 Botón inicio de práctica	87
Figura 4.51 Resultados de la práctica.	88
Figura 4.52 Tiempo de la práctica.....	89
Figura 4.53 Botón MRU	90
Figura 4.54 Control de inclinación.....	91
Figura 4.55 Colocación de módulo autónomo en rampa.	91
Figura 4.56 Encendido de módulo autónomo.	91

Figura 4.57 Conexión serial.....	92
Figura 4.58 Botón inicio de práctica.....	92
Figura 4.59 Resultados de la práctica.....	93

LISTA DE TABLAS

Tabla 3.1 Consumo de componentes	33
Tabla 3.2 Características Arduino Mega R3	34
Tabla 3.3 Características de Arduino Nano.	35
Tabla 3.4 Características Shield OpenSmart 3.2"	36
Tabla 3.5 Características L298N.....	37
Tabla 3.6 Características de LM7805	38
Tabla 3.7 Características principales de HCsr04	41
Tabla 3.8 Características de LCD 16x2	42
Tabla 3.9 Conexión entre celda y HX711	43
Tabla 3.10 Conexión MPU6050	44
Tabla 3.11 Conexión de Sensor TCRT5000	45
Tabla 3.12 Costos de implementación de proyecto	55
Tabla 4.1 Tabla comparativa de resultados.....	88
Tabla 4.2 Tabla comparativa de resultados.....	93
Tabla 4.3 Costo de implementación.....	94

RESUMEN

Este proyecto describe la realización de un módulo autónomo para el aprendizaje de los principios de la cinemática de una partícula en el ámbito universitario, en esta investigación realizada se determinó que las variables que intervienen en la cinemática tales como: el espacio, el tiempo, la gravedad, el peso y la fuerza son variables de fácil interpretación, cuando la captación de estos datos se realiza con dispositivos electrónicos como sensor Infrarrojo Fc51, Sensor MPU6050, sensor carga HX711, sensor de fuerza MF01 estos a su vez conectados a un Arduino nano Rev3, datos procesados y transmitidos vía Bluetooth hc-06 un software en LabVIEW que mostrara los datos en pantalla en tiempo real, así el estudiante podrá interpretar los datos y comparar los resultados entre los datos obtenidos de forma teórica y la forma práctica, para así asimilar mejor los conceptos de la cinemática. Para abordar esto, fue necesario el estado del arte y los diferentes dispositivos y componentes electrónicos y los diferentes softwares que permitan el desarrollo del proyecto. La conclusión más relevante es pese a que en el mercado existen este tipo de equipos de laboratorio, existen alternativas de bajo costo tanto el hardware como software que permiten desarrollar satisfactoriamente un módulo autónomo para el aprendizaje de los principios básicos de la cinemática de una partícula en el ámbito universitario

Palabras clave: módulo autónomo, aprendizaje, cinemática de una partícula, ámbito universitario

ABSTRACT

This project describes the realization of an autonomous module for learning the principles of kinematics of a particle in the university environment. In this research, it was determined that the variables involved in kinematics such as space, time, gravity, weight and force are variables of easy interpretation, when the capture of these data is done with electronic devices such as infrared sensor Fc51, sensor MPU6050, sensor HX711 load, force sensor MF01 these in turn connected to a nano Rev3 Arduino, data processed and transmitted via Bluetooth hc-06 a software in LabVIEW that will show the data on screen in real time, so the student will be able to interpret the data and compare the results between the data obtained in a theoretical way and the practical way, in order to better assimilate the concepts of kinematics. To address this, it was necessary the state of the art and the different electronic devices and components and the different software that allow the development of the project. The most important conclusion is that, despite the existence of this type of laboratory equipment in the market, there are low-cost alternatives, both hardware and software, which allow the satisfactory development of an autonomous module for learning the basic principles of kinematics of a particle in the university environment.

Translated with www.DeepL.com/Translator (free version)

Keywords: autonomous module, learning, kinematics of a particle, university environment

INTRODUCCIÓN

Antecedentes

Las nuevas tecnologías han generado grandes cambios en la educación, ya que los estudiantes de la actualidad son considerados nativos digitales, por lo que la utilizan sin ningún tipo de problemas, es por eso que al desarrollar herramientas tecnológicas con fines educativos los estudiantes se involucran más en los procesos de enseñanza – aprendizaje. Es así como el movimiento tecnológico se ha insertado en la educación, por otra parte, los docentes la consideran como una necesidad, para buscar nuevas experiencias en el ámbito educativo y pedagógico, por eso adquieren capacidades tecnológicas con la finalidad de formar a los nuevos profesionales como personas competitivas en el ámbito laboral y profesional.

Al integrar efectivamente la tecnología en las áreas temáticas, los roles de los docentes crecen (son asesores, expertos en contenido y entrenadores) y permite que el aprendizaje sea más significativo y divertido. Tener acceso a otra información fuera del aula ofrece a los estudiantes diferentes formas de aprender. Los docentes encuentran formas creativas de enseñar a sus estudiantes que los mantengan comprometidos.

Las nuevas tecnologías construyen al fomento de entornos de aprendizaje en un encuadre constructivista, que potencian tanto las expectativas de los estudiantes como su capacidad de análisis e interpretación al utilizar los modelos que explican el comportamiento de un fenómeno físico.

En el ámbito mundial varias empresas se han dedicado a investigar y crear soluciones tecnológicas que facilitan la experimentación en un laboratorio como es el caso de PASCO que, en la división de Física, ha creado una serie de interfaces, sensores y software de análisis para conectar a los estudiantes directamente con los conceptos y leyes de la física. Las construcciones completas de laboratorios tecnológicos para física integran la tecnología con la ciencia para desarrollar un nivel más profundo de conocimiento, dirigido al aprendizaje integral del profesional.

Así también, en los laboratorios de Vernier (USA) se han desarrollado soluciones basadas en la tecnología, que sirven a los educadores y estudiantes a nivel mundial, los

laboratorios presentan una solución completa de sensores, experimentos y recursos, que permiten el éxito en la enseñanza y el aprendizaje de los estudiantes.

En la Universidad de los Andes de Colombia, desarrolló una página web específicamente para los laboratorios de física de pregrado, en esta página se puede ver videos de varias prácticas de laboratorio para la enseñanza, además informan en que consiste el experimento, como se va a realizar y que se espera conseguir en la práctica, ellos utilizan la tecnología de PASCO para realizar las prácticas.

En la Universidad de San Pedro de Sula con sede en Honduras tiene laboratorios de física que refuerzan el aprendizaje de los estudiantes, porque proponen tres grandes metas, ilustrar el contenido de las clases teóricas, enseñar las técnicas experimentales y promover actitudes científicas. Ellos poseen una estructura tecnológica desarrollada en Honduras, debido a lo costoso de los equipos comercializados internacionalmente. De manera que, basado en los avances tecnológicos, se desarrollan laboratorios donde por medio de la acción y el descubrimiento los estudiantes aprendan haciendo, es decir se persigue que el estudiante pueda con la experiencia directa con un montaje práctico estimular su pensamiento y que con su conocimiento previo pueda generar un aprendizaje significativo.

En el marco de desarrollo de estas actividades se considera viable realizar estudios y análisis que permitan adelantar acciones para captar fenómenos y de procesos en un laboratorio, para el caso específico del campo de la física, en este caso el movimiento rectilíneo uniforme (MRU) y rectilíneo uniformemente acelerado (MRUV).

Planteamiento del problema

El desarrollo de la electrónica, control, automatización y comunicación ha permitido la implementación de instrumentos pedagógicos que aportan de manera significativa al proceso de enseñanza y aprendizaje. Los estudiantes aprenden y asimilan mejor los conceptos cuando experimentan y aplican la teoría, este método pedagógico conocido como aprendizaje experimental provee la capacidad del individuo para aprender de su propia experiencia, siempre respaldado de un marco conceptual, idóneo para el fenómeno específico.

No obstante, la falta de equipos de laboratorio que faciliten la experimentación para comprobar los conceptos teóricos expuestos por el docente no permite relacionar la teoría con la práctica, lo cual contrasta con la filosofía de Piaget, quien asegura que el aprendizaje es un proceso donde el individuo, a través de la experiencia, manipulación, y la interacción construye el conocimiento.

Por lo antes expuesto, se deduce que los estudiantes, se limitan a memorizar conceptos, no construyen su aprendizaje, lo que trae como consecuencia una desventaja con otros alumnos, que tuvieron la oportunidad de construir su conocimiento mediante la experiencia práctica en laboratorios.

En ese sentido, se pretende desarrollar un módulo autónomo para el aprendizaje de los principios básicos de la cinemática de una partícula en el ámbito Universitario, específicamente para el análisis de los movimientos MRU y MRUA, mediante riel de prueba, sensores para la adquisición de datos, una interfaz de acondicionamiento de datos, que conjuntamente con un software de análisis permitan el análisis de los datos del experimento desarrollado, y cuyas graficas sean mostradas para la interpretación de los mismos por los estudiantes. Esto coincide con la teoría constructivista, donde el estudiante mediante la experimentación pueda construir su propio aprendizaje basado en los conocimientos previos, cabe, aclarar que el termino Módulo Autónomo, debe ser interpretado, como un sistema integrado de experimentación donde no es necesario la manipulación del alumno para tomar los datos, ya que mediante sensores se encargará de realizar de manera autónoma la adquisición y procesamiento de los datos, también permite su almacenamiento y posterior entrega a los estudiantes, para que el mismo realice los cálculos necesarios para contrastar los resultados experimentales con los teórico y los producidos por el módulo autónomo.

En ese sentido, el aprendizaje de los principios básicos de la cinemática de una partícula en el ámbito Universitario, se realiza de una forma innovadora, facilitando el proceso enseñanza y de aprendizaje, de los principios de la dinámica y la institución estaría a la vanguardia en la enseñanza.

Objetivos

Objetivo general

Desarrollar un módulo autónomo para el aprendizaje de los principios básicos de la cinemática de una partícula en el ámbito universitario.

Objetivos específicos:

Establecer los recursos tecnológicos necesarios para su incorporación en las prácticas de laboratorio de Física.

Diseñar el sistema de carriles, poleas, accesorios de sujeción, carros de baja fricción, sensor de movimiento, contador de tiempo, para el estudio del Movimiento Rectilíneo Uniforme (MRU) y Movimiento Rectilíneo Uniforme Acelerado (MRUA).

Crear el módulo autónomo, con todos sus componentes para el estudio de la cinemática de una partícula MRU y MRUA.

Realizar pruebas de funcionamiento y validación del módulo autónomo para el aprendizaje de los principios básicos de la cinemática de una partícula.

Alcance

Para la realización de este proyecto se aplica conocimientos de física, electrónica, diseño electrónico y cinemática. Este laboratorio consta de un vehículo que simula a la partícula, el cual contiene los sensores que captan las variables como tiempo, distancia, peso, inclinación y aceleración, también posee un riel para desplazamiento del vehículo y estudiar el movimiento en un solo plano, también se desarrollará una interfaz de comunicación con el computador para que mediante un software se analice y grafique el fenómeno en el computador.

Justificación

Con el diseño y construcción de este módulo autónomo para el aprendizaje de la cinemática, se proporciona datos en tiempo real, que el estudiante podrá visualizar en la pantalla, los cuales le permitirá anotarlos en su cuaderno de prácticas para realizar los cálculos según el modelo matemático que rige el fenómeno estudiado y de esta forma logren comparar resultados prácticos con los teóricos, también podrán cambiar algunas condiciones del experimento (manipular las variables) para poder constatar el efecto de una variable sobre otra, a través de los resultados obtenidos.

En este proyecto se desarrollará un vehículo que en física se conoce como la partícula y que simula los movimientos en el espacio - tiempo y las fuerzas que le afectan o modifican, todo esto captado por sensores electrónicos y procesados por un microcontrolador (Arduino nano) y enviados a un software en Labview para luego ser mostrados en pantalla.

Referente al aporte científico esta investigación se justifica teóricamente mediante los conceptos de la cinemática y la electrónica que servirán como apoyo en la elaboración de nuevas plataformas de aprendizaje. Desde el punto de vista tecnológico se constituye como herramienta invaluable para la experimentación y conceptualización de los fenómenos de la cinemática en una dirección, debido a su versatilidad y bajo costo.

Desde la visión del impacto en el ámbito estudiantil, el proyecto es muy significativo ya que el estudiante podrá manipular variables que comprenden la cinemática y ver los resultados modificados en tiempo real en pantalla.

Dentro de las principales ventajas de esta investigación, está el aspecto económico ya que un laboratorio comercial con las mismas prestaciones es costoso tal como laboratorio de dinámica de Vernier que ronda los USD. 1200, en este aspecto este módulo se construye con materiales de fácil acceso y dispositivos electrónicos de bajo costo.

Descripción de los capítulos.

En el capítulo I se realiza la fundamentación teórica de la cinemática y del comportamiento de la partícula en el espacio-tiempo y las fuerzas que le afectan a la misma también de los componentes electrónicos que utilizaremos para la captación de datos. En el capítulo II, se describe la metodología que se utiliza para desarrollar este proyecto. Sigue el

capítulo III, donde se describe la propuesta de la investigación y se indica el diagrama de bloques de construcción del módulo, su estructura, los componentes electrónicos, los componentes de control, las partes de la partícula, plan de costos y ventajas de este proyecto.

Finalmente, en el capítulo IV, se realiza los diseños de las placas electrónicas, la programación y verificación de funcionamiento de los sensores, la comunicación entre Arduino Nano y LabVIEW y así mostrar la información en pantalla.

CAPÍTULO 1

1 FUNDAMENTACIÓN TEÓRICA

1.1 Aspectos generales de cinemática

La física es una ciencia que estudia los fenómenos naturales para establecer modelos analíticos que predicen leyes y comportamientos mediante observaciones experimentales y de mediciones cuantitativas de los procesos reales. El estudio de la física comienza desde los ámbitos de la experiencia cotidiana mediante el razonamiento que procede hacia lo abstracto. La cinemática es el estudio del movimiento relativo de una partícula que excluye la geometría (forma y tamaño) y las causas que generaron el movimiento (fuerzas externas principalmente) en un cuerpo. (Sears, Zemansky, & Young, 2010)

Aristóteles sostuvo que hay dos tipos de movimiento para la materia. El primero es el movimiento antinatural que se genera cuando se empuja un objeto, y en este caso la velocidad del movimiento es proporcional a la fuerza del empuje. El segundo es el movimiento natural que es cuando un cuerpo busca su lugar natural en el universo, como la caída de una piedra. Para el movimiento natural de los objetos pesados que caen a la tierra, Aristóteles afirmó que la velocidad de la caída era proporcional al peso e inversamente proporcional a la densidad del medio por el que el cuerpo estaba cayendo. También mencionó que hubo cierta aceleración, porque el cuerpo se acercó más a su propio elemento, su peso aumentó y se aceleró. Sin embargo, estos comentarios en Aristóteles son muy breves y vagos, y ciertamente no son cuantitativos. (Resnick, Halliday, & Krane, 2001)

En realidad, estos puntos de vista de Aristóteles no quedaron sin respuesta incluso en la antigua Atenas. Aproximadamente treinta años después de la muerte de Aristóteles, Estratón de Lampsaco señaló que una piedra caída desde una altura mayor tuvo un mayor impacto en

el suelo, lo que sugiere que la piedra recogió más velocidad a medida que cayó desde la altura más alta. (Resnick, Halliday, & Krane, 2001)

Después de XVII siglos, Galileo expuso ideas sobre la caída de cuerpos, y sobre proyectiles en general, en el libro "Dos nuevas ciencias". Los dos eran la ciencia del movimiento, que se convirtió en la piedra angular de la física, y la ciencia de los materiales y la construcción, una contribución importante a la ingeniería. (Serway & Jewett, 2014)

Galileo formuló la ley de caída libre que predecía que los cuerpos del mismo material que caen a través del mismo medio caerían a la misma velocidad. Esto era contrario a lo que Aristóteles había señalado. Galileo llegó a su hipótesis mediante un famoso experimento mental descrito en el libro Un Movimiento. Este experimento se desarrolla de la siguiente manera: imagina que dos objetos, uno ligero y otro más pesado que el otro, están conectados entre sí por una cuerda. Suelta este sistema de objetos desde lo alto de una torre. Al asumir que los objetos más pesados caen más rápido que los más ligeros (y, a la inversa, los objetos más ligeros caen más lentamente), la cuerda pronto se tensará a medida que el objeto más ligero retrase la caída del objeto más pesado. Pero el sistema considerado en su conjunto es más pesado que el objeto pesado solo y, por lo tanto, debería caer más rápido. Esta contradicción lleva a uno a concluir que la suposición es falsa. (Serway & Jewett, 2014)

1.1.1 Velocidad media

Al considerar una partícula moviéndose sobre una línea recta se supone que en el instante inicial (t_i) se encuentra en la posición (x_i) y después de un cierto intervalo (t_f) se encuentra en la posición (x_f), la velocidad media de la partícula se define como la relación entre la variación de la posición y el intervalo de tiempo (Serway & Jewett, 2014) como se muestra en la siguiente ecuación:

$$v_{media} = \frac{x_f - x_i}{t_f - t_i}$$

1.1.2 Movimiento rectilíneo uniforme

De acuerdo con la primera Ley de Newton, cualquier partícula puntual permanece en reposo o en movimiento rectilíneo uniforme cuando no hay fuerza externa neta que actúe sobre el cuerpo, porque las fuerzas actuales están en equilibrio, y se conserva el estado de

reposo o de movimiento rectilíneo uniforme (MRU) (Resnick, Halliday, & Krane, 2001). Esta es una situación ideal, porque siempre hay fuerzas que tienden a alterar el movimiento de las partículas.

El MRU es un movimiento que describe una trayectoria recta con respecto a un observador, es uniforme debido a que la velocidad (magnitud y dirección) es constante en el tiempo y la aceleración es nula. Un movimiento rectilíneo también puede ser no uniforme, y en ese caso la relación entre posición y tiempo es compleja. El movimiento puede considerarse en dos sentidos; una velocidad negativa representa un movimiento en la dirección opuesta al sentido convencionalmente adoptado como positivo (Resnick, Halliday, & Krane, 2001). La ecuación que define la posición de una partícula en MRU está dada por:

$$\lim_{\Delta t \rightarrow 0} \frac{x_f - x_i}{t_f - t_i} = \frac{dx}{dt}$$

$$\frac{dx}{dt} = v$$

$$dx = v dt$$

$$\int_{x_0}^{x_f} dx = \int_{t_0}^{t_f} v dt$$

$$x_f = x_0 + vt$$

Dónde: x representa la posición inicial de la partícula, v es la velocidad y t es el tiempo. Esta relación también es aplicable si la trayectoria no es recta, siempre que el módulo de la velocidad sea constante.

1.1.3 Aceleración media

En la mayoría de movimientos, la velocidad varía con el tiempo y en este instante interviene la aceleración. La aceleración media es la razón entre la diferencia de velocidades ($v_f - v_i$) después de un cierto intervalo ($t_f - t_0$) como se muestra en la siguiente ecuación:

$$a_{media} = \frac{v_f - v_0}{t_f - t_0}$$

1.1.4 Movimiento rectilíneo uniformemente variado

El movimiento rectilíneo uniformemente acelerado (MRUV) tiene una aceleración constante diferente de cero. Es un movimiento común en la vida diaria. Un objeto que puede caer y que no encuentra ningún obstáculo en el camino (caída libre), o un automóvil que desciende una pendiente, justo antes de llegar al área de salto, son buenos ejemplos de esto. El MRUV tiene las siguientes características: La trayectoria es una línea recta y, por lo tanto, la aceleración normal o centrípeta es cero, la velocidad instantánea cambia de magnitud (velocidad) de manera uniforme: aumenta o disminuye, esto implica que la aceleración tangencial es constante. Por lo tanto, la aceleración promedio es la misma que la aceleración instantánea para cualquier período estudiado ($a = a_m$) (Resnick, Halliday, & Krane, 2001). La velocidad para este movimiento se calcula con la siguiente ecuación:

$$\lim_{\Delta t \rightarrow 0} \frac{v_f - v_i}{t_f - t_i} = \frac{dv}{dt}$$

$$\frac{dv}{dt} = a$$

$$dv = a dt$$

$$\int_{v_o}^{v_f} dv = \int_{t_o}^{t_f} a dt$$

$$v_f = v_o + at$$

La posición de una partícula se encuentra mediante la siguiente igualdad:

$$a = \frac{dv}{dt} = d\left(\frac{dv}{dt}\right)$$

$$v dv = a dt \left(\frac{dx}{dt}\right) = adx$$

$$\int_{v_o}^{v_f} v dv = \int_{x_o}^{x_f} a dx$$

$$\frac{1}{2}v_f^2 - \frac{1}{2}v_o^2 = \int_{x_o}^{x_f} a dx$$

La relación final es importante cuando la posición y la aceleración de la partícula son conocidas, y se puedan evaluar en un determinado intervalo de tiempo.

1.2 Equipos de laboratorio para el estudio de movimientos rectilíneos

Los equipos más comunes en los laboratorios actualizados para el estudio de la cinemática relacionado al movimiento rectilíneo, emplean mecanismos electrónicos como sensores, temporizadores, cronómetros digitales, contadores y equipos de aire comprimido para simular un MRU al reducir la fricción (Gutiérrez, Jácome, & Larrea, 2016). El equipo más común en los laboratorios de física en universidades e instituciones de educación superior se muestra en la Figura 1.1.

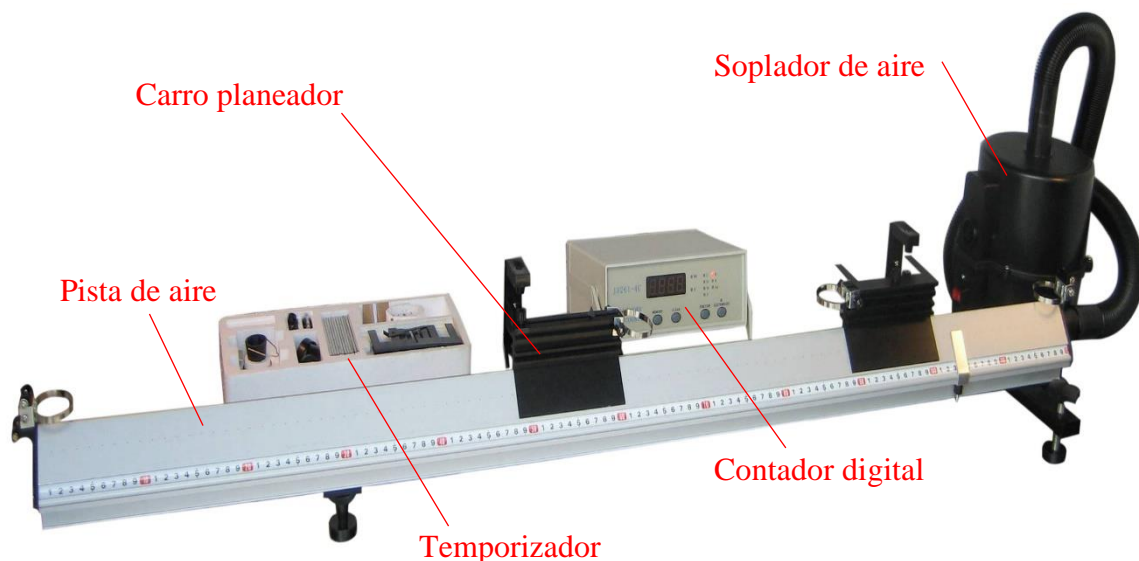


Figura 1.1 Equipos de laboratorio para el estudio de movimiento rectilíneos.
Fuente: *Lambda scientific (2019)*

El equipo experimental de vía aérea está diseñado con los principios de un planeador. Cuando se conecta una pista de aire con el soplador de aire, se genera una capa de aire comprimido desde los orificios de la superficie y forma una capa delgada de aire que disminuye la fricción (μ cercano a cero) entre la superficie interna del carro planeador y la pista de aire. Con el soplador de aire y el temporizador digital, la pista de aire se usa para estudiar experimentalmente conceptos físicos importantes como la velocidad, la aceleración, el momento, la colisión y la cinética. (Lambda Scientific Systems, Inc., 2019)

Estos equipos tienen conexión para computadora, Tablet o celular, mediante un plug de USB, posteriormente se inicia el programa desarrollado por cada fabricante, se encuentran precargados varios experimentos (Primera Ley de Newton, MRU. MRUV, aceleración gravitacional, segunda ley de Newton, tercera ley de Newton, teorema de la energía cinética, teorema del momento, ley de conservación del impulso, ley de conservación de energía mecánica, vibración armónica simple, coeficiente de rigidez de un resorte, vibración armónica simple, colisiones elásticas), posteriormente se calibra al equipo y se inicia un registro de las variables velocidad, aceleración, posición, tiempo conforme se realiza varias repeticiones, para posteriormente arrojar gráficos de velocidad vs tiempo, posición vs tiempo y aceleración vs tiempo en los experimentos de cinemática de movimientos rectilíneos. (Lambda Scientific Systems, Inc., 2019).

1.2.1 Celda de carga HX711

Es una celda de carga con un tamaño aproximado de 24 x 16 mm cuya interfaz de una celda de carga con Arduino se genera de la siguiente manera: La señal de salida producida por la celda de carga está en el rango de milivoltios ($\pm 40\text{mV}$), por lo que un amplificador que convierte la señal en un nivel que luego se transforma en una señal digital. Para este propósito, se utiliza un sensor amplificador HX711.

El sensor amplificador HX711 incluye un chip HX711 con frecuencia de actualización de 10/80 Hz y capacidad de conversión de analógico a digital (A/D) con una precisión de 24 bits. El módulo HX711 amplifica la salida de bajo voltaje de la celda de carga (2.7V a 5V DC) y una corriente de funcionamiento menor a 10 mA para posteriormente enviar al Arduino para que calcule el peso de estos datos. Se puede observar en la Figura 1.2. y su conexión en la Figura 1.3. (Naylamp Mechatronics SAC, 2019)

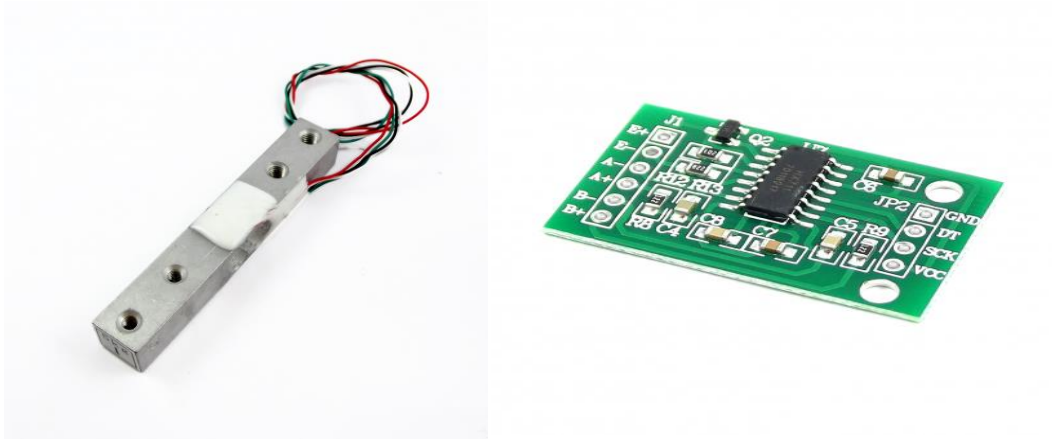


Figura 1.2. Celda de carga y transmisor.
Fuente: naylampmechatronics (2018).

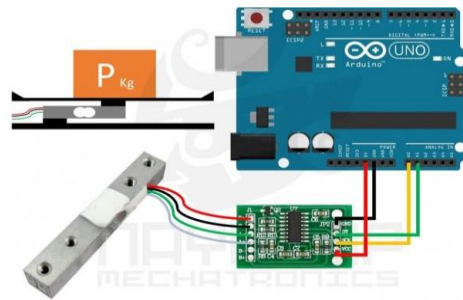


Figura 1.3. Conexión celda de carga.
Fuente: naylampmechatronics (2018).

1.2.3 Acelerómetro y giroscopio MPU6050

Un acelerómetro mide la aceleración (no gravitacional) de un objeto cuando pasa por una determinada posición a cualquier rapidez, el acelerómetro está diseñado para responder a las vibraciones asociadas con dicho movimiento. Utiliza cristales microscópicos que sufren tensión cuando se producen vibraciones, y a partir de esa tensión se genera un voltaje para crear una lectura en cualquier aceleración. Los acelerómetros son componentes importantes

para los dispositivos que rastrean variables físicas. El primer acelerómetro se llamó la máquina Atwood y fue inventado por el físico inglés George Atwood en 1783. (Hetpro, 2014)

Un giroscopio es un dispositivo que utiliza la gravedad terrestre para determinar la orientación. Consiste en un disco de rotación libre llamado rotor, montado en un eje giratorio en el centro de una rueda más grande y más estable. A medida que el eje gira, el rotor permanece estacionario para indicar la atracción gravitacional central y, por lo tanto, la dirección hacia el centro terrestre. Los giroscopios fueron inventados en el siglo XIX por el físico francés Jean-Bernard-León Foucault. (Prometec, 2018)

El sensor MPU 6050 es un acelerómetro y un giroscopio MEMS denominado dispositivo de 6 grados de libertad, estas características se usan en robots, teléfonos, posicionar objetos, orientar teléfonos inteligentes, rastrear el movimiento. Se puede trabajar con el acelerómetro y giroscopio por separado pero las mediciones disminuyen en precisión.

El giroscopio mide la variación de la posición angular con respecto al tiempo (ω), en coordenadas cartesianas (ejes X, Y y Z). Utiliza el efecto de la aceleración de Coriolis para obtener mediciones. El acelerómetro mide el cambio de la rapidez con respecto al tiempo (v) en coordenadas cartesianas y mediante cálculos se obtiene el ángulo en el que se coloca el sensor. Al combinar los datos del acelerómetro y el giroscopio, se obtiene información precisa sobre la orientación del sensor porque posee un hardware de conversión A/D de 16 bits. (García, 2018)

El sensor MPU 6050 se comunica con el Arduino a través del protocolo I2C como se muestra en el diagrama de la gráfica 3, el pin de 5V se conecta con la tarjeta o también se puede conectar al pin de 3.3V, a continuación, se conecta a la GND, se conecta el pin digital 2 del Arduino con el INT, se configura las líneas I2C mediante el SDA al pin 4 del Arduino, y el pin etiquetado como SCL al pin 5 de Arduino (SCL). Se descarga la biblioteca Arduino para MPU 6050, después se descomprime y extrae la biblioteca y se pega dentro de la carpeta de Arduino. Se puede observar los pines de conexión en las Figuras 1.4 y 1.5. (García, 2018)

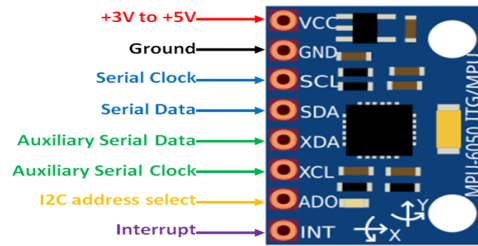


Figura 1.4. Sensor MPU6050.
Fuente: (García, 2018)

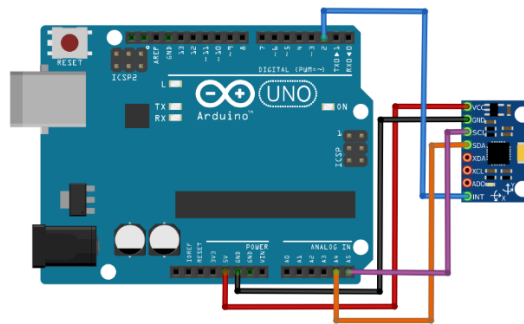


Figura 1.5. Comunicación entre sensor MPU 6050 y Arduino.
Fuente: (García, 2018).

1.2.4 Sensor de fuerza y presión FSR402

Son sensores de fuerzas y presión (FSR) robustos utilizados en baterías electrónicas, dispositivos de juegos portátiles, celulares, dispositivos electrónicos portátiles, en la industria en general porque son fáciles de implementar y al bajo costo. La tecnología utilizada en los sensores FSR está patentada por Interlink Electronics, los tipos más comunes son el FSR 402 y 406 con una región de detección de 1/2 diámetro. (Nares, 2015)

Los FSR constan de dos membranas de plástico y un adhesivo espaciador o lengüeta de conexión de un material delicado como se muestra en la Figura 1.6. Las membranas conductoras están separadas por un espacio de aire delgado cuando no se aplica presión. Una de las membranas contiene dos trazas que van desde el filo de la placa hasta el área de detección (la parte redonda). Estas huellas se marcan juntas, pero no se tocan. La otra membrana está recubierta con una tinta conductora. Cuando presiona el sensor, la tinta pone

en cortocircuito los dos trazos junto con una resistencia que depende de la presión. (Carrión, Ochoa, & Valverde, 2009)

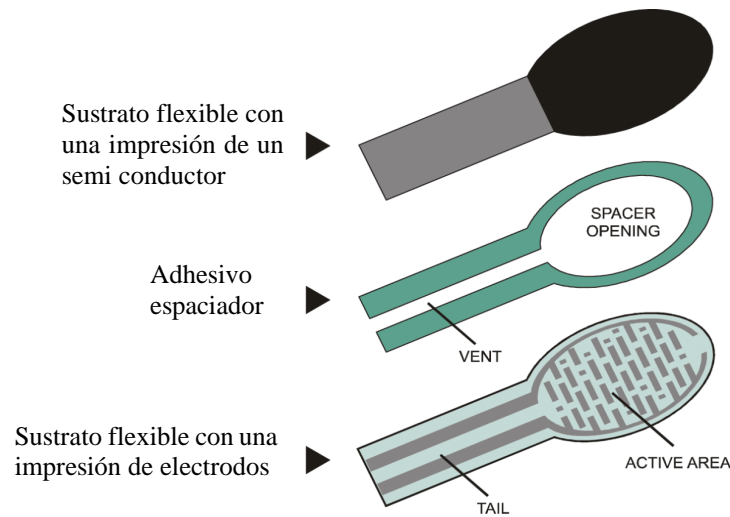


Figura 1.6. Sensor de fuerza y presión.
Fuente: (Carrión, Ochoa, & Valverde, 2009).

La mejor manera de comprobar el correcto funcionamiento del FSR es mediante un multímetro, al enchufar en una placa de pruebas o usar un conector de estilo de abrazadera como pinzas de cocodrilo, encabezado hembra o un bloque de terminales, al usar el sensor en modo de medición de resistencia (Ω) y evidencia que el valor de resistencia cambia cuando se presiona la membrana plástica. (Marqués, 2017)

Los sensores tienen una resistencia que depende de la presión aplicada en el área de detección, en una relación inversa, es decir que, a mayor presión, menor es la resistencia ($> 10 \text{ M}\Omega$ (sin presión) a $\sim 200 \Omega$ (presión máxima)). La mayoría de los sensores FSR pueden detectar fuerza en el rango de 100 g a 10 kg. Para medir la fuerza aplicada con un Arduino, se construye un circuito divisor de voltaje y una resistencia desplegable. Este circuito crea una salida de voltaje variable que puede ser leído por la entrada A / D del microcontrolador. (Carrión, Ochoa, & Valverde, 2009)

1.2.5 Sensor ultrasónico HCSR 04

Es un módulo ultrasónico formado por un transmisor y receptor de 4 pines: (Ground o tierra color negro, VCC de 5V color rojo, Echo Pin 12 color amarillo y Trigger Pin 11 color azul), el sensor es utilizado para medir distancias u detectar objetos, evitar y detectar

obstáculos, mapear objetos que rodean el sensor, medir la profundidad de ciertos lugares en radio máximo de 4,5 m, funciona con el principio del MRU al considerar que la velocidad del sonido es constante (330 m / s) en el aire y cuando es objetada por cualquier material, se refleja hacia el receptor ultrasónico. (Collaguazo, 2012)

El HC-SR04 sensor de distancia es usado en Arduino, ARM, PIC, Raspberry Pie, entre otras. Para el correcto funcionamiento se debe alimentar el sensor usando un + 5V regulado a través de los pines Vcc y Ground (tierra). La corriente consumida por el sensor es inferior a 15 mA y, por lo tanto, puede alimentarse directamente con los pines integrados de 5 V (si están disponibles). Los pines Trigger y Echo son pines de E / S y, por lo tanto, se pueden conectar a los pines de E / S del microcontrolador, se puede observar en la Figura 1.7. Para comenzar la medición, el pin del Trigger debe estar alto por 10uS y luego apagarse. (Equipo de laboratorio didáctico, 2018)



Figura 1.7. Sensor ultrasónico HCSR 04.
Fuente: (Equipo de laboratorio didáctico, 2018).

Esta acción activará una onda ultrasónica a una frecuencia de 40Hz desde el transmisor y el receptor esperará a que la onda regrese. Una vez que la onda regresa después de ser reflejada por cualquier objeto, el pin Echo se eleva durante un período de tiempo particular, que será igual al tiempo que tarda la onda en regresar al sensor. El microcontrolador mide la cantidad de tiempo durante el cual el pin Echo permanece alto, ya que proporciona la información sobre el tiempo que tarda la onda en volver al sensor. (Naylamp Mechatronics, 2019)

1.2.6 Módulo bluetooth HC06

Bluetooth es un protocolo de comunicación por radio que conecta dos o más dispositivos móviles. El módulo Bluetooth HC06 es un dispositivo ampliamente disponible y económico

que sirve para enviar datos entre un microcontrolador o microprocesador y un dispositivo equipado con bluetooth (teléfono inteligente, tablet Android, laptop). Presenta una interfaz de solo cuatro pines (VCC, para suministrar alimentación + 5V, GND ground o tierra para suministrar voltaje negativo de alimentación, TXD o pin TX de transmisión de datos que llegan desde el teléfono móvil o PC y RXD o pin de Rx de recepción de datos). El LED que posee sirve para verificar la conexión mediante parpadeo indica que no hay conexión y al mantenerse brillante significa que está conectado (Styger, 2013). Este módulo se muestra en la Figura 1.8:

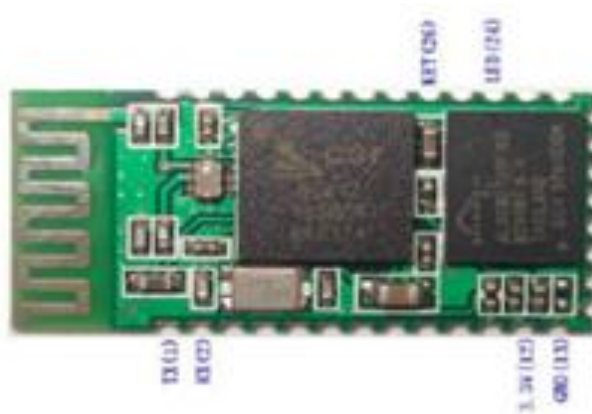


Figura 1.8. Módulo bluetooth HC06.
Fuente: (Styger, 2013).

El hardware que se requiere para el funcionamiento del dispositivo es el siguiente: una tarjeta Arduino, un módulo Bluetooth HC-06, 4 conectores macho a hembra. El circuito de construcción se realiza conectando el pin VCC a la salida de 5V, el pin GND con el mismo terminal en el microcontrolador, el pin TXD al pin -5 y el pin RXD al terminal -6 de la tarjeta respectivamente. (BlueBolt IINC, 2015)

1.2.7 Sensor infrarrojo TCRT5000

El sensor TCRT 5000 es una unidad infrarroja que tiene un fotodiodo y un fototransistor incorporado. El fotodiodo tiene dos pines (ánodo y cátodo) que se pueden usar para generar una señal IR y se alimenta a través de una resistencia limitadora de corriente de valor 220R, mientras que el transistor de fotos también tiene dos pines (colector y emisor) que se pueden usar para leer la señal IR que se refleja y está conectado a tierra a través de una resistencia de 10K. Se muestra en la Figura 1.9. (Mecatrónica LATAM, 2015)



Figura 1.9. Sensor infrarrojo TCRT 5000.

Fuente: (Mecatrónica LATAM, 2015).

Este sensor se puede usar para detectar la presencia de un objeto reflectante frente a él, también con cierto nivel de programación también se puede calcular la distancia del objeto frente a él. Pero la distancia solo se puede calcular para objetos de corto alcance, también puede distinguir fácilmente el color blanco y negro para ser utilizado en el seguimiento de líneas. (Nubbeo, 2019)

Una desventaja del sensor es que se ven afectados fácilmente por las condiciones ambientales. Al igual que el transistor fotográfico, no solo reaccionará a la luz IR del fotodiodo, sino que también reaccionará a la luz solar, las luces de la casa y otras fuentes comúnmente disponibles. (Mecatrónica LATAM, 2015)

1.2.8 Controlador L298N

El controlador de motor L298 usa una configuración de puente H, que es útil cuando se controla la dirección de rotación de un motor de CC. Un circuito de puente H contiene cuatro elementos de conmutación, transistores o MOSFET, con el motor en el centro formando una configuración tipo H. Al activar dos interruptores particulares al mismo tiempo, se cambia la dirección del flujo de corriente, y así cambiar la dirección de rotación del motor. (PROMETEC, 2017)

El L298N es un controlador de motor de puente H que permite el control de velocidad y dirección de dos motores de CC al mismo tiempo. El módulo puede conducir motores de CC

que tienen voltajes entre 5 y 35 V, con una corriente máxima de hasta 2A. El módulo tiene dos bloques de terminales de tornillo para el motor A y B, y otro bloque de terminales de tornillo para el pin de tierra, el VCC para el motor y un pin de 5V que puede ser una entrada o una salida, tiene un regulador de 5V integrado que se habilita o deshabilita mediante un puente. Si el voltaje de suministro del motor es de hasta 12 V, se habilita el regulador de 5 V y el pin de 5 V se puede usar como salida. Pero si el voltaje del motor es mayor a 12V, se desconecta el puente porque esos voltajes causarían daños al regulador de 5V a bordo. En este caso, el pin de 5V se usará como entrada, ya que se necesita conectar a una fuente de alimentación de 5V para que el IC funcione correctamente. Se observa en la Figura 1.10.

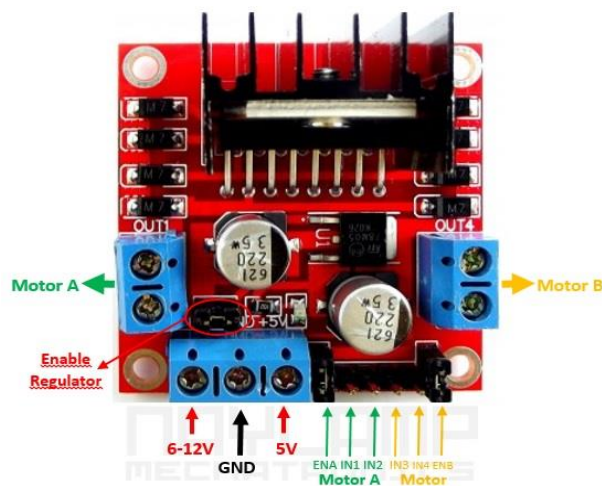


Figura 1.10. Puente H L298N.
Fuente (PROMETEC, 2017)

1.2.9 Arduino Nano y Mega

El código Arduino se puede transferir fácilmente a diferentes tableros prácticamente sin cambios, lo cual es muy beneficioso para cualquier diseño. El factor decisivo que usará cada proyecto se reduce a dos cosas básicas: hardware y costo. Si bien el aspecto de costo de cada Arduino es obvio (la opción de menor costo es siempre la preferencia), el hardware puede no ser tan fácil de determinar. Al mirar el hardware, se debe considerar lo siguiente: Dimensiones físicas, potencia de la CPU, tamaño de la memoria, capacidades de E / S, periféricos a bordo, peso y conectividad. (Ruiz, 2007)

El Arduino Nano es un microcontrolador procesador de perfil pequeño conveniente para espacios reducidos y proyectos que necesiten reducir el peso, trabaja con un procesador Atmega328 que funciona a 16MHz, incluye 32KB de memoria de programa, 1KB de EEPROM, 2KB de RAM, tiene 14 E / S digitales, 6 entradas analógicas y rieles de alimentación de 5V y 3.3V. No puede conectarse a los escudos Arduino, pero tiene encabezados de clavija que lo hacen útil para la creación de prototipos de placa de prueba o en PCB con el uso de un zócalo. A menudo, las placas Arduino Nano son la opción de placa más barata disponible, lo que las hace rentables para proyectos más grandes. Se observa en a Figura 1.11. (Marker Zona, 2014)

El Arduino Mega tiene 54 E / S es alimentado por un núcleo ATmega2560. La CPU tiene una velocidad de reloj de 16MHz e incluye 256KB de ROM, 8KB de RAM, 4KB de EEPROM, y funciona a 5V, lo que facilita su uso con la mayoría de los dispositivos electrónicos amigables para los aficionados. El Arduino Mega tiene 16 entradas analógicas, 15 canales PWM, un pinout similar al Due y es compatible con hardware con los escudos Arduino. Figura 1.12. (Torrente, 2013)

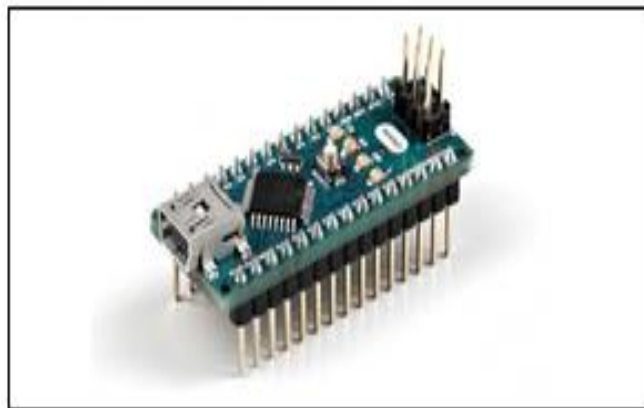


Figura 1.11. Arduino nano.
Fuente: (Marker Zona, 2014)



Figura 1.12. Arduino Mega2560 rev3.
Fuente. (Torrente, 2013)

1.2.10 Display LCD TFT 3.2” Open-Smart

Pantalla LCD TFT de 3.2” a colores con membrana sensible al tacto, también incluye lector de tarjetas SD y un sensor de temperatura incorporado, estas pantallas permiten mostrar texto o gráficas a color en diversas aplicaciones con microcontroladores como Arduino Raspberry Pi o Pic, ideal para crear una interfaz entre la maquina y el profesional.

Esta pantalla utiliza colores con una resolución de 240x400 pixeles, posee un controlador gráfico hx9327, controlador táctil y puede mostrar colores hasta 262144 colores RGB. Además, incluye un socket para memorias SD útil para almacenar imágenes en formato .bmp, también posee un sensor LM75A de temperatura que se comunica mediante el bus I2C su dirección es la 0x48. La instalación de este display es muy sencilla ya que por su distribución de pines se conecta en las placas Arduino Uno, Mega y Due. Trabaja a 5-12 VCD a una corriente de 150mA. Se muestra en Figura 1.13. (Torrente, 2013)

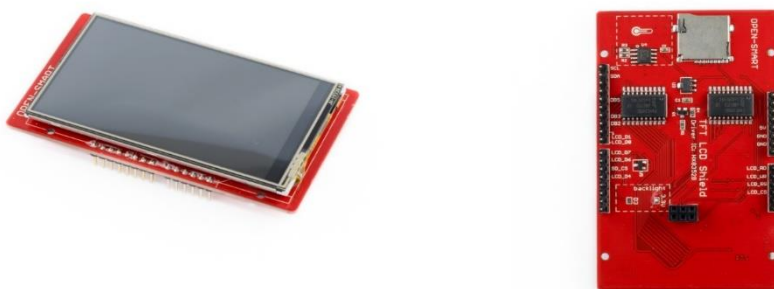


Figura 1.13. Display OpenSmart 3.2” TFT-LCD.
Fuente: (Torrente, 2013).

1.2.11 Fuente de alimentación

Fuente de alimentación conmutada de 12 v 30amp con una potencia de 360W la entrada es de 110v-220v el voltaje de salida puede modificarse entre +- 10% tiene una protección interna contra sobrecargas posee 3 terminales negativos y 3 positivos, con su respectiva entrada de voltaje, sus aplicaciones son diversas tales como maquinas cnc, tiras led y dispositivos que necesiten esta fuente de alimentación, tiene la certificación CE y RoHS y se observa en la Figura 1.14. Como (ONROM, 2020)



Figura 1.14. Fuente de Alimentación 12v 30 A.
Fuente: (ONROM, 2020)

1.2.12 Regulador de Voltaje 7805

Es un circuito integrado de la familia de reguladores de tensión positivo que comprenden entre 7805-7830 y es muy utilizado en fuentes de alimentación estabilizadas, anteriormente para la alimentación de voltaje estabilizado se utilizaba gran cantidad de componentes electrónicos, hoy en día se simplifica con este dispositivo, la instalación de este se realiza a la entrada (VIN) con el voltaje que se requiere estabilizar que debe ser mínimo mayor con 2 voltios al voltaje que se requiere el pin de tierra (GND) y la salida (VOUT) con el voltaje ya estabilizado, trabaja con una temperatura de operación entre 0- 125°C y la salida máxima es de 1A. Se puede observar la distribución en la Figura 1.15. (Torrente, 2013)

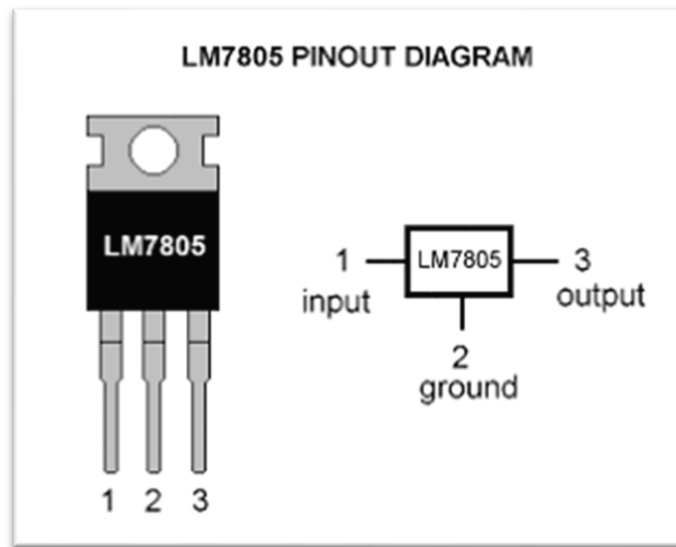


Figura 1.15. Regulador de voltaje 7805.
Fuente. (Torrente, 2013)

1.2.13 Pantalla LCD 16x2 con I2C

Pantalla LCD de 16x2 basada en los controladores HD44780 de Hitachi es controlado con dos líneas digitales a través de bus I2C que a su vez estos pines pueden ser compartidos por otros dispositivos estos pines son el SDA y SCL reduciendo así la utilización de cables permite visualizar mensajes o caracteres que provengan de cualquier microcontrolador, puede desplegar 16 caracteres en 2 filas en un total de 32 caracteres alfanuméricos, su alimentación es con 5v-12v. Se observa en la Figura 1.16. (amgkits, 2016)



Figura 1.16. Pantalla LCD 16x2.
Fuente: (amgkits, 2016)

1.2.14 Software LabVIEW

LabVIEW es un software para la ingeniería donde se requieren realizar análisis y pruebas con los componentes de forma virtual antes de llevar a la etapa de construcción para ello esta plataforma ofrece una programación gráfica, incluyendo configuraciones de hardware. Plataforma de programación LabVIEW Figura 1.17. (INSTRUMENTS, 2020).

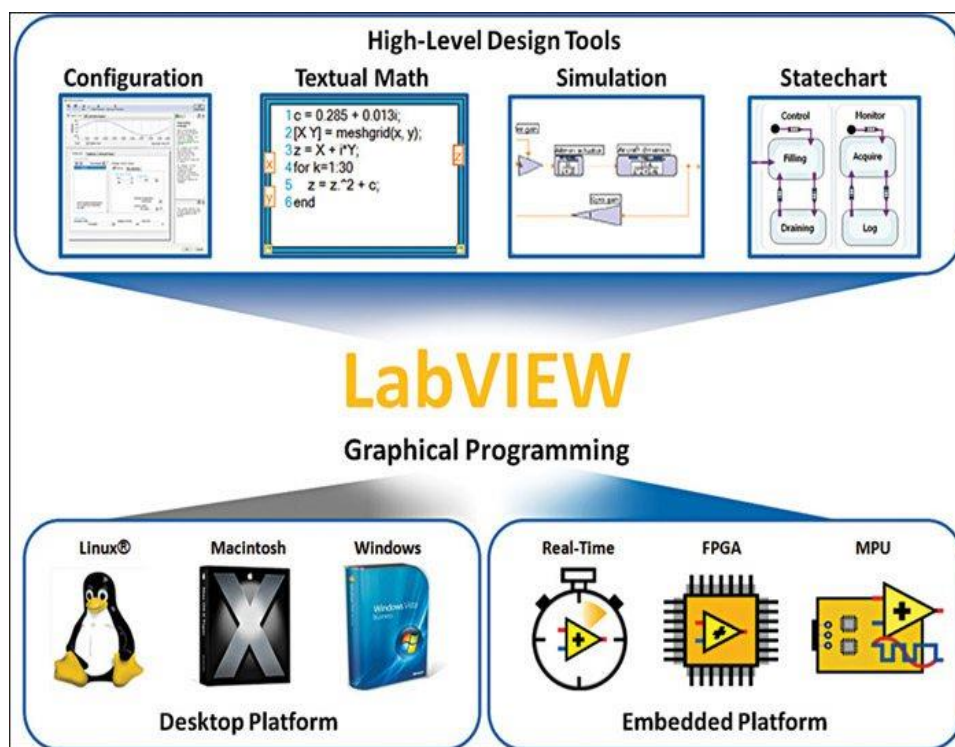


Figura 1.17. Plataforma LabVIEW.

Fuente: (INSTRUMENTS, 2020)

CAPÍTULO 2

2 MARCO METODOLÓGICO

En este capítulo se desarrolla los aspectos relacionados a los métodos de investigación y tipos de investigación dentro de la metodología en el desarrollo del proyecto “Módulo autónomo para el aprendizaje de los principios básicos de la cinemática de una partícula en el ámbito universitario”

2.1 Métodos de investigación

El método deductivo – inductivo sirvió en este proyecto porque durante la etapa de la revisión bibliográfica se partió desde una etapa general bajo dos las variables. Bernal (2010) define que “Este método de inferencia se basa en la lógica y estudia hechos particulares” (p. 60). La primera variable fue el estudio de la cinemática en los movimientos rectilíneos y la segunda variable correspondiente a los equipos y elementos electrónicos que se requieren para diseñar y construir la propuesta; para posteriormente estudiar casos más específicos. Según Bernal (2010) se indica que el método deductivo “consiste en tomar conclusiones generales para obtener explicaciones particulares. Se inicia con el análisis de los postulados, teoremas, leyes, principios, de aplicación universal y de comprobada validez, para aplicarlos a soluciones o hechos particulares” (p. 59).

También se utilizó el método inductivo en la etapa de análisis y diseño del prototipo de laboratorio. Según Bernal (2010) “El método inductivo utiliza el razonamiento para obtener conclusiones que parten de hechos particulares aceptados como válidos, para llegar a conclusiones cuya aplicación sea de carácter general”. (p. 59) Es decir, que mientras se diseñó el prototipo se solucionó problemas específicos en nuestro medio y se generalizo dentro del área de la cinemática.

El método cualitativo se utilizó porque se realizó mediciones y cuantificaciones de problemas específicos relacionados a MRU y MRUV que permitieron caracterizar y cuantificar las variables físicas como velocidad y aceleración. Bernal (2010) define al estudio cuantitativo “Método cuantitativo se fundamenta en la medición de las características de los fenómenos, lo cual supone derivar de un marco conceptual pertinente al problema analizado, una serie de postulados que expresen relaciones entre las variables estudiadas de forma deductiva” (p. 60).

2.2 Tipos de investigación

Esta investigación es de tipo explicativa porque explica las variables que interfieren dentro del análisis del MRU y MRUV como son: Velocidad, aceleración, tiempo, distancia recorrida y mediante datos cuantitativos poder caracterizar al fenómeno y poder idealizarlo dentro de la mecánica clásica. Según Bernal (2010) señala que la investigación explicativa “tiene como fundamento la prueba de hipótesis y busca que las conclusiones lleven a la formulación o al contraste de leyes o principios científicos”. (p. 122)

Así también, es una investigación experimental porque se desarrolló un prototipo que permita influenciar sobre las variables que caracterizan la cinemática y poder encontrar resultados fiables bajo un espacio controlado que permita relacionar la realidad del fenómeno con la teoría reduciendo el margen de error al mínimo posible. Bernal (2010) señala que la investigación experimental “actúa conscientemente sobre el objeto de estudio, en tanto que los objetivos de estos estudios son precisamente conocer los efectos de los actos producidos por el propio investigador como mecanismo o técnica para probar sus hipótesis. Es el método por excelencia de la investigación científica” (p. 123)

CAPÍTULO 3

3. PROPUESTA

Módulo autónomo para el aprendizaje de los principios básicos de la cinemática de una partícula en el ámbito universitario con captura de datos en tiempo real con proyección de información en pantalla.

3.1 Descripción general del proyecto.

En este capítulo se describe la propuesta en base a los lineamientos teórico y técnicos para el desarrollo del módulo y a su vez la estructura de que simula el espacio-tiempo, con su respectiva captación de datos en tiempo real y la proyección en pantalla de los resultados por medio de una interfaz realizada en LabVIEW, esta captación de datos se realiza por medio de sensores electrónicos que nos sirven para medir las variables que comprenden la cinemática tales como la velocidad, aceleración, tiempo y también fuerza, peso y la fuerza de gravedad comprendidos en la dinámica que serán procesados por un microcontrolador Arduino Nano y transmitidos por Bluetooth a LabVIEW.

Este proyecto se encuentra apoyado en los resultados obtenidos por la aplicación de las herramientas de investigación, los cuales han demostrado la importancia que tiene en el campo de la física la experimentación, dado que esta científicamente comprobado que esta experimentación de los fenómenos físicos refuerza los conocimientos teóricos del estudiante. Además, este módulo dota al docente de una herramienta más para transmitir los conocimientos y reforzar los mismos para que con ello el estudiante capte mejor los conceptos de a cinemática.

3.2 Diagrama de bloques.

En la siguiente figura (Figura 3.1) se muestra el diagrama de bloques de cada etapa que se necesita para la implementación del módulo autónomo para el aprendizaje de los conceptos básicos de la cinemática.

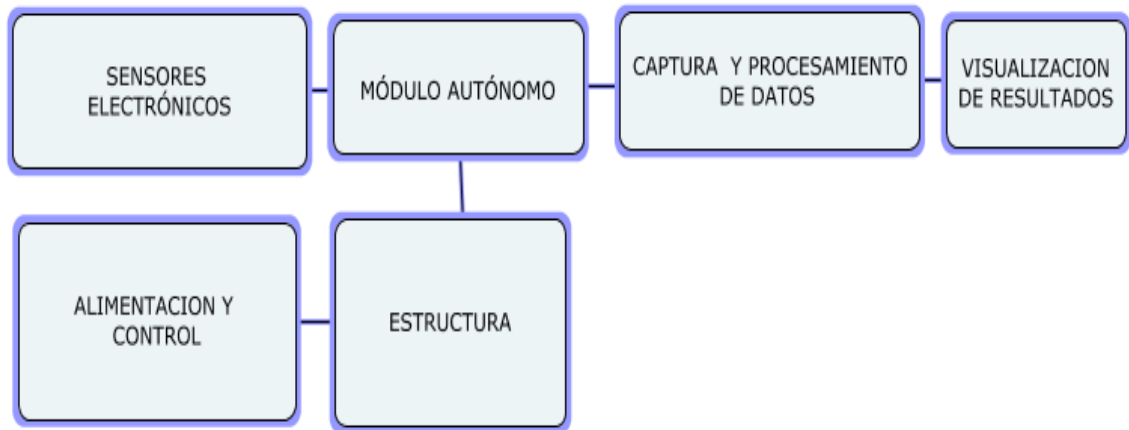


Figura 3.1 Diagrama de Bloques del módulo autónomo
Fuente: Autor

El funcionamiento del módulo autónomo que se muestra en el diagrama de bloques, está compuesto de sensores infrarrojos, sensor MPU6050 de aceleración, sensor de peso y sensor de fuerza conectados a un Arduino Nano para que procese la información, todos estos componentes están conectados a 9v, todos estos datos procesados son enviados a LabVIEW que a su vez muestra los datos en pantalla.

Para la simulación del movimiento de la partícula o módulo autónomo se ha preparado un entorno controlado como es la estructura la cual es controlada por la pantalla táctil que nos permite configurar la inclinación de las torres con una interfaz muy amigable para el operador, el movimiento de la pista se realiza por medio de motores de precisión y la distancia es censada continuamente por los sensores hc-sr04.

Los datos obtenidos son procesados, enviados a LabVIEW y mostrados en tiempo real en pantalla, así el estudiante puede observar el comportamiento modificando las variables del módulo autónomo y la inclinación de la rampa de pruebas.

En consecuencia, los pasos para la construcción del módulo autónomo y su plataforma de funcionamiento serían los siguientes:

- Construcción de estructura o plataforma de pruebas
- Control de inclinación de torres de la estructura por medio de motores paso a paso censados por Hc-sr04 (ultrasónico).
- Construcción de módulo autónomo con sensores infrarrojos, sensor de aceleración MPU6050, sensor de peso HX711, y sensor de fuerza MF-01, además con la pantalla LCD 16x2, todo esto conectado a un microcontrolador Arduino nano, y los datos transmitidos por bluetooth hc-06 a LabVIEW

3.3 Estructura de plataforma

La estructura o rampa está diseñada para controlar la inclinación de ambas torres y soporta el movimiento del módulo autónomo, con ello la simulación de los movimientos MRU y MRUV se realizan en un ambiente controlado, esta estructura está compuesta por dos torres y una rampa parametrizada con la medida espacio, los componentes de cada torre son el sensor Hc-sr04 que mide la distancia de la rampa y un motor de precisión paso a paso que a su vez controlan el desplazamiento del eje roscado haciendo el movimiento en forma vertical estos controlados por dos puentes H L298N, todos estos componentes electrónicos conectados a un microcontrolador Arduino mega rev3 que a su vez está conectado a la pantalla TFT-LCD OpenSmart de 3.2”, luego de esto se programa al Arduino para que en pantalla muestre los pasos a seguir para configurar la rampa. Todo esto alimentados por la fuente de 12v 30A, se observa en la Figura 3.1.

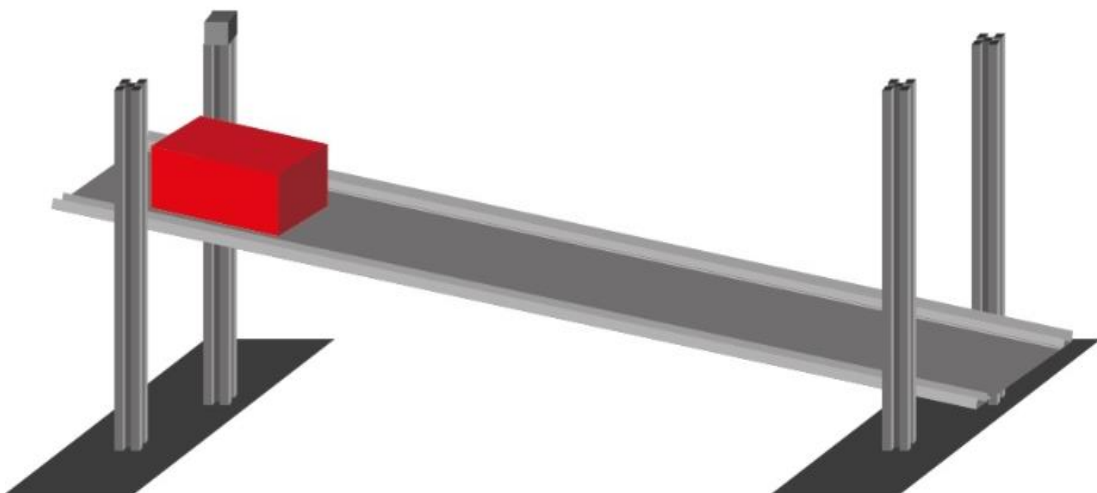


Figura 3.2 Estructura de control de inclinación
Fuente: Autor

3.4 Módulo autónomo

El módulo es el principal elemento de la presente investigación en el estarán integrados los sensores electrónicos y la placa principal todo esto alimentado por una batería de 9v DC, el procesamiento de datos realizados por el Arduino Nano serán enviados por bluetooth, puerto serial a una interfaz gráfica realizada en LabVIEW.



Figura 3.3 Módulo autónomo de simulación de partícula.
Fuente: (Vernier, 2020)

3.5 Fuente de alimentación 12v 30A

La fuente de alimentación del control de elevación de las torres es fabricada por la empresa española OMROM, dentro de las características principales tiene un voltaje ajustable entre $\pm 10\%$, tiene una protección de sobrecargas, la entrada de voltaje es de 110v a 220v y una potencia de 360W, su chasis es de aluminio resistente a caídas, su aplicación es para cualquier elemento que necesite el voltaje de salida, la conexión se realiza por medio de bornes de fácil acceso y posee una certificación CE, RoHS. Las dimensiones y la distribución se observa en la Figura 3.4. (ONROM, 2020)

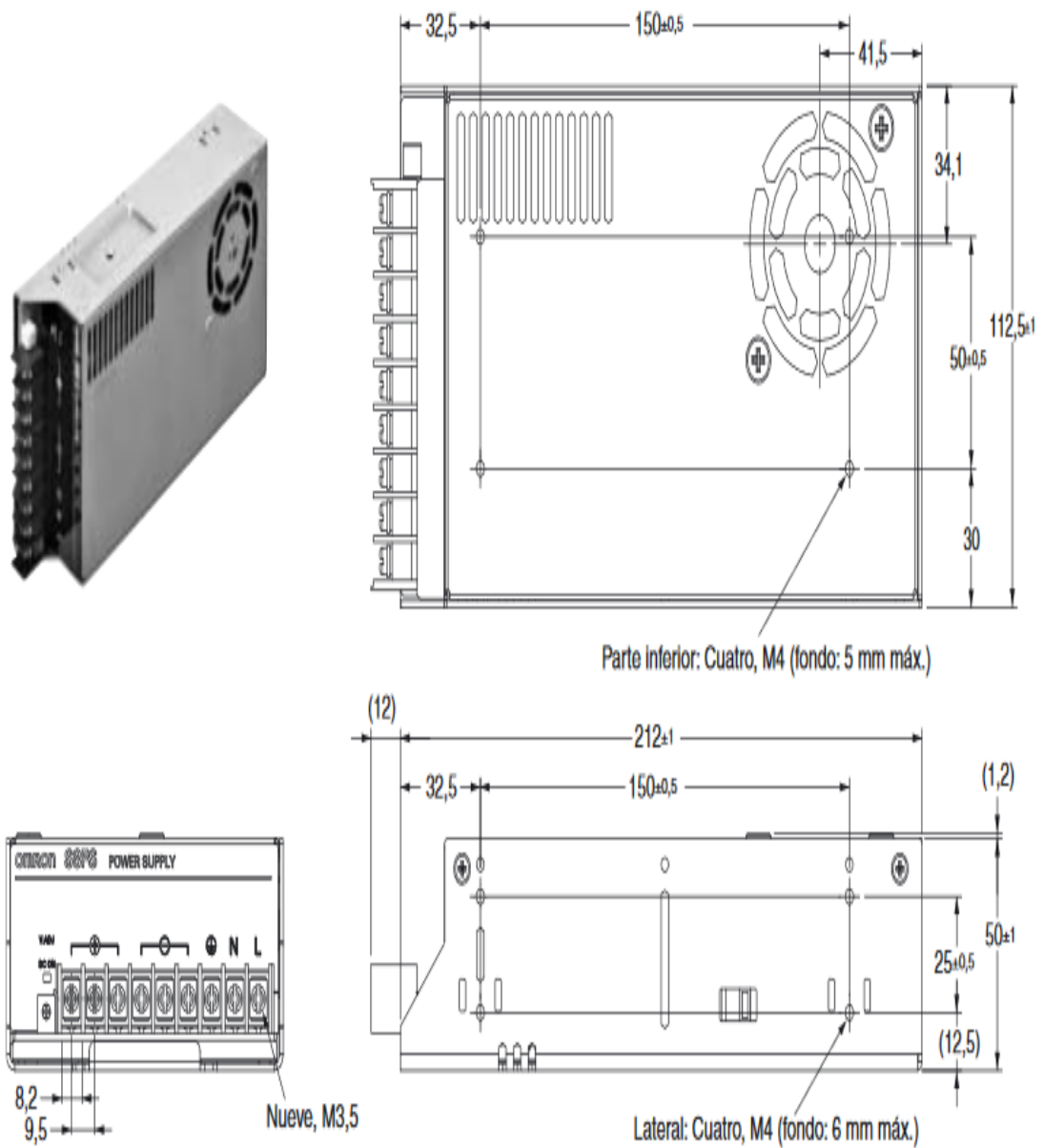


Figura 3.4 Fuente de alimentación 12v 30A.
Fuente. (ONROM, 2020)

Se realiza un cálculo de potencia consumida en la alimentación a las placas de procesamiento y control de los motores paso a paso que controlan la posición de la rampa donde se moverá la partícula

Tabla 3.1
Consumo de componentes

Características	Parámetros (W)
Pantalla OpenSmart 3.2”	0.75
Arduino Mega 2650 rev3.	2.4
Puentes H L298N	25
Regulador 7805	5
Selenoide	8
Motor bipolar	12
Total	45.95

Fuente: Autor

Por sus características y por el consumo de los componentes esta fuente de alimentación será suficiente para hacer funcionar todos los componentes en sus rangos óptimos.

3.6 Microcontrolador y sensores

3.6.1 Microcontrolador Arduino Mega 2560 Rev3

El Arduino Mega 2560Rev3 compuesto por el microcontrolador ATMEGA 2560, que viene incorporado en la tarjeta como se observa en la Figura 3.5

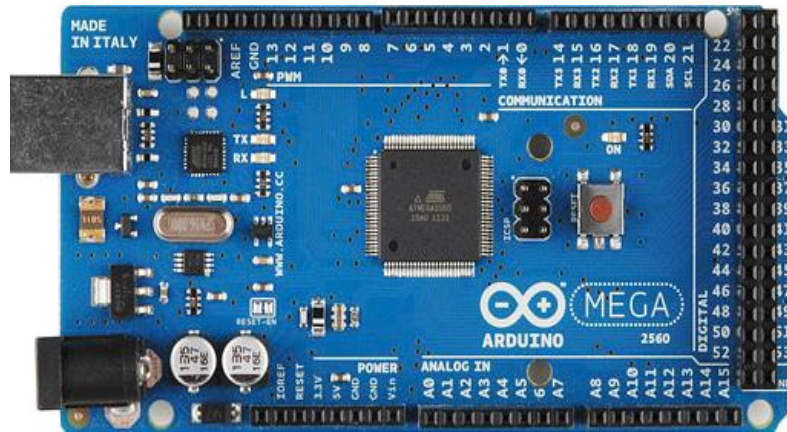


Figura 3.5 Arduino MEGA 2560R3.

Fuente: (Arduino, 2020)

Esta tarjeta fue seleccionada para la realización de este proyecto por la cantidad de puertos y características que permiten que interactúe con la pantalla TFT OpenSmart de 3.2” y la utilización de los puertos entre 22 y 53 para controlar los puentes H L298N y los sensores ultra sónicos SH-sr04. En la Tabla 3.2 se exponen las características principales.

Tabla 3.2
Características Arduino Mega R3

Características	Parámetros
Microcontrolador	Atmega 2560
Tensión de trabajo	5 V
Tensión de entrada (recomendada)	7-12 V
Tensión de entrada (límite)	6-20 V
Pines Digitales I/O	54 (15 salida PWM)
Pines de entradas Analógicas	16
DC Corriente por Pin I/O	20 mA
DC Corriente por Pin 3.3 V	50 mA
Memoria Flash	256 KB (8 KB para bootloader)
SRAM	8 KB
EEPROM	4 KB
Velocidad del reloj	16 MHz
Largo	101.52 mm
Ancho	53.3 mm
Peso	37 g

Fuente: (Arduino, 2020)

3.6.2 Microcontrolador Arduino Nano

Esta tarjeta lleva incorporado el microcontrolador ATMEGA 328 posee una memoria flash de 32KB posee 14 entradas y salidas digitales, la principal característica es que pese a su tamaño ofrece las mismas entradas y salidas que un Arduino Uno y para este proyecto se ha acoplado perfectamente para el procesamiento de la información recibida por los sensores, Se observa en la Figura 3.6 y se describe las especificaciones de esta tarjeta en la Tabla 3.3.



Figura 3.6 Arduino Nano.
Fuente: (Arduino, 2020)

Tabla 3.3
Características de Arduino Nano.

Características	Parámetros
Microcontrolador	ATmega328
Arquitectura	AVR
Voltaje de Operación	5 V
Flash Memory	32 KB // 2 KB used by bootloader
SRAM	2 KB
Velocidad de Reloj	16 MHz
Entrada Análoga	8
EEPROM	1 KB
DC Corriente	40 mA (I/O Pins)
Entrada de voltaje	7-12 V
Entrada digital	22 (6 of which are PWM)
PWM Salida	6
Consumo	19 mA
Tamaño Placa	18 x 45 mm
Peso	7 g

Fuente: (Arduino, 2020)

3.6.3 Pantalla TFT LCD OpenSmart 3.2" táctil

La Shield display LCD 3.2" es una pantalla táctil a colores con una resolución de 240x400 píxeles con un controlador ILI9327, táctil y puede mostrar hasta 262144 colores RGB distintos, apta para el montaje en Arduino Mega y Uno. En esta Shield viene incorporado un sensor de temperatura LM75A que se comunica por el bus I2C. A continuación, se muestra en la Figura 3.7 y en la Tabla 3.4 algunas especificaciones técnicas.



Figura 3.7 Pantalla TFT OpenSmart 3.2".
Fuente: (Naylamp Mechatronics SAC, 2019)

Tabla 3.4
Características Shield OpenSmart 3.2"

Características	Parámetros
Voltaje de operación	5 VDC
Regulador en tarjeta	3.3 V
Corriente de funcionamiento	150 mA
Controlador gráfico (driver)	ILI9327
Interface LCD	Data (8 pines) Control (5 pines)
Interface micro SD	SPI (SS,DI,DO,SCK)
Nivel lógico de SPI	3.3 - 5V
Tamaño de la pantalla (diagonal)	3.2" (81mm)
Resolución	240x400 píxeles
Profundidad de color o bits por pixel	16/18 bpp
Pantalla táctil	resistiva
Sensor de temperatura en placa	LM75
Fabricante	Open Smart

Fuente: (Naylamp Mechatronics, 2019)

3.6.4 Módulo Puente H L298n

Este driver H L298n es uno de los módulos más utilizados en el mercado para manejar motores en DC, soporta una corriente de 2A, este componente posee dos puentes H completos que a su vez permiten controlar 2 motores DC y en nuestro caso controlarán motores bipolares, con ello es factible hacer girar el motor en ambos sentidos y controlar su velocidad mediante señales TTL que se obtienen o generan del Arduino Mega 2560R3. Este control se realiza mediante la modulación de ancho de pulso o PWM, tiene integrado un regulador de voltaje de 5v, se observa el diagrama de conexión en la Figura 3.8 y las características en la Tabla 3.5.

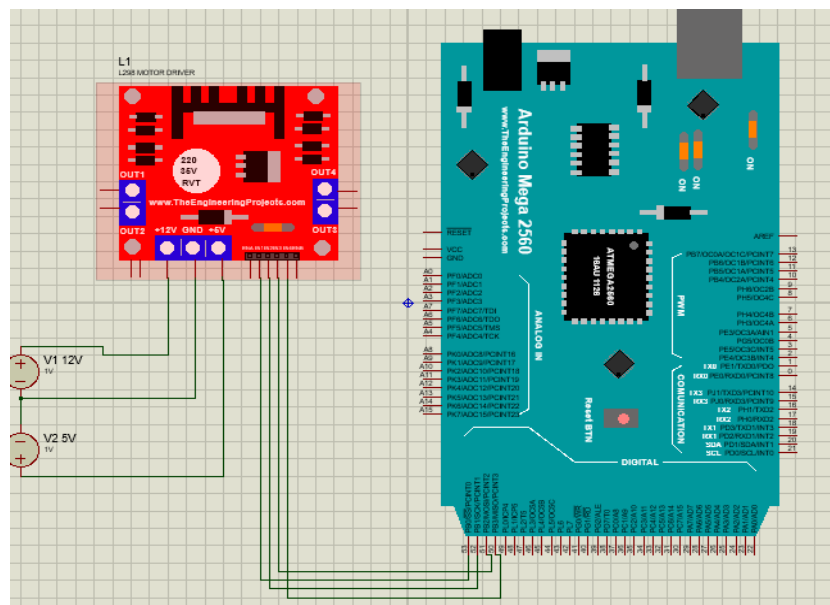


Figura 3.8 Conexión L298N
Fuente: Autor

Tabla 3.5
Características L298N

Características	Parámetros
Chip	L298N
Canales	2
Voltaje lógico	5v
Voltaje de potencia (V motor)	15v-35vDC
Consumo de corriente lógico	0-36mA
Capacidad de corriente	2 – 3A pico
Potencia max	25W
Peso	30g
Dimensiones	43*43*27 mm

Fuente (PROMETEC, 2017)

3.6.5 Regulador de Voltaje LM7805

Este dispositivo electrónico es ampliamente utilizado en las fuentes de alimentación y en circuitos que es necesario garantizar una fuente constante para no dañar los circuitos como los implementados en los proyectos Arduino, en nuestro caso se utilizará para alimentar 5v al módulo L298N. Este regulador trabaja con una capacidad de corriente de 1A y en su pin de entrada siempre tiene que existir 2v por encima de 5 para que este dispositivo trabaje, se muestra a continuación los pines de conexión en la Figura 3.9.

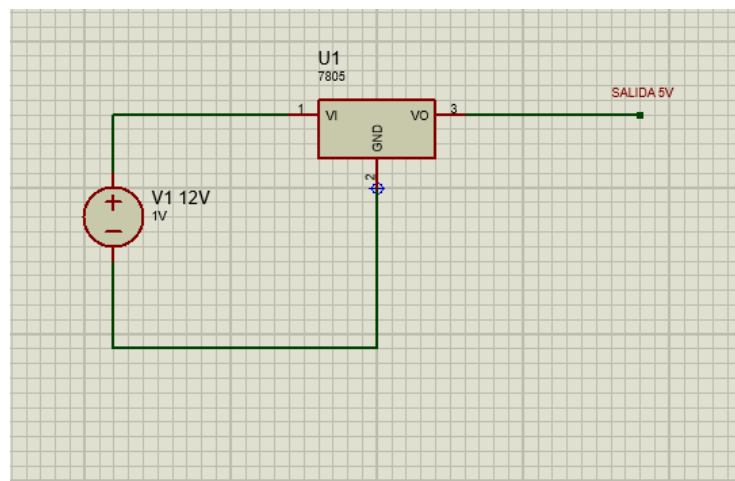


Figura 3.9 Conexión de regulador de voltaje 7805.
Fuente: Autor

En la siguiente tabla se muestra las principales características en la Tabla 3.6.

Tabla 3.6
Características de LM7805

Características	Parámetros
Salida de V	4.8 – 5.25
Línea de Regulación	8 -12V
Corriente	4-8 mA
Temperatura de trabajo	25°C

Fuente: (Torrente, 2013)

3.6.6 Motor Bipolar

Este motor se compone de dos principales elementos, la parte mecánica y la parte eléctrica. Trabaja convirtiendo los pulsos emitidos en nuestro caso por el Arduino en

desplazamientos angulares, una de las ventajas es que nos ayuda con movimientos de precisión y en ambos sentidos de giro, en este proyecto se aplicara para el movimiento en ambos sentidos de los ejes que transportaran la pista del módulo, trabaja en un rango de 8 a 12V, y soporta hasta 2ª, se muestra en la Figura 3.10.

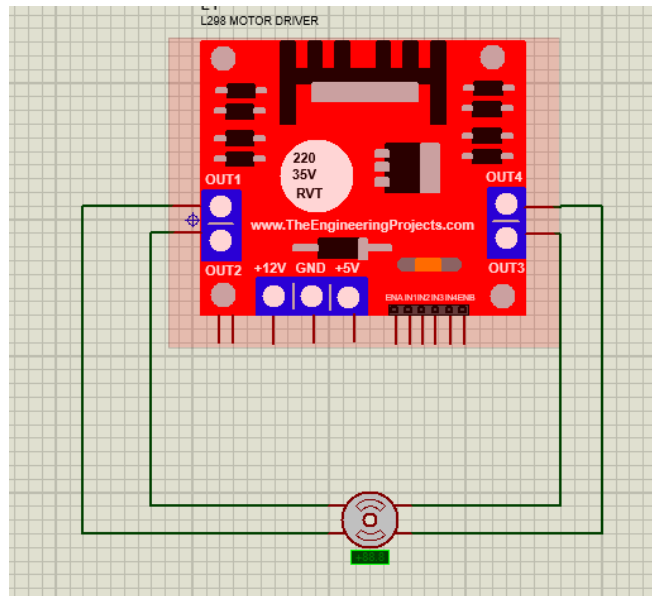


Figura 3.10 Conexión motor Bipolar
Fuente: Autor

Se muestra la Figura 3.11 el principio de funcionamiento por los impulsos emitidos por Arduino y controlados por el puente L298N.

Paso	A+	A-	B+	B-
1	+V	-V	+V	-V
2	+V	-V	-V	+V
3	-V	+V	-V	+V
4	-V	+V	+V	-V

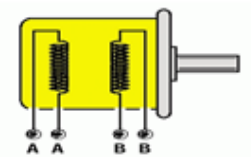


Figura 3.11 Funcionamiento de Motor.
Fuente: (INFmecafenix, 2020)

3.6.7 Sensor Ultrasónico HC-sr04

Este sensor permite medir distancia utilizando transductores un emisor y un receptor piezoeléctricos, estos funcionan de tal manera que el emisor propaga 8 impulsos de ultrasonido que rebotan en algún objeto y el receptor admite la onda y calcula el tiempo que

se ha demorado en regresar la misma y así tenemos la distancia del objeto en el cual ha rebotado la onda. El funcionamiento de este sensor no se ve afectado por la luz solar o el tipo de material.

La fórmula para calcular la distancia es:

$$D(m) = \frac{\text{tiempo de pulso} * \text{Velocidad sonido} \left(\frac{340m}{s}\right)}{2}$$

En este proyecto nos ayudará a medir la rampa del módulo para así variar la inclinación de la misma, se muestra el diagrama de conexión en la Figura 3.12. y las características principales en la Tabla 3.7.

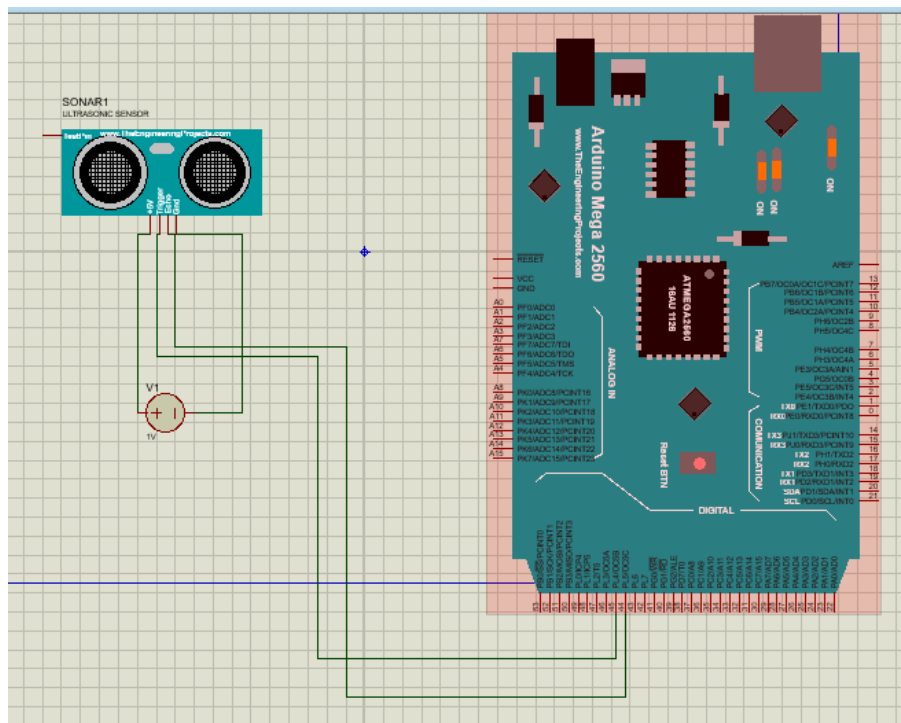


Figura 3.12 Conexión Sensor ultrasónico HCSR04

Fuente: Autor

Tabla 3.7
Características principales de HCsr04

Características	Parámetros
Voltaje de operación	5v
Corriente de trabajo	15mA
Rango de medición	2cm- 450cm
Precisión	+ - 3mm
Angulo de Apertura	15°
f de Ultrasonido	40KHz

Fuente: (Naylamp Mechatronics SAC, 2019)

3.6.8 Pantalla LCD 16x2 I2C

Esta es una pantalla que incorpora un controlador I2C ideal para controlar por dos cables la programación, ahorrando así varios pines del microcontrolador, permite mostrar 32 caracteres en 16 columnas y 2 filas, en este proyecto incorpora al módulo autónomo para mostrar la información de los datos obtenidos y procesador por el Arduino Nano, se muestra en la Figura 3.13.

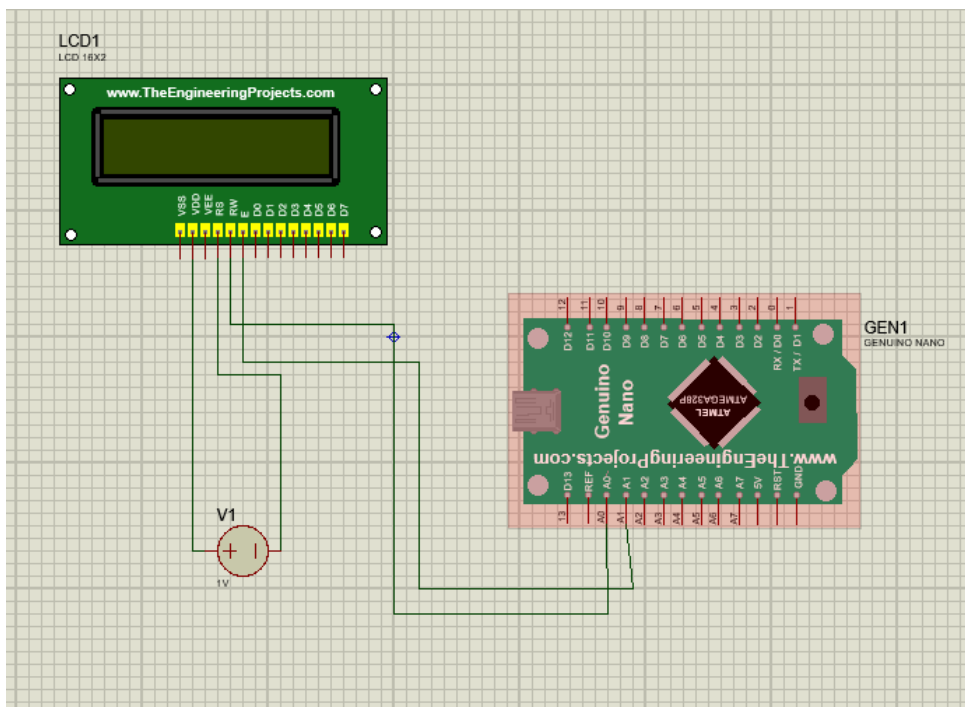


Figura 3.13 Conexión Pantalla LCD 16x2 I2C
Fuente: Autor

Dentro de las características principales se observa en la Tabla 3.8.

Tabla 3.8
Características de LCD 16x2

Características	Parámetros
Dirección I2C	0x20-0x27
Retroiluminación	Azul
Voltaje	5V
Jumper	Activa la pantalla
Angulo de visión	6H

Fuente: (amgkits, 2016)

3.6.9 Celda de carga y HX711

Este dispositivo es un transductor que convierte una fuerza a una señal eléctrica, lo realiza a través de varias galgas internas configuradas en un puente, existen varias en el mercado variando la carga de soportan variando entre 200gr hasta 20 Kg. El funcionamiento de esta celda se la realiza mediante una interface llamada HX711 dispositivo electrónico que se encarga de la lectura de la celda y convierte en datos analógicos interpretados por el microcontrolador. En nuestro caso la utilizaremos en la medición del peso colocado al módulo para realizar las diferentes prácticas. Se muestra la conexión en la Figura 3.14.

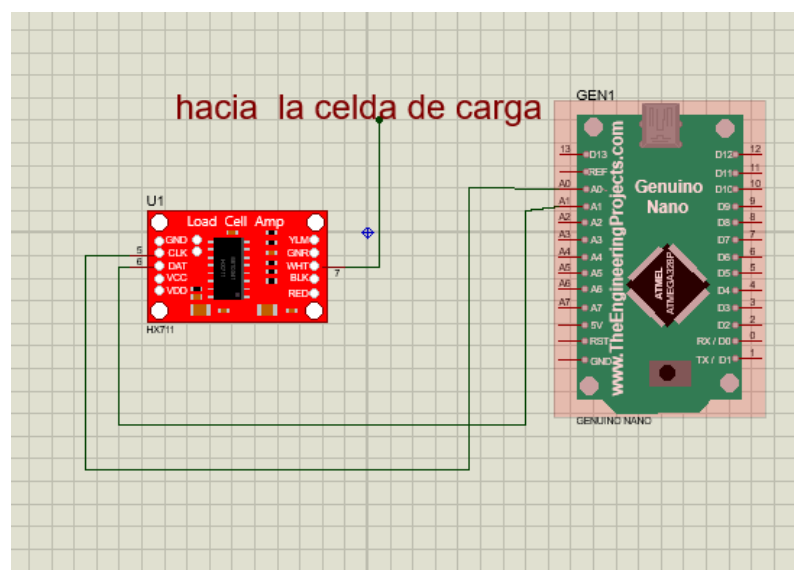


Figura 3.14 Conexión celda de carga y HX711

Fuente: Autor

Conexiones entre celda y modulo HX711 se muestra en la Tabla 3.9.

Tabla 3.9
Conexión entre celda y HX711

Celda de carga	HX711
Rojo	E+
Negro	E-
Verde	A-
Blanco	A+

Fuente: Autor

3.6.10 Sensor MPU6050 Acelerómetro y Giroscopio

Este sensor combina un acelerómetro y un giroscopio de 3 ejes, es utilizado frecuentemente en la navegación y estabilización de drones: Con el MPU6050 se puede hacer diversas mediciones tales como la velocidad, aceleración y velocidad angular, mide el movimiento realizado en x, y, z de un eje de coordenadas. En este proyecto será utilizado como elemento principal del módulo autónomo de la cinemática para realizar los diversos laboratorios.

Su conexión es muy sencilla y se muestra a continuación en la Figura 3.15.

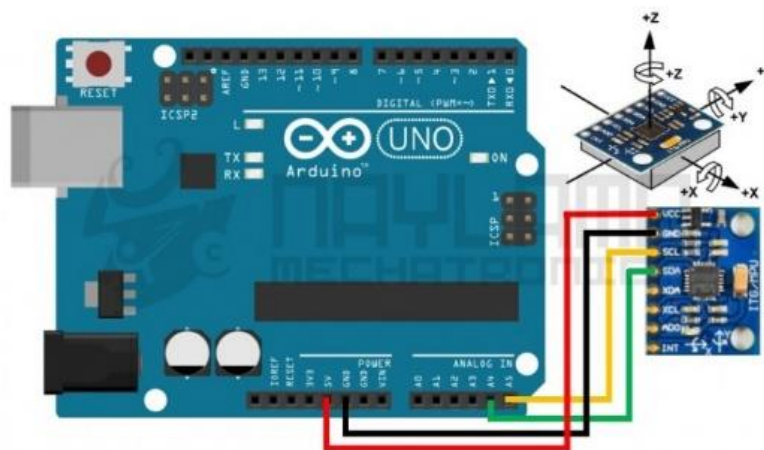


Figura 3.15 Conexión de MPU6050.

Fuente: (Naylamp Mechatronics SAC, 2019)

Se muestra en la Tabla 3.10 las conexiones.

Tabla 3.10
Conexión MPU6050

MPU6050	Nano Arduino
VCC	5v
GND	GNG
SCL	A5
SDA	A4

Fuente: Autor

3.6.11 Sensor TCRT5000 infrarrojo

El TCRT5000 es un sensor óptico reflectivo que utiliza la luz infrarroja como emisor y un fototransistor como receptor, este fototransistor detecta la luz reflejada cuando un objeto pasa frente al sensor.

En este proyecto serán utilizados dos sensores de este tipo para realizar un contador de la variable tiempo, uno de inicio y reset y el otro de fin. Su conexión es muy sencilla se utiliza Vcc y Gnd y un pin de la lectura, se muestra en la Figura 3.16 la conexión.

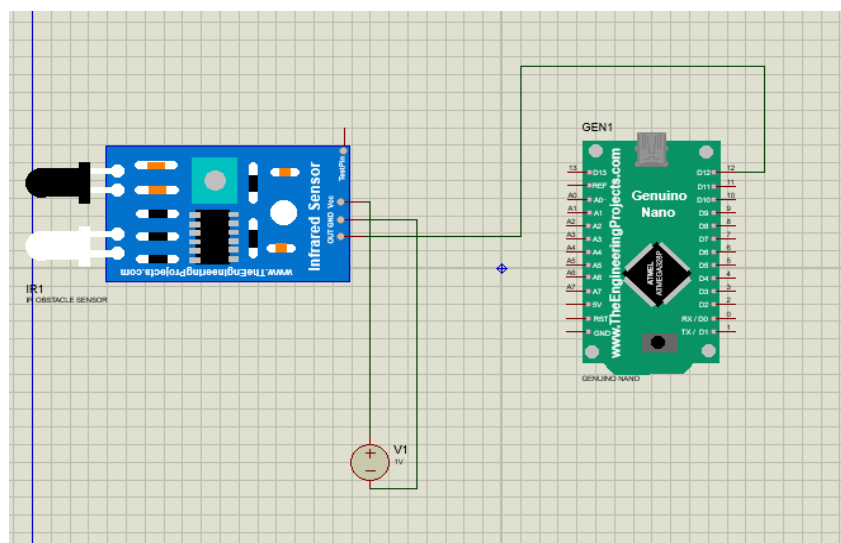


Figura 3.16 Conexión de Sensor Infrarrojo

Fuente: Autor

Conexiones de sensor infrarrojo con Arduino Nano se muestran en la siguiente Tabla 3.11.

Tabla 3.11
Conexión de Sensor TCRT5000

TCRT5000	Nano Arduino
VCC	VCC
GND	GNG
OUT	11/12

Fuente: Autor

3.6.12 Sensor de fuerza FSR402

Este sensor está compuesto por una delgada película de polímero, su funcionamiento se basa en el cambio de resistencia al aplicarle presión, su rango de medición es de 0.2N a 20N sus aplicaciones son diversas, en este proyecto será utilizado para la medición de la fuerza generada al impacto de nuestro modulo con la pared de la rampa de deslizamiento, se muestra en la Figura 3.17 la conexión para su funcionamiento.

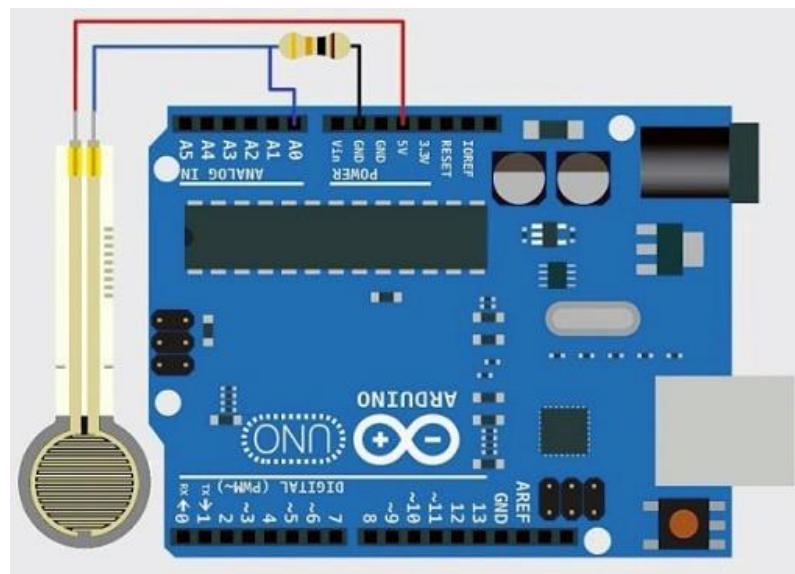


Figura 3.17 Conexión de FSR.

Fuente: (Carrión, Ochoa, & Valverde, 2009)

3.6.13 Bluetooth Hc-06

Este dispositivo comunica vía bluetooth a cualquier dispositivo a una velocidad de 2.4Mhz, la configuración de este dispositivo es sencillo y en este caso servirá para conexión entre el módulo de la partícula y la interfaz gráfica en el PC. Su conexión es en los terminales tx y rx en el Arduino y se conectan invirtiendo los pines, como podemos ver en la Figura 3.18.

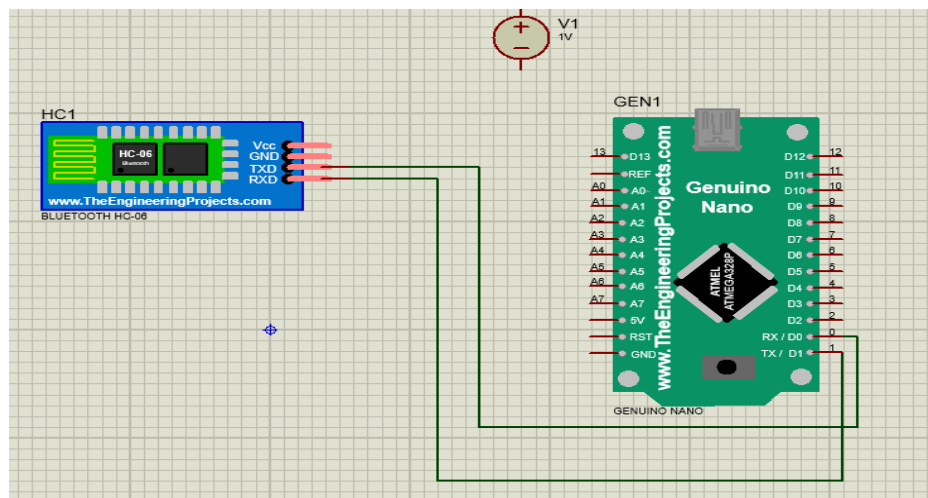


Figura 3.18 Conexión HC-06
Fuente: Autor

3.7 Diseño de placas

3.7.1 Placa base de control de torres

Esta placa fue diseñada de tal manera de que sus componentes sean de fácil acceso y reemplazo, su distribución de pines y terminales es evitando el acceso de cables y conexiones externas y optimizando el espacio.

El diseño de la placa base se realiza en proteus 8, se muestra en la Figura 3.19.

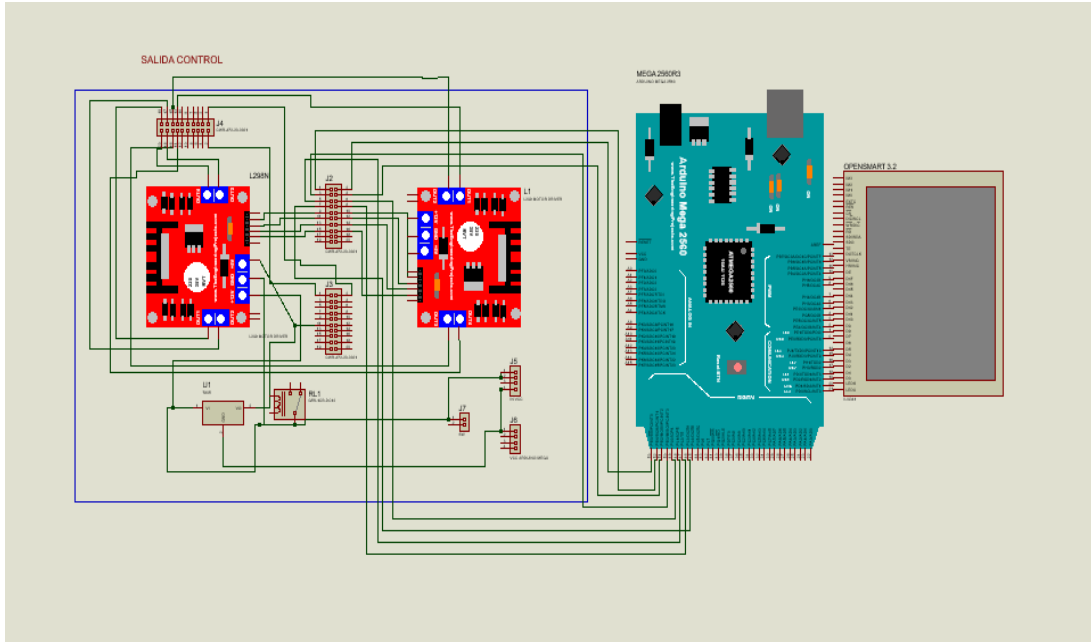


Figura 3.19 Diseño de Placa base Control torres
Fuente: Autor

El diseño de la PCB también se realiza en proteus 8, ver la Figura 3.20

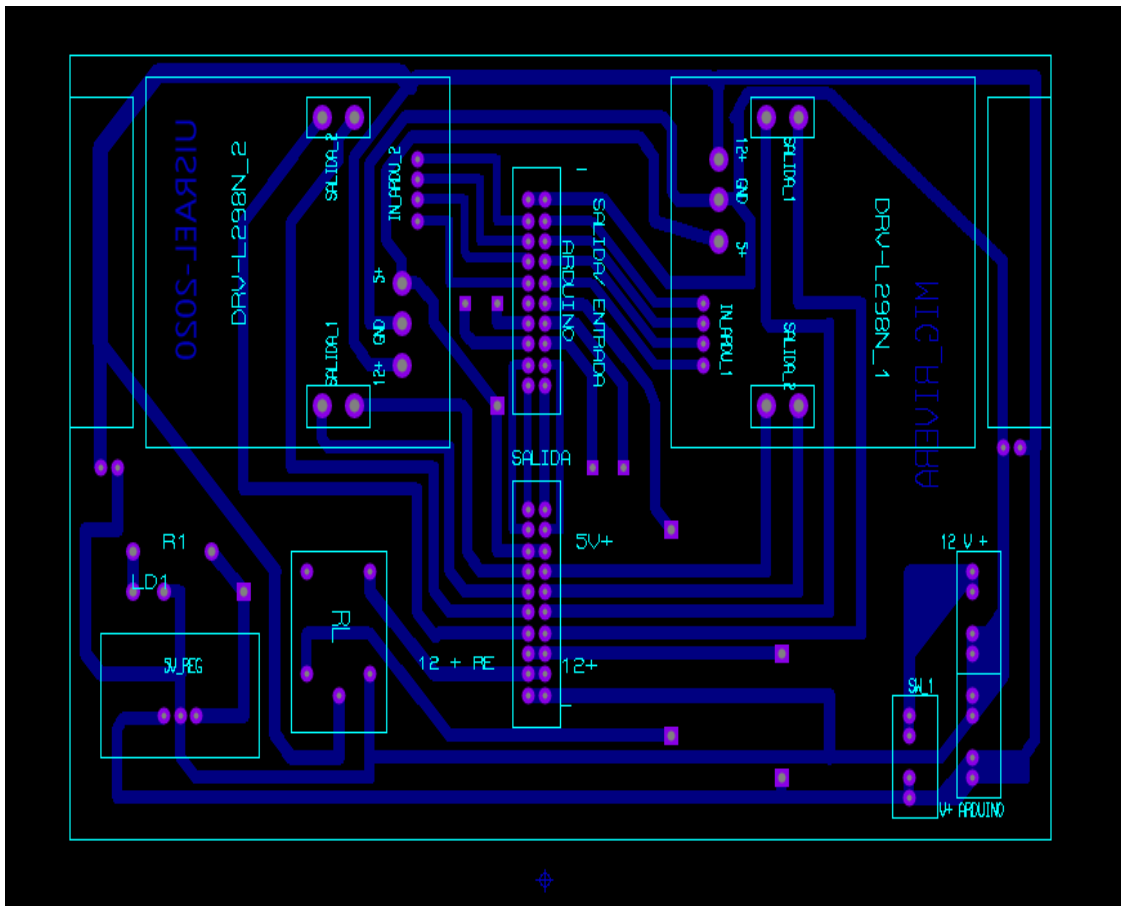


Figura 3.20 Diseño de PCB placa base Control Torres
Fuente: Autor

3.7.2 Placa base de módulo autónomo

Esta placa soporta los sensores MPU6050, el Arduino Nano, los sensores TRCT5000, el sensor de carga HX711, el sensor FSR402 y la pantalla LCD la para lo cual se ha realizado el diseño de esta placa de manera que todos sus componentes sean de fácil conexión, acceso y se muestra en la Figura 3.21.

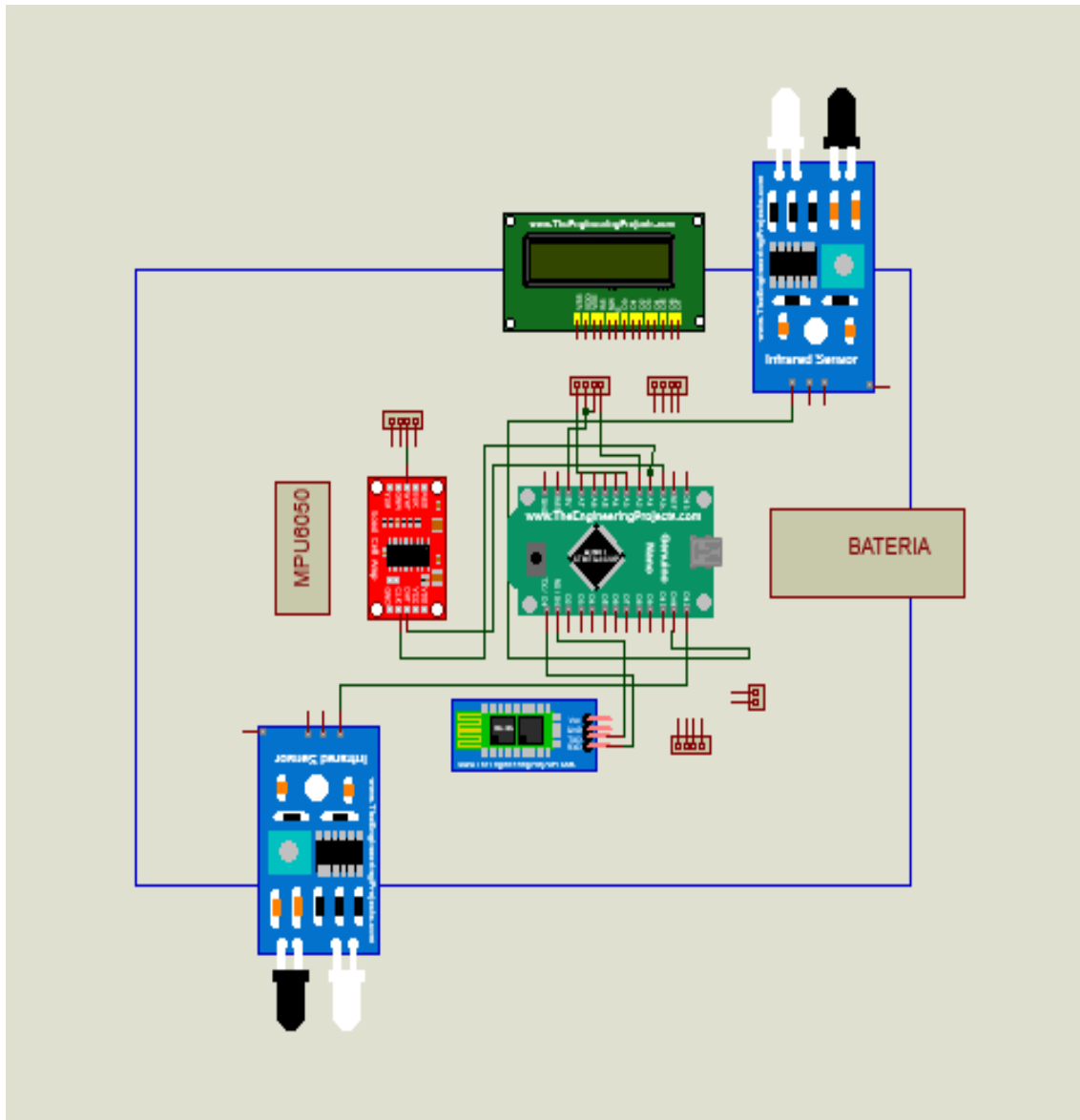


Figura 3.21 Diseño de módulo autónomo

Fuente: Autor

El diseño de PCB se adjunta en la Figura. 3.22.

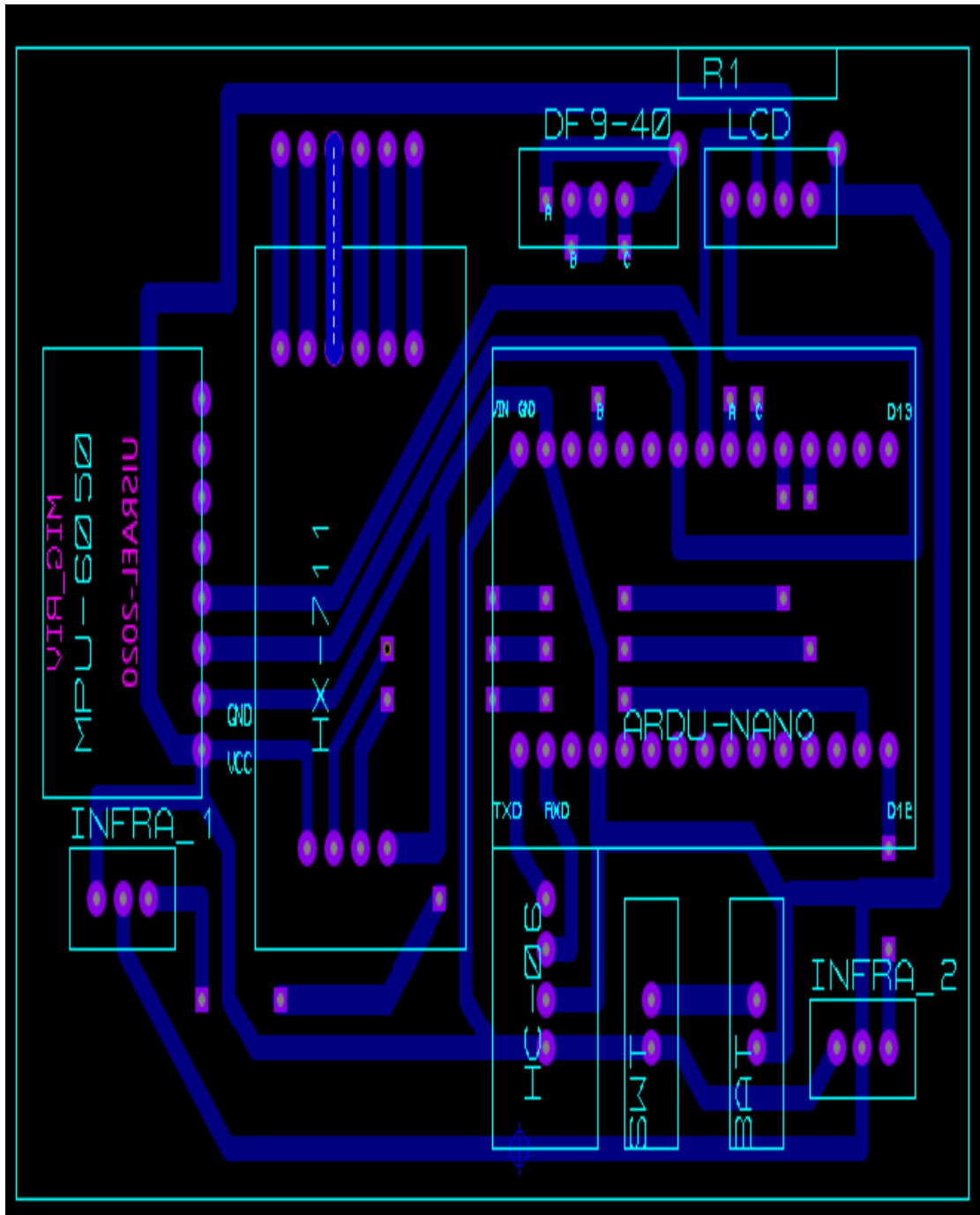


Figura 3.22 Diseño PCB módulo autónomo.
Fuente: Autor

3.7.3 Diseño de soporte de motores y carcasa de módulo autónomo.

En esta investigación se diseña con el software Adobe Ilustrador los componentes que contendrán los motores bipolares y los sensores ultrasónicos, se puede observar en la Figura 3.23.

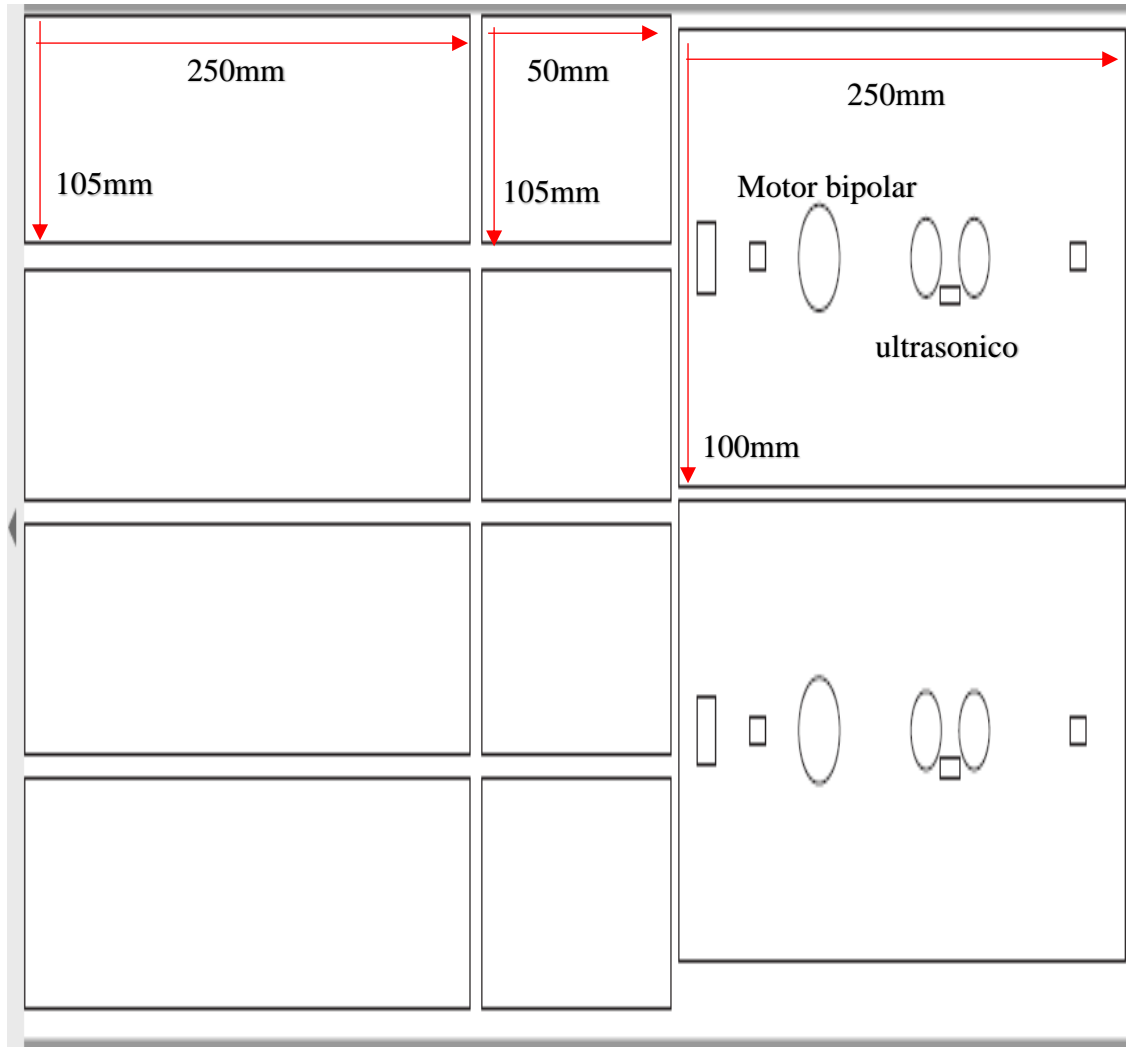


Figura 3.23 Contenedor de Motor bipolar y sensor ultrasónico
Fuente: Autor

En la siguiente figura se muestra el diseño y cortes de la estructura de la estructura módulo autónomo así como los cortes para los sensores, podemos observar en la Figura 3.24

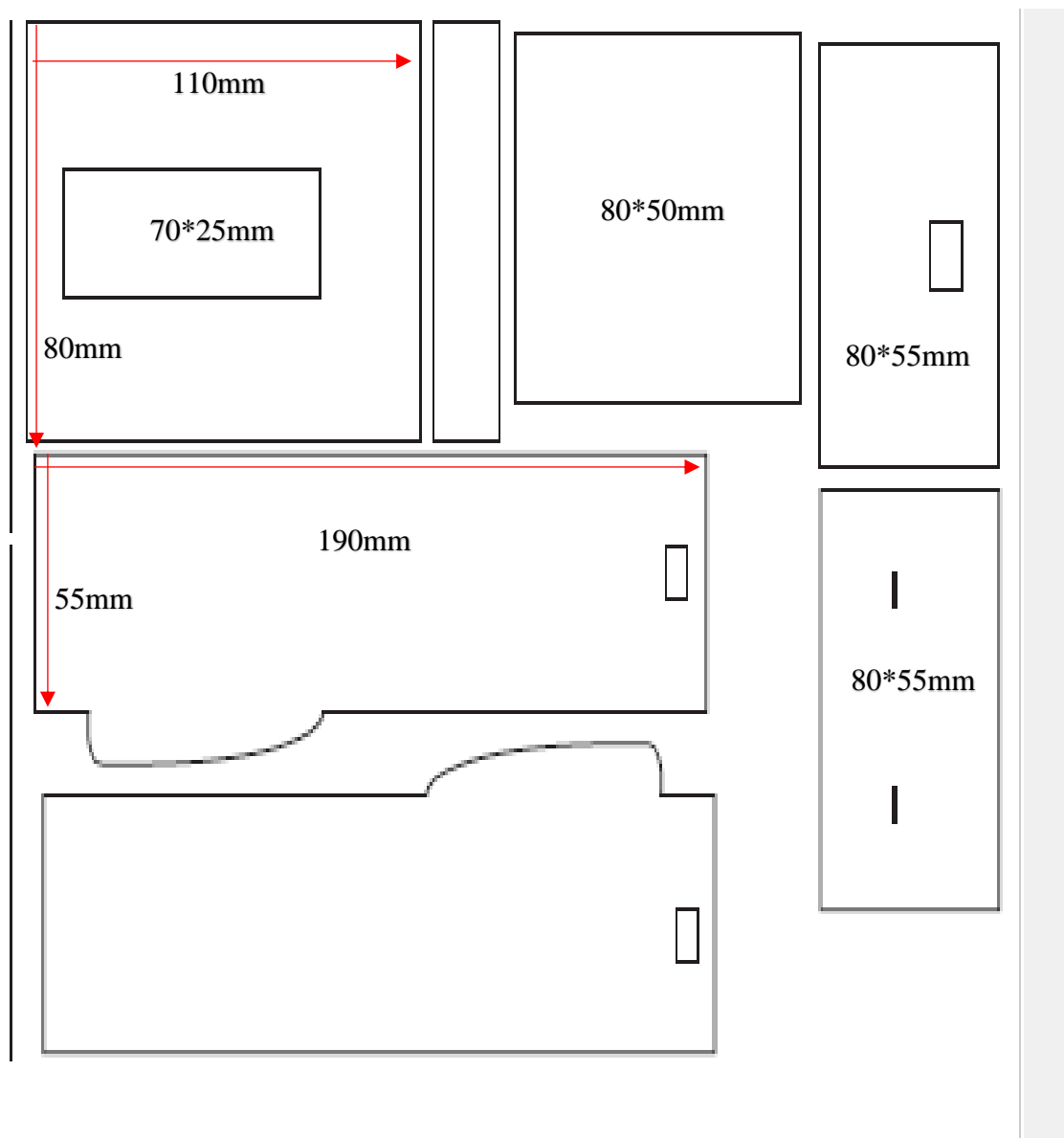


Figura 3.24 Diseño de módulo autónomo.
Fuente: Autor

La estructura se realiza con perfiles de aluminio de 35*35mm y la rampa de 82mm*2000mm, se puede observar en la Figura 3.25.

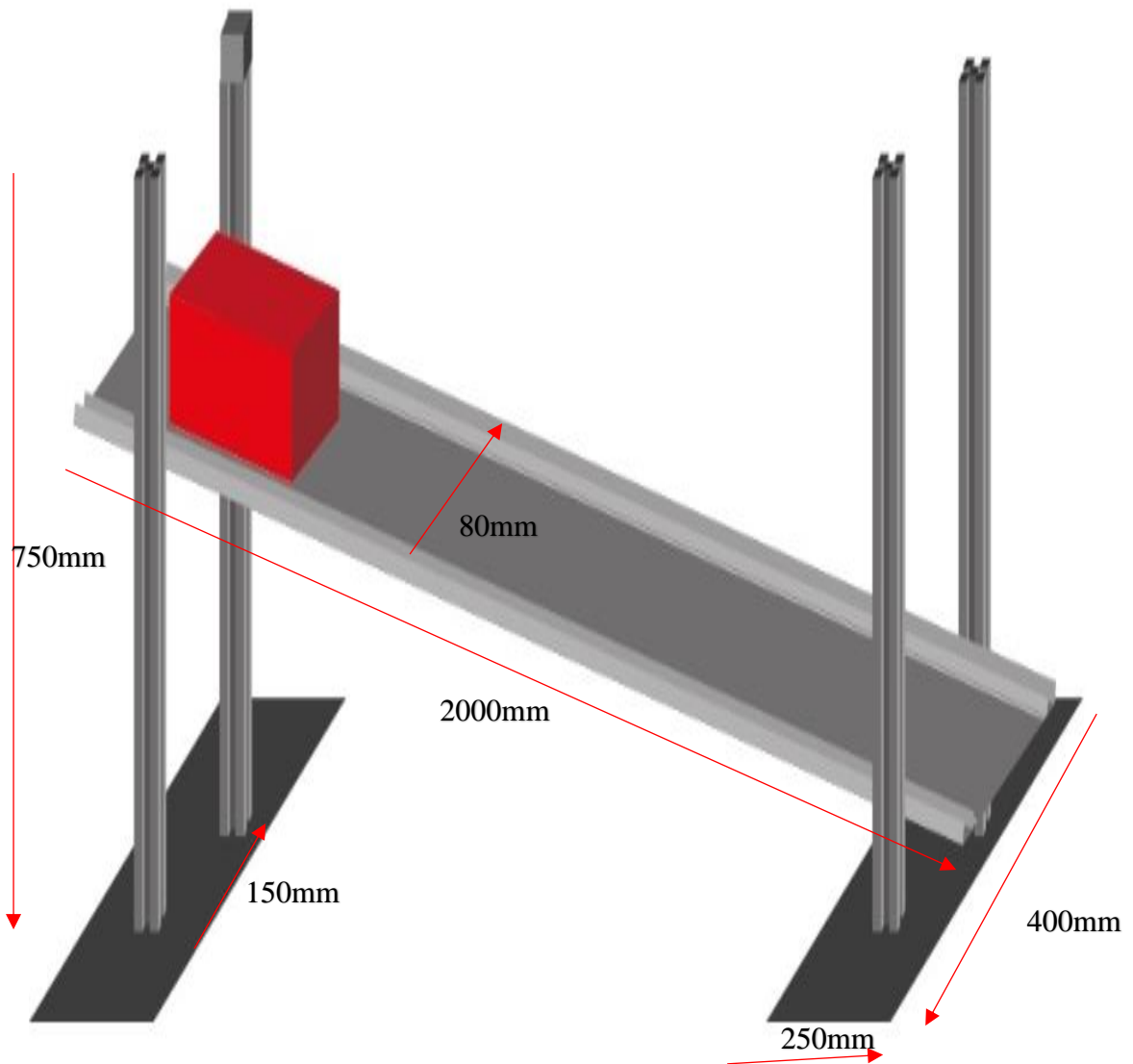


Figura 3.25 Diseño de torres y Rampa
Fuente: Autor

3.8 Diseño de interfaz gráfica en LabVIEW

Para la visualización de los datos obtenidos por el módulo autónomo para el estudio de la cinemática se realiza un aplicativo en el software LabVIEW, en este podremos realizar las pruebas y verificaciones antes de proceder al armado de nuestros dispositivos con lo cual se establece la programación de la siguiente manera.

Para la captación de datos por el puerto serial se coloca una herramienta llamada VISA Configure Serial Port VI la cual establecemos el puerto de comunicación COM y los baudios de transmisión de datos, es la principal herramienta para la comunicación entre nuestro módulo autónomo y la interfaz gráfica en el PC. A continuación, establecemos un nodo de comunicación y la lectura de los datos por medio de la herramienta VISA READ, que en conjunto procesan los datos enviados desde el módulo hasta la interfaz.

Para mostrar los datos es necesario utilizar un Array lo cual nos permitirá el separar los datos para luego mostrarlos en las casillas de nuestra interfaz.

Se muestra la configuración completa de la programación en la Figura 3.26.

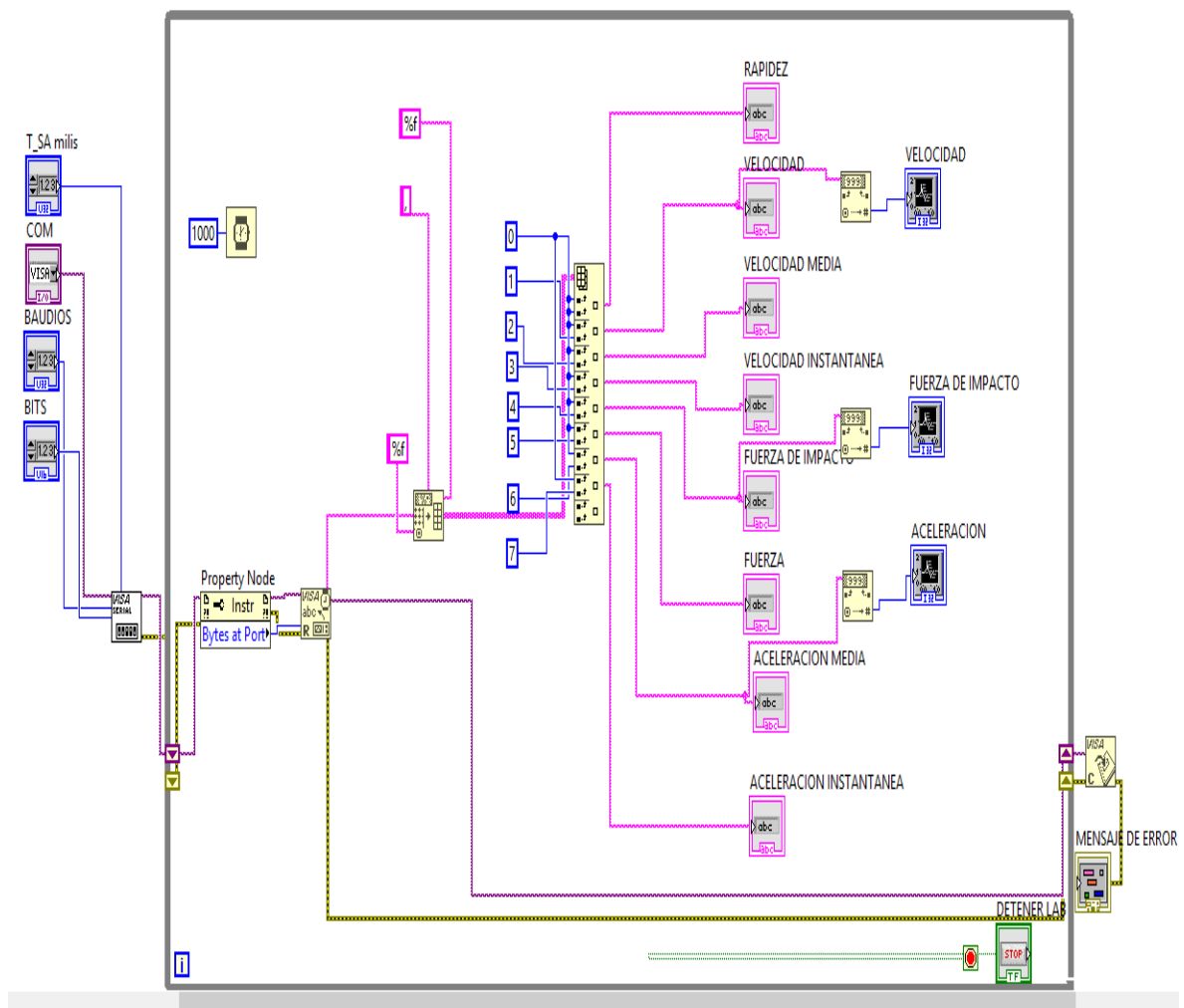


Figura 3.26 Programación de interfaz en LabVIEW

Fuente: Autor

La pantalla que muestra los datos obtenidos contiene las variables censadas tales como como la rapidez, velocidad final, velocidad media velocidad instantánea, fuerza de impacto, la fuerza generada en el módulo y la aceleración, observamos en la interfaz de la Figura 3.27.



Figura 3.27 Interfaz de captura de datos
Fuente: Autor

3.9 Análisis de costos

Según la investigación de este proyecto se realiza un análisis de costos para la implementación de este módulo autónomo para el estudio de la cinemática, los materiales requeridos fueron presupuestados en el mercado local, estos componentes cumplen con los estándares de resistencia y calidad para el mejor desarrollo de este proyecto.

Tabla 3.12

Costos de implementación de proyecto

COMPONENTES ELECTRÓNICOS				
Item	Descripción	Cantidad	VALOR U.	VALOR TOTAL
1	Arduino Mega 2560r3	1	19,00	19,00
2	Arduino Nano	1	8,00	8,00
3	Módulo HL298N	2	8,00	16,00
5	Fuente de alimentación 12v 30A	1	33,00	33,00
6	Regulador 7805	1	1,00	1,00
7	Pantalla TFT OpenSmart	1	45,00	45,00
8	Relé 5v	1	0,50	0,50
9	Motor bipolar	2	11,50	23,00
10	Sensor HC-sr04	2	2,50	5,00
11	Sensor Hx711 y celda de carga	1	10,00	10,00
12	Sensor MPU6050	1	4,00	4,00
13	Pantalla LCD 16x 2	1	6,00	6,00
14	Sensor infrarrojo	2	2,50	5,00
15	Sensor FSR402	2	12,00	24,00
16	Hc-06	1	7,00	7,00
17	Cajas de proyectos	2	15,00	30,00
18	Placas	2	4,00	8,00
19	Conectores y cables	1	10,00	10,00
20	Varios	1	20,00	20,00
			TOTAL_1	274.50
MATERIALES PARA ESTRUCTURA Y RAMPA				
1	Barra de aluminio	2	25,00	50,00
2	Perfil de aluminio	1	28,00	28,00
3	Cortes Varios	1	25,00	25,00
			TOTAL_2	103,00
<i>Total_1+Total_2</i>				377.50

Fuente: Autor

3.10 Ventajas del Módulo autónomo.

La implementación de este módulo conlleva grandes desafíos sin embargo se evalúa el enorme potencial y ventajas de la elaboración que detallamos a continuación:

- Dispositivos electrónicos de fácil acceso
- Fácil programación de sensores
- Interfaz gráfica intuitiva
- Simple manipulación de estructura
- Montaje y desmontaje rápido
- Módulo autónomo de fácil interpretación.
- Ayuda a la interpretación de conceptos de la cinemática de forma simple
- Gran herramienta de enseñanza para el profesorado.

CAPITULO 4

4. IMPLEMENTACIÓN

En este capítulo se describe el proceso de construcción y comprobación de funcionamiento del módulo autónomo para el aprendizaje de los principios básicos de la cinemática en el ámbito universitario, así como también la fabricación de la estructura y el control que soporta al módulo y finalmente validar los resultados de la investigación.

4.1 Ensamblaje, programación y comprobación de los componentes

Una vez definidos los componentes a utilizar se procede al ensamblaje, programación y comprobación de cada uno de los sensores en sus respectivas carcasas.

4.1.1 Control torres.

Está compuesto de la placa base que soporta los puentes HL298N, las entradas y salidas del Arduino Mega 2560rev3 que a su vez está conectada a la pantalla OpenSmart TFT 3.2” que ayudará a crear una interfaz amigable e intuitiva con el usuario y controla los sensores de HC-sr04 Ultrasónicos, es alimentada por la fuente de 12V DC a una corriente de 30A, estos dispositivos nos ayudan a controlar la inclinación de la rampa en donde se desplazará el módulo autónomo. Para ver la programación completa ver Anexo 5

A continuación, se muestra la distribución de los componentes en la placa base y el diseño de la PCB realizada en proteus 8, Figura 4.1 y 4.2.

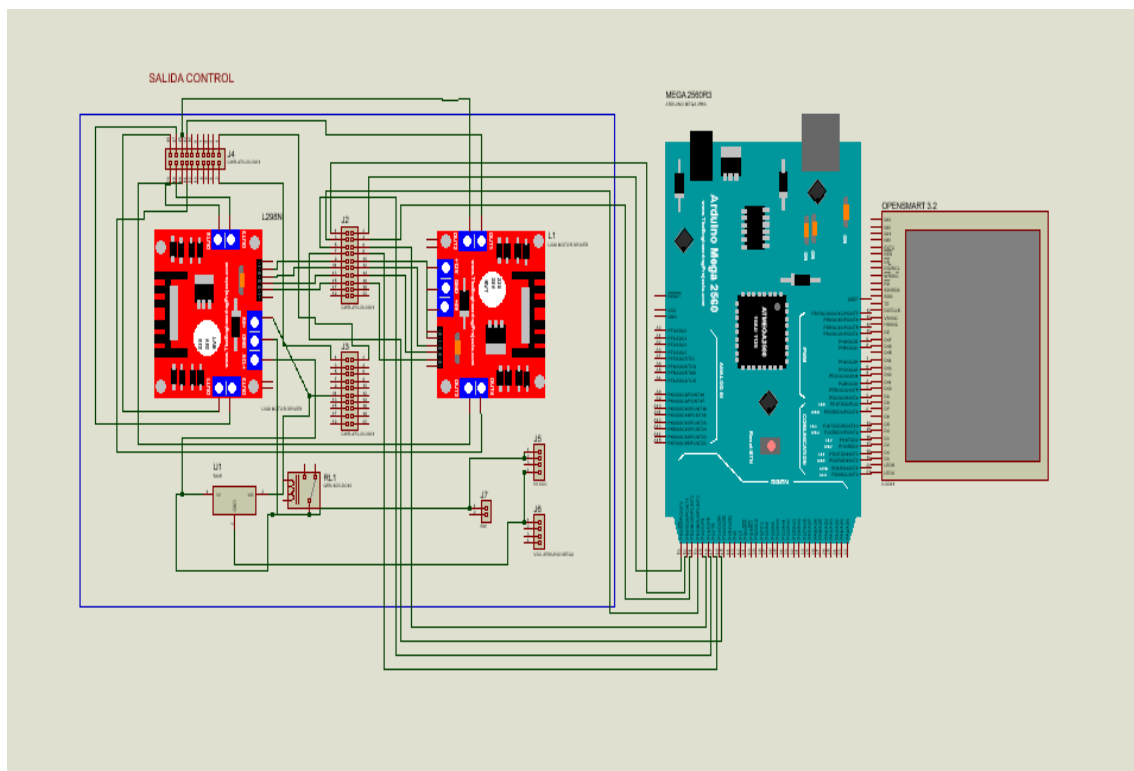


Figura 4.1 Distribución de componentes de la placa base

Fuente: Autor

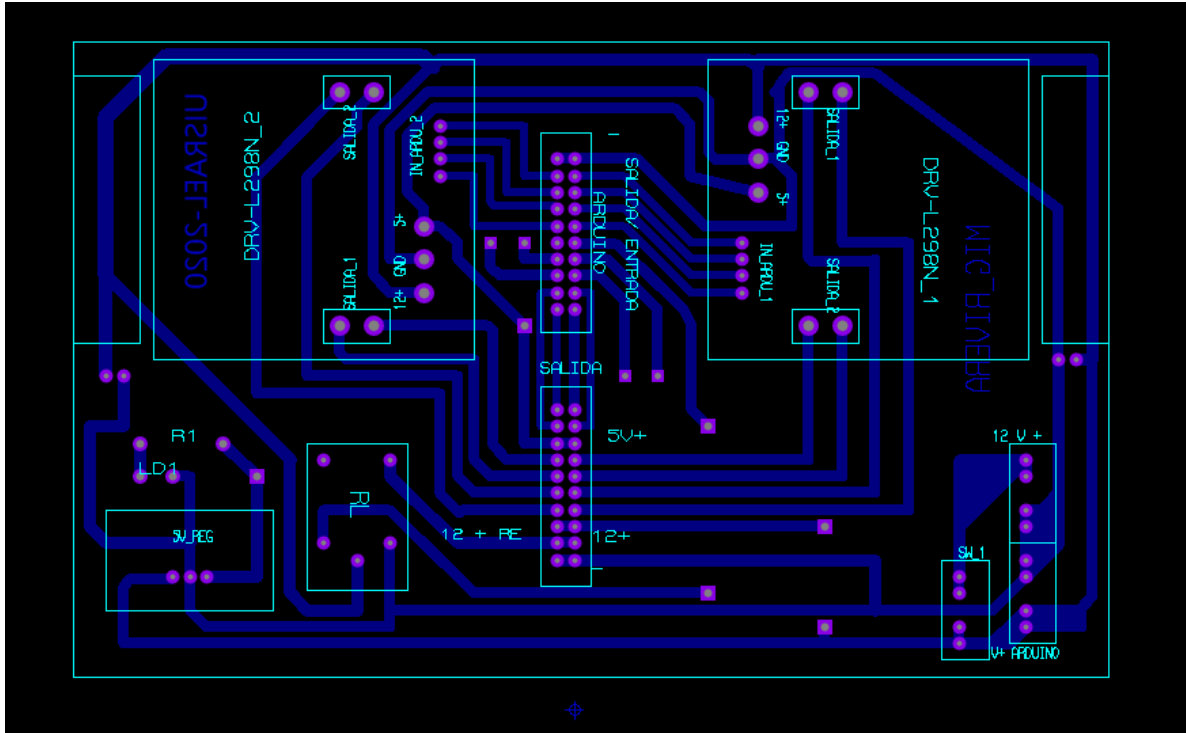


Figura 4.2 Diseño PCB placa base
Fuente: Autor

La función de esta placa base es la de soportar los puentes HL298N, los cuales reciben los impulsos de la placa Arduino Mega 2560 R3 para controlar los motores bipolares de las torres de elevación esto se consigue con la programación del Arduino mega 2560 Rev3.

4.1.2 Instalación, programación de los puentes L298N y motores bipolares

Los dos módulos L298N son instalados en la placa base del control de Torres los cuales son controlados con la programación en el Arduino Mega 2560rev3, en esta se establecen parámetros de funcionamiento como son los pasos de los motores, la velocidad y sentido de giro. Esto se realiza con la ayuda de la librería Stepper.h de Arduino que nos ayuda a controlar los motores bipolares. Se adjunta la programación en la Figura 4.3 y la instalación en la placa base. Figura 4.4

```
motor_paso_a_paso

#include <Stepper.h> //Importamos la librería para controlar motores paso a paso
#define STEPS 48//Ponemos el número de pasos que necesita para dar una vuelta. 200 en nuestro caso

// Ponemos nombre al motor, el número de pasos y los pins de control
Stepper M_elevador_1(STEPS,50,51,52,53); //M_elevador nombre motor (número de pasos por vuelta, pins de control)
Stepper M_elevador_2(STEPS,46,47,48,49);
void setup()
{
  // Velocidad del motor en RPM
  M_elevador_1.setSpeed(500);
  M_elevador_2.setSpeed(500);
}

void loop()
{
  M_elevador_1.step(100000);}
  delay(500); //Pequeña pausa
  M_elevador_1.step(100000);
  delay(500);
  //////////////////////////////////////
  M_elevador_2.step(100000);
  delay(500); //Pequeña pausa
  M_elevador_2.step(100000);
  delay(500);
}

Compilado

El Sketch usa 2268 bytes (7%) del espacio de almacenamiento de programa. El máximo es 30720 bytes.
Las variables Globales usan 61 bytes (2%) de la memoria dinámica, dejando 1987 bytes para las variables locales. E
```

Figura 4.3 . Programación para los L298N

Fuente: Autor

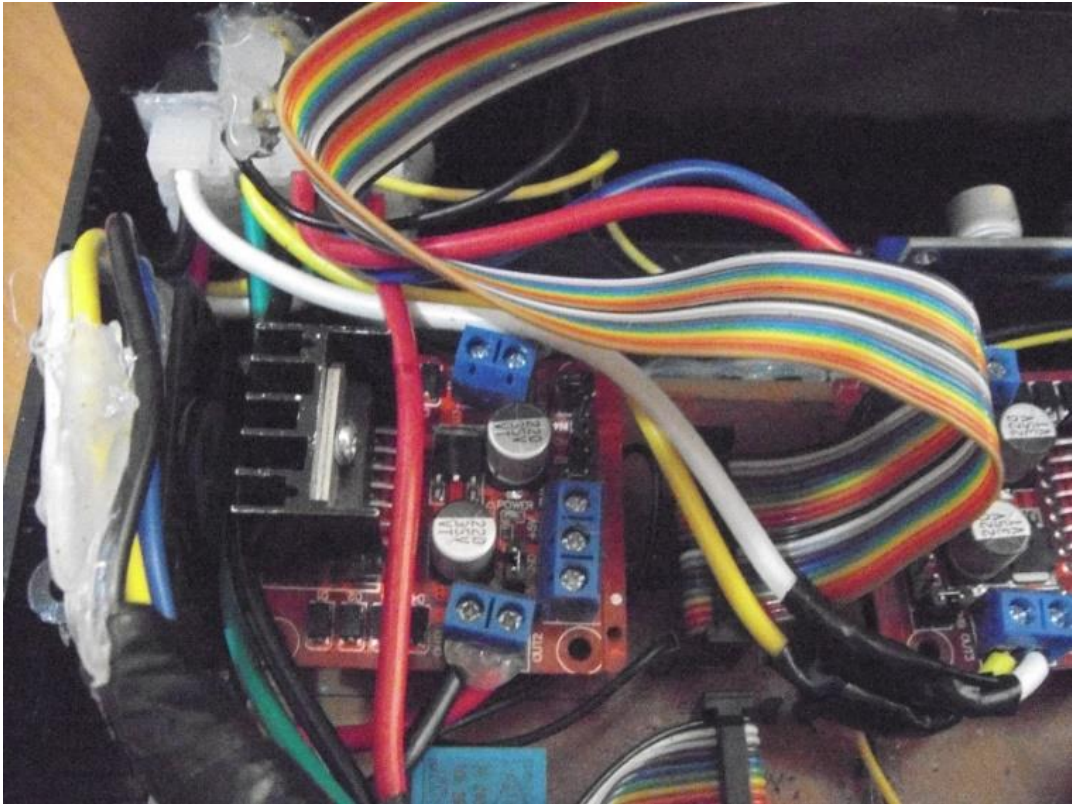


Figura 4.4 Montaje del L298N en la placa base control

Fuente: Autor

Los motores son instalados en la parte superior de cada torre que a su vez están conectados al perno roscado que al girar mueven las bases de la rampa, colocando así en la inclinación deseada, ver la Figura 4.5



Figura 4.5 Instalacion de motor Bipolar

Fuente: Autor

4.1.3 Instalación y programación de pantalla OpenSmart TFT 3.2”

Esta pantalla es de fácil instalación ya que la distribución de pines se conecta directamente al Arduino Mega 2560 rev3 y la programación del táctil con su respectiva acción se lo realiza con la ayuda de las librerías TouchScreen.h y OPENSMTART_kbv.h, luego definiendo los pines del táctil, el tamaño de la pantalla para delimitar el uso del táctil, los pines de LCD donde muestra los gráficos. A continuación, se declaran variables y se procede en el void setup el inicio de la pantalla con su respectivo controlador. El en el void loop se configura las entradas y salidas para controlar la inclinación de torres.

Se adjunta la programación y el montaje entre el Arduino Mega en la Figura 4.6 y la pantalla OpenSmart TFT 3,2” en la Figura 4.7.



```

#include <Stepper.h>
#include <TouchScreen.h>
#include <OPENSMTART_kbv.h>
#include <LM75.h>
#include <Wire.h>
// Pines necesarios para los 4 pines del panel tactil
#define YP A1 // Pin analogico A1 para ADC
#define YM A2 // Pin analogico A2 para ADC
#define YM 7
#define YP 6
#define MINPRESSURE 1
#define MAXPRESSURE 1500
TouchScreen ts = TouchScreen(XP, YP, YM, YM, 660);
short TS_MINX = 5; // Coordenadas del panel tactil para delimitar
short TS_MINY = 5; // el tamaño de la zona donde podemos presionar
short TS_MAXX = 950; // y que coincida con el tamaño del LCD
short TS_MAXY = 950;
#define LCD_CS A3 // Definimos los pines del LCD
#define LCD_CD A2 // para poder visualizar elementos graficos
#define LCD_WR A1
#define LCD_RD A0
#define LCD_RESET A4
#define NEGRO 0x0000 // Definimos los colores
#define AZUL 0x001F // que utilizaremos para
#define ROJO 0xF800 // el texto y los elementos graficos
#define VERDE 0x07E0
#define CYAN 0x07FF
#define MAGENTA 0xF81F
#define AMARILLO 0xFFE0
#define BLANCO 0xFFFF
OPENSMTART_kbv pantalla(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET);
int X; // Variables que almacenaran la coordenada
int Y; // X, Y donde presionemos y la variable Z
int Z; // almacenara la presion realizada
//CONTROL TORRES
const int Trigger_A = 38; //Pin digital 2 para el Trigger del sensor
const int Echo_A = 36; //Pin digital 3 para el Echo del sensor
const int Trigger_B = 38; //Pin digital 2 para el Trigger del sensor
const int Echo_B = 37;

void lecturaPanel()
{
  digitalWrite(13, HIGH);
  TSPoint p = ts.getPoint(); // Realizamos lectura de las coordenadas
  digitalWrite(13, LOW);

  pinMode(XM, OUTPUT); // La libreria utiliza estos pines como entrada y salida
  pinMode(YP, OUTPUT); // por lo que es necesario declararlos como salida justo
                        // despues de realizar una lectura de coordenadas.

  // Mapeamos los valores analogicos leidos del panel tactil (0-1023)
  // y los convertimos en valor correspondiente a la medida del LCD 320x240
  // X = map(p.x, TS_MAXX, TS_MINX, tft.width(), 0);
  // Y = map(p.y, TS_MAXY, TS_MINY, tft.height(), 0);
  //si la configuracion anterior de pantalla no funciona
  //ensaya esta
  X = map(p.x, TS_MAXX, TS_MINX, pantalla.width(), 0);
  Y = map(p.y, TS_MINY, TS_MAXY, pantalla.height(), 0);
  Z = p.z;
  /* Serial.println("////////////////////////");
  Serial.println(X);
  Serial.println(Y);
  Serial.println(Z);
  delay(500);*/
}

```

Figura 4.6 Programación de pantalla OpenSmart TFT 3.2”

Fuente: Autor

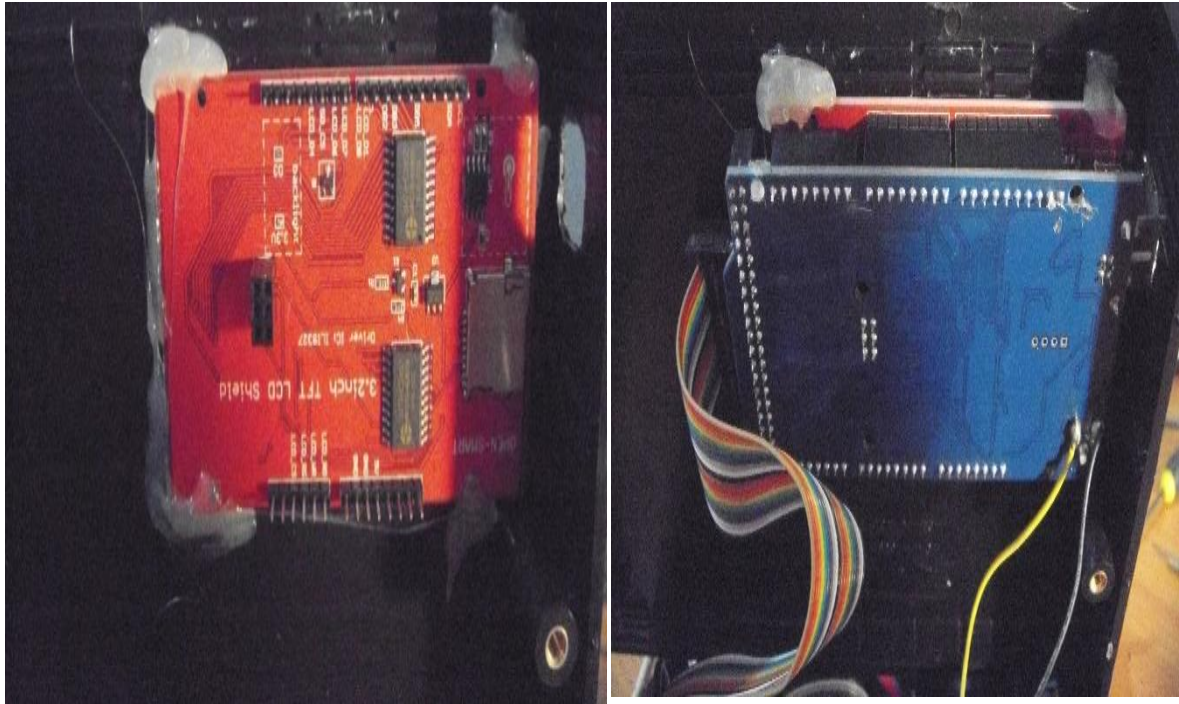


Figura 4.7 Montaje de OpenSmart con Arduino Mega2560r3

Fuente: Autor

4.1.4 Regulador de voltaje 7805

Este dispositivo es primordial para el funcionamiento de los puentes L298N los cuales necesitan alimentación de 5v DC para poder trabajar con normalidad, el voltaje de entrada en este dispositivo es el de la fuente 12v y lo estabiliza a 5v. Se ha instalado con un disipador de calor para que trabaje en su rango de temperatura de 125°C, ver Figura 4.8.



Figura 4.8 Regulador de voltaje 5v DC

Fuente: Autor

4.1.5 Sensor HC-sr04 Ultrasonido.

Este sensor nos ayuda medir la posición de la rampa del módulo, se han colocado dos sensores uno en cada torre para que con las lecturas ordene a los motores el movimiento y ubique la rampa en la posición deseada, ver la Figura 4.10.

La programación de este dispositivo se realiza declarando variable como Echo y Trig, la variable Echo envía un impulso de alta frecuencia y el Trig lo recibe calculando el tiempo que se demora entre el envío y la recepción de la onda, observamos en la Figura 4.9.

pantalla_tactil_control_torres-15-12-2019-ok	LecturaPanel	PANTALLA_INICIO	PANTALLA_MRU	PANTALLA_MRUV	Salir	inicio_de_practica	me
--	--------------	-----------------	--------------	---------------	-------	--------------------	----

```

void medir_torres()
{
  long t_A; //timepo que demora en llegar el eco
  long d_A; //distancia en centimetros
  digitalWrite(Trigger_A, HIGH);
  delayMicroseconds(10); //Enviamos un pulso de 10us
  digitalWrite(Trigger_A, LOW);
  t_A = pulseIn(Echo_A, HIGH); //obtenemos el ancho del pulso
  d_A = (t_A/59); //escalamos el tiempo a una distancia en cm
  Serial.print("Distancia TORRE_A: ");
  Serial.print(d_A); //Enviamos serialmente el valor de la distancia
  Serial.print("cm");
  Serial.println();
  delay(10); //Hacemos una pausa de 100ms
  ////////////////////////////////////////////////////////////////////
  long t_B; //timepo que demora en llegar el eco
  long d_B; //distancia en centimetros
  digitalWrite(Trigger_B, HIGH);
  delayMicroseconds(10); //Enviamos un pulso de 10us
  digitalWrite(Trigger_B, LOW);
  t_B = pulseIn(Echo_B, HIGH); //obtenemos el ancho del pulso
  d_B = (t_B/59); //escalamos el tiempo a una distancia en cm
  Serial.print("Distancia TORRE_B: ");
  Serial.print(d_B); //Enviamos serialmente el valor de la distancia
  Serial.print("cm");
  Serial.println();
  delay(10);

  if ((d_B>50)&&(d_B<50)){
    M_elevador_B.step(-100);
  } else if ((d_B<50)&&(d_B>50)){
    M_elevador_B.step(100);
  }
}

```

Compilado

Figura 4.9 Programación de HC-sr04

Fuente: Autor



Figura 4.10 Instalación de HC-sr04

Fuente: Autor

4.1.6 Ensamblaje del control torres.

Una vez instalados todos los componentes electrónicos en la placa base, se procede a el ensamblaje de las torres y la estructura que servirá de soporte para el movimiento del módulo autónomo de cinemática para la cual se arman las torres de forma individual acoplando en la parte superior el motor y el sensor HC-sr04 ultrasónico, el motor acopla al eje roscado obteniendo una alineación a 90°, ver la Figura 4.11.

Se detalla a continuación:

- Se instalan los ejes de las torres

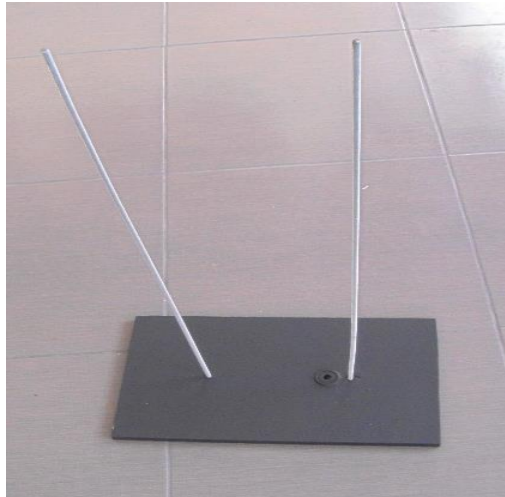


Figura 4.11 Instalación ejes de torres
Fuente: Autor

- Se colocan las bases de las torres, ver Figura 4.12.



Figura 4.12 Bases de las torres
Fuente: Autor

- Acoplar la parte superior de las torres, ver la Figura 4.13.



Figura 4.13 Instalacion de parte superior de torres
Fuente: Autor

- Colocación de soportes horizontales, ver Figura 4.14.



Figura 4.14 Soportes laterales
Fuente: Autor

- Instalación de rampa, instalación 4.15



Figura 4.15 Instalación de rampa
Fuente: Autor

- Colocación de marcas métricas y cableado, ver la Figura 4.16

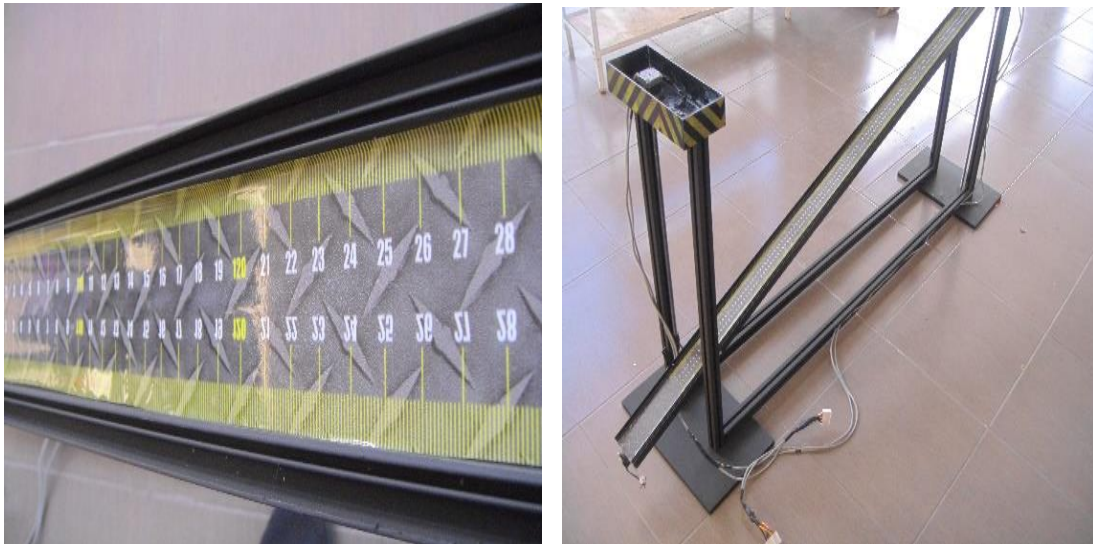


Figura 4.16 Marcas métricas y Cableado
Fuente: Autor

- Instalación de módulo de control y fuente de alimentación, ver la Figura 4.17

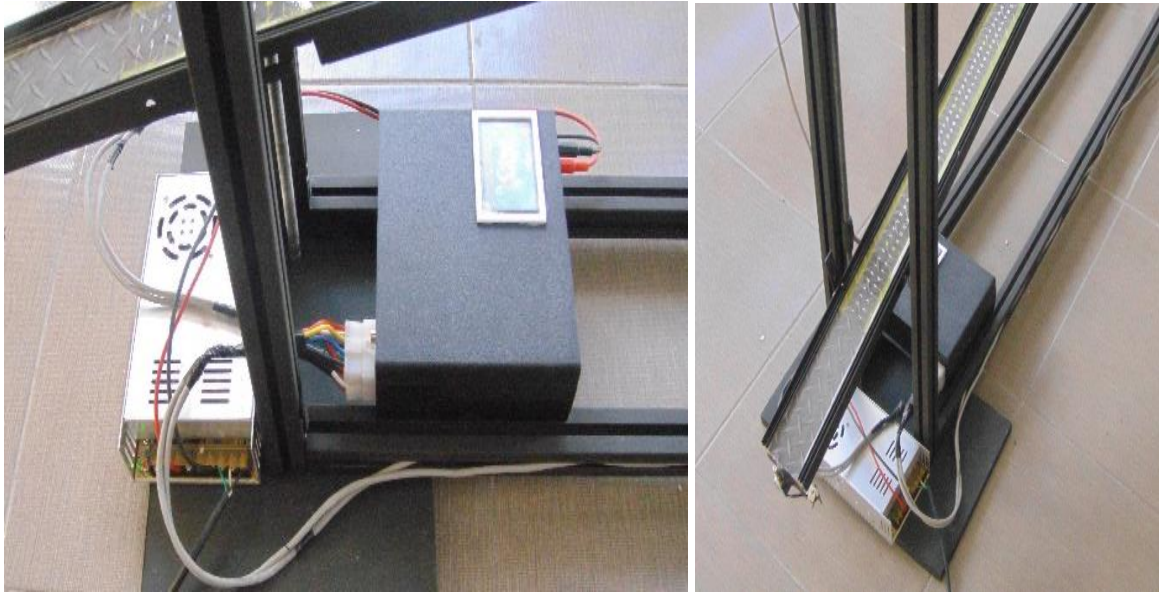


Figura 4.17 Instalación de Fuente de alimentación y Control Torres
Fuente: Autor

4.1.7 Ensamblaje de estructura.

Luego de integrar todos los componentes de la estructura se puede observar en la Figura 4.18 y 4.19 todo el prototipo de la estructura ensamblado y listo para su utilización.



Figura 4.18 Estructura terminada
Fuente: Autor



Figura 4.19 Estructura terminada

Fuente: Autor

4.1.8 Interfaz gráfica de módulo

Para el control la inclinación y de la rampa de pruebas el módulo control torres se ha diseñado una interfaz gráfica y táctil con la ayuda de la pantalla OpenSmart TFT 3.2" y el Arduino Mega 2560rev3 de tal manera que el usuario pueda interactuar con los mecanismos de inclinación de la estructura para ello se nombra a las torres como torres A y torre B, además las inclinaciones generadas por los motores bipolares son simétricas en las dos torres.

Al iniciar el módulo control de torres se presenta las primeras imágenes las cuales indican del inicio del módulo, ver en la Figura 4.20



Figura 4.20 . Pantalla de inicio de módulo
Fuente: Autor

Luego de esta presentación muestra la pantalla de inicio donde podremos decidir si realizar el movimiento rectilíneo Uniforme o el movimiento rectilíneo uniformemente acelerado, ver la Figura 4.21.



Figura 4.21 Pantalla principal del módulo del control
Fuente: Autor

Presionando en el apartado MRU se muestra en pantalla la opción que permite nivelar la rampa a una distancia de 63 cm de la base inferior y a 10 cm de la base superior las torres A y B donde se coloca en la partida el módulo autónomo para el estudio de la cinemática, ver la Figura 4.22.



Figura 4.22 Pantalla MRU nivelación de la rampa
Fuente: Autor

Al terminar la nivelación presionamos OK y pasaremos a la pantalla previa al inicio de la práctica.

Presionando el táctil en el botón inicio se realiza el disparo del Selenoide que libera al módulo autónomo del estudio de la cinemática, esto muestra en pantalla un tiempo de 1,5 segundos hasta que termine la práctica Muestra en la Figura 4.23 el botón de inicio y en la Figura 4.24 el tiempo hasta que termine la práctica.

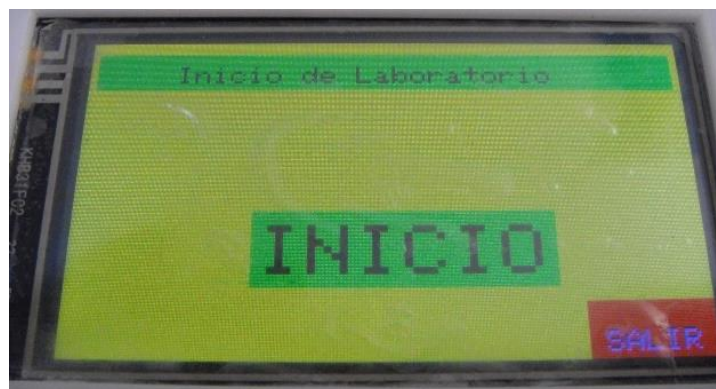


Figura 4.23 Pantalla lista para iniciar la práctica
Fuente: Autor



Figura 4.24 . Inicio de la práctica
Fuente: Autor

Para iniciar la práctica en MRUV presionamos en el táctil para ingresar a la pantalla de configuración de parámetros de este movimiento, ver en la Figura 4.25



Figura 4.25 Ingreso a configuración MRUV
Fuente: Autor

En la siguiente pantalla podremos configurar la altura tanto en la torre A como la torre B para lograr con ello la inclinación deseada en la rampa para el módulo autónomo para el estudio de la cinemática, ver en la Figura 4.26.



Figura 4.26 Pantalla de configuración de MRUV

Fuente: Autor

En esta pantalla podemos observar las opciones de inclinación para facilitar la captura de datos de los sensores para ello se ha dispuesto de la siguiente manera.

- **A--B1** pulsando este botón la rampa se inclina a un ángulo de 30° respecto a la torre A, el punto A se mantiene y en la torre B se desplaza hacia abajo generando este ángulo.
- **A--B2** cuando se pulsa este botón la rampa del lado B baja hasta generar un ángulo de 60° respecto a la torre A.
- **A1--B** este botón invierte la posición de A--B1, con lo cual se genera un ángulo de 30° respecto a la torre B.
- **A2--B** la función de este botón es de invertir la posición de A--B2 provocando un ángulo de 60° respecto a la torre B.

Luego de parametrizar estos valores se prepara el módulo autónomo para la ejecución del laboratorio. En ambos casos como en el MRU y MRUV las pantallas de ejecución son las mismas.

4.2 Ensamblaje, programación y comprobación de módulo autónomo.

Para el ensamblaje del módulo autónomo que simula la partícula se ha dispuesto el material el acrílico de 3 mm para la estructura, se muestra en la Figura 4.27. Además, en se realiza el proceso químico a la baquelita con el diseño de la placa base.

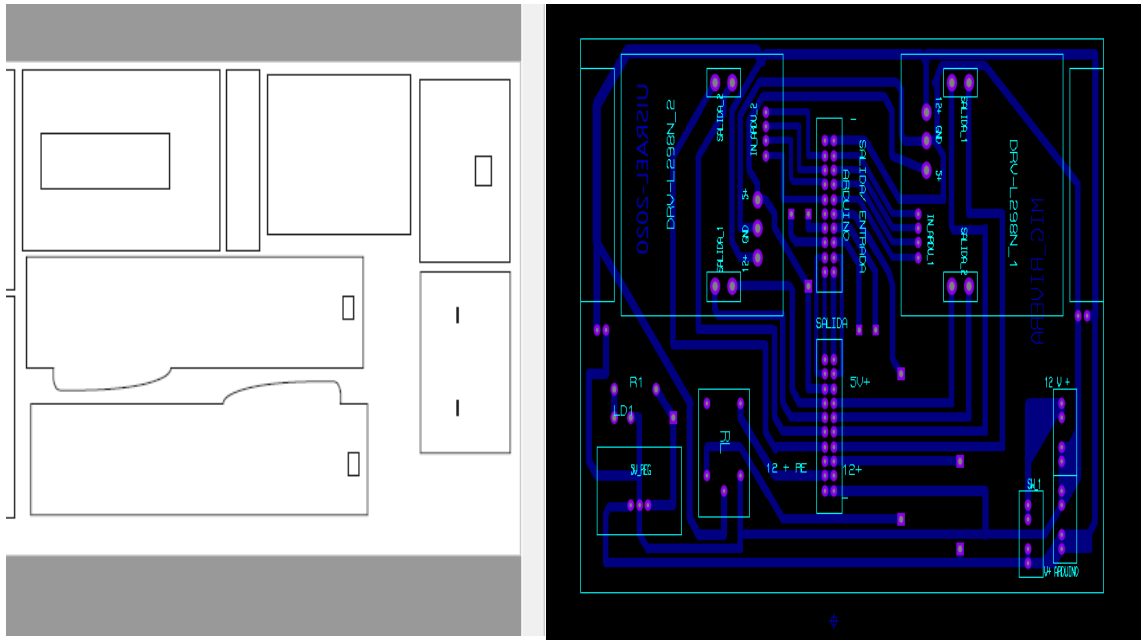


Figura 4.27 Diseño de la estructura del módulo autónomo y placa base PCB.
Fuente: Autor

Con los cortes realizados se arma las partes para construir la estructura se puede observar en la Figura 4.28.



Figura 4.28 Estructura externa del módulo autónomo
Fuente: Autor

Una vez que la estructura es sólida se procede al armado de los componentes electrónicos y se detallan a continuación:

a) Colocamos la placa base donde van conectados todos nuestros dispositivos electrónicos, como se puede observar la distribución de cada socket es para cada sensor., ver la Figura 4.29.

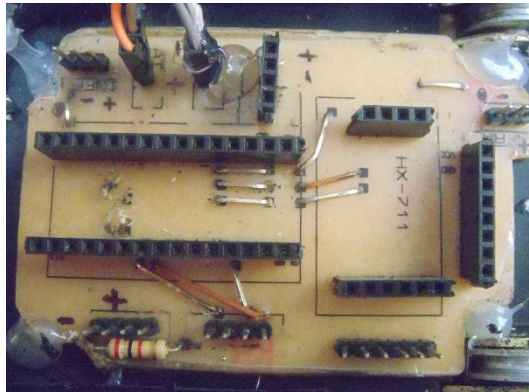


Figura 4.29: Placa base módulo autónomo

Fuente: Autor

b) Montamos en la ranura la pantalla LCD I2C 16x2 tal como se muestra en la Figura 4.30.

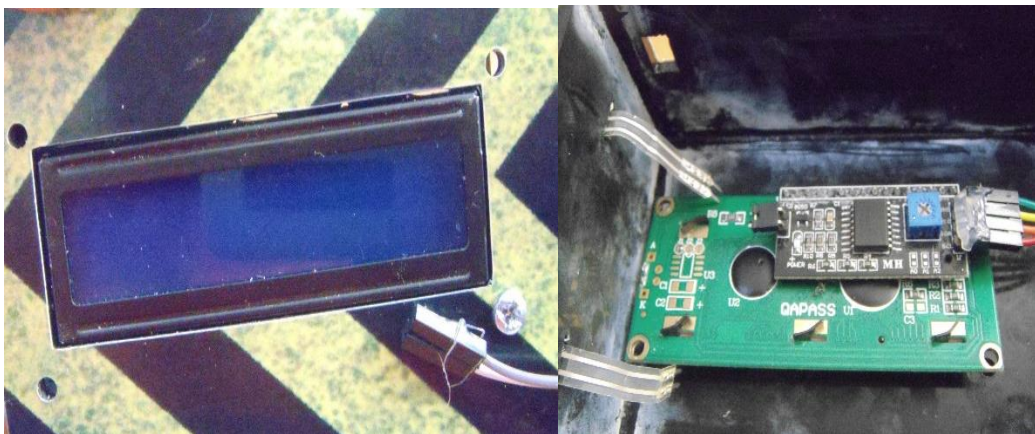


Figura 4.30: Pantalla LCD 16x2

Fuente: Autor

c) Instalamos sensor de carga en la base y el transductor en el sócalo de la placa base, se observa en la Figura 4.31.

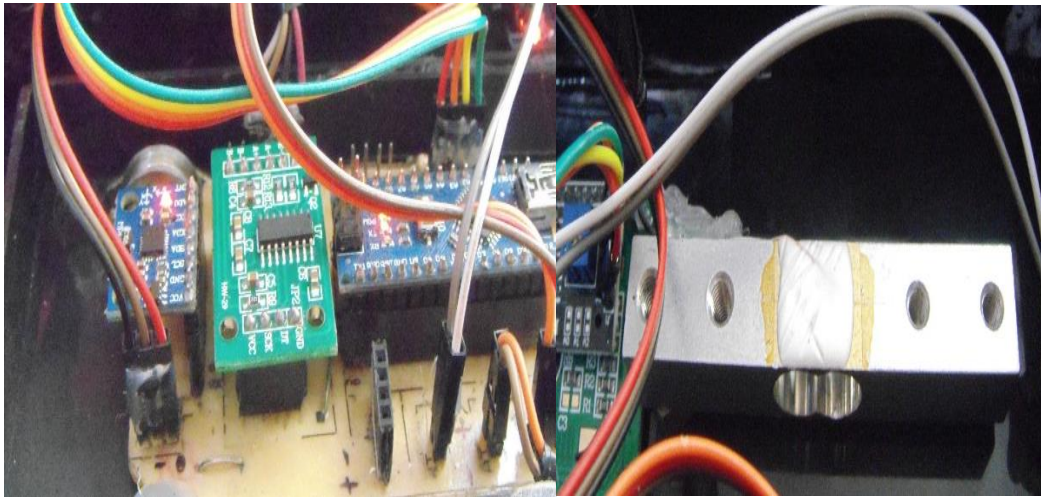


Figura 4.31 Celda de carga y transductor
Fuente: Autor

d) Colocamos los sensores Tcrt 5000 en las ranuras previamente establecidas en el chasis, ver Figura 4.32.



Figura 4.32: Instalación de Sensor TCRT5000

Fuente: Autor

e) Se coloca en la ranura el sensor MPU6050 fijándolo en coordenadas x y, ver la Figura 4.33

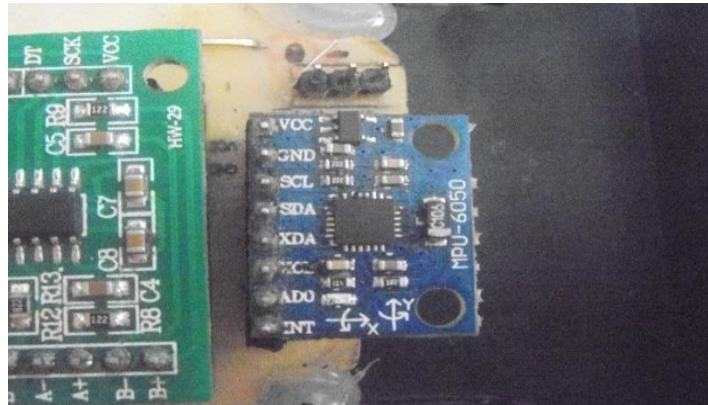


Figura 4.33 Sensor MPU6050

Fuente: Autor

f) Instalamos en la parte frontal los sensores resistivos FSR402 para la captura de datos por el impacto, ver Figura 4.34



Figura 4.34: Sensor Fsr402

Fuente: Autor

g) Procedemos a la colocación del microcontrolador Arduino nano y el Hc-06 tx y rx bluetooth, ver Figura 4.35.

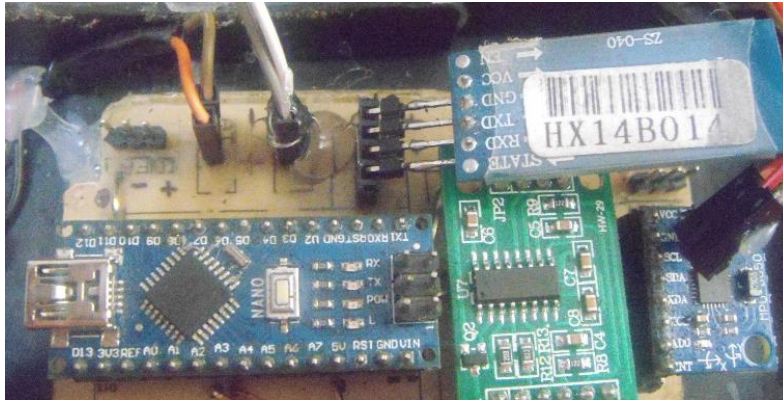


Figura 4.35 Colocación de Arduino Nano y Hc-06 Bluetooth

Fuente: Autor

h) Para finalizar la instalación de los componentes se procede a conectar todos los dispositivos a la placa base, estos serán alimentados por una batería de 9V DC, ver Figura 3.6.

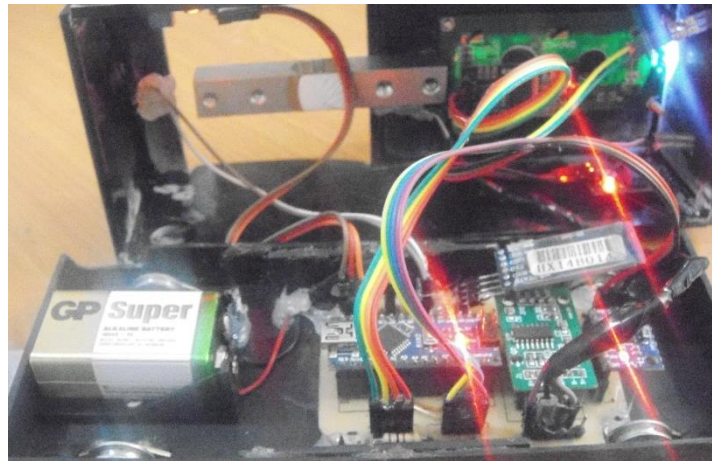


Figura 4.36: Funcionamiento de todos los componentes módulo autónomo

Fuente: Autor

- i) Se muestra el módulo ensamblado con todos los dispositivos internos en la Figura 4.37

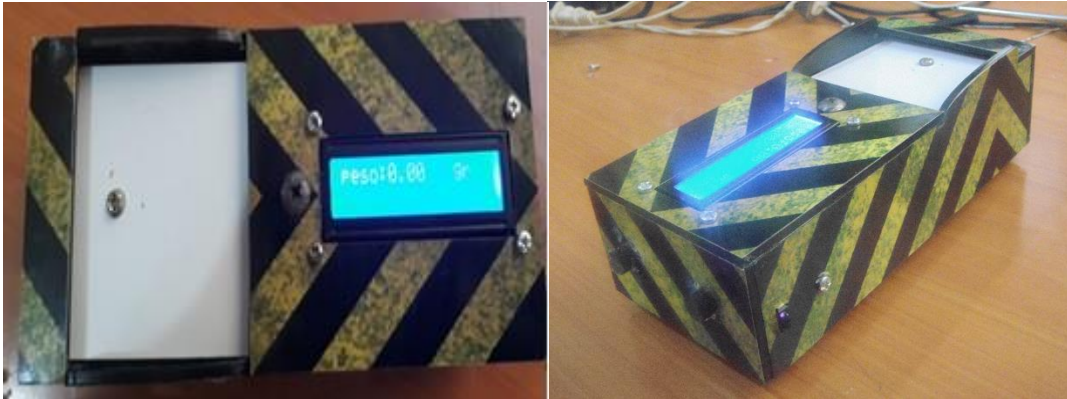


Figura 4.37 Módulo autónomo
Fuente: Autor

4.2.1 Programación de módulo autónomo

En esta sección se integra todos los sensores del módulo autónomo con su respectiva programación, pero en un solo código los cuales nos muestra los resultados en pantalla de las variables comprendidas en la cinemática.

4.2.2 Calibración de modulo MPU6050

Este sensor es primordial para la lectura de datos en nuestro módulo autónomo por lo tanto es necesario realizar una calibración previa antes de cargar la programación de la obtención de los datos ya que este dispositivo es muy sensible a los movimientos que al ser ensamblado y colocado en nuestra placa base se descalibra, para esto se procede a cargar el código para recalibrarlo en una posición horizontal sin realizar ningún movimiento, se puede observar en la Figura 4.38.

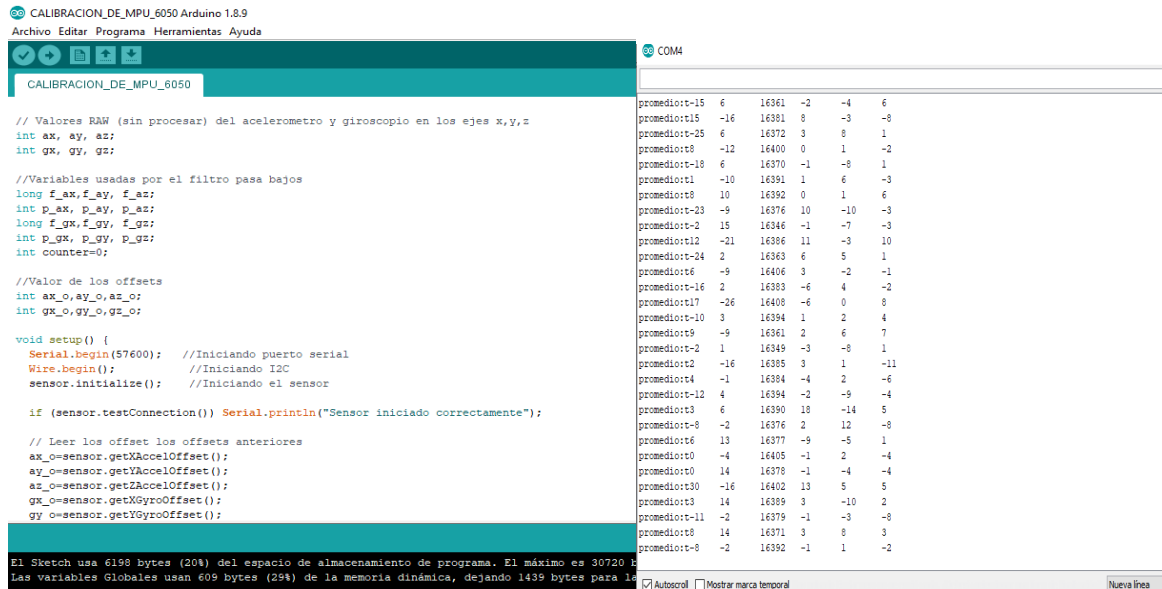


Figura 4.38 Calibración de MPU6050

Fuente: Autor

Cuando los valores de calibración se acerquen al 0 y el tercer valor sea aproximado al 16384 que es el valor de en z de la aceleración procedemos a desconectarlo para luego cargar el código de lectura de nuestro módulo autónomo.

4.2.3 Programación de módulo autónomo.

Una vez finalizado el ensamblaje y la calibración del sensor MPU6050 se procede a cargar la programación Arduino para controlar todos los sensores y componentes de módulo autónomo para el estudio de la cinemática. La programación completa estará en anexo 4 de esta investigación, ver Figura 4.39

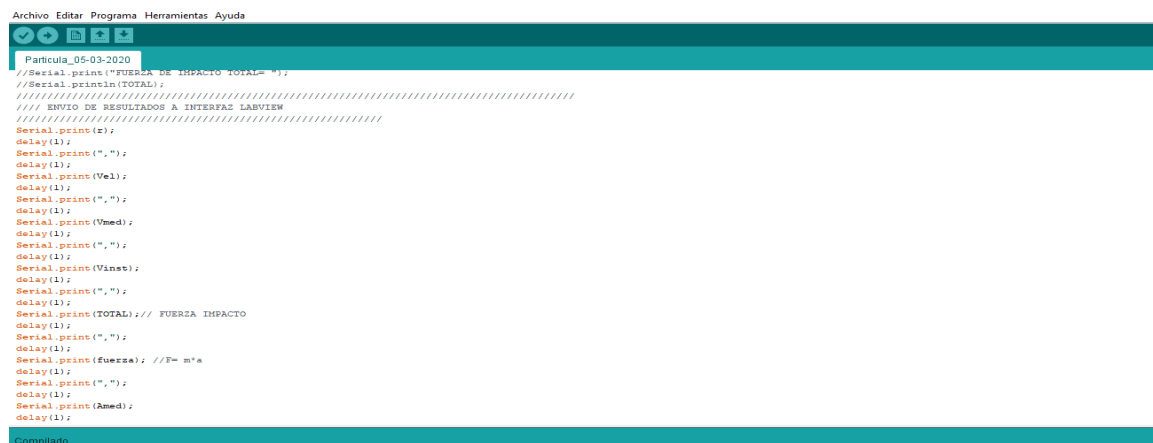


Figura 4.39 Programación de módulo autónomo.

Fuente: Autor

4.2.4 Aplicativo LabVIEW para la lectura de datos.

Una vez realizada la comprobación de funcionamiento de los dispositivos y de la interfaz gráfica procedemos a la conexión entre el módulo autónomo y el aplicativo en el PC en plataforma Windows 10, para lo cual se enciende la el módulo y abrimos el aplicativo, establecemos la comunicación bluetooth seleccionando el puerto serie comunicación y observamos datos sin procesar. en pantalla, ver Figura 4.40.

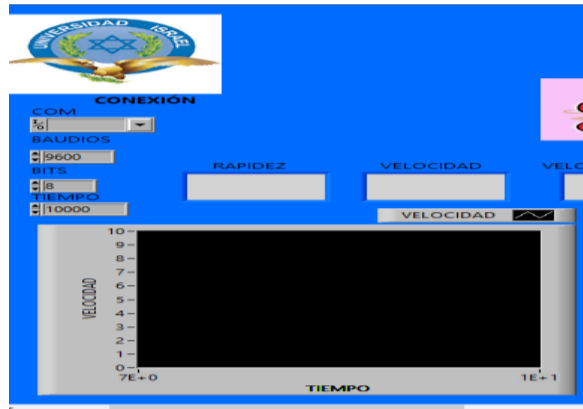


Figura 4.40 Configuración de conexión serie
Fuente: Autor

Establecida la comunicación se procede a la verificación de recepción de datos, como se muestra en la Figura 4.41.

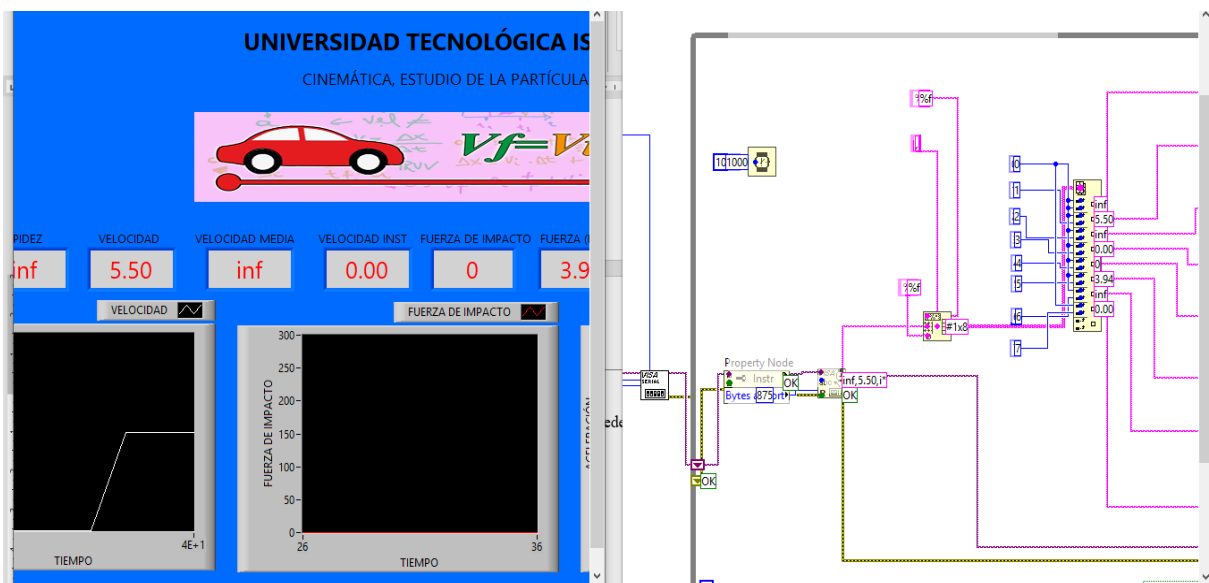


Figura 4.41: Conexión y datos obtenidos
Fuente: Autor

Se visualiza en la figura siguiente la captura de los datos emitidos por el módulo autónomo. Cabe mencionar que el envío y recepción de datos se realiza en formato String que se procesan y se convierten en numéricos tipo float., se puede observar en la Figura 4.42.

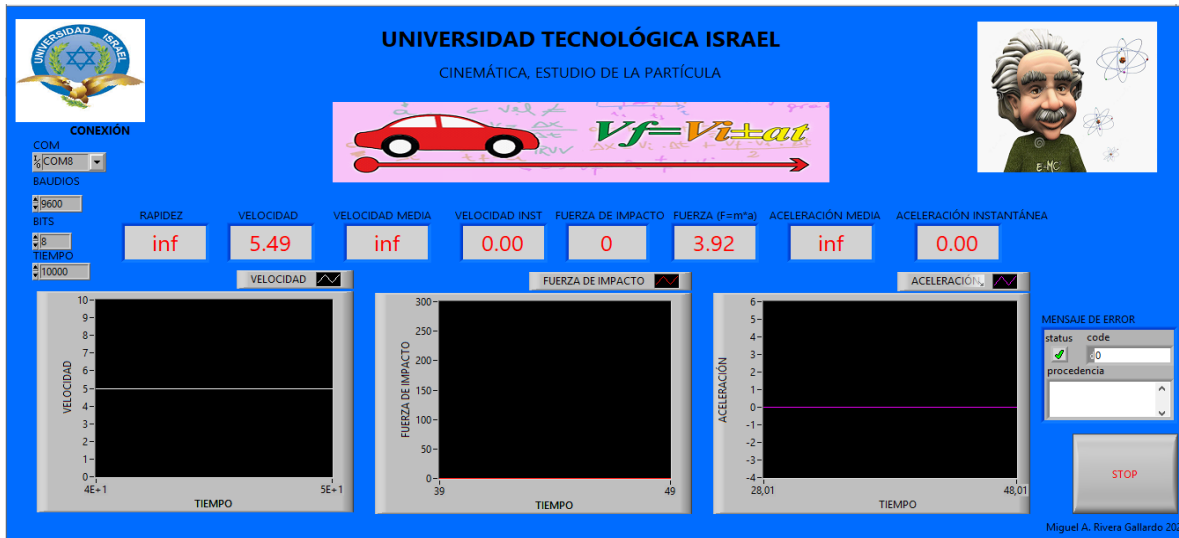


Figura 4.42: Pantalla del aplicativo
Fuente: Autor

Para un mejor funcionamiento de la interfaz gráfica se ha construido un ejecutable para abrirlo y muestra en pantalla solo la conexión y los datos recibidos del módulo autónomo, ver Figura 4.43.

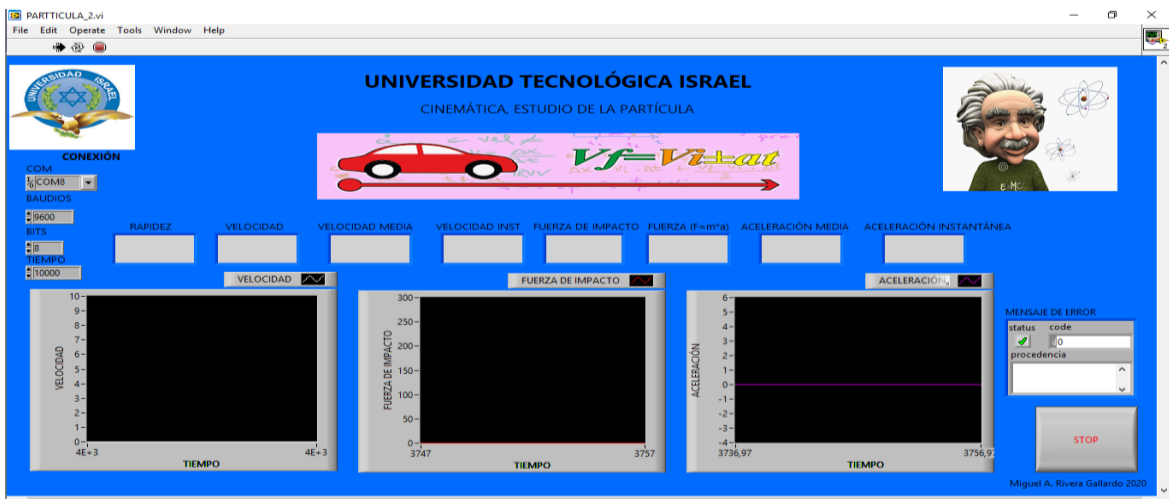


Figura 4.43: Aplicativo LabVIEW de captura de datos
Fuente: Autor

4.2.5 Practica MRU

Para realizar la simulación de este movimiento utilizaremos la siguiente fórmula.

- $$v = \frac{\Delta x}{\Delta t}$$

Donde V es la velocidad, Δx el desplazamiento y Δt el tiempo transcurrido.

Dadas nuestras variables procedemos a la práctica de forma manual y teórica, la rampa tiene una medida establecida de 1.80m, para la toma del tiempo lo realizamos de forma cronometrada, ver Figura 4.44 y 4.45.

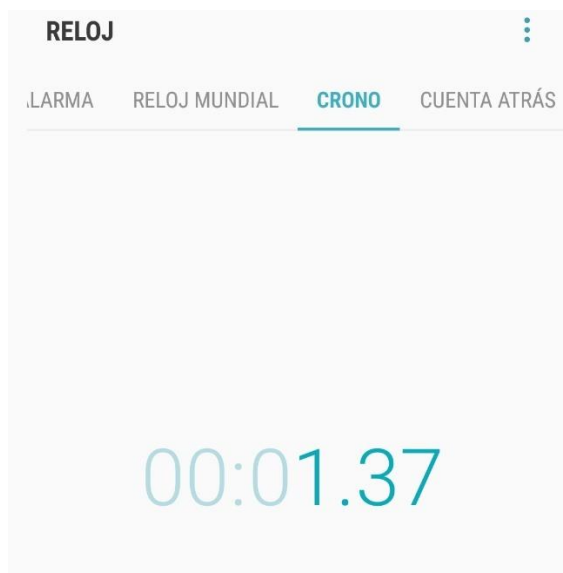


Figura 4.44: Toma de tiempo del movimiento
Fuente: Autor



Figura 4.45: Movimiento del Módulo autónomo

Fuente: Autor

Dados los valores de los parámetros recogido se procede al cálculo teórico.

- $x = 1.80\text{m}$
- $t = 1.24\text{s}$
- $v = ?$

$$V = \frac{x}{t} ; V = \frac{1.80}{1.24} = 1.45 \frac{m}{s}$$

Donde, x es el desplazamiento,

t = tiempo,

v = Velocidad

Se determina de manera teórica que la velocidad de nuestro módulo autónomo es de **1.45m/s**.

El cálculo de manera práctica se realiza de la siguiente forma:

- Preparamos la rampa con el módulo de control torres en línea horizontal, lo conseguimos presionando la pantalla táctil en el apartado MRU, ver Figura 4.46.



Figura 4.46: Botón MRU
Fuente: Autor

- Colocamos nuestro módulo autónomo al inicio de la rampa y anclándolo en el disparador, ver en la Figura 4.47.



Figura 4.47: Colocación de módulo autónomo en rampa.
Fuente: Autor

- Encendemos el módulo autónomo, ver Figura 4.48.



Figura 4.48: Encendido de módulo autónomo.
Fuente: Autor

- Preparamos el aplicativo LabVIEW de captura de datos, estableciendo la configuración de conexión serial, ver Figura 4.49.

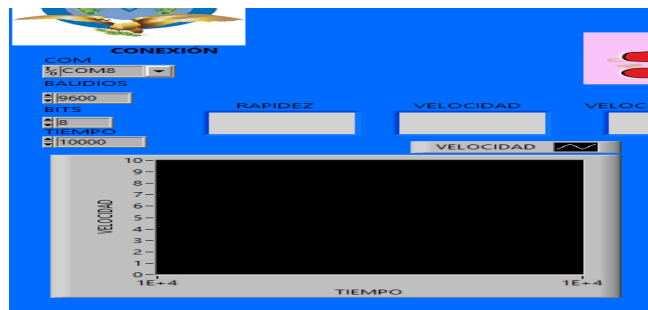


Figura 4.49: Conexión serial.
Fuente: Autor

- Procedemos en el módulo de control torres a presionar el botón inicio, ver Figura 4.50.



Figura 4.50: Botón inicio de práctica
Fuente: Autor

- Se muestra los resultados obtenidos en la Figura 4.51.



Figura 4.51: Resultados de la práctica.

Fuente: Autor

Tabla comparativa entre los valores teóricos y práctico se muestran en la Tabla 4.1

Tabla 4.1

Tabla comparativa de resultados.

VARIABLES	TEÓRICO	PRÁCTICO	%ERROR
VELOCIDAD (v)	1.45 m/s	1.51m/s	3,97
TIEMPO (t)	1.24 s	1.19 s	5,00
DISTANCIA (x)	1.80 m	1.80m	0.00

Fuente: Autor

Se concluye que comparando los resultados teóricos con los prácticos existe una diferencia de 3.97% décimas en la velocidad y 5% en el tiempo con lo cual los resultados afirman que es correcto las mediciones hechas por nuestro módulo autónomo.

4.2.6 Practica MRUV

Para realización de esta práctica utilizaremos las siguientes fórmulas.

- $V_f = V_o \pm at$
- $V_f^2 = V_o^2 + 2ax$
- $x = V_o t + \frac{at^2}{2}$

Donde V_f es la velocidad, x el desplazamiento, t el tiempo transcurrido a es la aceleración.

Para esta práctica se procede inclinar las torres a un ángulo de 60° respecto a la base de la torre A y dados los valores de los parámetros recogidos se procede al cálculo teórico. La toma de tiempo con el cronómetro se puede observar en la Figura 4.52.



Figura 4.52: Tiempo de la práctica
Fuente: Autor

- $x = 1.80\text{m}$
- $t = 0.99\text{s}$
- $v_f = ?$
- $a = ?$

Lo primero determinaremos la aceleración de nuestro módulo autónomo para lo cual utilizaremos la siguiente fórmula.

$$a = \frac{2x}{t^2} ; a = \frac{1.80 * 2}{0.99^2} = 3.67 \frac{\text{m}}{\text{s}^2}$$

Donde, v_f velocidad final,

v_o = Velocidad inicial,

a = aceleración.

t = tiempo

Para encontrar la velocidad final (V_f) utilizaremos la siguiente formula.

$$V_f = V_o + at ; V_f = 0 + 3.67 \frac{m}{s^2} * 0.99 s = 3.63 \frac{m}{s}$$

El cálculo de manera práctica se realiza de la siguiente forma:

- Preparamos la rampa con el módulo de control torres en línea horizontal, lo conseguimos presionando la pantalla táctil en el apartado MRUV, ver en la Figura 4.53.



Figura 4.53: Botón MRU
Fuente: Autor

Preparamos la rampa con el módulo de control torres con la inclinación que deseemos en este caso le colocamos a 60° , se muestran las opciones en la Figura 4.54.

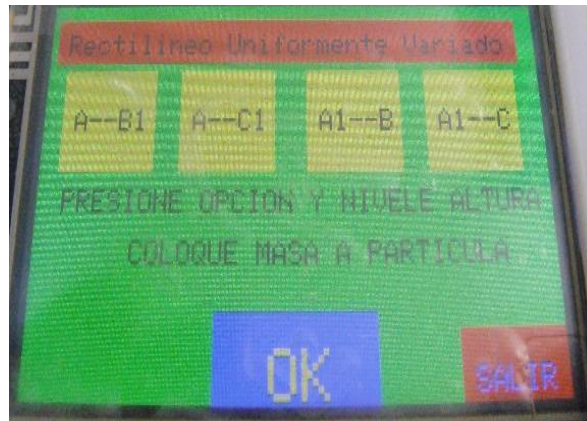


Figura 4.54: Control de inclinación.
Fuente: Autor

- Colocamos nuestro módulo autónomo al inicio de la rampa y anclándolo en el disparador, ver en la Figura 4.55.



Figura 4.55: Colocación de módulo autónomo en rampa.
Fuente: Autor

Encendemos el módulo autónomo, ver en la Figura 4.56.



Figura 4.56: Encendido de módulo autónomo.
Fuente: Autor

- Preparamos el aplicativo LabVIEW de captura de datos, estableciendo la configuración de conexión serial, ver en la Figura 4.57.

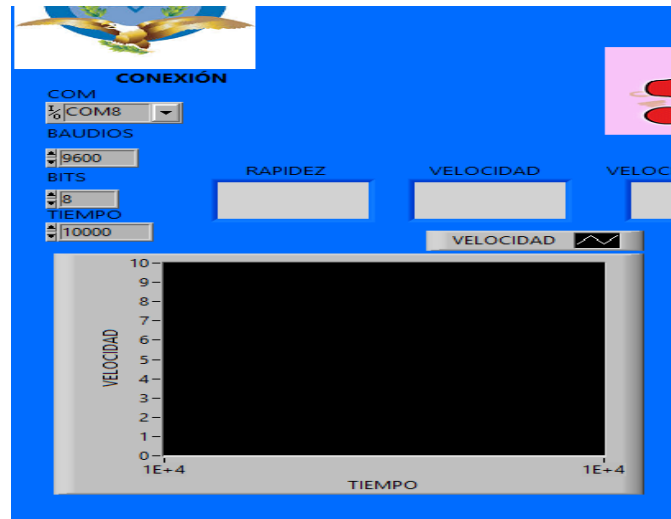


Figura 4.57: Conexión serial.
Fuente: Autor

- Procedemos en el módulo de control torres a presionar el botón inicio, ver en la Figura 4.58.

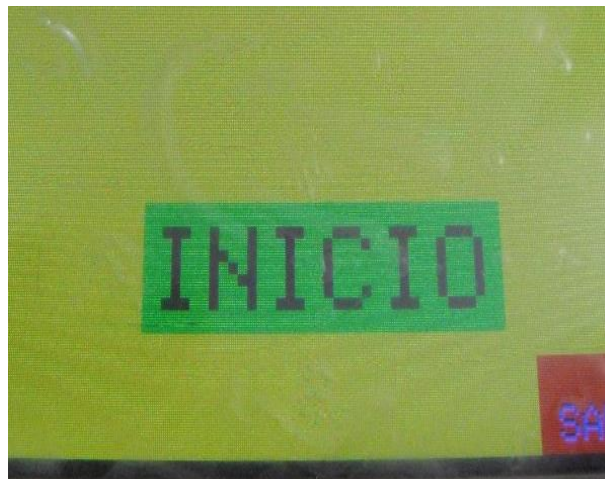


Figura 4.58: Botón inicio de práctica
Fuente: Autor

Se muestra los resultados obtenidos en la Figura 4.59



Figura 4.59: Resultados de la práctica.

Fuente: Autor

Tabla comparativa entre los valores teóricos y práctico se puede observar en la tabla 4.2

Tabla 4.2

Tabla comparativa de resultados.

VARIABLES	TEÓRICO	PRÁCTICO	%ERROR
VELOCIDAD (V)	3.63 m/s	3.71m/s	2.15
TIEMPO (t)	0.99 s	0.93 s	6.45
DISTANCIA (x)	1.80 m	1.80m	0.00
ACELERACION (a)	3.67m/s ²	3.82m/s ²	3.92

Fuente: Autor

Se concluye que comparando los resultados teóricos con los prácticos existe una diferencia de 2.15% décimas en la velocidad y 6.45% en el tiempo y en la aceleración debido a nuestro sensor MPU6050 es de 3.92%, con lo cual los resultados afirman que es correcto las mediciones hechas por nuestro módulo autónomo.

Tabla 4.3 :
Costo de implementación

COMPONENTES ELECTRÓNICOS				
Item	Descripción	Cantidad	VALOR U.	VALOR TOTAL
1	Arduino Mega 2560r3	1	19,00	19,00
2	Arduino Nano	1	8,00	8,00
3	Módulo HL298N	2	8,00	16,00
5	Fuente de alimentación 12v 30A	1	33,00	33,00
6	Regulador 7805	1	1,00	1,00
7	Pantalla TFT OpenSmart	1	45,00	45,00
8	Relé 5v	1	0,50	0,50
9	Motor bipolar	2	11,50	23,00
10	Sensor HC-sr04	2	2,50	5,00
11	Sensor Hx711 y celda de carga	1	10,00	10,00
12	Sensor MPU6050	1	4,00	4,00
13	Pantalla LCD 16x 2	1	6,00	6,00
14	Sensor infrarrojo	2	2,50	5,00
15	Sensor FSR402	2	12,00	24,00
16	Hc-06	1	7,00	7,00
17	Cajas de proyectos	2	15,00	30,00
18	Placas	2	4,00	8,00
19	Conectores y cables	1	10,00	40,00
20	Varios	1	20,00	20,00
			TOTAL_1	274.50
MATERIALES PARA ESTRUCTURA Y RAMPA				
1	Barra de aluminio	2	25,00	65,00
2	Perfil de aluminio	1	28,00	28,00
3	Cortes Varios	1	25,00	25,00
1	Hora/ Hombre	100	4,00	400,00
			TOTAL_2	503,00
			<i>Total_1+Total_2</i>	777.50

Fuente: Autor

CONCLUSIONES

La presente investigación está basada en el desarrollo y construcción de un módulo autónomo para el aprendizaje de los principios básicos de la cinemática de una partícula, para el estudio de las leyes de movimiento, donde se han cumplido todos los objetivos propuestos por cuanto se ha logrado desarrollar un módulo autónomo para el aprendizaje de los principios básicos de la cinemática de una partícula en el ámbito universitario del Ecuador.

En ese sentido, se han considerado diferentes recursos tecnológicos tanto de hardware como de software, en ese sentido, se pudo concluir que los componentes electrónicos que se utilizaron como fueron sensores infrarrojos, TRCT 5000, ultrasónicos HCsr04, y como principal el MPU6050 determinando que son los más idóneos, por considerar que su costo en el mercado son bajos, por otra parte, con respecto a la parte mecánica se establecieron que los componentes idóneos son de aluminio debido a ligereza y a la condición de inoxidable, considerando el software el de mejor rendimiento fue la plataforma de programación de LabVIEW, ya que permite realizar pruebas de funcionamiento antes del desarrollo del proyecto, además con las herramientas se implementó una interfaz gráfica fácilmente.

Así mismo, se logró diseñar el sistema de carriles, accesorios de sujeción, carros de baja fricción, sensor de movimiento, contador de tiempo, para el estudio del Movimiento Rectilíneo Uniforme (MRU) y Movimiento Rectilíneo Uniforme Acelerado (MRUA), utilizando componentes asequibles en el mercado local, quedando un diseño óptimo y atemperado con los requerimientos de la institución.

Cabe destacar, que, con las herramientas y elementos establecidos, y tomando en cuenta las prácticas a desarrollar, se pudo concluir que el producto desarrollado cumple con todas las características para el estudio del MRU y MRUA, por cuanto se compara los cálculos teóricos con la experimentación arrojando datos con una variación entre el 2% y el 6%. Finalmente, con respecto a las pruebas de funcionamiento del módulo autónomo, se llega a la conclusión que el mismo cumple con los requerimientos de la institución, ya que los

experimentos y medidas realizadas se comprobaron con elementos de medición disponibles en la institución, lo cual garantiza el funcionamiento del mismo.

RECOMENDACIONES

Una vez concluido este trabajo de grado, se considera importante investigar sobre otros aspectos relacionados con el estudio y la utilización de otro tipo de sensores con la finalidad de tener mayor precisión en las lecturas, también se recomienda desarrollar otro tipo de módulos para estudiar otras leyes de la física clásica, por tanto, se recomienda:

La utilización de sensores con otros protocolos de comunicación para el mejor funcionamiento en la conectividad entre el módulo y la interfaz gráfica.

Crear interfaz gráfica de comunicación con aplicativos Open Source en las plataformas Raspberry con sistemas operativos Raspbian para lograr que este módulo sea portable y aplicable en cualquier aula de clases.

Basándonos en esta investigación, los recursos, los datos se pueden utilizar estos sensores y estructuras para experimentar en nuevas prácticas en caída libre, tiro parabólico y fuerzas

Referencias bibliográficas

amgkits. (ENERO de 2016). *<https://amgkits.com/>*. Obtenido de <https://amgkits.com/home/:https://amgkits.com/home/142-display-lcd-16x2-con-interfaz-i2c.html>

Arduino. (02 de 2020). *store Arduino* . Obtenido de <https://store.arduino.cc/usa/>

Bernal, C. (2010). *Metodología de la investigación* (Tercera ed.). Bogotá: Pearson Educación.

BlueBolt IINC. (2015). *Módulo Bluetooth HC-06 con puerto serial. Guía fácil*. Obtenido de <https://www.puntofloante.net/MANUAL-BLUETOOTH-HC-06-CON-PUERTO-SERIAL.pdf>

Carrión, L., Ochoa, D., & Valverde, J. (2009). *Análisis del funcionamiento del sensor de fuerza resistivo (FSR) con LabVIEW*. Cuenca: Universidad Politécnica Salesiana.

Collaguazo, S. (2012). *Implementación de un sistema de frenado automático progresivo para prevenir colisiones frontales y posteriores en un vehículo liviano dentro de la ciudad de Quito*. Quito: Universidad Tecnológica Equinoccial.

Equipo de laboratorio didáctico. (2018). *Curso de Arduino: Módulo sensor ultrasónico*. PROSERQUISA.

García, V. (2 de Marzo de 2018). *Configurar el MPU6050*. Obtenido de Configurar el MPU6050: <https://www.diarioelectronicohoy.com/blog/configurar-el-mpu6050>

Gutiérrez, L., Jácome, J., & Larrea, P. (2016). *Laboratorio de física general I*. Quito: Escuela Politécnica Nacional.

Hetpro. (25 de Abril de 2014). *MPU6050 Arduino, Acelerómetro y Giroscopio*. Obtenido de Sensores digitales: <https://hetpro-store.com/TUTORIALES/modulo-acelerometro-y-giroscopio-mpu6050-i2c-twi/>

INFmecafenix. (02 de 2020). *INGmecafenix*. Obtenido de <https://www.ingmecafenix.com/electricidad-industrial/motor-paso-a-paso/>

INSTRUMENS, N. (02 de 2020). <https://www.ni.com/>. Obtenido de <https://www.ni.com/es-cr/shop/labview.html>

Lambda Scientific Systems, Inc. (2019). *Lab equipment for teaching physics*. Obtenido de LEMI-45 Complete Air Track Apparatus: <https://lambdasys.com/products/detail/15>

Lozano, R. (23 de Enero de 2019). *Celda de carga HX711*. Obtenido de Celda de carga HX711: <https://www.taloselectronics.com/blogs/tutoriales/celda-de-carga-hx711>

Marker Zona. (2014). *Manual de Arduino: Programación y conceptos básicos*. Arduino.

Marqués, J. (2017). *Prototipo instrumental de medición de presión de torniquetes*. Valencia: Escuela Técnica Superior de Ingeniería del Diseño, Universitat Politècnica de València.

Mecatrónica LATAM. (2015). *ARDUINO: Sensor TCRT5000*. Obtenido de <https://www.mecatronicalatam.com/es/tarjeta/arduino/sensor-ir-tcrt5000>

Nares, C. (16 de Febrero de 2015). *Sensor de Fuerza o Presión MF01*. Obtenido de <https://hetpro-store.com/TUTORIALES/sensor-de-fuerza-o-presion-mf01/>

Naylamp Mechatronics. (4 de Septiembre de 2019). *Tutorial de Arduino y sensor ultrasónico HC-SR04*. Obtenido de https://naylampmechatronics.com/blog/10_Tutorial-de-Arduino-y-sensor-ultras%C3%B3nico-HC-S.html

Naylamp Mechatronics SAC. (2019). *Tutorial transmisor de celda de carga HX711*. Obtenido de Tutorial transmisor de celda de carga HX711: https://naylampmechatronics.com/blog/25_Tutorial-trasmisor-de-celda-de-carga-HX711-Ba.html

Nubbeo. (2019). *Sensor Optico Reflectivo Infrarrojo Tcrt5000 Arduino Nubbeo*. Obtenido de <https://www.nubbeo.com.ar/sensor-optico-reflectivo-infrarrojo-tcrt5000-arduino-nubbeo-549563122xJM>

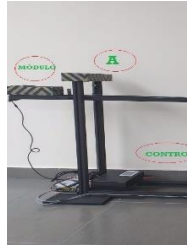
-
- ONROM. (02 de 2020). *ONROM FUENTES DE ALIMENTACION*. Obtenido de https://assets.omron.eu/downloads/datasheet/es/v5/t062_s8fs-c_switching_power_supply_datasheet_es.pdf
- PROMETEC. (16 de Abril de 2017). *EL MÓDULO CONTROLADOR DE MOTORES L298N*. Obtenido de <https://www.prometec.net/l298n/>
- Prometec. (17 de Febrero de 2018). *USANDO EL MPU6050*. Obtenido de USANDO EL MPU6050: <https://www.prometec.net/usando-el-mpu6050/>
- Raspberry pi 3 b+*. (25 de 02 de 2020). Obtenido de <https://static.raspberrypi.org/files/product-briefs/200206+Raspberry+Pi+3+Model+B+plus+Product+Brief+PRINT&DIGITAL.pdf>
- Resnick, R., Halliday, D., & Krane, K. (2001). *Física* (Cuarta ed., Vol. I). John Wiley & Sons Inc.
- Ruiz, J. (2007). *Manual de programación Arduino*. San Francisco: Attribution-Noncommercial-Share.
- Sears, F., Zemansky, M., & Young, H. (2010). *Física universitaria* (Undécima ed., Vol. I). Texas: Pearson.
- Serway, R., & Jewett, J. (2014). *Física para ciencias e ingeniería* (Novena ed., Vol. I). México DF: Brooks / Cole.
- Styger, E. (19 de Junio de 2013). *Using the HC06 Bluetooth Module*. Obtenido de <https://cdn.instructables.com/ORIG/FX1/N43N/IMXQP4WO/FX1N43NIMXQP4WO.pdf>
- Torrente, O. (2013). *Curso práctico de formación*. Madrid: RC Libros.
- Vernier, L. (02 de 2020). *Lab. Vernier*. Obtenido de Dynamics Cart and Track System with Go Direct® Sensor Cart: <https://www.cienytec.com/edu2-vernier-fisica-para-universidad.htm>

ANEXOS

Anexo 1 Manual Usuario.



1. Establezca la ubicación considerando base plana, fija y a una altura aproximada de 1m.
2. Coloque las torres a una distancia aproximada de 2 m de distancia entre ellas y coloque la rampa en las ranuras. Siempre considerando la torre A y B.
3. Coloque el módulo control torres y la fuente alimentación en la base de torre A



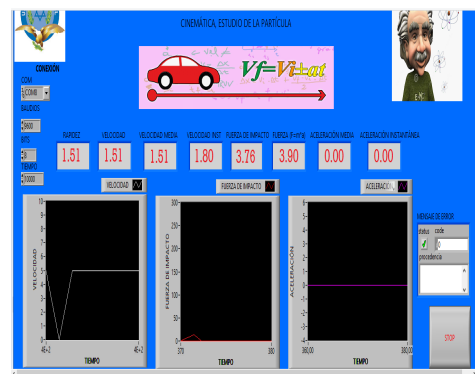
4. Inserte los cables de alimentación y control en las clavijas del módulo control torres.
5. Coloque el módulo autónomo sobre la rampa y ancle al disparador.
6. Revise que el aplicativo LabVIEW se encuentre abierto.
7. La conexión entre el módulo autónomo y el aplicativo se establece cuando encendemos el módulo, verificar que este

en aplicativo el puerto serie detectado en el dispositivo Bluetooth.

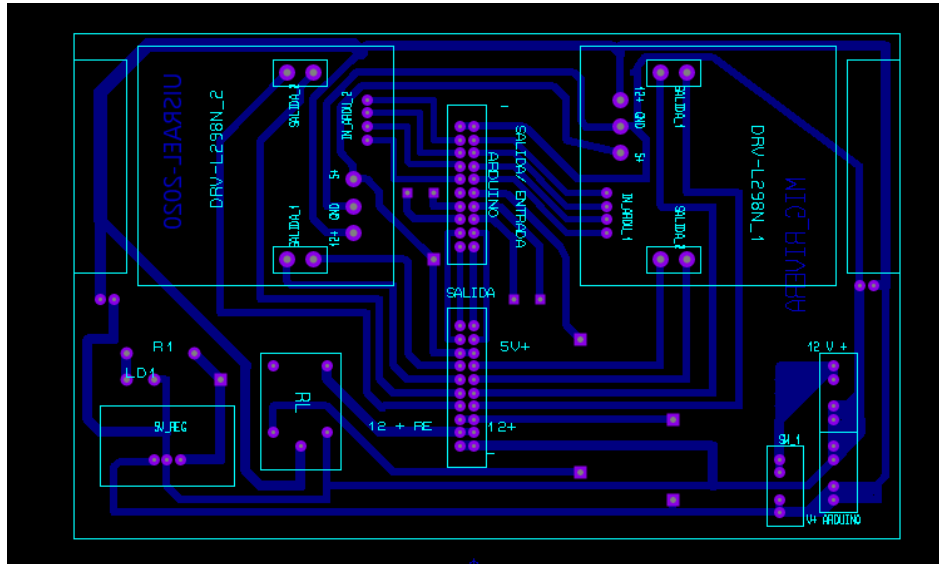
8. Al iniciar la práctica verifique que todo este adecuado con los parámetros de experimentación
9. Para iniciar el experimento diríjase al módulo control torres e inicie la práctica



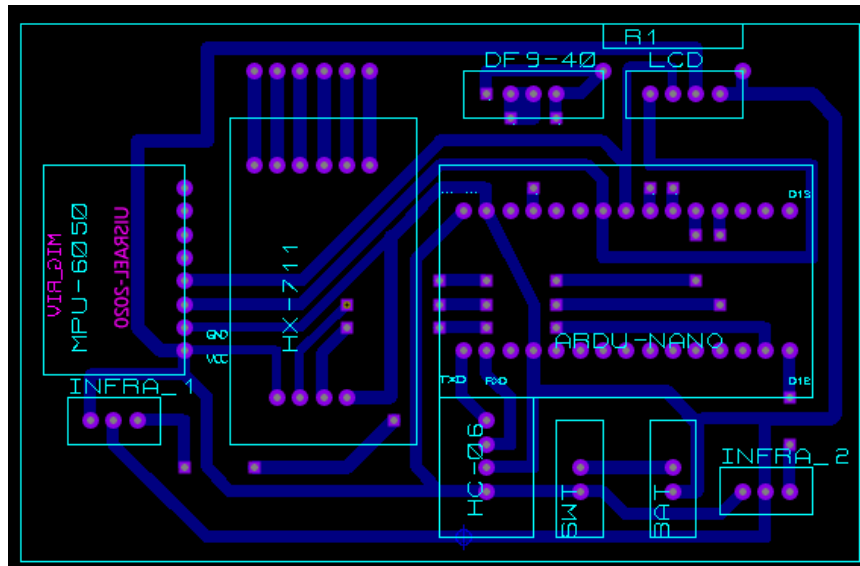
10. En el aplicativo se mostrará datos erróneos hasta que se inicie la práctica.
11. Al finalizar la práctica observaremos los resultados obtenidos y actualizados cada 1s.



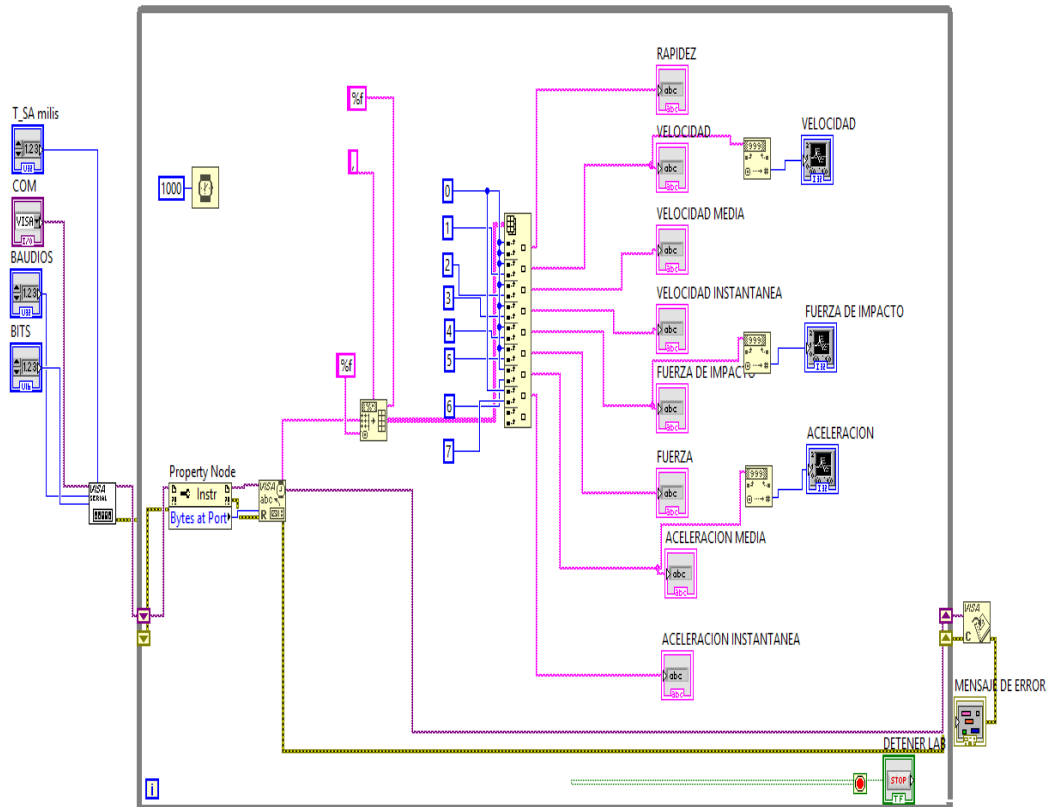
Anexo 2 Diseño PCB módulo control torres.



Anexo 3 Diseño PCB módulo autónomo



Anexo 4 Diseño LabVIEW



Anexo 4 Programación módulo autónomo

```

// Librerias I2C para controlar el mpu6050
// la libreria MPU6050.h necesita I2Cdev.h, I2Cdev.h necesita Wire.h
#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"
#include <HX711_ADC.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
HX711_ADC celda_C(6, 7);
LiquidCrystal_I2C lcd(0x27,16,2);
/////////////////////////////////////////////////////////////////
//sensor mpu6050
/////////////////////////////////////////////////////////////////
// La dirección del MPU6050 puede ser 0x68 o 0x69, dependiendo
// del estado de AD0. Si no se especifica, 0x68 estará implícito
MPU6050 sensor;
// Valores RAW (sin procesar) del acelerometro y giroscopio en los ejes x,y,z
int ax, ay, az;
int gx, gy, gz;
double Aini, Afin, Atranscurrido;
float Ang_x;
int Angulo;
/////////////////////////////////////////////////////////////////
double r;//rapidez
double Vi=0; //Velocidad Inicial
double Ax = 1.60;// Desplazamiento
double Vel;//velocidad// velocidad final
double Vmed; //Velocidad med
double Vinst;//Velocidad Instantanea
double Amed;//aceleracion media
double Ainst; //aceleracion instantanea
double Acel;//aceleracion
double Peso_Total;
double tiempo;
double fuerza;
double M_auto;
float Tara_m = 557;// tara de módulo autónomo en gr
/////////////////////////////////////////////////////////////////
//CRONOMETRO
/////////////////////////////////////////////////////////////////
double inicio, finalizado, Ttranscurrido;
unsigned long sensor_1 = 11;
unsigned long sensor_2 = 12;
unsigned long terminado;
double Tseg;
/////////////////////////////////////////////////////////////////
//Peso
/////////////////////////////////////////////////////////////////
int sen1;
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
//FUERZA
/////////////////////////////////////////////////////////////////
int P_SENSOR_1 = A2; // Sensor conectado a Analog 2
int P_SENSOR_2 = A3; // Sensor conectado a Analog 2
int L_SENSOR_1; // La Lectura de la Resistencia por División de Tensión

```



```

int L_SENSOR_2;
int TOTAL;

void setup() {
  ///////////////////////////////////////////////////////////////////
  //inicio
  ///////////////////////////////////////////////////////////////////
  Serial.begin(9600); //Iniciando puerto serial
  ///////////////////////////////////////////////////////////////////
  //mpu 6050
  ///////////////////////////////////////////////////////////////////
  Wire.begin(); //Iniciando I2C
  sensor.initialize(); //Iniciando el sensor
  ///////////////////////////////////////////////////////////////////
  //cronometro
  ///////////////////////////////////////////////////////////////////
  pinMode(sensor_1, INPUT); // Boton de inicio //////////////// cronometro
  pinMode(sensor_2, INPUT); // Boton para Parar

  ///////////////////////////////////////////////////////////////////
  //Peso
  ///////////////////////////////////////////////////////////////////
  lcd.print("Espere");// pantalla
  lcd.init();
  lcd.backlight();
  celda_C.begin();// celda de carga
  long stabilisingtime = 1; //
  celda_C.start(stabilisingtime);
  celda_C.setCalFactor(1550); // FACTOR DE CALIBRACION

}

void loop() {
  ///////////////////////////////////////////////////////////////////
  //calculo de angulo
  ///////////////////////////////////////////////////////////////////
  sensor.getAcceleration(&ax, &ay, &az);
  Ang_x=-(atan(ax/sqrt(pow(ay,2) + pow(az,2)))*(180.0/3.14)-2.65);
  Angulo = Ang_x;
  delay(10);
  ///////////////////////////////////////////////////////////////////
  //lcd
  ///////////////////////////////////////////////////////////////////
  celda_C.update();
  int sen1=celda_C.getData();
  lcd.setCursor(0,0);
  lcd.print("P:");
  lcd.print(((sen1*16.71)/10));
  M_auto = (((sen1*16.71)/10)+Tara_m);
  lcd.print("gr");
  lcd.setCursor(0,1);
  lcd.print("t:");
  lcd.print(Tseg);
  lcd.print("s");
  lcd.setCursor(10,0);
  lcd.print("F:");
  lcd.print(TOTAL);
  lcd.print("N");
}

```

```

lcd.setCursor(10,1);
lcd.print("<");
lcd.print(Angulo);
lcd.print("o");

////////////////////////////////////
//tiempo
////////////////////////////////////
if (digitalRead(sensor_1) == LOW)
{
  inicio = millis();

  Serial.println("contando...");
  delay(300);
  sensor.getAcceleration(&ax, &ay, &az);
  float ax_m_s2 = ax * (9.81/16384.0);
  float ay_m_s2 = ay * (9.81/16384.0);
  float az_m_s2 = az * (9.81/16384.0);

  Aini=sqrt(pow(ax_m_s2,2)+pow(ay_m_s2,2));/+pow(az_m_s2,2));
  L_SENSOR_1= analogRead(P_SENSOR_1);
}
if (digitalRead(sensor_2) == LOW)
{
  finalizado = millis();
  float ax_m_s2 = ax * (9.81/16384.0);
  float ay_m_s2 = ay * (9.81/16384.0);
  float az_m_s2 = ay * (9.81/16384.0);
  Afin =(sqrt(pow(ax_m_s2,2)+pow(ay_m_s2,2))-0.42);/+pow(az_m_s2,2));
  L_SENSOR_2= analogRead(P_SENSOR_2);
  Serial.println("fin...");
  delay(10);
}

Ttranscurrido = finalizado - inicio;
Tseg = Ttranscurrido/1000;
/*Serial.println(Tseg);
Serial.println(Afin);
delay(100);*/

////////////////////////////////////
// RAPIDEZ
////////////////////////////////////
r=Ax/Tseg;
////////////////////////////////////
//VELOCIDAD
////////////////////////////////////
Vel=Ax/Tseg;
////////////////////////////////////
// Velocidad instantanea
////////////////////////////////////
// se determina cada cm del desplazamiento del mÓdulo
////////////////////////////////////
//Ax=1cm= 0.01m

Vinst = Afin*Tseg;
////////////////////////////////////
// Velocidad media
////////////////////////////////////

```

```

Vmed = Ax/Tseg;
////////////////////////////////////
// Aceleracion media
////////////////////////////////////
Amed = Vel/Tseg;
////////////////////////////////////
// Aceleracion Instantanea
////////////////////////////////////
// Aceleracion instantanea es igual a al tiempo

Ainst = Tseg;
////////////////////////////////////
//VF
////////////////////////////////////

Vel=sqrt(2*Afin*1.6);
////////////////////////////////////
//FUERZA
////////////////////////////////////
fuerza = ((M_auto*Afin)/1000);
////////////////////////////////////

L_SENSOR_1 += analogRead(P_SENSOR_1); // La Resistencia es igual a la lectura del sensor (Analog 0)
L_SENSOR_2 += analogRead(P_SENSOR_2);
TOTAL =((L_SENSOR_1+L_SENSOR_2)/2);

////////////////////////////////////
//// ENVIO DE RESULTADOS A INTERFAZ LABVIEW
////////////////////////////////////
Serial.print(Tseg);/// TIEMPO
Serial.print(",");
Serial.print(Angulo); ///ANGULO DE INCLINACION
Serial.print(",");
Serial.print(r);///RAPIDEZ
Serial.print(",");
Serial.print(Vmed);///VELOCIDAD MEDIA
Serial.print(",");
Serial.print(Vinst);/// VELOCIDAD INSTANTANEA
Serial.print(",");
Serial.print(Amed ); ///ACELERACION MEDIA
Serial.print(",");
Serial.print(Ainst);///ACELERACION INSTANTANEA
Serial.print(",");
Serial.println(fuerza);/// F=m*a
delay(10);

```

Anexo 5 Programación módulo control torres

```

#include <Stepper.h>
#include <TouchScreen.h>
#include <OPENSIMART_kbv.h>
#include <LM75.h>
#include <Wire.h>
// Pines necesarios para los 4 pines del panel tactil
#define YP A1 // Pin analogico A1 para ADC

```

```

#define XM A2 // Pin analogico A2 para ADC
#define YM 7
#define XP 6
#define MINPRESSURE 1
#define MAXPRESSURE 1500
TouchScreen ts = TouchScreen(XP, YP, XM, YM, 663);
short TS_MINX = 5; // Coordenadas del panel tactil para delimitar
short TS_MINY = 5; // el tamaño de la zona donde podemos presionar
short TS_MAXX = 950; // y que coincida con el tamaño del LCD
short TS_MAXY = 930;
#define LCD_CS A3 // Definimos los pines del LCD
#define LCD_CD A2 // para poder visualizar elementos graficos
#define LCD_WR A1
#define LCD_RD A0
#define LCD_RESET A4
#define NEGRO 0x0000 // Definimos los colores
#define AZUL 0x001F // que utilizaremos para
#define ROJO 0xF800 // el texto y los elementos graficos
#define VERDE 0x07E0
#define CYAN 0x07FF
#define MAGENTA 0xF81F
#define AMARILLO 0xFFE0
#define BLANCO 0xFFFF
OPENS_MART_kbv pantalla(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET);
int X; // Variables que almacenaran la coordenada
int Y; // X, Y donde presionemos y la variable Z
int Z; // almacenara la presion realizada
////////////////////////////////////
//CONTROL TORRES
////////////////////////////////////
const int Trigger_A = 38; //Pin digital 2 para el Trigger del sensor
const int Echo_A = 36; //Pin digital 3 para el Echo del sensor
const int Trigger_B = 39; //Pin digital 2 para el Trigger del sensor
const int Echo_B= 37;
int led_1= 51;
int led_2= 9;
#define STEPS 48
Stepper M_elevador_A(STEPS,47,49,51,53);
Stepper M_elevador_B(STEPS,46,48,50,52);
////////////////////////////////////
int a =44; //disparador selenoide
////////////////////////////////////
LM75 TEMP; // TEMPERATURA
////////////////////////////////////

void setup() {
  pantalla.begin(0x9327);
  pantalla_arranque();
  pantalla_inicio ();
  //////////////////////////////////////

  //////////////////////////////////////
  pinMode(a, OUTPUT);
  // Iniciamos el LCD especificando el controlador ILI9341.
  Serial.begin(9600);
  //////////////////////////////////////
  pinMode(Trigger_A, OUTPUT); //pin como salida
  pinMode(Echo_A, INPUT); //pin como entrada
  digitalWrite(Trigger_A, LOW); //Inicializamos el pin con 0

```

```

pinMode(Triquer_B, OUTPUT); //pin como salida
pinMode(Echo_B, INPUT); //pin como entrada
digitalWrite(Triquer_B, LOW); //Inicializamos el pin con 0
////////////////////////////////////
////////////////////////////////////
M_elevador_A.setSpeed(660);
M_elevador_B.setSpeed(660);
////////////////////////////////////

}

void loop()
{

lecturaPanel();

if((X>20&& X <190) && (Y>140&& Y<220)&&(Z > MINPRESSURE && Z < MAXPRESSURE))
{

pantalla_MRU();

}

if((X>65&& X <235) && (Y>20&& Y<70)&&(Z > MINPRESSURE && Z < MAXPRESSURE))
{

pantalla_MRUV();

}

if((X>53&& X <95) && (Y>1&& Y<31)&&(Z > MINPRESSURE && Z < MAXPRESSURE))
{
pantalla_inicio();
}
////////////////////////////////////
//CONTROL MRU
////////////////////////////////////

if((X>180&& X <250) && (Y>70&& Y<140)&&(Z > MINPRESSURE && Z < MAXPRESSURE))
{

long t_A; //timepo que demora en llegar el eco
long d_A; //distancia en centimetros
digitalWrite(Triquer_A, HIGH);
delayMicroseconds(5); //Enviamos un pulso de 10us
digitalWrite(Triquer_A, LOW);
t_A = pulseIn(Echo_A, HIGH); //obtenemos el ancho del pulso
d_A = (t_A/59); //escalamos el tiempo a una distancia en cm
Serial.print("Distancia TORRE_A: ");
Serial.print(d_A); //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
delay(2); //Hacemos una pausa de 100ms
////////////////////////////////////
long t_B; //timepo que demora en llegar el eco
long d_B; //distancia en centimetros
digitalWrite(Triquer_B, HIGH);
delayMicroseconds(5); //Enviamos un pulso de 10us
digitalWrite(Triquer_B, LOW);

```

```

t_B = pulseIn(Echo_B, HIGH); //obtenemos el ancho del pulso
d_B = (t_B/59); //escalamos el tiempo a una distancia en cm
Serial.print("Distancia TORRE_B: ");
Serial.print(d_B); //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
delay(2);

if ((d_B>8)&&(d_B>8)){
  M_elevador_B.step(-100);

} else if ((d_B<8)&&(d_B<8)){
  M_elevador_B.step(100);
}

if((d_A>9)&&(d_A>9)){

  M_elevador_A.step(100);

} else if ((d_A<9)&&(d_A<9)) {
  M_elevador_A.step(-100);
}

}

////////////////////////////////////
if((X>50&& X <91) && (Y>80&& Y<115)&&(Z > MINPRESSURE && Z < MAXPRESSURE))
{
  inicio_de_practica();
}
////////////////////////////////////
if((X>125&& X <175) && (Y>75&& Y<135)&&(Z > MINPRESSURE && Z < MAXPRESSURE))
{

  practica_proceso();
  digitalWrite(a, HIGH);
  delay(3000);
  digitalWrite(a, LOW);
  delay(3000);
  //delay(1000);
  pantalla_inicio ();
}

////////////////////////////////////
//CONTROL MRUV A--B1
////////////////////////////////////
if((X>254&& X <315) && (Y>170&& Y<198)&&(Z > MINPRESSURE && Z < MAXPRESSURE))
{

long t_A; //timepo que demora en llegar el eco
long d_A; //distancia en centimetros
digitalWrite(Trigger_A, HIGH);
delayMicroseconds(5); //Enviamos un pulso de 5us
digitalWrite(Trigger_A, LOW);
t_A = pulseIn(Echo_A, HIGH); //obtenemos el ancho del pulso
d_A = (t_A/59); //escalamos el tiempo a una distancia en cm
Serial.print("Distancia TORRE_A: ");
Serial.print(d_A); //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
delay(10); //Hacemos una pausa de 100ms

```

```

////////////////////////////////////
long t_B; //timepo que demora en llegar el eco
long d_B; //distancia en centimetros
digitalWrite(Triquer_B, HIGH);
delayMicroseconds(5); //Enviamos un pulso de 10us
digitalWrite(Triquer_B, LOW);
t_B = pulseIn(Echo_B, HIGH); //obtenemos el ancho del pulso
d_B = (t_B/59); //escalamos el tiempo a una distancia en cm
Serial.print("Distancia TORRE_B: ");
Serial.print(d_B); //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
delay(10);

if ((d_B>32)&&(d_B<32)){
  M_elevador_B.step(-100);

} else if ((d_B<32)&&(d_B>32)){
  M_elevador_B.step(100);
}

if((d_A>9)&&(d_A<9)){

  M_elevador_A.step(100);

} else if ((d_A<9)&&(d_A>9)) {
  M_elevador_A.step(-100);
}
}

////////////////////////////////////
//CONTROL MRUV A--B2
////////////////////////////////////
if((X>254&& X <315) && (Y>120&& Y<145)&&(Z > MINPRESSURE && Z < MAXPRESSURE))
{

long t_A; //timepo que demora en llegar el eco
long d_A; //distancia en centimetros
digitalWrite(Triquer_A, HIGH);
delayMicroseconds(5); //Enviamos un pulso de 10us
digitalWrite(Triquer_A, LOW);
t_A = pulseIn(Echo_A, HIGH); //obtenemos el ancho del pulso
d_A = (t_A/59); //escalamos el tiempo a una distancia en cm
Serial.print("Distancia TORRE_A: ");
Serial.print(d_A); //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
delay(10); //Hacemos una pausa de 100ms
////////////////////////////////////
long t_B; //timepo que demora en llegar el eco
long d_B; //distancia en centimetros
digitalWrite(Triquer_B, HIGH);
delayMicroseconds(5); //Enviamos un pulso de 10us
digitalWrite(Triquer_B, LOW);
t_B = pulseIn(Echo_B, HIGH); //obtenemos el ancho del pulso
d_B = (t_B/59); //escalamos el tiempo a una distancia en cm
Serial.print("Distancia TORRE_B: ");
Serial.print(d_B); //Enviamos serialmente el valor de la distancia
Serial.print("cm");
}

```

```

Serial.println();
delay(10);

if ((d_B>53)&&(d_B>53)){
  M_elevador_B.step(-100);

} else if ((d_B<53)&&(d_B<53)){
  M_elevador_B.step(100);
}

if((d_A>9)&&(d_A>9)){

  M_elevador_A.step(100);

} else if ((d_A<9)&&(d_A<9)) {
  M_elevador_A.step(-100);
}
}

////////////////////////////////////
//CONTROL MRUV A1--B
////////////////////////////////////
if((X>254&& X <315) && (Y>60&& Y<90)&&(Z > MINPRESSURE && Z < MAXPRESSURE))
{

  long t_A; //timepo que demora en llegar el eco
  long d_A; //distancia en centimetros
  digitalWrite(Trieger_A, HIGH);
  delayMicroseconds(5); //Enviamos un pulso de 10us
  digitalWrite(Trieger_A, LOW);
  t_A = pulseIn(Echo_A, HIGH); //obtenemos el ancho del pulso
  d_A = (t_A/59); //escalamos el tiempo a una distancia en cm
  Serial.print("Distancia TORRE_A: ");
  Serial.print(d_A); //Enviamos serialmente el valor de la distancia
  Serial.print("cm");
  Serial.println();
  delay(10); //Hacemos una pausa de 100ms
  //////////////////////////////////////
  long t_B; //timepo que demora en llegar el eco
  long d_B; //distancia en centimetros
  digitalWrite(Trieger_B, HIGH);
  delayMicroseconds(5); //Enviamos un pulso de 10us
  digitalWrite(Trieger_B, LOW);
  t_B = pulseIn(Echo_B, HIGH); //obtenemos el ancho del pulso
  d_B = (t_B/59); //escalamos el tiempo a una distancia en cm
  Serial.print("Distancia TORRE_B: ");
  Serial.print(d_B); //Enviamos serialmente el valor de la distancia
  Serial.print("cm");
  Serial.println();
  delay(10);

  if ((d_B>9)&&(d_B>9)){
    M_elevador_B.step(-100);

  } else if ((d_B<9)&&(d_B<9)){
    M_elevador_B.step(100);
  }

  if((d_A>32)&&(d_A>32)){

```

```

M_elevador_A.step(100);

} else if ((d_A<32)&&(d_A<32)) {
  M_elevador_A.step(-100);
}
}

////////////////////////////////////
//CONTROL MRUV A1--B
////////////////////////////////////
if((X>254&& X <315) && (Y>5&& Y<36)&&(Z > MINPRESSURE && Z < MAXPRESSURE))
{

long t_A; //timepo que demora en llegar el eco
long d_A; //distancia en centimetros
digitalWrite(Trieger_A, HIGH);
delayMicroseconds(5); //Enviamos un pulso de 10us
digitalWrite(Trieger_A, LOW);
t_A = pulseIn(Echo_A, HIGH); //obtenemos el ancho del pulso
d_A = (t_A/59); //escalamos el tiempo a una distancia en cm
Serial.print("Distancia TORRE_A: ");
Serial.print(d_A); //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
delay(10); //Hacemos una pausa de 100ms
////////////////////////////////////
long t_B; //timepo que demora en llegar el eco
long d_B; //distancia en centimetros
digitalWrite(Trieger_B, HIGH);
delayMicroseconds(5); //Enviamos un pulso de 10us
digitalWrite(Trieger_B, LOW);
t_B = pulseIn(Echo_B, HIGH); //obtenemos el ancho del pulso
d_B = (t_B/59); //escalamos el tiempo a una distancia en cm
Serial.print("Distancia TORRE_B: ");
Serial.print(d_B); //Enviamos serialmente el valor de la distancia
Serial.print("cm");
Serial.println();
delay(10);

if ((d_B>9)&&(d_B>9)){
  M_elevador_B.step(-100);

} else if ((d_B<9)&&(d_B<9)){
  M_elevador_B.step(100);
}

if((d_A>52)&&(d_A>52)){

  M_elevador_A.step(100);

} else if ((d_A<52)&&(d_A<52)) {
  M_elevador_A.step(-100);
}
}

}

```