



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA:

INGENIERÍA ELECTRÓNICA

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO
DE:**

INGENIERO ELECTRÓNICO

PROYECTO TÉCNICO

**DISEÑO E IMPLEMENTACIÓN CON ALGORITMO FUZZY PARA CONTROL
DE VELOCIDAD EN MOTOR DE CORRIENTE CONTINUA EN SISTEMAS
EMBEBIDOS LATTEPANDA CON MONITOREO EN PLATAFORMA IOT**

AUTORES:

KEVIN RODRÍGUEZ - MICHAEL BRIONES

TUTOR:

PHD. VÍCTOR HUILCAPI

GUAYAQUIL- ECUADOR

2023

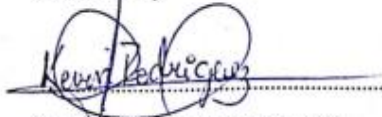
**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN**

Yo, Kevin Efrén Rodríguez Morocho con documento de identificación N° 0952471977 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 8 de agosto de 2023

Atentamente,



Kevin Efrén Rodríguez Morocho

CI: 0952471977


**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN**

Yo, Michael Rolando Briones García con documento de identificación N° 0927728147 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 8 de agosto de 2023

Atentamente,

.....

Michael Rolando Briones García

CI: 0927728147

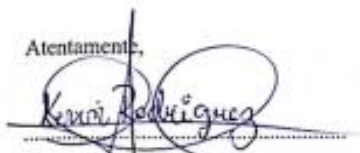
**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Kevin Rodríguez Morocho con documento de identificación N° 0952471977, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del: Proyecto Técnico: "Diseño e implementación con algoritmo Fuzzy para control de velocidad en motor de corriente continua en sistemas embebidos LattePanda con monitoreo IOT", el cual ha sido desarrollado para optar por el título de: Ingeniero electrónico mención en Automatización Industrial, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 8 de agosto de 2023

Atentamente,



Kevin Efraín Rodríguez Morocho

CI: 0952471977

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Michael Briones García con documento de identificación N° 0927728147, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto Técnico: "Diseño e implementación con algoritmo Fuzzy para control de velocidad en motor de corriente continua en sistemas embebidos LattePanda con monitoreo IOT", el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico mención en Telecomunicaciones, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 8 de agosto de 2023

Atentamente,

Michael Briones.....

Michael Rolando Briones García

CI: 0927728147

CERTIFICADO DE DIRECCIÓN DE TRABAJO DE TITULACIÓN

Yo, Víctor Huilcapi S, con documento de identificación N° 0917206294, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO E IMPLEMENTACION CON ALGORITMO FUZZY PARA CONTROL DE VELOCIDAD EN MOTOR DE CORRIENTE CONTINUA EN SISTEMAS EMBEBIDOS LATTEPANDA CON MONITOREO EN PLATAFORMA IOT , realizado por Kevin Rodríguez con documento de identificación N° 0952471977 y Michael Briones con documento de identificación N° 0927728147, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 8 de agosto de 2023

Atentamente,



PHD Víctor Huilcapi S.

CI: 0917206294

DEDICATORIA
KEVIN RODRIGUEZ

Este trabajo va dedicado en primer lugar a Dios pues he podido fortalecerme en la fe y retribuir los dones y talentos que me ha entregado, a mi familia, padres, amigos quienes con su esfuerzo y colaboración han estado presentes en este proceso de preparación que he tomado para enfrentar con éxitos mi vida profesional, brindándome consejos y recursos necesarios para poder enfrentar este desafío de culminar mi proyecto de titulación y así poder servir a mi país y a la sociedad.

A mi Tío Luchito que en vida se alegraría de este triunfo para ti dedico este proyecto

DEDICATORIA
MICHAEL BRIONES

A Dios por todo lo bueno que me dejo vivir esta Carrera con sus grandes enseñanzas, a mis Padres Cynthia y Rolando que han formado parte de este gran proceso totalmente nuevo en sus inicios, fueron ese motor e inspiración cada vez que no podía más, a ellos les dedico mis logros Personales y Profesionales que he alcanzado y estaré por cumplir porque este paso es un pequeño peldaño de los muchos que vienen.

AGRADECIMIENTO (KEVIN RODRIGUEZ)

Agradezco a mi familia que con su aliento constante me ayudaron a superar este gran desafío, a todas las personas que conocí en este duro camino, pero no imposible que me ayudaron a forjar el carácter y no darme por vencido.

Agradezco a mis amigos de vida y de camino que siempre han estado pendientes y animándome para cumplir con esta meta tan anhelada.

Agradezco a mis padres Efrén y Noralbita por su amor incondicional, paciencia y recordarme que con constancia se llega al éxito.

A mi abuelita Delia que con sus oraciones y bendiciones me han permitido llegar a la culminación de mi titulación.

A nuestro tutor de tesis PHD. Víctor Huilcapi por su constante apoyo y sapiencia en la dirección de nuestro proyecto de titulación que el Buen Dios le de salud y larga vida y continúe encaminando a miles de jóvenes en proyectos innovadores.

AGRADECIMIENTO (MICHAEL BRIONES)

Agradezco a Dios por permitirme cumplir este logro, ese sueño que tiene todo bachiller al terminar el colegio y empezar una carrera, una vida diferente sin saber todo lo que se puede conocer, aprender y ejercer en esta hermosa carrera. A mis Padres Cynthia García y Rolando Briones, quienes con todo su esfuerzo pudieron darme el empujón y apoyo que necesitaba para seguir con la meta trazada. A mi abuela que puede ver este logro gracias a sus oraciones y ejemplos de perseverancia, a mis amigos cercanos que con el pasar del tiempo han demostrado ser esa familia en otra parte de la tierra que uno encuentra.

Quiero extender un profundo agradecimiento a quienes estuvieron junto a mí en todo momento y siempre fueron ese apoyo que uno necesita; Dayannara Crespo, los chicos del barrio, Derecho de Autor.

Y a los docentes de la carrera de Ing. Electrónica por compartir todos sus conocimientos, experiencias. A mi tutor PHD. Víctor Huilcapi por su asesoría en este gran proyecto.

ÍNDICE GENERAL

TITULACIÓN.....	¡Error! Marcador no definido.
TITULACIÓN.....	¡Error! Marcador no definido.
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA	¡Error! Marcador no definido.
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA	¡Error! Marcador no definido.
CERTIFICADO DE DIRECCIÓN DE TRABAJO DE TITULACIÓN	¡Error! Marcador no definido.
INTRODUCCIÓN	1
CAPÍTULO 1: EL PROBLEMA.....	3
1.1 Planteamiento del problema	3
1.2 Delimitación del problema	3
1.3 Objetivos	4
1.3.1 Objetivo general	4
1.3.2 Objetivos específicos	4
1.4 Justificación del proyecto	4
1.5 Hipótesis.....	5
1.6 Variable e Indicadores.....	5
1.6.1 Variables	5
1.6.2 Indicadores	6
1.7 Metodología	6
1.7.1 Método Inductivo	6
1.7.2 Método Teórico y Sistemático.....	6
1.7.3 Técnica Documental.....	6
1.8 Población y Muestra	7
1.8.1 Población.....	7
1.8.2 Muestra	7
1.9 Descripción de la propuesta	8
1.10 Beneficiarios	8
1.11 Impacto	9
1.12 Fundamentos Teóricos.....	10

CAPÍTULO 2 : MARCO TEÓRICO.....	13
2.1 Microcontroladores.....	13
2.1.1 Microprocesadores.....	13
2.1.2 Sistemas Embebidos.....	14
2.1.3 Ventajas de los Sistemas Embebidos.....	15
2.1.4 Desventajas de los Sistemas Embebidos.....	16
2.2 PID.....	17
2.2.1 Proporcional (P).....	19
2.2.2 Integral (I).....	20
2.2.3 Proporcional Derivativo (PD).....	20
2.2.4 Proporcional Integral Derivativo (PID).....	21
2.3 Sistemas de control Difusos.....	21
2.3.2 Fusificación.....	22
2.3.3 Inferencia.....	22
2.4 Lattepanda.....	23
2.4.1 Lattepanda Alpha.....	23
2.4.1 Atmel 32UA.....	24
2.5 Software de instrumentación LabView.....	25
2.6 Motores de Corriente Continua.....	25
2.7 Codificadores Rotativo (Encoder).....	27
2.7.1 Encoder Incrementales.....	28
2.7.2 Encoder absolutos.....	30
2.8 Fuente de alimentación.....	31
2.9 Pulsadores y selectores.....	32
2.10 Internet de las cosas.....	32
2.10.1 Características del Internet de las cosas.....	32
3. CAPITULO 3: MARCO METODOLÓGICO.....	34
3.1 Recopilación de información y revisión bibliográfica.....	34
3.2 Diseño del algoritmo difuso.....	34
3.3 Implementación del controlador PID.....	34
3.4 Integración en sistemas embebidos LattePanda.....	35
3.5 Conexión y monitoreo en plataforma IoT.....	35
3.6 Validación y pruebas.....	35
3.7 Diseño e Implementación de modulo didáctico.....	35
4. CAPITULO 4: GUÍA DE PRÁCTICAS DE LABORATORIOS.....	38

Práctica 1: Programación de señales discretas analógicas de entrada y salida en LabView con supervisión y control en plataforma IOT	38
Práctica 2: Diseño de un controlador PID clásico para control de velocidad de motor con supervisión y control en plataforma IOT	43
Práctica 3: Diseño de un controlador PID clásico para control de velocidad de motor con supervisión y control en plataforma IOT	46
Práctica 4: Diseño de un controlador difuso para control de velocidad de motor con supervisión y control en plataforma IOT.	55
Práctica 5: Comparación de un control PID convencional y un control difuso aplicado a velocidad de un motor.	63
5. CAPITULO 5: ANÁLISIS Y RESULTADOS	665
5.1 Resultados Obtenidos.....	66
5.2 Diseño y elaboración de los elementos del módulo didáctico	66
5.3 Guía o Manual de Prácticas de Laboratorio	67
Diseño de controlador PID en lattepanda.....	72
Diseño de controlador difuso en lattepanda	77
Creación de variables lingüísticas	77
Creación de reglas de correspondencia.....	81
Conexión entre LabView y lattepanda.....	83
CONCLUSIONES.....	80
RECOMENDACIONES.....	88
ANEXOS.....	89
Resultados de las prácticas.....	89
BIBLIOGRAFÍA.....	93

ÍNDICE DE FIGURAS

Figura 1: Microcontroladores	13
Figura 2: Microprocesador INTEL.....	14
Figura 3: Sistema Embebido.....	15
Figura 4: Control PID de una planta.....	18
Figura 5: Esquema interno de un variador de frecuencia	19
Figura 9: Diagrama de control Difuso	22
Figura 11: Lattepanda.....	23
Figura 12: Lattepanda Alpha	24
Figura 13: Atmel 32UA	25
Figura 14: Motor DC	26
Figura 15: Motor DC	26
Figura 16: Funcionamiento de disco rotativo del Encoder	28
Figura 17: Disposición de un Encoder incremental	29
Figura 18: Salidas A y B de un Encoder y valor (Q) del Biestable D según su sentido de giro	30
Figura 19: Codificador absoluto binario de 4bits. (a) Construcción Interna. (b) Señales de salida	31
Figura 20: Fuente de alimentación.....	31
Figura 21: Variedades de pulsadores	32
Figura 22: Módulo de entrenamiento.....	33
Figura 23: Conexiones módulo entrenamiento	36
Figura 24: Conexiones módulo entrenamiento	36
Figura 25: Diseño e implementación del módulo de entrenamiento.....	37
Figura 26: Conexiones y etiquetas de los elementos	66
Figura 27: Maleta de control para velocidad de un motor de corriente continua.....	67
Figura 28: Resultados Práctica 1 -Interfaz LabView	67
Figura 29: Resultados Práctica 1 -Interfaz en Ubidots.....	68
Figura 30: Resultados Práctica 2 -Interfaz LabView	68
Figura 31: Resultados Práctica 2 -Interfaz en Ubidots.....	69
Figura 32: Resultados Práctica 3 – Interfaz LabView	69
Figura 33: Resultados Práctica 3 – Interfaz Ubidots.....	70

Figura 34. Resultados Práctica 4 – Interfaz LabView	70
Figura 35. Resultados Práctica 4 – Interfaz Ubidots.....	71
Figura 36. Resultados Práctica 5 – Interfaz Ubidots.....	71
Figura 37. Diagrama de control del PID clásico.....	73
Figura 38. Señal paso en el motor DC	73
Figura 39. Ingreso de variables para análisis	74
Figura 40. estimación del modelo.....	74
Figura 41. Estimación con un retardo y un polo.....	75
Figura 42. Estimación con un retardo y un polo.....	75
Figura 43. Función de transferencia.....	76
Figura 44. Auto tuning Matlab.....	76
Figura 45. Las ganancias del PID	77
Figura 46. Complemento de LabView Fuzzy system Designer.....	77
Figura 47. Fuzzy System Designer	78
Figura 48. . Variables lingüísticas para velocidad	78
Figura 49. . Variables lingüísticas para el error	79
Figura 50. . Variables lingüísticas para la salida.....	81
Figura 51. Creación de reglas para el conjunto difuso.....	82
Figura 52. . Evaluación del sistema difuso en LabView.....	83
Figura 53. Conexión interna entre la lattepanda y LabView	83

RESUMEN

El presente proyecto técnico tiene como objetivo el diseño de un prototipo didáctico de pruebas para el control de la velocidad de un motor de corriente continua teniendo como núcleo una tarjeta Lattepanda Alpha con el sistema operativo Windows y el programa de instrumentación virtual LabView.

El objetivo principal es elaborar prácticas de aprendizaje para el correcto uso del módulo didáctico enfocándose en dos controladores, tal como el control PID clásico y un control basado en lógica difusa teniendo como variable de control la velocidad de un motor de corriente continua.

El equipo consta de una conexión a un servidor alojado en la nube permitiendo el almacenamiento de datos y visualización desde un entorno lejano de las variables que actúan en el sistema y los periféricos (Lattepanda, tarjeta de control de motor y encoder de motor), para el control se utiliza como entrada del sistema el valor de la velocidad mediante el encoder acoplado al motor, y realizar el control mediante una salida Pwm del Lattepanda.

Palabras claves: Lattepanda, Arduino, IoT, Motor DC, Módulo Didáctico, control PID, control difuso.

ABSTRACT

The purpose of this technical project is to design a didactic prototype of tests for the control of the speed of a current motor, having as its core a Lattepanada Alpha card with the Windows operating system and the LabView virtual instrumentation program.

The main objective is to develop learning practices for the correct use of the didactic module, focusing on two individually controls such as the classic PID control and a control based on Fuzzy logic having the speed of a direct current motor as a control variable.

The equipment consists of a connection to a server hosted in the cloud allowing the data storage and visualization from a distant environment of the variables that act in the system and the peripherals (Lattepanada, motor control card and motor encoder), for the control is used as the system input the speed value by means of the encoder coupled to the motor, and control by means of a Pwm output of the Lattepanada.

Keywords: Lattepanada, Arduino, IoT, CC Motor, Learning Module, PID control, Fuzzy control.

INTRODUCCIÓN

Las tecnologías informáticas enfocadas en el área del control inteligente desempeñan un papel importante en el proceso de la formación de los estudiantes en el campo de la electrónica y automatización.

La Universidad Politécnica Salesiana cuenta con laboratorios necesarios para el manejo de procesos industriales enfocados en la automatización y control; sin embargo, no se han explotado en su totalidad por el constante avance tecnológico, un campo que está ganando terreno es integrar sistemas de supervisión basados en la nube o internet de las cosas en procesos continuos.

La carrera de Ingeniería Electrónica y Automatización tiene pocos módulos didácticos para realizar prácticas con algoritmos de control inteligente que apliquen la lógica difusa, por lo general los estudiantes realizan prácticas con controles básicos PID.

En la actualidad los sistemas embebidos están en todas partes, debido a que es cualquier sistema que no es una computadora en sí, ni un servidor, ni clúster, en otras palabras, están diseñados para cumplir diversas funciones específicas, y que esta embebida como componente de un dispositivo físico completo, por ejemplo, un router es un dispositivo que forma parte de un sistema embebido de telecomunicaciones. (Sol Pedre, 2012)

El objetivo principal es diseñar e implementar un módulo didáctico para pruebas de un algoritmo difuso que ayuden a controlar la velocidad de un motor de corriente continua utilizando un sistema embebido o micro computador LattePanda a la par con una interfaz de monitoreo y control alojada en la nube Ubidots, el cual servirá como instrumento de apoyo para reforzar su formación académica en la parte teórica y práctica, logrando así, complementar los conocimientos en materias como electiva I, microcontroladores, sensores y/o alguna materia con fines de enseñanzas en el campo del control inteligente.

La implementación del proyecto surgió en base a la búsqueda de ampliar las opciones de equipos disponibles para el aprendizaje que brinden a los estudiantes el complemento necesario en la parte teórica y experimental para la formación de los futuros ingenieros , logrando así que las prácticas se realicen de manera efectiva y se

ahorre tiempo al momento de obtener conocimientos para el control de la velocidad de un motor enfocado en un controlador con lógica difusa adquiriendo nuevas destrezas al momento de la creación de variables lingüísticas de entrada y salida en este tipo de sistemas y la configuración de las reglas de correspondencia que dominen el sistema implementado, enfocados en el área de sistemas de control muy utilizados en el campo de la ingeniería electrónica, mecatrónica y robótica.

Con este proyecto el estudiante innovará en el uso de nuevas tecnologías basadas en la supervisión mediante la nube, tendrá como comprobar, los resultados y visualizar en una plataforma IOT, aprender a configurar una plataforma IOT para el envío y recepción de datos mediante el software LabView por un protocolo de comunicación HTTP (Hypertext Transfer Protocol).

1: EL PROBLEMA

1.1 Planteamiento del problema

La Universidad Politécnica Salesiana cuenta con laboratorios necesarios para el manejo de procesos industriales enfocados en la automatización y control; sin embargo, no se han explotado en su totalidad por el constante avance tecnológico, un campo que está ganando terreno es integrar sistemas de supervisión basados en la nube o internet de las cosas en procesos continuos, por lo cual es necesario un módulo base que guíe al estudiante en la conexión con una plataforma en la nube, para el cumplimiento de este objetivo se necesitan equipos muy específicos que permitan lograr un mejor entendimiento de estas áreas.

La universidad no cuenta actualmente con un módulo donde se haya realizado un estudio adecuado acerca de los sistemas embebidos en hardware y su aplicación en el área de la ingeniería de la carrera, lo cual no permite a los futuros ingenieros aplicar y obtener conocimientos sólidos acerca del tema realizado.

En la actualidad existe la necesidad de reforzar los laboratorios con un equipo el cual ayude al estudiante, mediante el uso de recursos prácticos, de tal manera motivar a los alumnos al uso de nuevas tecnologías como el internet de las cosas, permitiendo al estudiante de la carrera de Ingeniería Electrónica y Automatización, implementar controladores inteligentes integrándose con aplicaciones alojadas en la nube.

La carrera de Ingeniería Electrónica y Automatización tiene pocos módulos didácticos para realizar prácticas con algoritmos de control inteligente que apliquen la lógica difusa, por lo general los estudiantes realizan prácticas con controles básicos PID, por tal motivo se propone realizar un módulo el cual ayude al entendimiento de controladores con lógica difusa.

1.2 Delimitación del problema

El Proyecto de Titulación se desarrolló en la Universidad Politécnica Salesiana sede Guayaquil, Ecuador en el período 2022-2023.

1.3 Objetivos

1.3.1 Objetivo general

Elaborar prácticas de aprendizaje para el manejo y correcto uso del módulo didáctico enfocándose en dos controles individualmente tal como el control PID clásico y un control basado en lógica difusa teniendo como variable de control la velocidad de un motor de corriente continua.

1.3.2 Objetivos específicos

1. Diseñar la estructura y la circuitería electrónica de un módulo de pruebas con un núcleo LattePanda para la conexión con plataforma basada en el internet de las cosas para la visualización y control de variables.
2. Diseñar un controlador difuso para la velocidad de un motor de corriente directa.
3. Diseñar el algoritmo PID Clásico para la velocidad de un motor de corriente directa.
4. Comparar el comportamiento de los controladores planteados previamente.
5. Realizar un manual de 5 prácticas didácticas orientadas al área del control y el internet de las cosas con LabView y LattePanda.

1.4 Justificación del proyecto

En la búsqueda de ampliar las opciones de equipos disponibles para el aprendizaje que brinden a los estudiantes el complemento necesario en la parte teórica y experimental necesaria para la formación de los futuros ingenieros teniendo como principal objetivo que el estudiante no requiera de comprar elementos externos y tener todo en el equipo de manera que las prácticas se realicen de manera efectiva y se ahorre tiempo al momento de obtener conocimientos para el control de la velocidad de un motor enfocado en un controlador con lógica difusa y lograr adquirir destrezas al momento de la creación de variables lingüísticas de entrada y salida en este tipo de sistemas y la

configuración de las reglas de correspondencia que dominen al sistema a implementar, enfocadas en el área de sistemas de control muy utilizados el campo de la ingeniería electrónica, mecatrónica y robótica.

El presente proyecto de titulación permite al estudiantado aplicar los conocimientos adquiridos a lo largo de la carrera en el ámbito de los sistemas embebidos y las tecnologías que se pueden utilizar y aprender de la misma, haciendo más interactivas y acorde las experiencias educativas de la carrera de ingeniería.

Con este proyecto el estudiante innovará en el uso de nuevas tecnologías enfocadas en la supervisión mediante la nube, tendrá como comprobar, los resultados y visualizar en una plataforma IOT, aprender a configurar una plataforma IOT para el envío y recepción de datos mediante el software LabView por un protocolo de comunicación HTTP (Hypertext Transfer Protocol).

Se desarrolla un complemento académico mediante la utilización de controladores difusos mediante el software LabView utilizando un sistema embebido con su propio sistema operativo disminuyendo el uso de recursos para la universidad como uso de computador, ya que el Lattepanada posee un sistema operativo con el cual trabajar y realizar la conexión con una plataforma IOT.

1.5 Hipótesis

Los dispositivos de programación de procesos serán capaces de mantener una comunicación continua entre ellos mediante una red de internet de las cosas para monitoreo y supervisión del proceso de control de motor.

Estos dispositivos de programación de procesos se comunicarán creando una red IOT y transmitida a la aplicación de monitoreo y a la PC.

1.6 Variable e Indicadores

1.6.1 Variables

- Señales analógicas, luces indicadoras, selectores, indicadores.
- Comunicación con internet.
- Señales Digitales.

1.6.2 Indicadores

- Salidas digitales y analógicas.
- Visualización de parámetros IOT.
- Visualización de estados digitales por medio de luces indicadoras

1.7 Metodología

1.7.1 Método Inductivo

Con este método se pondrá a prueba los conocimientos de materias como; Instalaciones Industriales, Automatización Industrial I y II, Sistemas Microprocesados, Sensores y Transductores, Teoría de Control, Informática Industrial, Arduino IDE para programación de estación Arduino, Matlab y LabView para modelado matemático necesario para control PID y Fuzzy Logic Control System y que servirán para el desarrollo del módulo didáctico y su programación.

1.7.2 Método Teórico y Sistemático

- Funcionamiento de Lattepanda.
- Funcionamiento de Motor DC.
- Funcionamiento de Microcontroladores y Arduino.
- Funcionamiento de entornos de programación.
- Software de programación.
- Investigación de información referente al tema en internet.

Estos puntos mencionados han sido considerados para la realización de este proyecto.

1.7.3 Técnica Documental

Se recopiló información para tener en cuenta fundamentos teóricos de todos los elementos que conforman el proyecto a implementar y que se basó en libros y páginas encontradas en internet para esta técnica.

1.8 Población y Muestra

1.8.1 Población

La población son todos los estudiantes de la carrera Ingeniería Electrónica de la Universidad Politécnica Salesiana.

1.8.2 Muestra

La población objetivo de este proyecto está compuesta por estudiantes de la carrera de Ingeniería Electrónica que se encuentran cursando asignaturas relacionadas con el control de procesos y sistemas industriales. Específicamente, se considerarán los estudiantes inscritos en las siguientes materias:

- Automatización I y II: Estas asignaturas abordan la programación básica de controladores lógicos programables, lo cual brinda a los estudiantes conocimientos fundamentales para el control de procesos industriales.
- Microprocesados I y II: En estas materias, los alumnos adquieren habilidades en la programación de microcontroladores, lo cual es relevante para comprender y analizar los diferentes usos de plataformas de código de microcontroladores en aplicaciones industriales.
- Redes de Computadora II y III: Estas asignaturas se centran en el estudio de protocolos de red y protocolos de comunicación industrial. Los estudiantes obtienen conocimientos sobre la configuración y comunicación de dispositivos en entornos industriales, lo cual es relevante para la integración de sistemas embebidos en una plataforma de IoT.
- Teoría de Control e Informática Industrial: En estas asignaturas, los estudiantes adquieren conocimientos en el diseño y aplicación de técnicas de control en sistemas industriales. La aplicación de estos conocimientos resulta relevante para el desarrollo del algoritmo difuso utilizado en el control de velocidad del motor.

1.9 Descripción de la propuesta

La presente propuesta de proyecto tiene como objetivo el diseño e implementación de un algoritmo difuso y PID para el control de velocidad en un motor de corriente continua utilizando el sistema embebido LattePanda. Además, se busca habilitar el monitoreo del motor en tiempo real a través de una plataforma de IoT (Internet de las Cosas).

El control de velocidad preciso en motores de corriente continua es fundamental en diversas aplicaciones industriales y sistemas automatizados. El uso de algoritmos PID y difusos permite adaptarse a las variaciones y perturbaciones del sistema, mejorando la estabilidad y el rendimiento del control. Además, la integración de sistemas embebidos LattePanda con una plataforma de IoT brinda la posibilidad de monitorear y gestionar el motor de forma remota, lo que amplía su potencial de aplicación en entornos industriales.

En este proyecto, se llevará a cabo el diseño del algoritmo PID y difuso que permita controlar la velocidad del motor de corriente continua. Para ello, se realizará una revisión exhaustiva de los fundamentos teóricos del control PID y difuso, se investigarán las técnicas más adecuadas para su implementación en sistemas embebidos LattePanda, se realizarán los circuitos correspondientes para cada práctica propuesta en este proyecto y se crearán interfaces de usuario para visualización de gráficas comparativas entre los distintos controles utilizados.

1.10 Beneficiarios

Los beneficiarios del proyecto serían principalmente los estudiantes de Ingeniería Electrónica que participen en las actividades relacionadas con la maleta didáctica y la estación de trabajo. Estos estudiantes se beneficiarán al tener acceso a un entorno práctico y realista donde podrán adquirir conocimientos y habilidades relevantes en áreas como comunicaciones industriales, fundamentos de electrónica, internet de las cosas y automatización industrial.

1.11 Impacto

El impacto del proyecto puede tener diversas dimensiones:

- Impacto educativo: El proyecto proporcionará a los estudiantes de Ingeniería Electrónica una experiencia de aprendizaje enriquecedora y práctica. Podrán aplicar los conocimientos teóricos adquiridos en las aulas a través de la manipulación y programación de dispositivos reales. Esto les permitirá desarrollar habilidades prácticas y aumentar su comprensión de los conceptos relacionados con el control de velocidad en motores de corriente continua y la implementación de algoritmos PID y difusos. Este impacto educativo los preparará mejor para enfrentar desafíos en su futura carrera profesional.
- Impacto en la industria: Al brindar a los estudiantes la oportunidad de trabajar con esta maleta didáctica y una estación PC que simulan situaciones industriales reales, el proyecto los familiarizará con equipos y tecnologías utilizados en la industria. Esto les permitirá adquirir habilidades prácticas y conocimientos relevantes que los harán más aptos para enfrentar los desafíos tecnológicos en el campo del control de velocidad en motores de corriente continua. Como resultado, los graduados estarán mejor preparados para contribuir de manera efectiva en proyectos de automatización industrial y en la implementación de soluciones de IoT.
- Impacto en la investigación y desarrollo: El proyecto puede generar nuevos conocimientos y resultados significativos que contribuyan al avance del campo del control PID y difuso, la implementación de algoritmos en sistemas embebidos y la integración de IoT en aplicaciones industriales. Los hallazgos y resultados obtenidos durante el proyecto pueden ser compartidos a través de publicaciones, presentaciones o conferencias, lo que permitiría la difusión del conocimiento y su aplicación en otros contextos similares.
- Impacto social y económico: La implementación de soluciones de control de velocidad más eficientes y precisas en motores de corriente continua puede tener un impacto positivo en diversos sectores industriales. Esto puede resultar en un mayor rendimiento y eficiencia en los procesos de producción, lo que a su vez podría tener beneficios económicos para las empresas y contribuir al desarrollo económico en general. Además, al integrar la plataforma de IoT, se abre la posibilidad de monitorear

y gestionar el motor de forma remota, lo que puede proporcionar a las empresas mayor flexibilidad y control en sus operaciones.

1.12 Fundamentos Teóricos

TEMA 1:

Resumen: El trabajo explorado siguiente., detalla el diseño e implementación de un módulo para el control de caudal de líquidos, integrado a una interfaz hombre máquina. Este módulo será parte del equipo didáctico para las materias de instrumentación y control industrial, con el fin de complementar el aprendizaje de los estudiantes en sistemas de control, mediante la interacción con la interfaz gráfica y con los diferentes componentes del módulo. Está estructurado en cinco capítulos, de los cuales en el primer capítulo se plantea el problema de investigación, este hace referencia a la insuficiente aplicación práctica de los conocimientos adquiridos en el aula por parte de los estudiantes, se propone objetivos que estén encaminados a contribuir con una parte de la solución del problema de investigación, y se justifica el trabajo de grado por medio de la implementación de un módulo didáctico para el control de caudal de líquidos. En el segundo capítulo se describe el marco teórico, este contiene toda la información necesaria para el diseño del módulo didáctico tales como: propiedades de los fluidos, tipos de medidores de caudal, tipos de bombas, cálculo para la selección del sistema de bombeo, conceptos generales sobre los sistemas de control, sintonización del PID por el método de Ziegler-Nichols, una pequeña introducción a la identificación de sistemas para posteriormente aplicarlo en el programa de MATLAB, dispositivos de control, comunicación y programación del PLC, y una breve descripción del entorno grafico de LabView. En el tercer capítulo se hace referencia a la metodología, explica la investigación documental, tecnológica y práctica. Igualmente, de los métodos utilizados en la elaboración del presente trabajo de grado. En el cuarto capítulo se detalla la propuesta tecnológica, en esta sección se describe: el proceso de diseño e implementación del módulo didáctico, el análisis de las pérdidas en la tubería para el dimensionamiento de la bomba, se justifica la selección de cada uno de sus componentes mediante un criterio técnico, la configuración de los parámetros del variador de frecuencia, la comunicación y configuración del PLC, el cálculo de los coeficientes del controlador PID utilizando el

método de Ziegler-Nichols por medio de las herramientas del programa de MATLAB, y la elaboración de la interfaz gráfica para el monitoreo y control del módulo didáctico. Finalmente, en el quinto capítulo se detalla las conclusiones y recomendaciones obtenidas en el transcurso de la elaboración del módulo didáctico. (Aza Saraguro, Lennin Ramiro, Bastidas Carlosama, Danilo Ramiro, 2016)

TEMA 2:

Resumen: Un controlador PID (Controlador Proporcional-Integral-Derivativo) es un mecanismo de control por realimentación ampliamente usado en sistemas de control industrial. Este calcula la desviación o error entre un valor medido y un valor deseado. En este proyecto se aplicó un controlador PID para un prototipo de planta con un control de nivel por presión hidrostática. Usando un transmisor de presión para medir la presión en bares (bar) dentro del tanque y un variador de frecuencia para controlar la velocidad rotacional de un motor de corriente alterna (AC), se programó un controlador PID para controlar el nivel en centímetros (cm) del fluido dentro del tanque mediante el software MYOPENLAB, estos valores medidos fueron corroborados usando la fórmula de la presión para calcular la altura del fluido dentro del tanque. Viendo la necesidad de la lubricadora “El Alemán” en que requiere un tipo de control para no operar manualmente el aceite residual de los carros, este proyecto permite reducir los riesgos de accidentes graves como quemaduras de diferentes tipos por la manipulación de estos aceites. (Cando Sevilla, Jazmani Esterlin, Calberto Oyarvide, Erick Andrés, 2018)

TEMA 3:

Resumen: En el tema de proyecto explorado: “Diseño de un prototipo de sistema de control para temperatura y humedad en el almacenamiento de arroz, utilizando sistemas embebidos”, consiste en la implementación a escala de un modelo de silo para el almacenamiento de arroz y un módulo el cual permite monitorear las variables de temperatura y humedad importantes para evitar proliferaciones de bacterias y agentes biológicos como insectos y microorganismos. En aplicaciones enfocadas en sistemas embebidos se tiene en cuenta la microelectrónica que con el uso de una lattepanda por su reducido tamaño y su versatilidad permite la conexión de pantallas gráficas y la instalación de programas como LabView. Dirigido principalmente para el sector Agroindustrial y sus beneficiarios así de igual manera para estudiantes de la carrera de

Ingeniería Electrónica de la Universidad Politécnica Salesiana para que desarrollen habilidades prácticas de programación y control en el módulo teniendo como sistema operativo Windows 10 Pro, permitiendo el uso de programas de instrumentación virtual como LabView. (Chichande Solano, Daniel Jean Pierre, Matamoros Galán, Ingrid Maylin, 2021)

2: MARCO TEÓRICO

2.1 Microcontroladores

Los microcontroladores o MCU se caracterizan por poseer una arquitectura tipo Harvard con memorias de programas independientes, por su simplicidad al momento de acceder a un espacio de memoria (Angulo Usategui, Romero Yesa, & M., 2006) tal como se muestra en la figura 1.



Figura 1: Microcontroladores (*Microchip, 2016*)

2.1.1 Microprocesadores

Los microprocesadores son aquellos que contienen un oscilador de cuarzo que genera los pulsos a un ritmo constante de manera que se pueden generar varios ciclos en un segundo, la velocidad de los microprocesadores se mide en frecuencias y estas pueden ser mega hertzios (MHz) o Giga hertzios (GHz) es decir, miles de millones o millones de ciclos por segundo superando a los microcontroladores (GONZALEZ, 2009).

Poseen subsistemas de entrada/salida y memoria que pueden ser combinados con un subsistema de CPU para formar una computadora o sistema embebido completo. Estos subsistemas se interconectan mediante los buses de sistema formados a su vez por el bus de control, el bus de direcciones y el bus de datos (Heath, 2003) tal como se muestra en la figura 2.



Figura 2: Microprocesador INTEL
(Heath, 2003)

2.1.2 Sistemas Embebidos

Los sistemas embebidos tienen una respuesta en tiempo real con un intervalo restringido, se clasifican como blandos o duros. Si un sistema de tiempo real blando no cumple con sus restricciones de tiempo, simplemente se degrada el rendimiento del sistema, pero si el sistema es de tiempo real duro y no cumple con sus restricciones de tiempo, el sistema fallará. Este fallo puede tener posiblemente consecuencias catastróficas (Heath, 2003).

Dichos sistemas usan como núcleo un microprocesador o microcontrolador, FPGA o DSPS, estos tienen sistemas simples y complejos (Cruz & Lutenberg, 2012) tal como se muestra en la figura 3.



Figura 3: Sistema Embebido (*Heath, 2003*)

2.1.3 Ventajas de los Sistemas Embebidos

- **Control total:** Estos sistemas posibilitan personalizar de forma casi total, permitiendo a los programadores emplear código propio para cambiar la interfaz del sistema, de igual manera la funcionalidad y las tareas que cumple cada pin del microprocesador, siendo así adaptable a cualquier ambiente.
- **Conectividad y adaptabilidad:** Si la ocasión lo amerita, estos dispositivos pueden conectarse a un ordenador para modificar o extraer datos del código, por otro parte su unificación entre sistemas embebidos y demás dispositivos es muy fácil, siendo de esta manera útil en el monitoreo de funcionalidad de las herramientas industriales automatizadas.
- **Reducción de costes:** están compuestos por módulos electrónicos desistiendo de emplear PLC o controladores lógicos programables, esto permite ahorrar costos, facilitando el mantenimiento al ser de fácil reemplazo de sus componentes.
- **Diseño modular:** los componentes incrustados son de fácil traslado, es decir se desmontan y reorganizar de manera sencilla, permitiendo integrarlos en cualquier posición y cualquier otro sistema electrónico.

- **Corto tiempo de respuesta:** Por lo general funcionan en tiempo real, ejecutan acciones en tiempos inmediatos, lo que significa que su tiempo de respuesta es muy rápido.
- **Accesibilidad:** En la actualidad, son de fácil acceso para que cualquier persona y/o institución puedan usarlos, herramientas de esta índole como Arduino o Raspberry Pi logran diseñar sistemas de control de luces, sensores de puertas automatizados e inclusive videoconsolas, todo esto de manera muy sencilla y de bajo costo. (Oasys, 2022)

2.1.4 Desventajas de los Sistemas Embebidos

- **Cifrado débil:** La criptografía en la que se basa la tecnología de los sistemas embebidos es débil, estas sencillas herramientas se basan en estándares poco seguros, por lo cual la información compartida entre dos diferentes dispositivos puede ser vulnerados e interceptados.
- **Falta de certificados:** La gran parte de estos servicios emplean certificados poco reconocidos en el sector, por lo cual un atacante puede fácilmente falsificarlo engañando al usuario.
- **Backdoors:** Los fabricantes tienen acceso a los cambios que el cliente haya hecho por medio de las puertas traseras de los sistemas embebidos, por lo cual, si un hacker tiene acceso a las mismas puede controlar el dispositivo.
- **Puertos de entrada y salida:** En el caso de haber puertos sobrantes se pueden emplear para violar la seguridad del sistema, entrar y controlar el dispositivo, debido a que por lo general no se deshabilitan los puertos empleados para las comunicaciones.
- **Ataques DOS/DDOS:** Posibilidad de una petición repetida de información que pueda bloquear los canales de comunicación, ya que los sistemas embebidos no saben tener controles de tramas de red y frecuentemente hay ataques de denegación de servicios. (Oasys, 2022)

2.2 PID

Un controlador o regulador PID es un dispositivo que permite controlar un sistema en lazo cerrado para que alcance el estado de salida deseado. El controlador PID está compuesto de tres elementos que proporcionan una **acción**

Proporcional, Integral y Derivativa. Estas tres acciones son las que dan nombre al controlador PID.

Los controladores PID poseen una función de transferencia que se muestra en la ecuación 1 (Dorf & Bishop, 2005), (Huilcapi, et al, 2017)

$$G(s) = K_p + \frac{K_i}{s} + K_d * s \quad (1)$$

Donde:

- K_p es la ganancia proporcional, que ajusta la respuesta en función del error presente.
- K_i es la ganancia integral, que corrige el error acumulado a lo largo del tiempo.
- K_d es la ganancia derivativa, que ajusta la respuesta en función de la tasa de cambio del error.

La estructura de control es por retroalimentación, esta operación permite calcular el error entre la entrada, el punto de comparación y la salida del sistema para un funcionamiento ideal. (Dorf & Bishop, 2005), (Huilcapi, et al, 2019) tal como se muestra en la ecuación 2 y en la figura 4.

$$G(s) = K_p \left(1 + \frac{1}{T_i * s} + T_d * s \right) \quad (2)$$

Donde:

- K_p es la ganancia proporcional, que ajusta la respuesta del controlador en función del error presente.
- T_i es el tiempo integral, que determina cómo el controlador corrige el error acumulado a lo largo del tiempo.
- T_d es el tiempo derivativo, que ajusta la respuesta del controlador en función de la tasa de cambio del error.

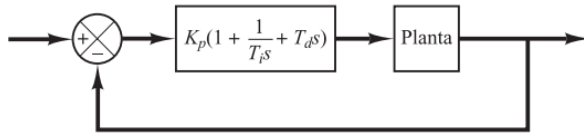


Figura 4: Control PID de una planta (Ogata, 2010)

El PID contiene acciones, una proporcional (P), integral (I) y derivativa (D). Se puede realizar diferentes tipos de controladores tales como P, PI, PD y PID (Morales & Zafra Siancas, 2013), (Lima, et al, 2017).

El control PID es un algoritmo el cual consta de tres parámetro o ganancias proporcional, integral y derivativo, el valor proporcional depende del error entre el valor medido y la referencia representada por la fórmula 3, el valor integral consta de los valores de los errores pasados representados por la fórmula 4 y el valor derivativo depende de una predicción de los valores futuros representado por la fórmula 5, fue patentado en 1939 por Albert Calender y Allan Stevenson de la firma Imperial Chemicoles (Åström, 1995), la formula del algoritmo PID está representado por la fórmula 6.

$$K_p e(t) \quad (3)$$

$$\frac{1}{T_i} \int_0^t e(\tau) d\tau \quad (4)$$

$$T_d \frac{de(t)}{dt} \quad (5)$$

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (6)$$

Comentado [VMHS1]: Esta gráfica tiene la función de transferencia de un PID. Escriba esa ecuación en (1).

Donde:

K_p es la acción proporcional

T_i es la acción intergral

T_d es la acción derivativa

En la figura 5 se muestra el esquema interno de un variador de frecuencia.

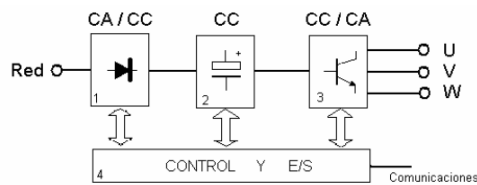


Figura 5: Esquema interno de un variador de frecuencia (Sevillano, 2011).

2.2.1 Proporcional (P)

La salida del controlador es proporcional al error, como se muestra en la ecuación 7 (Mazzone, 2002).

$$u(t) = K_p e(t) \quad (7)$$

Donde:

$u(t)$: salida del controlador.

$e(t)$: Error del sistema.

K_p : Ganancia proporcional al ajuste.

Dicha función de transferencia del controlador proporcional se muestra en la ecuación 8.

$$C_p(s) = K_p \quad (8)$$

Los controladores proporcionales sirven para una planta estable, las limitaciones son un trabajo limitado y error en un régimen permanente (Mazzone, 2002).

2.2.2 Integral (I)

La salida del controlador es proporcional al error acumulado como se visualiza en la siguiente ecuación 9.

$$U(t) = K_i \int_0^t e(t) dt \quad (9)$$

Donde:

K_i : Ganancia integral.

2.2.3 Proporcional Derivativo (PD)

Se define mediante la fórmula 10.

$$u(t) = K_p \cdot e(t) + K_p \cdot T_d \cdot \frac{d e(t)}{dt} \quad (10)$$

La T_d es una variable definida como tiempo derivativo, la cual tiene una acción en la respuesta del sistema de ser más rápida, cuyas desventajas son la amplificación de ruidos y saturación sobre el actuador. Esta acción derivativa no se usa sola ya que es funcional durante tiempo transitorios que se muestra en la ecuación 11 (Mazzone, 2002):

$$C_{pd}(t) = K_p \cdot e(t) + K_p \cdot T_d \cdot \frac{d e(t)}{dt} \quad (11)$$

Una acción de control derivativa se agrega a un controlador proporcional para obtener un controlador de alta sensibilidad, es decir la velocidad del cambio del error y se logra que el error se vuelva alto de esta manera el error no se vuelve de estado estacionario, agregando un amortiguamiento al sistema permitiendo una precisión estable al proceso (Mazzone, 2002).

2.2.4 Proporcional Integral Derivativo (PID)

La acción combinada de proporcional integral y derivativo logra un control eficaz de alto nivel con la cual la ecuación 12 resultante que se obtiene es:

$$U(t) = K_p e(t) + K_i \int_0^t e(t) dt \quad (12)$$

La función de transferencia resultantes se muestra en la siguiente ecuación:

$$CPID(t) = K_p \cdot \left(1 + \frac{1}{T_{ids}} + T ds \right) \quad (13)$$

2.3 Sistemas de control Difusos

Los sistemas difusos son una extensión de la inteligencia artificial y tiene dos tipos de estructuras una es el control directo y otra el control. (González Macías Bryan Xavier, 2018).

La capacidad de los sistemas difusos para describir la operación de un sistema complejo mediante reglas simples expresadas en lenguaje natural ha motivado su aplicación a numerosos problemas de automatización y control industrial, dando lugar a lo que ha venido en denominarse controladores difusos (Santiago Sánchez-Solano Alejandro J. Cabrera, 2015) , (Huilcapi, et al, 2023)

La estructura de un control difuso está definida por la parte de Fusificación, sistema de inferencia, Desfusificación y la base de conocimientos como se muestra en la figura 9.

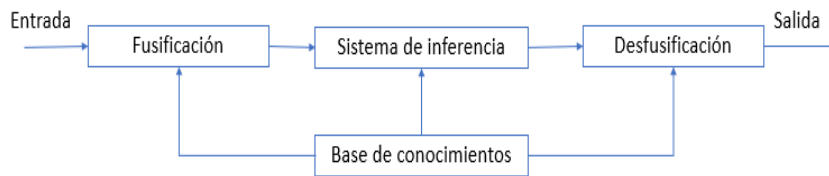


Figura 6: Diagrama de control Difuso *(Los Autores)*

El selector es un elemento de conmutación monoestable con dos o más posiciones que el operador puede elegir, y que para su accionamiento suelen llevar una palanca o llave giratoria que puede ser extraíble.

2.3.1 Desfusificación

Es el método de procesar un valor cuantificable en Lógica clásica, tomada de un conjunto difusos y sus correspondientes grados de membresía.

Poseen una serie de reglas que transforman un valor numérico de variables en un resultado difuso, el cual depende de las reglas de membresía dadas en el sistema (Leekwijck, 1999), (Huilocapi, et al, 2022)

2.3.2 Fusificación

Es el método de convertir valores reales a variables difusas que son dadas por los grados de pertenencia de cada una de las variables de entrada previamente definidos utilizados en los conjuntos difusos. (Ramos, 2008)

2.3.3 Inferencia

Es la relación que existe entre los conjuntos difusos de entrada y salida que son representadas por las reglas que gobiernan el sistema, se usa a la par con la base de conocimientos para generar reglas mediante casos o condiciones (Ramos, 2008).

2.4 Lattepanda

Lattepanda es un embebido que consta con un procesador Cherry Trail Atom y un microprocesador atmega32ua con puertos de salidas GPIO, Lattepanda posee edición completa de Windows 10, que incluye potentes herramientas como Visual Studio, NodeJS, Java, producción, y mucho más. Utilización de las API existentes, puede desarrollar sus propios proyectos de software y hardware en Lattepanda como lo haría en un PC normal – C #, JavaScript, Ruby y así sucesivamente (Electronilab, 2018).

En la figura 11 se aprecia al embebido lattepanda y sus características:

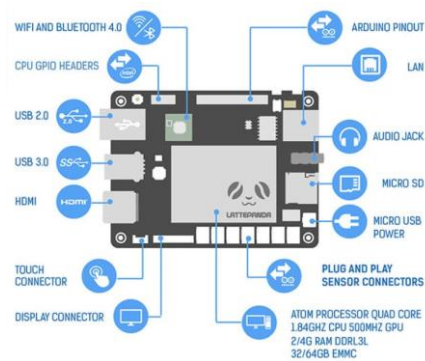


Figura 7: Lattepanda (DfRobot, 2017)

2.4.1 Lattepanda Alpha

Lattepanda Alpha es la versión actualidad del Lattepanda, está compuesta de un procesador Core m3 de séptima generación de Intel junto con 8 GB LPDDR3 y 64 GB de almacenamiento interno por medio de tarjeta eMMC, la versión Delta consta de un procesador Intel Celeron N4100 cuenta con un coprocesador Arduino, 3 puertos USB 3.0, uno Tipo-C, salida HDMI, Wi-Fi y Bluetooth 4.0 GB LPDDR4 y 32 GB eMMC ambas con graficas integradas Intel HD 615. (Chichande Solano, Daniel Jean Pierre, Matamoros Galán, Ingrid Maylin, 2021)

En la figura 12 se aprecia al embebido lattepanda alpha y sus características:

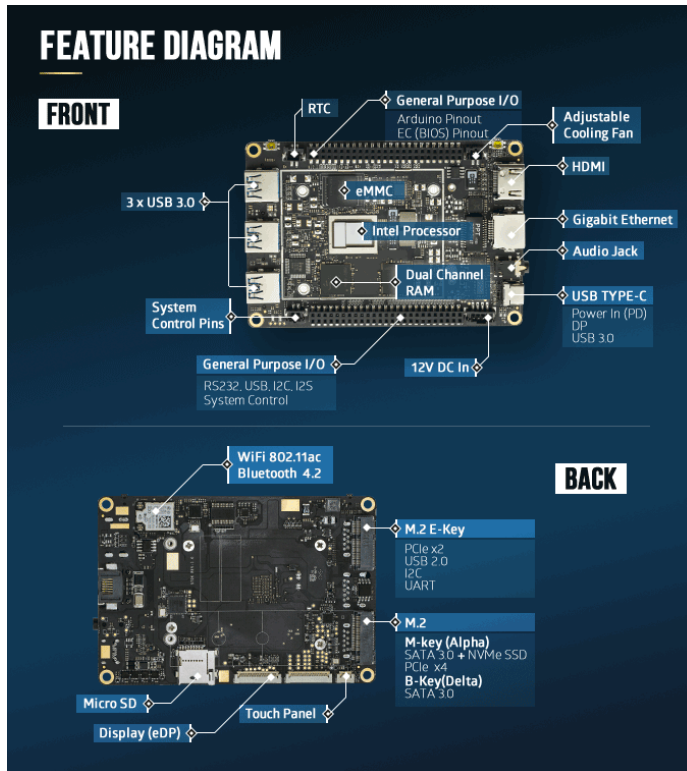


Figura 8. LattePanda Alpha (DfRobot, 2017)

2.4.1 Atmel 32UA

El Atmel 32ua es un microcontrolador el cual tiene una arquitectura Harvard, posee 8 bits, 32Kb de memoria Flash y un empaquetado VFQFN-44 (Microchip, 2016).

Posee 20 pines de entrada y salida digitales de los cuales 7 pueden ser usados como salidas PWM y 12 como entradas análogas. (Arduino, 2008).

En la figura 13 se muestra el Atmel 32ua con sus respectivas salidas GPIO en la lattepanda Alpha.

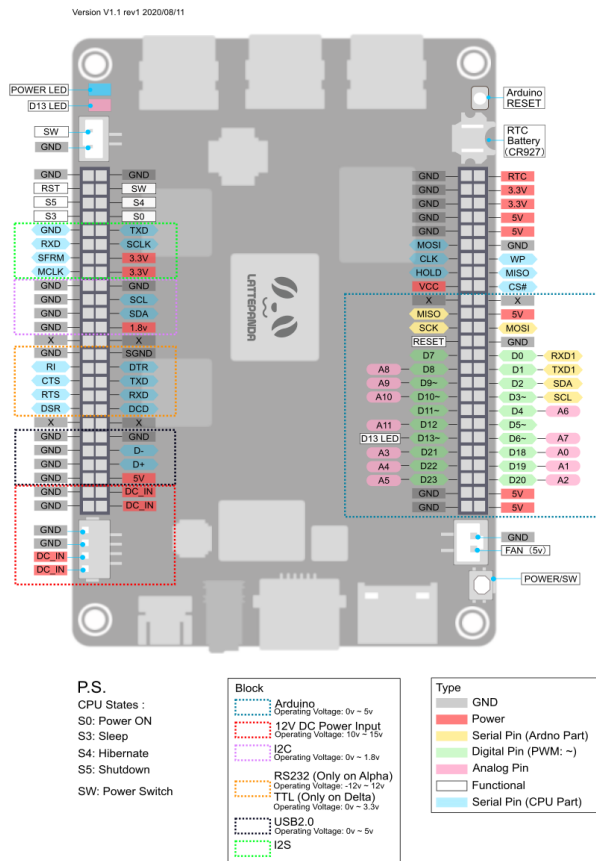


Figura 9. Atmel 32UA (Dorf & Bishop, 2005)

2.5 Software de instrumentación LabVIEW

LabView es un software de instrumentación virtual el cual trabaja con un lenguaje gráfico, ya que nos permite diseñar interfaces de usuario mediante una consola interactiva basada en software (VERGARA, 2012).

2.6 Motores de Corriente Continua

Una definición básica de motor DC, podría estar en la función que este realiza, la cual es convertir la energía eléctrica en energía mecánica rotacional y lo hace siguiendo los fenómenos magnéticos (Poznyak, 2005).

Los motores de corriente continua se manejan mediante la aplicación de voltaje entre sus bornes los cuales comenzaran a girar en un sentido, si se desea cambiar el giro se invertirá la conexión a los bornes (Calderon, 2004).

Este motorreductor que se muestra en la Figura 14 y 15 es un potente motor de CC de 12V con caja de engranajes metálicas 50: 1 y un codificador de cuadratura integrado que proporciona una resolución de 64 conteos por revolución del eje del motor, que corresponde a 3200 conteos por revolución del eje de salida de la caja de engranajes (Pololu, 2006).



Figura 10. Motor DC (Pololu, 2006)

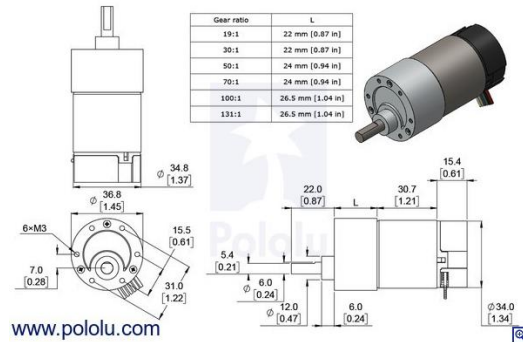


Figura 11. Motor DC (Pololu, 2006)

La tabla 1 contiene los tipos de relación, torque y velocidad de cada motor.

Caja engranajes	Velocidad	Torque
1:1	11,000 RPM	5 oz-in
19:1	500 RPM	84 oz-in
30:1	350 RPM	110 oz-in
50:1	200 RPM	170 oz-in
70:1	150 RPM	200 oz-in
100:1	100 RPM	220 oz-in
131:1	80 RPM	250 oz-in

Tabla 1: Tabla tipos de motores Pololu (Pololu, 2006)

2.7 Codificadores Rotativo (Encoder)

En aplicaciones de automatización, los codificadores rotativos sirven como medidores de ángulo, posición, velocidad y aceleración. Los codificadores rotativos transforman un valor de rotación mecánica en una señal eléctrica que puede ser procesada por contadores, tacómetros, controladores lógicos y Pc industriales. Trabajan con un disco de vidrio o de plástico con campos alternos oscuros y claros, con una fuente de luz en un lado y un sensor sensible a la luz en el otro. A medida que el disco gira, la fuente de luz se bloquea de manera alternativa y se revela al sensor. Siempre que la fuente de luz llegue al sensor, el codificador envía un impulso eléctrico que puede ser interpretado por un controlador. El pulso finaliza cuando un campo oscuro del disco no permite el paso de la luz. La rotación del disco produce una salida de pulso de onda cuadrada. (Principles of rotary encoders, 2004) como se muestra en la figura 16.

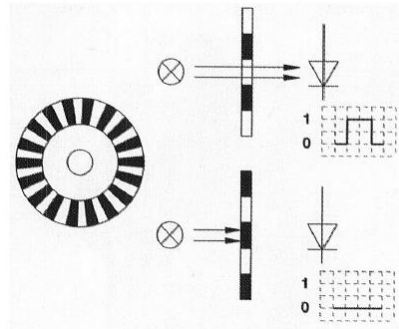


Figura 12. Funcionamiento de disco rotativo del Encoder (Principles of rotary encoders, 2004)

2.7.1 Encoder Incrementales

Un Encoder, también denominado generador de pulsos, es un elemento electromecánico, es decir relaciona la energía eléctrica con la mecánica, que transforma el movimiento, es decir energía mecánica, en una señal eléctrica, que pueden ser pulsos, ondas, entre otras. Esta señal eléctrica puede ser leída por algún tipo de dispositivo con el fin de controlar este movimiento, los Encoder incrementales tienen en su forma más sencilla un disco transparente con una serie de marcas opacas colocadas radialmente y a la misma distancia entre sí, de un sistema de iluminación en el que la luz es colimada de forma correcta, y de un dispositivo fotorreceptor. El eje, cuya posición se quiere calcular, va acoplado al disco transparente. Para que a medida que el eje gire se irán generando pulsos en el receptor cada vez que la luz atraviese, llevando una cuenta de estos pulsos se podrá conocer la posición del eje. (Barrientos, Antonio, Peñín, Luis Felipe, & Balag, 2007) como se muestra en la figura 17.

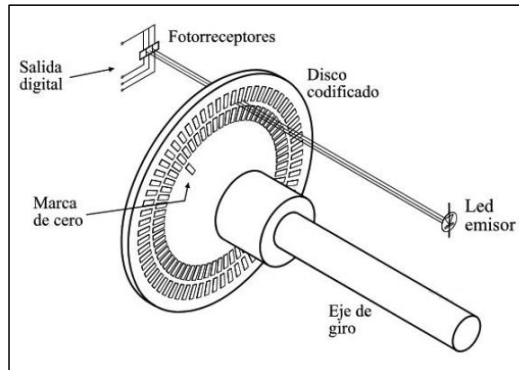


Figura 13. Disposición de un Encoder incremental (Barrientos, Antonio, Peñín, Luis Felipe, & Balag, 2007)

Los pulsos que se producen como resultado del giro, no indican por sí mismos del sentido de éste. Para poder conocer esta información, se coloca una segunda franja de marcas desplazada de la primera, de manera que el tren de pulsos que con ella se genere tenga un desfase de 90° con respecto al creado por la primera franja. Es así como mediante el uso de un biestable de tipo D, en el que la salida mantiene el valor de la entrada leída durante el flanco positivo del reloj, se puede obtener una señal adicional (Q) que indicara cuál es el sentido de giro, y que actúe sobre el contador correspondiente indicando que incrementa o decremente la cuenta que se esté realizando. (Barrientos, Antonio, Peñín, Luis Felipe, & Balag, 2007).

Debido a que el contador es incremental, se deberá indicar cuál es el cero, para lo cual se incorpora una marca de referencia adicional (Z) única en todo el trazado del disco, que indicara que se ha dado una vuelta completa y que, por tanto, se debe reinicializar el contador. (Barrientos, Antonio, Peñín, Luis Felipe, & Balag, 2007) como se muestra en la figura 18.

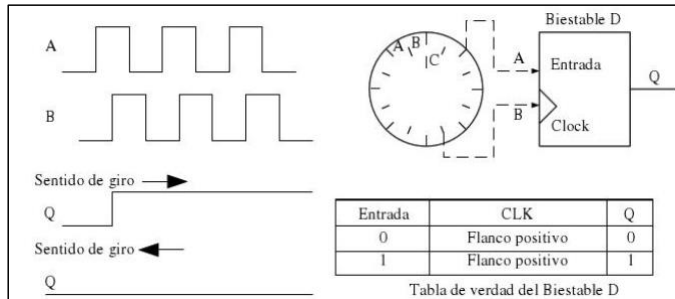


Figura 14. Salidas A y B de un Encoder y valor (Q) del Biestable D según su sentido de giro (Barrientos, Antonio, Peñín, Luis Felipe, & Balag, 2007)

2.7.2 Encoder absolutos

Un Encoder absoluto genera a la salida una palabra binaria en la que está codificada la posición absoluta que se desea medir. Soluciona las desventajas del codificador incremental, pero su precio es muy superior, entrega en su salida una señal digital codificada correspondiente a la posición de un disco móvil con respecto a una referencia interna. Para ello, el disco móvil se divide en sectores (2^n , donde n es el número de bits del codificador) y en cada sector se codifica el valor de su posición con zonas claras (a las que se asigna el valor binario 0) y oscuras (a las que se asigna el valor binario 1) situadas de acuerdo con el código utilizado (Gray, binario natural...). A diferencia de los codificadores incrementales, hay varias pistas con zonas diferenciadas y están agrupadas de tal forma que el sistema de lectura obtiene directamente en cada posición del elemento móvil el valor codificado de su posición. Cada pista simboliza un bit de salida, siendo la pista interna la correspondiente al bit de mayor peso. La salida se lee en paralelo para generar una representación binaria de la posición angular. (Granda Miguel, Mercedes, & Mediavilla Bolado, 2015) como se muestra en la figura 19.

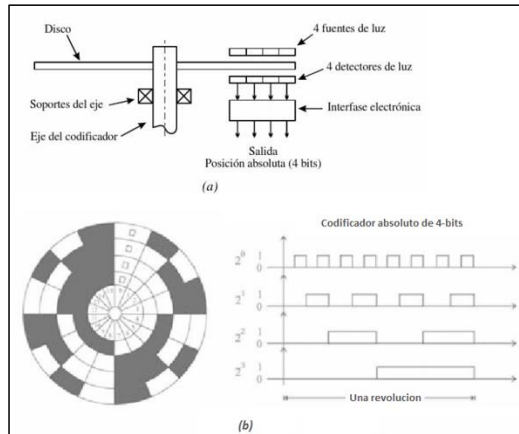


Figura 15. Codificador absoluto binario de 4bits. (a) Construcción Interna. (b) Señales de salida

(Granda Miguel, Mercedes, & Mediavilla Bolado, 2015)

2.8 Fuente de alimentación.

Una fuente de alimentación convierte la corriente alterna (AC) en una forma continua de energía que los componentes del ordenador necesitan para funcionar (Navas, 2017) como se muestra en la figura 20.

Las fuentes de alimentación utilizan la tecnología de conmutación para convertir la entrada de corriente alterna a voltajes de corriente continua más bajos. Los voltajes más usados son 3.3, 5 y 12 voltios (Navas, 2017).



Figura 16. Fuente de alimentación (Brico Geek, 2020).

2.9 Pulsadores y selectores.

Un pulsador es un interruptor o switch cuya función es permitir u obstaculizar el paso de la corriente eléctrica. Existen pulsadores normalmente cerrado y normalmente abierto (Shoptronica, 2019) como se muestra en la figura 21.



Figura 17. Variedades de pulsadores (Hoyos, 2020)

2.10 Internet de las cosas

El Internet de las cosas (IoT) es el mecanismo que permite la conexión de elementos físicos del día a día a Internet, desde los más simples como un foco hasta equipos médicos y sistemas caudales de ciudades inteligentes, son de 2 tipos los dispositivos de IoT los interruptores aquellos que envían las instrucciones a un objeto y los tipos sensores que reúnen los datos y los envían a otro destino. (Red Hat, 2023)

2.10.1 Características del Internet de las cosas

- **Accede a la tecnología de sensores de bajo costo y potencia:** La accesibilidad de sensores confiable permiten a más fabricantes el uso de la tecnología IoT.
- **Conectividad:** Las conexiones de sensores a la nube y demás cosas para envío de datos de forma eficiente ha sido permitido gracias a el número de protocolos de red para Internet.
- **Plataformas de informática en la nube:** La alta disponibilidad e incremento de plataformas en la nube permite a los consumidores y

empresas el acceso a infraestructura necesaria para poder escalar sin una administración completa.

- **Aprendizaje automático y analítica:** Las empresas pueden reunir información de manera fácil y rápida, gracias a los avances en aprendizaje automático en conjunto con el acceso a gran cantidad de datos en la nube.
- **Inteligencia artificial (IA) conversacional:** Los avances en las redes neuronales han llevado el procesamiento del lenguaje natural (PLN) a los dispositivos IoT (como los asistentes personales digitales Alexa, Cortana y Siri) y los han hecho atractivos, asequibles y viables para uso doméstico. (Oracle, 2022)

2.11 Módulo de entrenamiento

En la actualidad existe la necesidad de reforzar los laboratorios con un equipo el cual ayude al estudiante a implementar recursos prácticos, motivándolos al uso de nuevas tecnologías como el internet de las cosas, permitiendo al estudiante de la carrera de Ingeniería Electrónica y Automatización, implementar controladores inteligentes integrándose con aplicaciones alojadas en la nube, la finalidad de este Proyecto es contribuir a diseñar e implementar una herramienta de trabajo, que incluya un manual para el desarrollo del proyecto relacionado con el estudio en el campo práctico de la Electrónica, específicamente a las asignaturas de la Automatización. Los beneficiarios directos a largo o mediano plazo de este proyecto serán los estudiantes de la carrera de Ingeniería Electrónica y especialidades afines, debido a que este nuevo módulo didáctico como se muestra en la figura 22 se implementó para el laboratorio de Sensores y Actuadores para las prácticas correspondientes.



Figura 18. Módulo de entrenamiento (El Autor)

3: MARCO METODOLÓGICO

En este capítulo, se describe la metodología empleada en el diseño e implementación del algoritmo difuso y PID para el control de velocidad en un motor de corriente continua en sistemas embebidos LattePanda con monitoreo en plataforma IoT. Se presentan los pasos y enfoques utilizados para lograr los objetivos del proyecto.

3.1 Recopilación de información y revisión bibliográfica

En esta etapa, se lleva a cabo una extensa revisión de la literatura científica y técnica relacionada con el control de velocidad en motores de corriente continua, algoritmos difusos, controladores PID, sistemas embebidos LattePanda, IoT y aplicaciones similares. Se consultan libros, artículos científicos, conferencias y trabajos previos en el área, con el objetivo de obtener una comprensión profunda de los fundamentos teóricos y las mejores prácticas en el campo.

3.2 Diseño del algoritmo difuso

En esta fase, se diseña el algoritmo difuso para el control de velocidad en el motor de corriente continua. Se definen las variables de entrada y salida lingüísticas, así como las funciones de membresía correspondientes que describen el comportamiento de estas variables en el contexto del control de velocidad. Se establecen las reglas difusas basadas en el conocimiento experto y la experiencia en el campo. Además, se definen los métodos de inferencia y defuzzificación que se utilizarán para obtener una salida nítida a partir de las reglas difusas.

3.3 Implementación del controlador PID

En esta etapa, se lleva a cabo la implementación del controlador PID como una alternativa al algoritmo difuso. Se seleccionan los parámetros del controlador, como las constantes proporcional, integral y derivativa, y se realiza un ajuste adecuado para obtener un rendimiento óptimo del controlador. Se utiliza un lenguaje de programación apropiado en este caso el análisis en MATLAB y la presentación en LabView, para implementar el controlador PID y se realizan pruebas preliminares para verificar su funcionamiento.

3.4 Integración en sistemas embebidos LattePanda

En esta fase, se integra tanto el algoritmo difuso como el controlador PID en el sistema embebido LattePanda. Se desarrolla el software necesario para la comunicación con los componentes del motor de corriente continua, como los sensores de velocidad y los actuadores de control. Se asegura la compatibilidad y la correcta interacción entre el hardware del sistema embebido y el software implementado para el control de velocidad.

3.5 Conexión y monitoreo en plataforma IoT

En esta etapa, se establece la conexión y comunicación entre el sistema embebido LattePanda usando LabView y la plataforma IoT seleccionada. Se configuran los protocolos de comunicación, como MQTT o HTTP, para permitir la transmisión de datos entre el sistema embebido y la plataforma IoT. Se desarrolla una interfaz de usuario en la plataforma IoT para el monitoreo en tiempo real de la velocidad del motor y otros parámetros relevantes, lo cual permitirá a los usuarios supervisar y controlar el motor desde cualquier ubicación remota.

3.6 Validación y pruebas

En esta etapa final, se llevan a cabo pruebas exhaustivas del sistema implementado. Se realizan pruebas funcionales para evaluar el rendimiento del algoritmo difuso y el controlador PID en términos de respuesta, estabilidad y seguimiento de la velocidad deseada en diferentes condiciones de operación. Se comparan los resultados obtenidos con los objetivos establecidos al inicio del proyecto y se realizan ajustes adicionales si es necesario. También se realizan pruebas de conectividad y monitoreo en la plataforma IoT para verificar la correcta transmisión de datos y la funcionalidad del monitoreo remoto.

3.7 Diseño e Implementación de módulo didáctico

Para el diseño del módulo se optó por utilizar un maletín metálico con dimensiones de 35cm de anchura, 45 cm de largo, 18 cm de altura en el interior del maletín se diseñó una lámina metálica donde ubicar los periféricos tales como borneros para conexiones

de entradas y salidas, pulsadores, luces pilotos, potenciómetros, el motor acoplado al encoder, botones de encendido, pantalla HMI táctil y un host USB. En las figuras 23, 24 , 25 se puede observar el conexcionamiento e implementación del módulo didáctico.



Figura 19. Conexiones módulo entrenamiento

(Los Autores)



Figura 20. Conexiones módulo entrenamiento


(Los Autores)

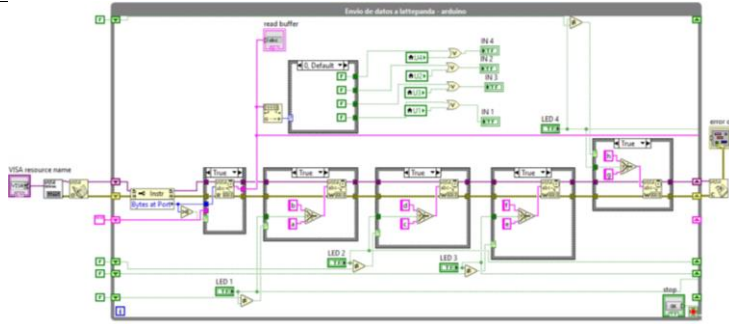


Figura 21. Diseño e implementación del módulo de entrenamiento
(El Autor)

4: GUÍA DE PRÁCTICAS DE LABORATORIO

Práctica 1: Programación de señales discretas analógicas de entrada y salida en LabView con supervisión y control en plataforma IOT

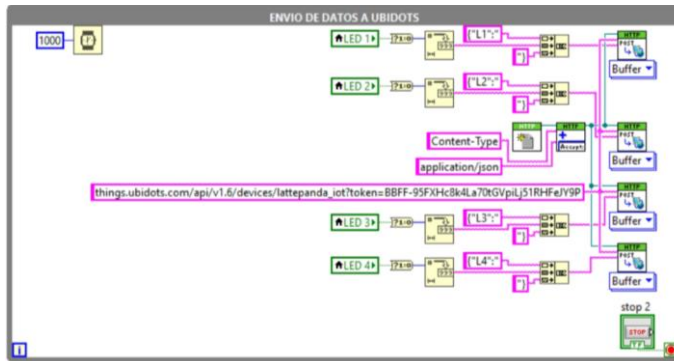
		GUÍA DE PRÁCTICA DE LABORATORIO
CARRERA: Ingeniería Electrónica		ASIGNATURA: Redes Industriales
No.: 1	TÍTULO: Programación de señales discretas analógicas de entrada y salida en LabView con supervisión y control en plataforma IOT	
OBJETIVOS: <ul style="list-style-type: none"> • Diseñar una conexión entre lattepanda y LabView. • Programar un aplicativo para la adquisición de valores analógicos con el microcontrolador Atmel 32UA. • Programar el envío de valores analógicos de salida simulados con el Ubidots con LabView. • Implementar la comunicación serial para el envío y recepción de datos al lattepanda y Arduino leonardo integrado. 		
INSTRUCCIONES: <ol style="list-style-type: none"> 1. Analizar la solución propuesta de la práctica#2 ubicada en los anexos de la memoria técnica. 2. Preparar el módulo didáctico mediante las conexiones establecidas en el presente documento. 3. Realizar las conexiones eléctricas y verificar las mismas previo a energizar el módulo didáctico. 		
DESARROLLO <ol style="list-style-type: none"> 1. Programación en LabView: <p>El siguiente bloque permite abrir el puerto serial desde LabView para enviar y recibir datos desde Arduino a el programa de adquisición Labview.</p> <p>Se utilizan los bloques Select que permite obtener una señal booleana y obtener valores si condición se cumple en verdadero o falso.</p> <p>Estos valores de cadena son escritos a Arduino para realizar las acciones de encendido de señales digitales.</p> 		



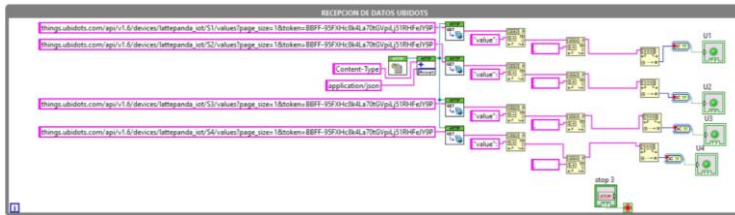
El envío de datos a plataforma IoT de UBIDOTS se realiza con envío de datos usando el método HTTP POST que envía datos a los recursos de la plataforma IoT por medio de la nube.

Estos datos compartidos se reciben con el formato de texto tipo .JSON que es un formato de tipo diccionario que usa subconjuntos de notación literal.

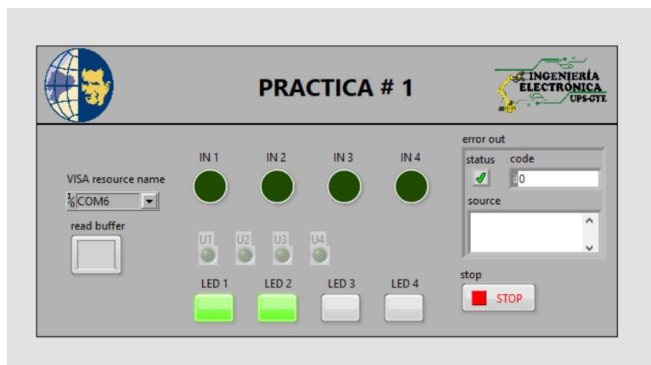
Para construir el formato de envío, se usan bloques de concatenación de strings en LabView, que construyen el formato obteniendo individualmente valores de elementos booleanos que en este caso son los indicadores de esta práctica, y se une el texto a la URL del Dashboard IOT.



El recibo de datos desde UBIDOTS se da con peticiones HTTP GET al servidor de la plataforma IoT, quien trae una URL con los datos de la petición, en este caso los datos son S1 hasta S4 que son los datos de los Switches de encendido remoto, y se toma su valor para luego con conversiones de string a booleanos se logra encender los indicadores cumpliendo la función de encendido remoto desde plataforma de IoT.



Interfaz de la práctica 1:



En la programación de Arduino se declaran las variables a utilizar y los puertos de los indicadores, switches y pulsadores, así como asignación de puertos de entradas y salidas.



```
1 //Declaración de pines de entrada digital
2 const int entrada_1 = 2; //PINO-2
3 const int entrada_2 = 3; //PINO-3
4 const int entrada_3 = 4; //PINO-4
5 const int entrada_4 = 5; //PINO-5
6 //Declaración de pines de salida digital
7 const int salida_1 = 7; //PINO-7
8 const int salida_2 = 8; //PINO-8
9 const int salida_3 = 9; //PINO-9
10 const int salida_4 = 10; //PINO-10
11
12 char inChar;
13 int q;
14 void setup() {
15
16   Serial.begin(9600);
17
18   pinMode(entrada_1, INPUT);
19   pinMode(entrada_2, INPUT);
20   pinMode(entrada_3, INPUT);
21   pinMode(entrada_4, INPUT);
22
23   pinMode(salida_1, OUTPUT);
24   pinMode(salida_2, OUTPUT);
25   pinMode(salida_3, OUTPUT);
26   pinMode(salida_4, OUTPUT);
27
28 }
29
30 void loop() {
31   inChar = Serial.read();
32   q = inChar - '0';
33   digitalWrite(salida_1, q);
34   digitalWrite(salida_2, q);
35   digitalWrite(salida_3, q);
36   digitalWrite(salida_4, q);
37 }
```

Las condiciones para esta práctica se toman de la lectura del puerto serial, estas reciben el comando enviado desde LabView y se comparan con el dato que se desea obtener para tomar las decisiones de encendido de los indicadores.



```
28
29
30 void loop() {
31   while (Serial.available()) {
32     inChar = Serial.read();
33     if (inChar == 'a')
34     {
35       digitalWrite(salida_1, HIGH);
36     }
37     else if (inChar == 'b')
38     {
39       digitalWrite(salida_1, LOW);
40     }
41     else if (inChar == 'c')
42     {
43       digitalWrite(salida_2, HIGH);
44     }
45     else if (inChar == 'd')
46     {
47       digitalWrite(salida_2, LOW);
48     }
49     else if (inChar == 'e')
50     {
51       digitalWrite(salida_3, HIGH);
52     }
53     else if (inChar == 'f')
54     {
55       digitalWrite(salida_3, LOW);
56     }
57   }
58 }
```


Por último, la variable “c” envía los datos por puerto serial de las entradas digitales, usando una función que permite obtener el valor del switch accionado y envía ese dato para que se visualicen los indicadores en LabView y en la plataforma IoT.

```

17.2
49:   while (1) {
50:     if (digitalRead(2) == HIGH) {
51:       digitalWrite(3, LOW);
52:     } else if (digitalRead(2) == LOW) {
53:       digitalWrite(3, HIGH);
54:     }
55:     if (digitalRead(3) == HIGH) {
56:       digitalWrite(4, LOW);
57:     } else if (digitalRead(3) == LOW) {
58:       digitalWrite(4, HIGH);
59:     }
60:     if (digitalRead(4) == HIGH) {
61:       digitalWrite(5, LOW);
62:     } else if (digitalRead(4) == LOW) {
63:       digitalWrite(5, HIGH);
64:     }
65:     if (digitalRead(5) == HIGH) {
66:       digitalWrite(6, LOW);
67:     } else if (digitalRead(5) == LOW) {
68:       digitalWrite(6, HIGH);
69:     }
70:     c = digitalRead(digitalPinToAnalog(11)) * 10 + digitalRead(10);
71:     Serial.println(c);
72:     delay(100);
73:   }

```

ACTIVIDADES POR DESARROLLAR

1. Definir la velocidad de transmisión del puerto serial.
2. Elaborar la programación para realizar la adquisición de datos por el puerto serial y comunicarse con el LabView
3. Realizar el cableado en la maleta didáctica.

RESULTADOS:

- Programación en el ide de Arduino para la declaración de puertos analógicos como entrada.
- Conexión entre la plataforma Ubidots y LabView mediante protocolo http para el control de las salidas analógicas virtuales.
- Envío de datos a plataforma Ubidots y visualización en navegador web o aplicación móvil.


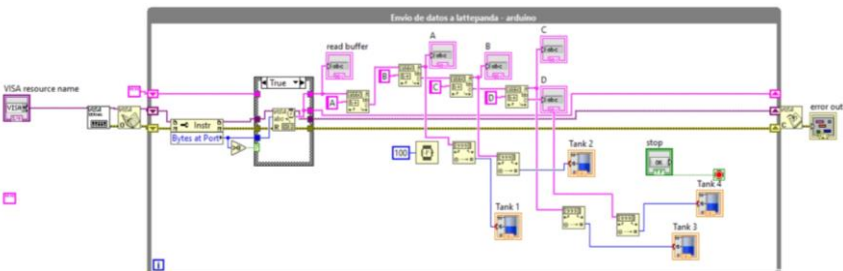
CONCLUSIONES:

- En la comunicación con la plataforma Ubidots con LabView se debe dirección a la dirección URL de la plataforma con el token y nombre del dispositivo.
- Para la comunicación entre LabView y lattepanda declarar la misma velocidad de transmisión en ambos programas.

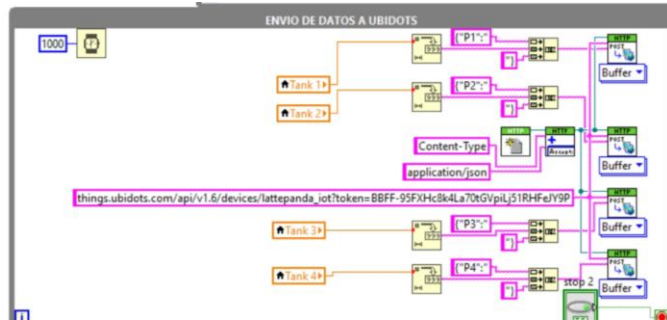
RECOMENDACIONES:

- Considerar el correcto funcionamiento de la red para poder acceder a Ubidots se necesita una red con acceso a internet.
- Realizar las conexiones descritas en el presente documento.

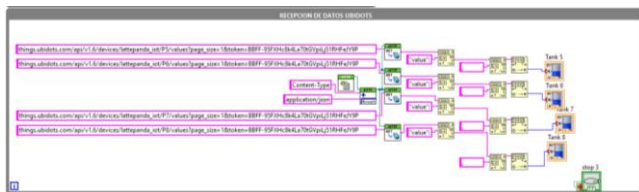
Práctica 2: Diseño de un controlador PID clásico para control de velocidad de motor con supervisión y control en plataforma IOT

 UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR		GUÍA DE PRÁCTICA DE LABORATORIO
CARRERA: Ingeniería Electrónica		ASIGNATURA: Redes Industriales
No.: 2	TÍTULO: Diseño de un controlador PID clásico para control de velocidad de motor con supervisión y control en plataforma IOT	
OBJETIVOS: <ul style="list-style-type: none"> • Diseñar una conexión entre lattepanda y LabView. • Programar un control PID clásico aplicado al control de velocidad de un motor. • Encontrar la función de transferencia del sistema mediante la herramienta de Matlab. • Programar un método para la adquisición del valor del encoder y envío de valores entre lattepanda y LabView. 		
INSTRUCCIONES: <ol style="list-style-type: none"> 1. Analizar la solución propuesta de la práctica #2 ubicada en los anexos de la memoria técnica. 2. Encontrar la función de transferencia mediante análisis en lazo abierto con Matlab. 3. Realizar las conexiones eléctricas y verificar las mismas previo a energizar el módulo didáctico. 		
DESARROLLO <p>Para esta práctica se realiza la adquisición de datos desde Arduino a LabView para visualización de datos analógicos que son representados por medio de tanques de líquido en la interfaz LabView.</p> <p>Para ello se leen las señales como en la práctica #1, pero los datos son convertidos de cadena string a número que vienen acompañados de un comando que son letras, para poder identificar cual es el valor que se va a asignar en cada tanque, la programación se puede observar en la siguiente imagen:</p>		
		

Los datos de los valores de los tanques son enviados a UBIDOTS como se indicó en la práctica #1 con la diferencia que se debe colocar la referencia de los tanques con las variables P1 hasta P4 que corresponden a cada valor de los tanques.



Para recibir datos, se realiza el mismo procedimiento de la práctica #1 con la diferencia que el dato se convierte a dato de tipo entero para poder ser leído por los tanques indicadores de LabView, estos datos son recibidos desde la plataforma IoT UBIDOTS con el método HTTP GET



Interfaz en LabView:



Para la programación de Arduino, se declaran las variables analógicas a enviar a LabView, estas variables se realizan un escalado con la función “map”, la cual permite escalar un dato “x” a una salida “y” solo colocando los rangos de entrada y salida.

Los valores son enviados con un identificador de letra para recibirlos en LabView y luego deconstruir el arreglo en una sola variable leída.

```

1 //Declaracion de pines de entrada analogica
2 const int entrada_1 = A0; //D00-0
3 const int entrada_2 = A1; //D00-1
4 const int entrada_3 = A2; //D00-2
5 const int entrada_4 = A3; //D00-3
6
7 char lcharr;
8 int v1,v2,v3,v4;
9 void setup() {
10
11   Serial.begin(9600);
12
13 }
14
15 void loop() {
16   v1=map(analogRead(entrada_1),0,1023,0,100);
17   v2=map(analogRead(entrada_2),0,1023,0,100);
18   v3=map(analogRead(entrada_3),0,1023,0,100);
19   v4=map(analogRead(entrada_4),0,1023,0,100);
20   Serial.print("A"); Serial.print(v1);
21   Serial.print("B"); Serial.print(v2);
22   Serial.print("C"); Serial.print(v3);
23   Serial.print("D"); Serial.print(v4);
24   delay(200);
25 }

```

1. Definir la velocidad de transmisión del puerto serial.
2. Encontrar la función de transferencia mediante el análisis de lazo abierto con la herramienta de Matlab.
3. Realzar el cableado en la maleta didáctica.

RESULTADOS:

- Controlar la velocidad de un motor de corriente directa mediante un controlador PID clásico.
- Conexión entre la plataforma Ubidots y LabView mediante protocolo http para el control de las salidas analógicas virtuales.
- Envío de datos a plataforma Ubidots y visualización en navegador web o aplicación móvil.


CONCLUSIONES:

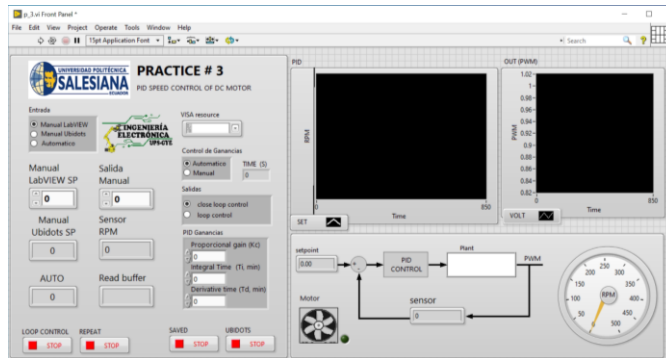
- Al realizar el control de lazo cerrado PID es necesario la obtención de la función de transferencia para utilizar la herramienta de Matlab.
- Para la comunicación entre LabView y lattepada declarar la misma velocidad de transmisión en ambos programas.

RECOMENDACIONES:

- Considerar el correcto funcionamiento de la red para poder acceder a Ubidots se necesita una red con acceso a internet.
- Realizar las conexiones descritas en el presente documento.

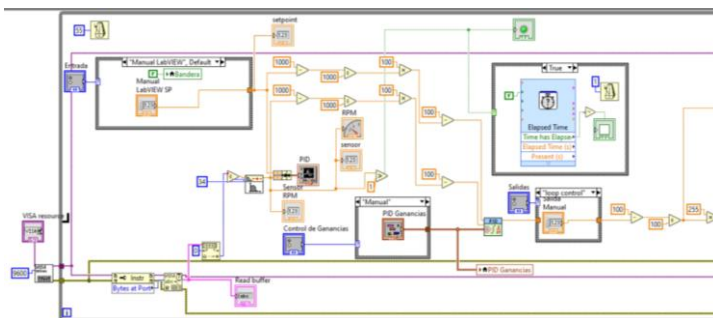
Práctica 3: Diseño de un controlador PID clásico para control de velocidad de motor con supervisión y control en plataforma IOT

		GUÍA DE PRÁCTICA DE LABORATORIO	
		CARRERA: Ingeniería Electrónica	ASIGNATURA: Redes Industriales
No.:	3	TÍTULO: Diseño de un controlador PID clásico para control de velocidad de motor con supervisión y control en plataforma IOT	
OBJETIVOS:			
<ul style="list-style-type: none"> • Diseñar una conexión entre lattepanda y LabView. • Programar un control PID clásico aplicado al control de velocidad de un motor. • Encontrar la función de transferencia del sistema mediante la herramienta de Matlab. • Programar un método para la adquisición del valor del encoder y envío de valores entre lattepanda y LabView. 			
INSTRUCCIONES:		1. Analizar la solución propuesta de la práctica#3 ubicada en los anexos de la memoria técnica.	
		2. Encontrar la función de transferencia mediante análisis en lazo abierto con Matlab.	
		3. Realizar las conexiones eléctricas y verificar las mismas previo a energizar el módulo didáctico.	
DESARROLLO			
Interfaz de usuario para control PID:			
En la pantalla para control PID, puede elegir el tipo de control que se desee usar, el control automático PID, control desde plataforma IoT UBIDOTS, y manual con LabView para controlar el Setpoint de la planta.			
En los indicadores se observan los valores de la planta, el sensor RPM, la lectura serial, el estado del motor y los controles de ganancias del PID, así como la entrada del Setpoint.			
Se tienen también las gráficas de la salida del control, la salida PWM para aplicar en Arduino.			

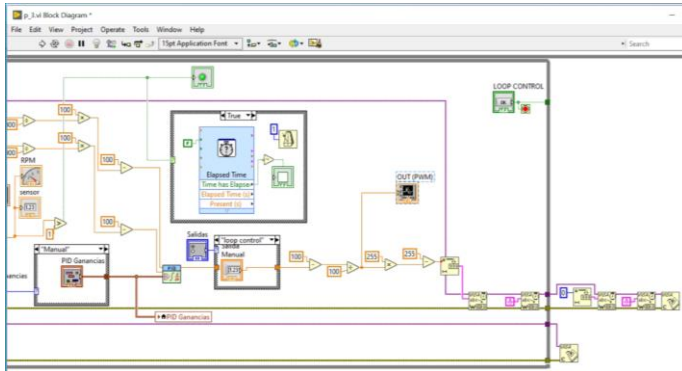


Los datos de entrada se obtienen desde Arduino con la lectura serial en LabView los mismos son escalados para obtener los valores deseados en el control PID.

El dato obtenido es el de revoluciones por minuto “RPM” del motor, será la variable de proceso, el SetPoint se obtiene de forma manual en la interfaz LabView, de forma remota desde UBIDOTS, de forma automática realizando una secuencia de control con el bloque de secuencia, y las ganancias del PID que son: ganancia proporcional, ganancia derivativa y ganancia integral.

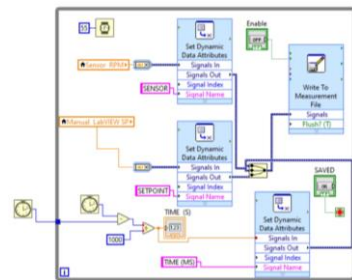


La segunda parte de la programación de control PID es la salida de control PID dada en señal PWM en dato decimal de 0 a 255, esto quiere decir que Arduino para su control PWM usa una resolución analógica de 8 bits que es enviada desde LabView por puerto serial con el control PID.

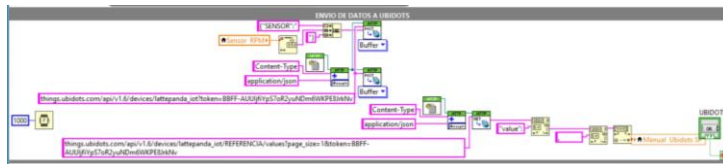


En esta sección se guardan los datos a Excel para luego ser usados como muestreo de datos y así poder estimar un modelo matemático.

Los datos de RPM, setpoint y el tiempo de ejecución se envían a un bloque que configura los atributos de los datos a guardar, luego son construidos en un arreglo para ser escritos en el bloque "Write to Measurement File".



Este bloque se utiliza para el envío de datos a UBIDOTS, con la diferencia que ahora se construye la URL con los tags del sensor RPM y se recibe el dato de operación manual desde UBIDOTS.



Este bloque permite crear una automatización de los valores del SetPoint para observar cómo cambia el control debido a las perturbaciones del sistema.



Para el código en Arduino, se debe declarar las variables a usarse en el sistema, el pin del sensor RPM, la salida analógica para control de motor, el envío de dato de sensor RPM a LabView, tiempos de interrupción, variables de entrada y salida y variables internas.

```

p.3 | Arduino 1.8.8
File Edit Sketch Tools Help
#_3_3
1 #include <math>math.h</math>
2 #include <math>math.h</math>
3 #include <math>math.h</math>
4 int RPM;
5 int Setpoint = 42;
6 #define Setpoint RPM;
7 #define Setpoint RPM;
8 #define Setpoint RPM;
9
10 #define SER1 10
11 #define SER2 11
12 void funcionRPM()
13 {
14   Setpoint++;
15 }
16 void setup() {
17   Serial.begin(9600);
18   pinMode(Setpoint, OUTPUT);
19   pinMode(SER1, OUTPUT);
20   pinMode(SER2, OUTPUT);
21   digitalWrite(SER1, 255);
22   digitalWrite(SER2, 0);
23   attachInterrupt(0, funcionRPM, RISING);
24   Setpoint = 0;
25   RPM = 0;
26   Setpoint = 0;
27 }
28 void loop()
  
```

Para que el control se active en Arduino, se lee desde el puerto serial el dato decimal para velocidad del motor en señal analógica.

Recordar que el control lo realiza LabView, se activa el sensor RPM realizando la conversión de datos, ya que el sensor es capaz de activar una señal cada vez que detecta en el motor una variación de posición, esta señal es leída y convertida en RPM para luego ser enviada con un comando por puerto serial de Arduino a LabView.


```
p_3 | Arduino 1.8.8
File Edit Sketch Tools Help
p_3
24 RPMcontador = 0;
25 RPM = 0;
26 tiempoAtras = 0;
27 }
28 void loop()
29 {
30   if (Serial.available()) {
31     data_labview = Serial.read();
32     datos += data_labview;
33     if (data_labview == '\n')
34     {
35       datoint = datos.toInt();
36       PWM = datoint;
37       analogWrite(motor1,PWM);
38       datos = "";
39     }
40     if (millis() - tiempoAtras >= 5) {
41       detachInterrupt(0);
42       RPM = (40 / 11) * 1000 / (millis() - tiempoAtras);
43       tiempoAtras = millis();
44       RPMcontador = 0;
45       Serial.println(RPM);
46       attachInterrupt(0, FuncionRPM, RISING);
47     }
48     delay(50);
49   }
50 }
51 }
```

Para estimar la función de transferencia en Matlab se debe realizar lo siguiente:

1. Importar los datos: Primero, se deben importar los datos experimentales a Matlab. Esto se puede hacer utilizando la función "importdata", que carga los datos desde un archivo de texto o de Excel en una variable en el espacio de trabajo de Matlab.

Import Data

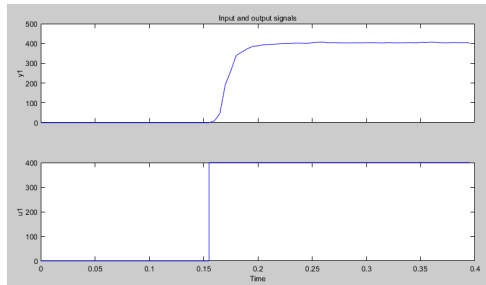
Data Format for Signals
Time-Domain Signals

Workspace Variable
Input: set
Output: rpm

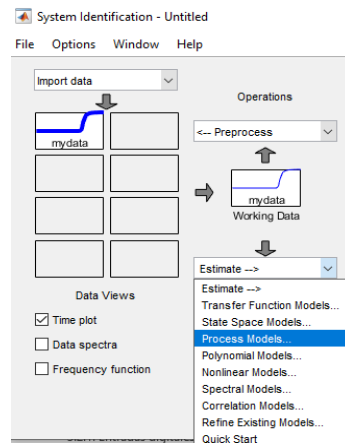
Data Information
Data name: mydata
Starting time: 0
Sample time: 0.005
More

Import Reset
Close Help

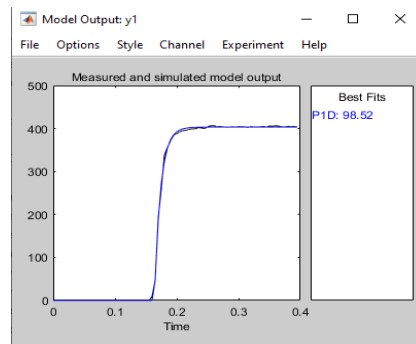
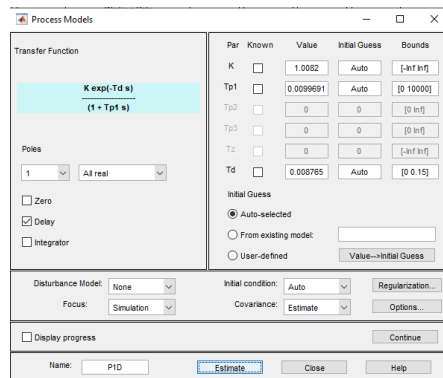
2. Visualizar los datos: Una vez que se han importado los datos, se pueden visualizar para asegurarse de que se han leído correctamente. Se pueden utilizar las funciones de gráficos de Matlab, como "plot" o "scatter", para visualizar los datos.



3. Identificar la función de transferencia: Después de visualizar los datos, se puede proceder a identificar la función de transferencia que mejor se ajuste a los datos. La identificación de la función de transferencia se puede hacer utilizando la función "Process Models" en Matlab. Esta función toma los datos experimentales como entrada y devuelve un objeto de datos identificado que se puede utilizar para identificar la función de transferencia.



4. Visualizar los resultados: Finalmente, se pueden visualizar los resultados utilizando las funciones de gráficos de Matlab. Se pueden trazar la respuesta del modelo ajustado junto con los datos experimentales para visualizar cómo se ajusta el modelo a los datos. Además de poder ajustar valores como delay, integrator o zero a la función de transferencia, para luego estimar y obtener el mejor porcentaje de respuesta del sistema.

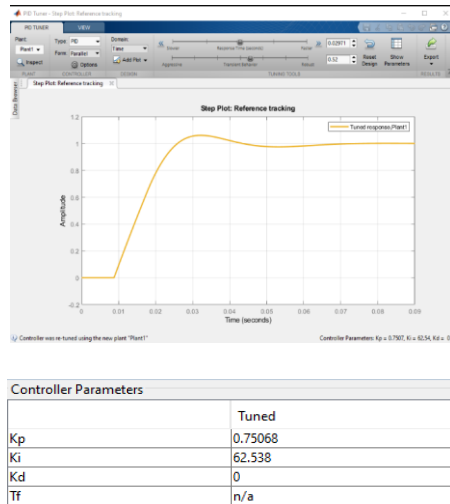


```
>> H=tf(P1D)
```

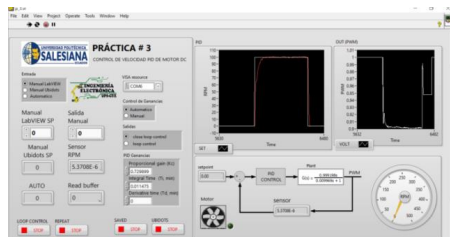
```
H =
```

```
From input "u1" to output "y1":
          1.008
exp(-0.00877*s) * -----
                0.009969 s + 1
```

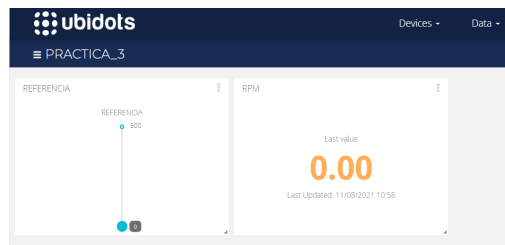
- Para obtener y ajustar los valores PID, se abre la herramienta PID Tuner, y luego ir ajustando la respuesta del sistema con respecto al tiempo y obtener los valores de K_p , K_i y K_d .



Finalmente, luego de ingresar los valores de K_p , K_i , K_d convertidos en ganancia proporcional, tiempo integral, y tiempo derivativo.



Desde UBIDOTS, se puede ajustar la referencia del sistema y las revoluciones por minuto RPM.



ACTIVIDADES POR DESARROLLAR
1. Definir la velocidad de transmisión del puerto serial.
2. Encontrar la función de transferencia mediante el análisis de lazo abierto con la herramienta de Matlab.
3. Realizar el cableado en la maleta didáctica.
RESULTADOS:
<ul style="list-style-type: none"> • Controlar la velocidad de un motor de corriente directa mediante un controlador PID clásico. • Conexión entre la plataforma Ubidots y LabView mediante protocolo http para el control de las salidas analógicas virtuales. • Envío de datos a plataforma Ubidots y visualización en navegador web o aplicación móvil.
CONCLUSIONES:
<ul style="list-style-type: none"> • Al realizar el control de lazo cerrado PID es necesario la obtención de la función de transferencia para utilizar la herramienta de Matlab. • Para la comunicación entre LabView y lattepanda declarar la misma velocidad de transmisión en ambos programas.
RECOMENDACIONES:
<ul style="list-style-type: none"> • Considerar el correcto funcionamiento de la red para poder acceder a Ubidots se necesita una red con acceso a internet. • Realizar las conexiones descritas en el presente documento.

Práctica 4: Diseño de un controlador difuso para control de velocidad de motor con supervisión y control en plataforma IOT.



GUÍA DE PRÁCTICA DE LABORATORIO

CARRERA: Ingeniería Electrónica

ASIGNATURA: Redes Industriales

No.: 4

TÍTULO: Diseño de un controlador difuso para control de velocidad de motor con supervisión y control en plataforma IOT.

OBJETIVOS:

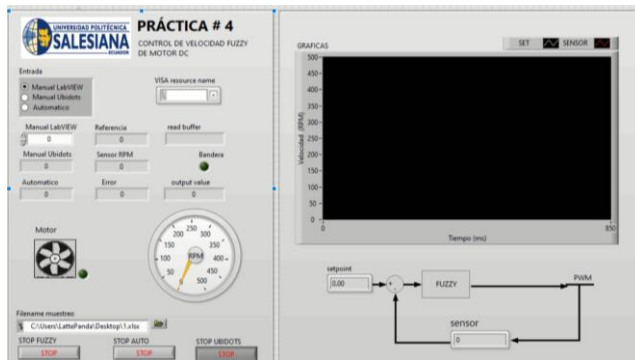
- Diseñar una conexión entre lattepanda y LabView.
- Programar un control basado en lógica difusa para el control de velocidad de un motor.
- Realizar el sistema difuso en la herramienta de LabView fuzzy designer.
- Programar un método para la adquisición del valor del encoder y envío de valores entre lattepanda y LabView.

INSTRUCCIONES:

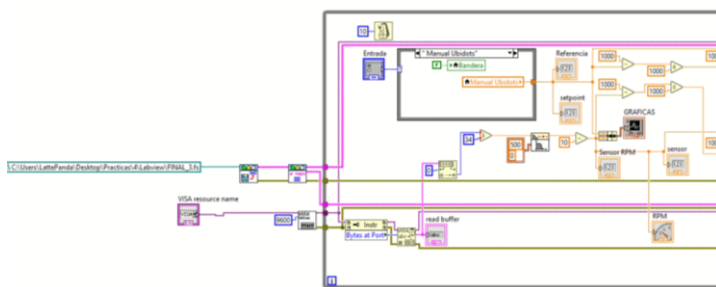
1. Analizar la solución propuesta de la práctica#4 ubicada en los anexos de la memoria técnica.
2. Utilizar la herramienta de LabView para la creación de sistemas difusos.
3. Realizar las conexiones eléctricas y verificar las mismas previo a energizar el módulo didáctico.

DESARROLLO

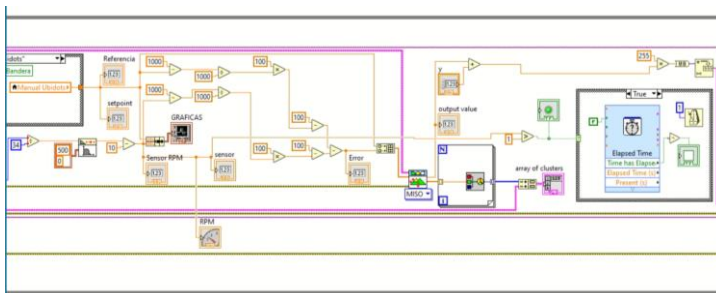
El control Fuzzy es una técnica de control que se basa en la lógica difusa para permitir que un sistema de control pueda tomar decisiones a partir de datos imprecisos o incompletos. En LabVIEW, se puede implementar un control Fuzzy utilizando el toolkit de lógica difusa.



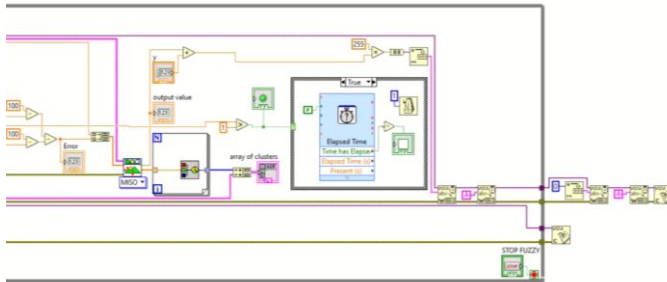
1. Implementar el controlador: se implementa el controlador utilizando un control "Fuzzy Control System". El control se encarga de realizar el cálculo del sistema de control utilizando el controlador difuso que se va a definir y las señales del sensor de RPM del motor. La salida del sistema de control se puede utilizar para controlar la velocidad del motor y mantenerla constante en un valor deseado donde esta es enviada por puerto serie hasta su recepción en Arduino.
2. El archivo donde se encuentran las reglas del controlador se define su ruta como se muestra en la imagen.
3. Las señales son escaladas para una mejor comprensión.
4. Se define un bloque booleano para definir de donde se podrá elegir el control del SetPoint, si desde UBIDOTS o desde LabView.



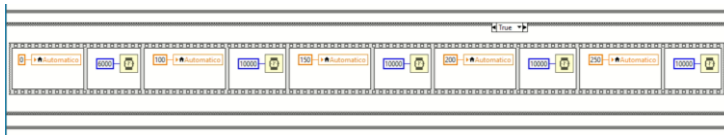
5. Los datos son presentados en una gráfica usando un constructor de datos enteros a datos de arreglos.



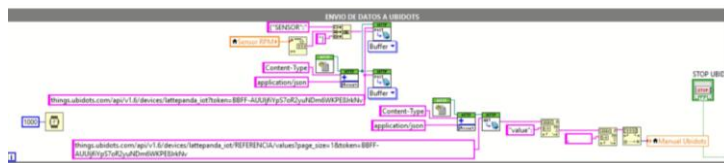
- Finalmente, los datos del control de motor en valor decimal de 0 a 255, son enviados por el puerto serial.



- Se crea un loop de secuencia donde los valores del SetPoint cambian para poder observar cómo se comporta el sistema.

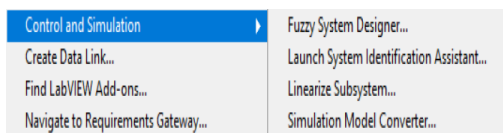


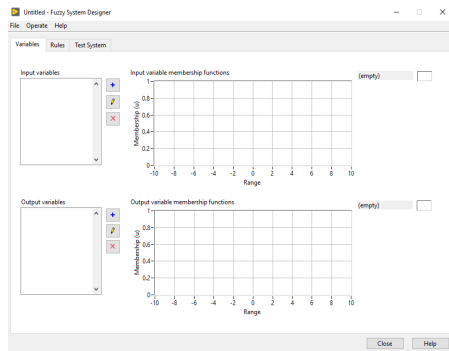
- Los datos son escritos a UBIDOTS, el sensor de RPM y la consigna de manual UBIDOTS para manejo remoto del SetPoint.



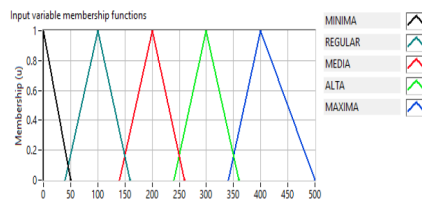
El diseño de un control difuso de velocidad de un motor con sensor de RPM en LabVIEW implica la definición de variables lingüísticas, la creación de reglas y la implementación de un controlador difuso. A continuación, se describen los pasos necesarios para implementar un control difuso de velocidad de un motor en LabVIEW:

Se abre la herramienta Fuzzy System Designer.

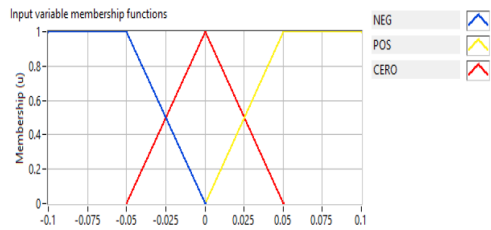




Definir las variables lingüísticas: En este paso, se deben definir las variables lingüísticas que se utilizarán en el sistema de control difuso. En este caso, se pueden definir dos variables lingüísticas: la velocidad del motor y el error de velocidad. Cada variable lingüística se divide en conjuntos difusos que representan los valores posibles de la variable, por ejemplo, "mínima", "Regular", "media", "alta" y "Máxima" para la velocidad del motor y "negativo", "cero" y "error" para el error de velocidad.



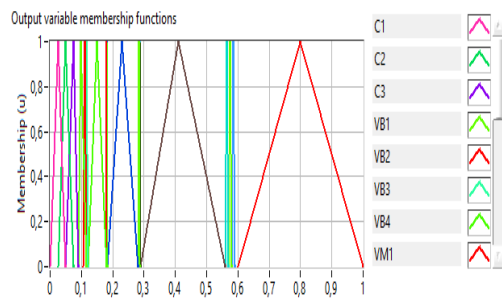
Nombre	Tipo	Puntos		
MINIMA	Triangular	0	0	50
REGULAR	Triangular	40	100	160
MEDIA	Triangular	140	200	260
ALTA	Triangular	240	300	360
MAXIMA	Triangular	340	400	500



Nombre	Tipo	Puntos		
NEGATIVO	Triangular	-100	-0.1	-0.05
CERO	Triangular	-0.05	0	0.05
ERROR	Triangular	0.05	0.1	100

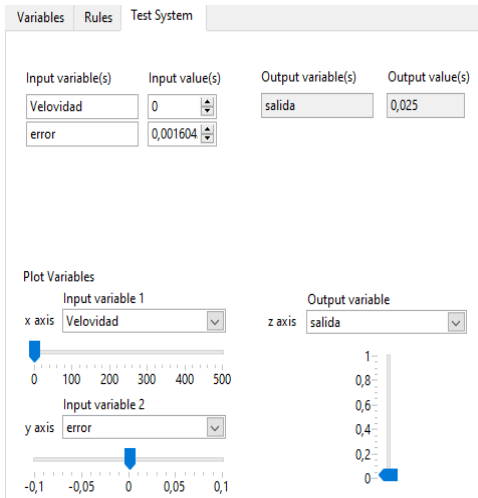
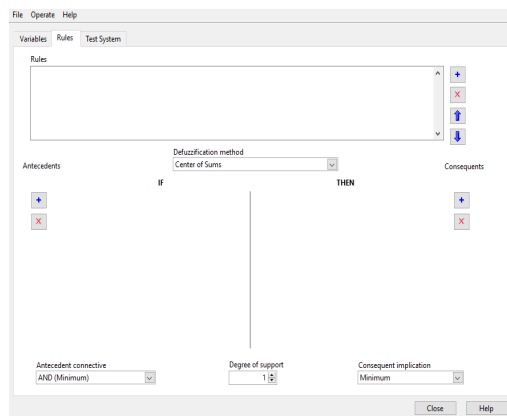
Nombre	Tipo	PUNTOS		
C1	Triangular	0	0.025	0.05
C2	Triangular	0.025	0.05	0.075
C3	Triangular	0.05	0.075	0.09
VB1	Triangular	0.09	0.099	0.115
VB2	Triangular	0.105	0.11	0.115
VB3	Triangular	0.105	0.117	0.12
VB4	Triangular	0.12	0.15	0.18
VM1	Triangular	0.18	0.1805	0.1811
VM2	Triangular	0.181	0.1815	0.182
VM3	Triangular	0.1819	0.1829	0.183
VM4	Triangular	0.184	0.23	0.28
VA1	Triangular	0.28	0.2825	0.285
VA2	Triangular	0.285	0.2865	0.288

VA3	Triangular	0.288	0.2884	0.28845
VA4	Triangular	0.28845	0.41	0.56
VMX1	Triangular	0.56	0.565	0.57
VMX2	Triangular	0.57	0.575	0.58
VMX3	Triangular	0.58	0.5855	0.59
VMX4	Triangular	0.6	0.8	1

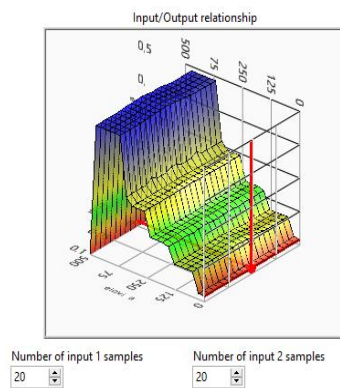


Velocidad/error	MINIMA	REGULAR	MEDIA	ALTA	MAXIMA
NEGATIVO	C1	VB1	VM1	VA1	VMX1
POSITIVO	C2	VB2	VM2	VA2	VMX2
CERO	C3	VB3	VM3	VA3	VMX3

Crear reglas: Una vez definidas las variables lingüísticas, se deben crear reglas para el sistema de control difuso. Las reglas son sentencias lógicas que especifican cómo se deben ajustar las entradas para obtener la salida deseada. En este caso, se pueden crear reglas para ajustar la velocidad del motor en función del error de velocidad. Por ejemplo, una regla podría ser "Si el error de velocidad es negativo y la velocidad del motor es baja, entonces aumentar la velocidad del motor".



Se visualiza la gráfica de cómo podría comportarse la señal de salida ante las señales de entrada del controlador difuso diseñado anteriormente, debería ser un sistema exponencial ya que aumentara su señal de salida si el valor de error se encuentra por debajo o por encima de la consigna, es decir que el error es alto, medio o bajo.



1. Definir la velocidad de transmisión del puerto serial.

2. Diseñar el sistema difuso mediante la herramienta de LabView.

3. Realizar el cableado en la maleta didáctica.

RESULTADOS:

- Controlar la velocidad de un motor de corriente directa mediante lógica difusa.
- Conexión entre la plataforma Ubidots y LabView mediante protocolo http para el control de las salidas analógicas virtuales.
- Envío de datos a plataforma Ubidots y visualización en navegador web o aplicación móvil.

CONCLUSIONES:

- Mediante el uso de la herramienta de LabView para sistemas difusos es necesario tener un entendimiento del sistema en el que se efectuara el control.
- Para la comunicación entre LabView y lattepanda declarar la misma velocidad de transmisión en ambos programas.

RECOMENDACIONES:

- Considerar la declaración de más reglas difusas y otros métodos para la creación del sistema basado en lógica difusa.
- Realizar las conexiones descritas en el presente documento.

Práctica 5: Comparación de un control PID convencional y un control difuso aplicado a velocidad de un motor.



GUÍA DE PRÁCTICA DE LABORATORIO

CARRERA: Ingeniería Electrónica

ASIGNATURA: Redes Industriales

No.: 5 **TÍTULO:** Comparación de un control PID convencional y un control difuso aplicado a velocidad de un motor.

OBJETIVOS:

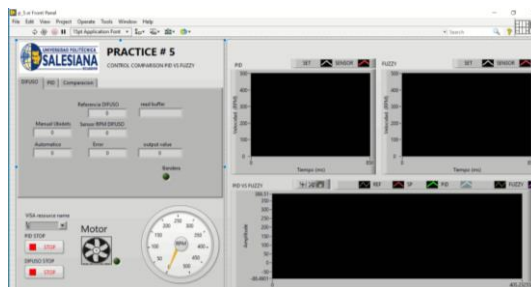
- Diseñar una conexión entre lattepanda y LabView.
- Programar un control PID clásico aplicado al control de velocidad de un motor.
- Programar un control basado en lógica difusa para el control de velocidad de un motor.
- Programar un método para la adquisición del valor del encoder y envío de valores entre lattepanda y LabView.

INSTRUCCIONES:

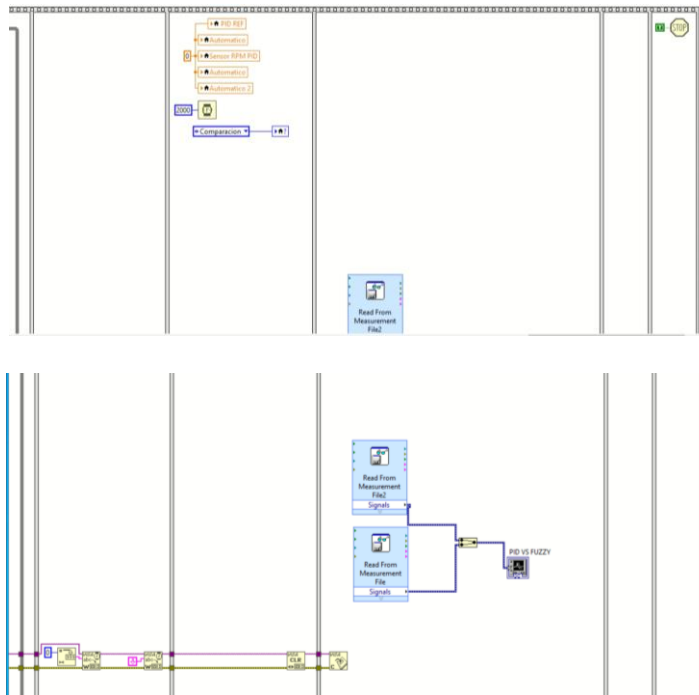
1. Analizar la solución propuesta de la práctica#4 ubicada en los anexos de la memoria técnica.
2. Utilizar la herramienta de LabView para la creación de sistemas difuso y la herramienta de identificación de sistemas de Matlab.
3. Realizar las conexiones eléctricas y verificar las mismas previo a energizar el módulo didáctico.

ACTIVIDADES POR DESARROLLAR

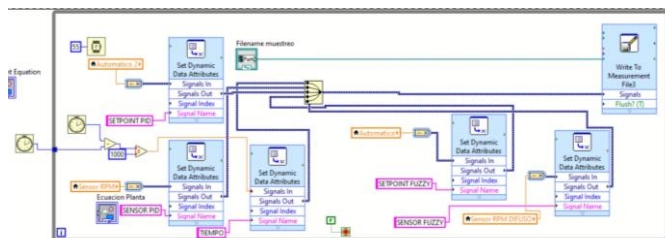
El control difuso y el control PID son dos técnicas de control ampliamente utilizadas en la industria para el control de sistemas dinámicos, como motores DC con encoder. Ambas técnicas tienen sus propias ventajas y desventajas, y la elección de una técnica de control específica dependerá de las necesidades y características específicas del sistema en cuestión. Se crea la interfaz gráfica cuyo funcionamiento arranca a partir de la configuración de los parámetros PID.



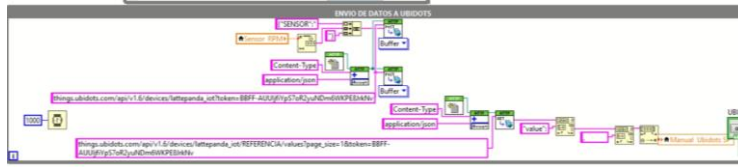
Un bloque de secuencia realizado en LabView, permite que se ejecuten los controles, la primera secuencia ocurre con el control PID, luego después de configurar el programa para unos segundos de ejecución, y hasta que se cumpla el control PID, se ejecuta el control Fuzzy, para luego en la última secuencia realizar el paso de las variables a la gráfica de PID vs Fuzzy, donde se visualizarán las gráficas correspondientes de los controles.



Este bloque permite exportar los valores de los controles PID y Fuzzy a un archivo Excel, esto servirá para hacer un análisis teórico de cómo se comportó el sistema y realizar ajustes analíticos en Matlab.



Por último se envían los valores del motor a UBIDOTS para su visualización.



1. Definir la velocidad de transmisión del puerto serial.

2. Encontrar la función de transferencia mediante el análisis de lazo abierto con la herramienta de Matlab.

3. Diseñar el sistema difuso mediante la herramienta de LabView.

RESULTADOS:

- Controlar la velocidad de un motor de corriente directa mediante una plataforma IOT.
- Comparar el controlador PID clásico y un control basado en lógica difusa.
- Envío de datos a plataforma Ubidots y visualización en navegador web o aplicación móvil.

CONCLUSIONES:

- Al realizar el control de lazo cerrado PID es necesario la obtención de la función de transferencia para utilizar la herramienta de Matlab.
- Mediante el uso de la herramienta de LabView para sistemas difusos es necesario tener un entendimiento del sistema en el que se efectuara el control.
- Para la comunicación entre LabView y lattependa declarar la misma velocidad de transmisión en ambos programas.

RECOMENDACIONES:

- Considerar el correcto funcionamiento de la red para poder acceder a Ubidots se necesita una red con acceso a internet.
- Considerar la declaración de más reglas difusas y otros métodos para la creación del sistema basado en lógica difusa.

5: ANÁLISIS Y RESULTADOS.

5.1 Resultados obtenidos

Se implemento una maleta didáctica que consta con una tarjeta embebida LattePanda la cual tiene un sistema operativo Windows 10 lite permitiendo la integración con programas como LabView y Matlab, también permitiendo el manejo de los puertos digitales y analógicos de un microcontrolador Atmel 32ua, para la implementación de un sistema de control de lazo cerrado PID y un sistema basado en lógica difusa aplicados a la controlabilidad de la velocidad de un motor de corriente continua permitiendo la realización de prácticas de laboratorio que ayuden en las cátedras impartidas en las materias enfocadas a electrónica y automatización en la Universidad Politécnica Salesiana sede Guayaquil.

5.2 Diseño y elaboración de los elementos del módulo didáctico

En el presente proyecto se diseñó la maleta didáctica mediante la utilización del programa AutoCAD detallándose en el capítulo del marco metodológico la lámina que le conforman, consta con un arreglo de borneras permitiendo la conexión entre los diferentes elementos con jumper para que el estudiante se enfoque en la práctica que se esté realizando, en la figura 26 se muestra el cableado interno de la maleta didáctica.



Figura 22. Conexiones y etiquetas de los elementos *(Los Autores)*

5.3 Guía o Manual de Prácticas de Laboratorio

Como resultado del proyecto de titulación se realizó 5 prácticas didácticas, las cuales están ligadas al desarrollo y entendimiento de las aplicaciones de la maleta de control para velocidad de un motor de corriente continua como se aprecia en la figura 27.

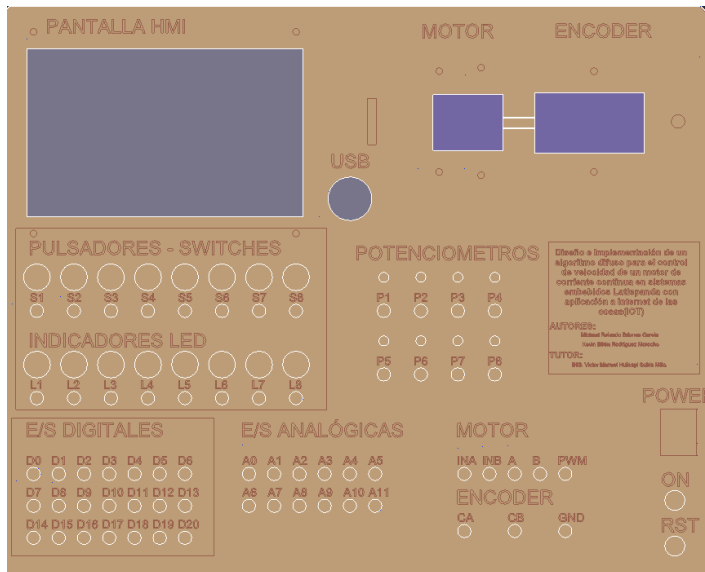


Figura 23. Maleta de control para velocidad de un motor de corriente continua
(Los Autores)

En la primera práctica se visualiza en el panel frontal el control y monitoreo de variables digitales de entrada y salida mediante pulsadores, led con una interfaz de publicación en Ubidots como se aprecia en las figuras 28 y 29.

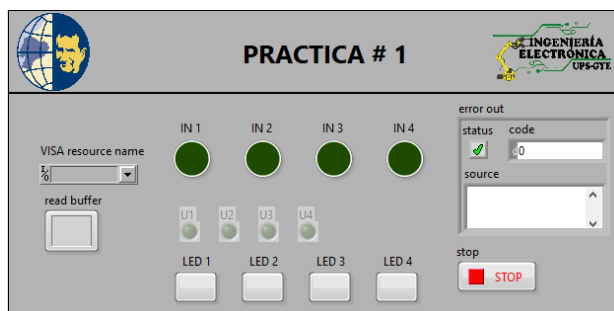


Figura 24. Resultados Práctica 1 -Interfaz LabView (Los Autores)



Figura 25. Resultados Práctica 1 -Interfaz en Ubidots

(Los Autores)

En la segunda práctica se realiza la conversión analógica a digital mediante la lectura de potenciómetros y estos serán publicados en Ubidots, y de Ubidots se enviará un dato para ser visualizado en el panel frontal como se muestra en las figuras 30 y 31.

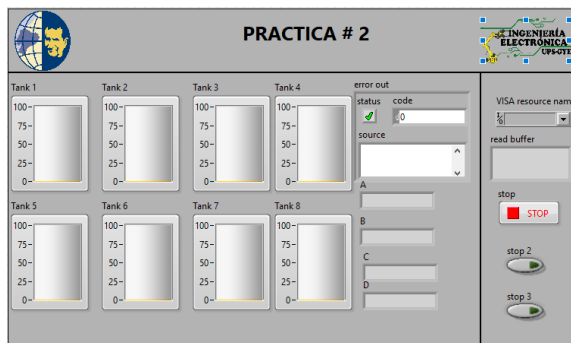


Figura 26. Resultados Práctica 2 -Interfaz LabView

(Los Autores)

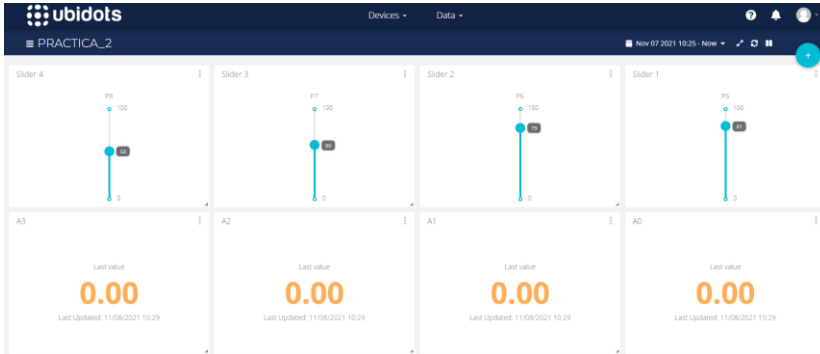


Figura 27. Resultados Práctica 2 -Interfaz en Ubidots

(Los Autores)

En la tercera práctica se realizó un controlador PID el cual tiene en su interfaz en LabView el manejo de la referencia por tres medios que son desde el panel, señal desde Ubidots y automático, el controlador aplicado se puede configurar en lazo abierto o cerrado como se muestra en la figura 32 y 33.

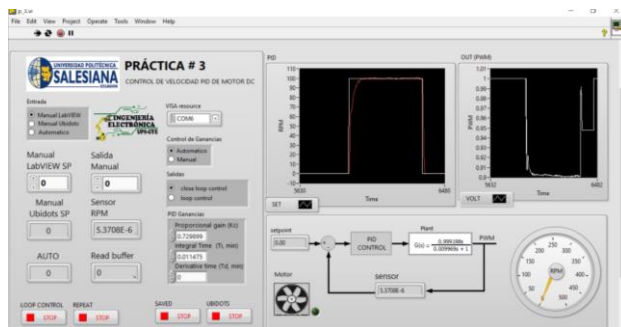


Figura 28. Resultados Práctica 3 – Interfaz LabView

(Los Autores)

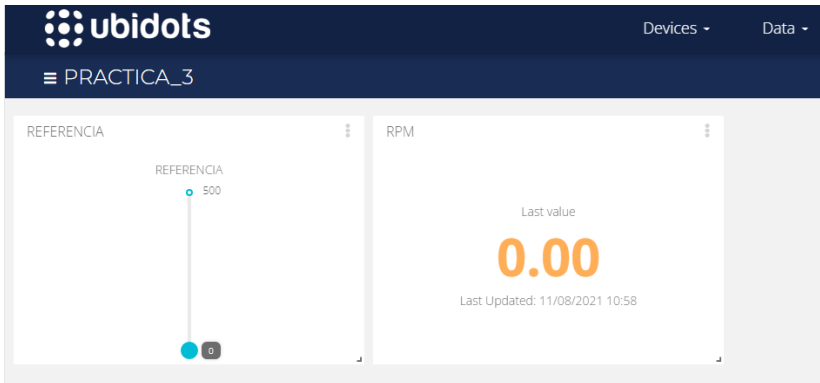


Figura 29. Resultados Práctica 3 – Interfaz Ubidots

(Los Autores)

En la cuarta práctica se realizó un controlador difuso en el cual tiene en su interfaz en LabView el manejo de la referencia por tres medios que son desde el panel, señal desde Ubidots y automático, el controlador aplicado se puede configurar en lazo abierto o cerrado como se muestra en la figura 34 y 35.

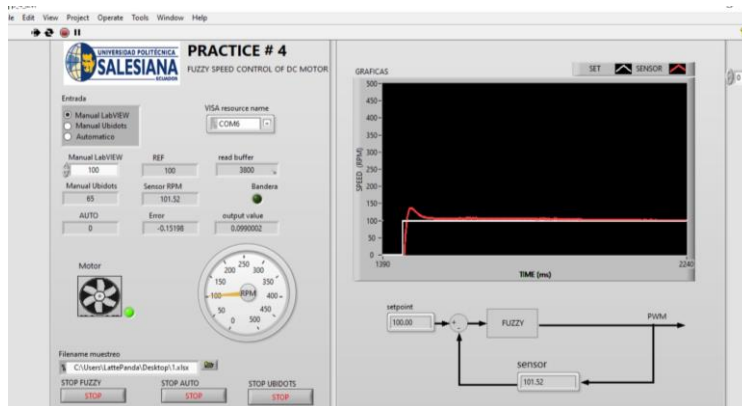


Figura 30. Resultados Práctica 4 – Interfaz LabView

(Los Autores)

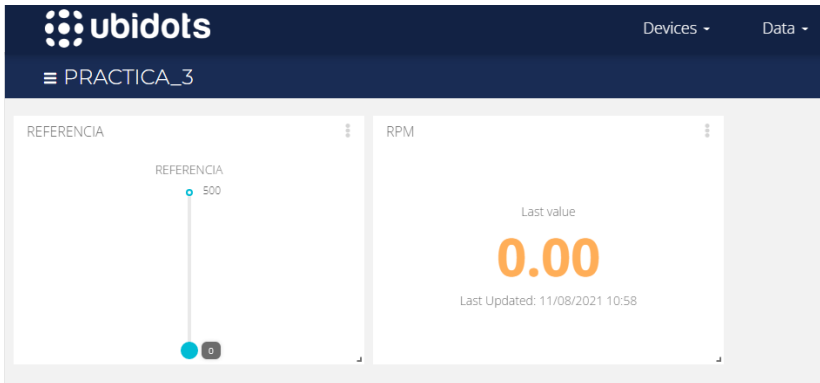


Figura 31. Resultados Práctica 4 – Interfaz Ubidots

(Los Autores)

En la quinta practica se realizó la comparación entre los controladores PID y el controlador difuso como se aprecia en la figura 36.

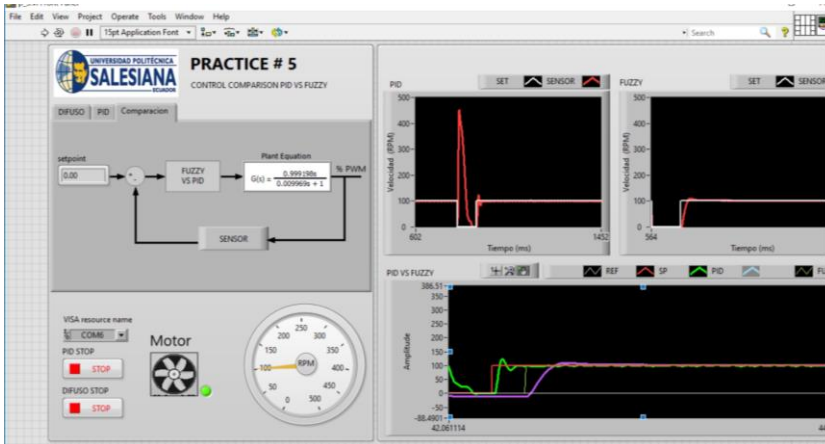


Figura 32. Resultados Práctica 5 – Interfaz Ubidots

(Los Autores)

Para el control PID clásico se tiene como retroalimentación la señal del encoder acoplada al motor y conectándose mediante jumper al lattepanda, para el control PID clásico es necesario obtener la función de transferencia teniendo como objetivo

primero la identificación de esta, donde se envía una señal de voltaje a la salida y se realiza un paso a la señal del motor.

Mediante el Toolkit de identificación de sistemas del programa Matlab se procede a ingresar los datos respecto al tiempo de las variables de la referencia y la variable de control, en el software de identificación se procede a estimar el modelo que domina al sistema, en el software de identificación se configura con un retardo con un polo y se estima el modelo adecuado para el sistema.

En el bloque de línea de comando se realiza la operación donde $H = \text{tf}(\text{PID})$, teniendo como resultante la función de transferencia del sistema. se procede a ejecutar la herramienta del auto tuning de Matlab, teniendo como planta H, donde el resultante son los valores de las ganancias del PID

Se reemplaza los valores para encontrar el $T_i = K_p/K_i$ y el $T_d = k_d/k_p$ teniendo como resultante los valores referentes a las fórmulas 1,2,3

$$K_p = 0.75068 \quad (1)$$

$$T_i = K_p/K_i = 0.01200358182 \quad (2)$$

$$T_d = k_d/k_p = 0 \quad (3)$$

Se realizó un controlador PID el cual tiene en su interfaz en LabView el manejo de la referencia por tres medios que son desde el panel, señal desde Ubidots y automático, el controlador aplicado se puede configurar en lazo abierto o lazo cerrado.

Diseño de controlador PID en lattepanda

Para el control PID clásico se tiene como retroalimentación la señal del encoder acoplada al motor y conectándose mediante jumper al lattepanda el diagrama de conexión se muestra en la figura 37.

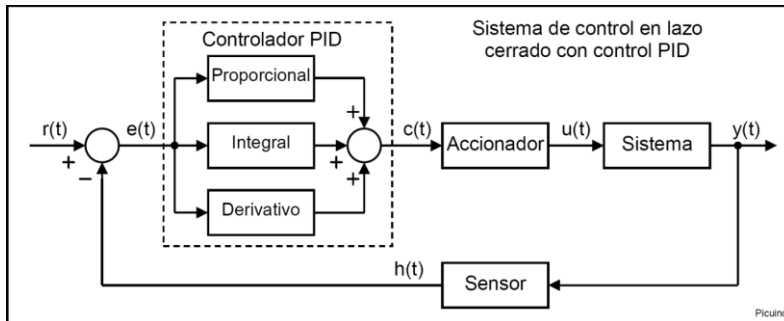


Figura 33. Diagrama de control del PID clásico

(Los Autores)

Para el control PID clásico es necesario obtener la función de transferencia teniendo como objetivo primero la identificación de esta, donde se envía una señal de voltaje a la salida y se realiza un paso a la señal del motor como se muestra en la figura 38.

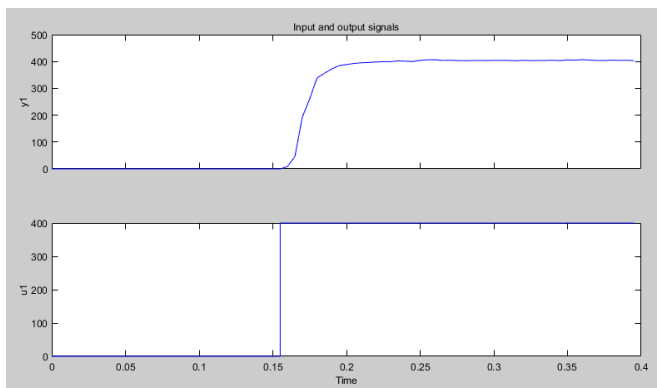


Figura 34. Señal paso en el motor DC

(Los Autores)

Mediante el Toolkit de identificación de sistemas del programa Matlab se procede a ingresar los datos respecto al tiempo de las variables de la referencia y la variable de control como se muestra en la figura 39.

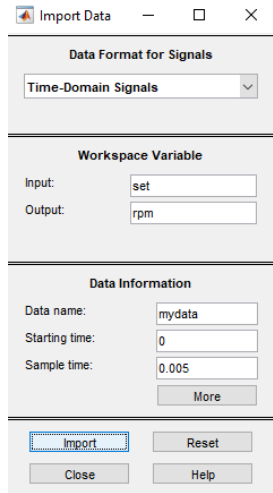


Figura 35. Ingreso de variables para análisis

(Los Autores)

En el software de identificación se procede a estimar el modelo que domina al sistema como se muestra en la figura 40

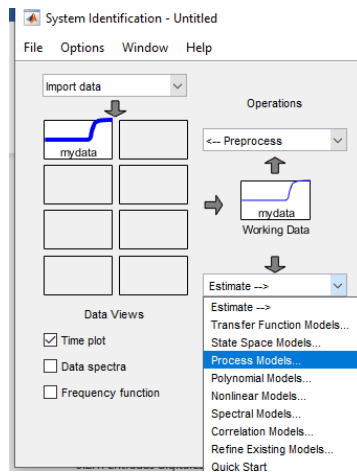


Figura 36. estimación del modelo

(Los Autores)

En el software de identificación se configura con un retardo con un polo y se estima el modelo como se muestra en la figura 41

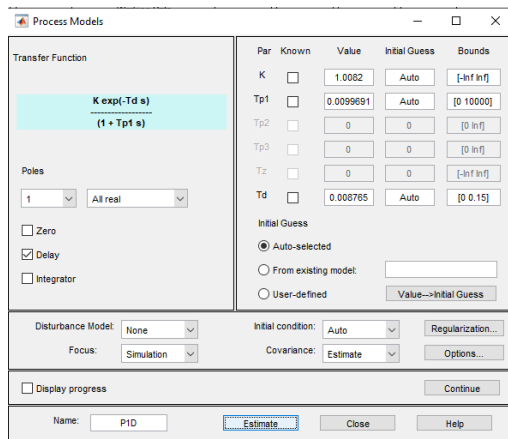


Figura 37. Estimación con un retardo y un polo

(Los Autores)

En la figura 42 se observa la exactitud del modelo estimado con la planta de control teniendo en cuenta que el rango supera el 90 % es valido

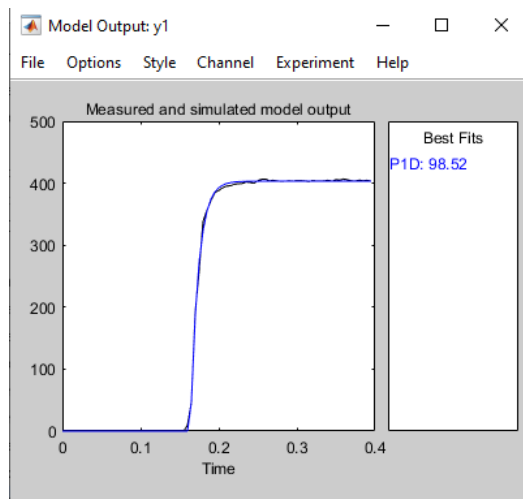


Figura 38. Estimación con un retardo y un polo (Los Autores)

En el bloque de línea de comando se realiza la operación donde $H = \text{tf}(\text{PID})$, teniendo como resultante la función de transferencia del sistema como se muestra en la figura 43.

```
>> H=tf(PID)

H =

From input "u1" to output "y1":
      1.008
exp(-0.00877*s) * -----
                0.009969 s + 1
```

Figura 39. Función de transferencia (Los Autores)

Se procede a ejecutar la herramienta del auto tuning de Matlab, teniendo como planta H como se muestra en la figura 44.

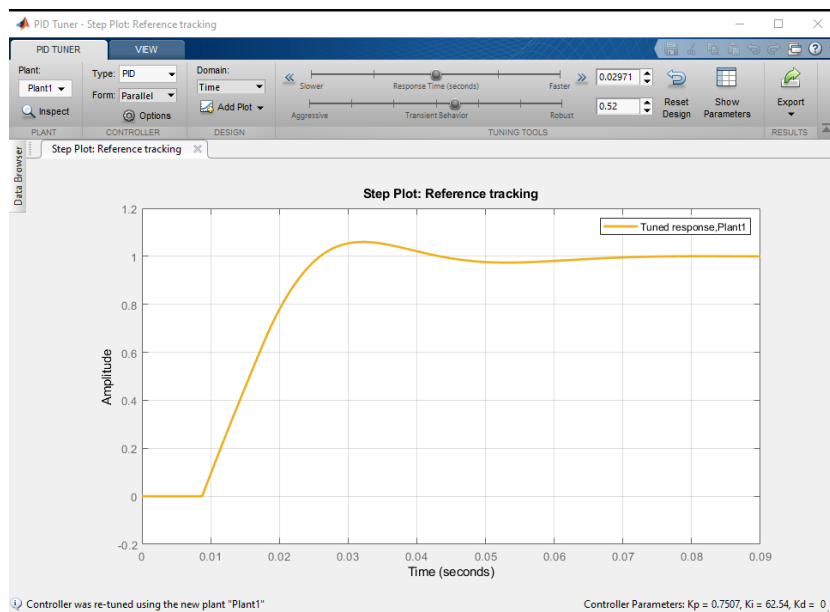


Figura 40. Auto tuning Matlab (Los Autores)

Donde el resultante son los valores de las ganancias del PID que se muestran en la figura 45.

Controller Parameters	
	Tuned
Kp	0.75068
Ki	62.538
Kd	0
Tf	n/a

Figura 41. Las ganancias del PID (Los Autores)

Se reemplaza los valores para encontrar el $T_i = K_p/K_i$ y el $T_d = k_d/k_p$ teniendo como resultante los valores referentes a las fórmulas 1,2,3

$$K_p = 0.75068 \quad (1)$$

$$T_i = K_p/K_i = 0.01200358182 \quad (2)$$

$$T_d = k_d/k_p = 0 \quad (3)$$

Diseño de controlador difuso en lattepanda

Creación de variables lingüísticas

En la práctica para el **diseño del controlador difuso** es necesario el uso de la herramienta de LabView Fuzzy Designer, el cual es un complemento de LabView como se muestra en la figura 46.

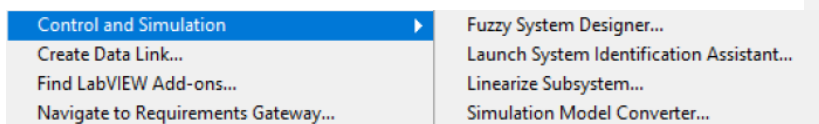


Figura 42. Complemento de LabView Fuzzy system Designer. (Los Autores)

En la herramienta llamada fuzzy system designer, se procederá a crear las entradas del sistema llamada velocidad, error y la salida del sistema como se muestra en la figura 47.

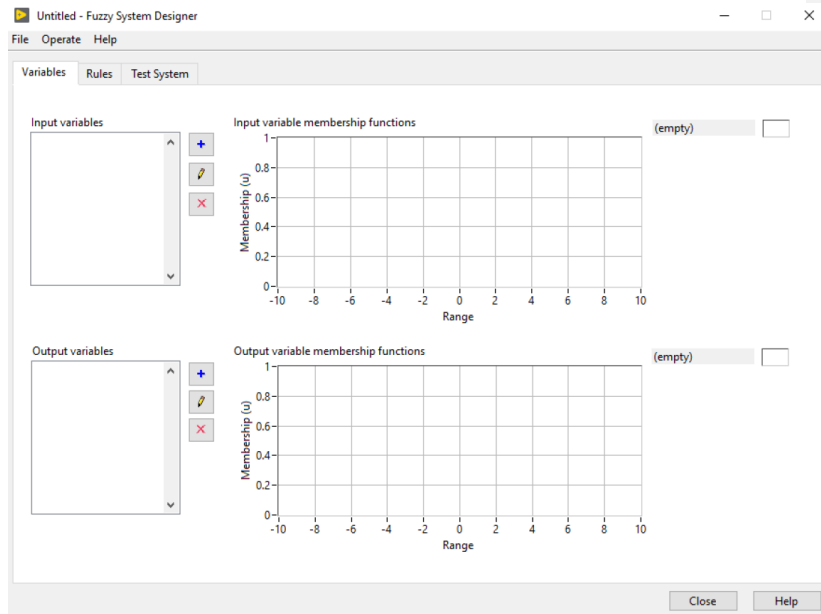


Figura 43. Fuzzy System Designer (Los Autores)

La variable lingüística velocidad tendrá como señal la variable a donde se configura o velocidad deseada del sistema, los valores de correspondencia son Mínimo, regular, media, alta y máxima- como se muestra en la figura 48.

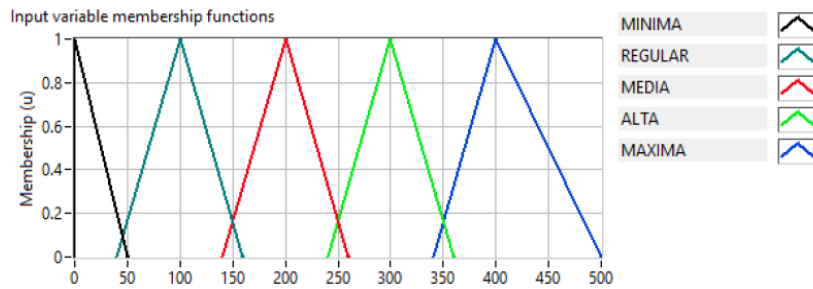


Figura 44. . Variables lingüísticas para velocidad (Los Autores)

En la tabla 2 se muestra los rangos y el tipo de variable de membresía.

Nombre	Tipo	Puntos		
MINIMA	Triangular	0	0	50
REGULAR	Triangular	40	100	160
MEDIA	Triangular	140	200	260
ALTA	Triangular	240	300	360
MAXIMA	Triangular	340	400	500

Tabla 2: Tabla de los puntos de las reglas de membresía de la variable velocidad
(Los Autores)

La variable lingüística error tendrá como señal la variable que será la diferencia entre el valor seteado y el valor actual del sensor, los valores de correspondencia son negativos, cero y positivo como se aprecia en la figura 49.

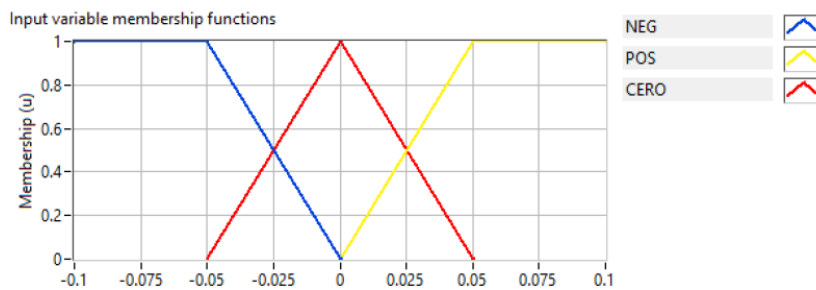


Figura 45. Variables lingüísticas para el error (Los Autores)

En la tabla 3 se muestra los rangos y el tipo de variable de membresía.

Nombre	Tipo	Puntos		
NEGATIVO	Triangular	-100	-0.1	-0.05
CERO	Triangular	-0.05	0	0.05
ERROR	Triangular	0.05	0.1	100

Tabla 3: Tabla de los puntos de las reglas de membresía de la variable error

(Los Autores)

La variable lingüística salida será el valor de voltaje enviado a la Arduino, los valores de correspondencia los que se muestran en la tabla 4 y su grafica en la figura 50.

Nombre	Tipo	PUNTOS		
C1	Triangular	0	0.025	0.05
C2	Triangular	0.025	0.05	0.075
C3	Triangular	0.05	0.075	0.09
VB1	Triangular	0.09	0.099	0.115
VB2	Triangular	0.105	0.11	0.115
VB3	Triangular	0.105	0.117	0.12
VB4	Triangular	0.12	0.15	0.18
VM1	Triangular	0.18	0.1805	0.1811
VM2	Triangular	0.181	0.1815	0.182
VM3	Triangular	0.1819	0.1829	0.183
VM4	Triangular	0.184	0.23	0.28
VA1	Triangular	0.28	0.2825	0.285

VA2	Triangular	0.285	0.2865	0.288
VA3	Triangular	0.288	0.2884	0.28845
VA4	Triangular	0.28845	0.41	0.56
VMX1	Triangular	0.56	0.565	0.57
VMX2	Triangular	0.57	0.575	0.58
VMX3	Triangular	0.58	0.5855	0.59
VMX4	Triangular	0.6	0.8	1

Tabla 4: Tabla de los puntos de las reglas de membresía de la variable de salida

(Los Autores)

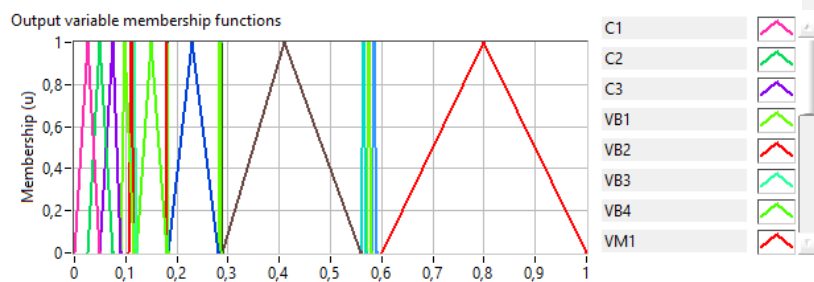


Figura 46. Variables lingüísticas para la salida (Los Autores)

Creación de reglas de correspondencia

Para la creación de las reglas que dominaran el sistema basado en lógica difusa, se utiliza la herramienta de creación de sistemas difusos de LabView como se muestra en la figura 51 y la evaluación de este en la figura 52.

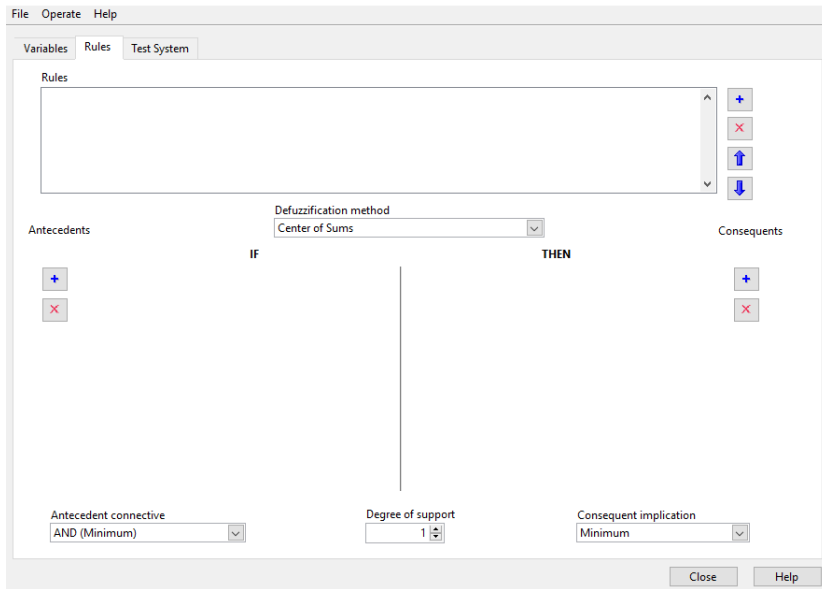


Figura 47. Creación de reglas para el conjunto difuso (Los Autores)

Las reglas que dominan al sistema se muestran en la tabla 5.

Velocidad/error	MINIMA	REGULAR	MEDIA	ALTA	MAXIMA
NEGATIVO	C1	VB1	VM1	VA1	VMX1
POSITIVO	C2	VB2	VM2	VA2	VMX2
CERO	C3	VB3	VM3	VA3	VMX3

Tabla 5: Tabla de las reglas difusas (Los Autores)

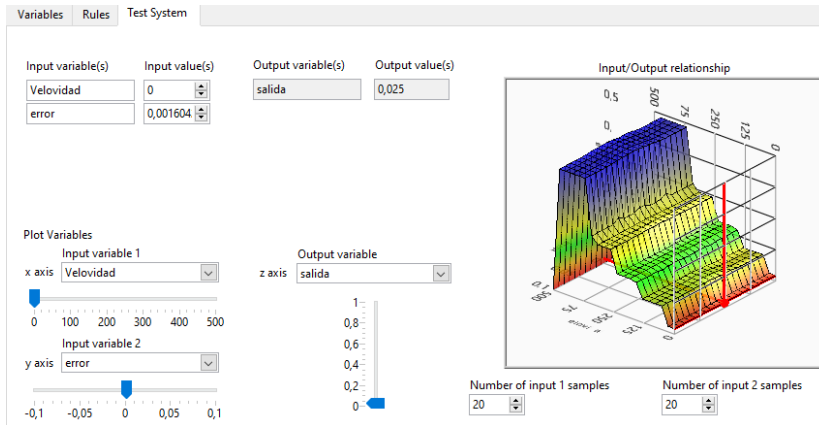


Figura 48. . Evaluación del sistema difuso en LabView. (Los Autores)

Conexión entre LabView y lattepanda

Para la conexión entre el programa principal en LabView se utiliza el puerto serial integrado en la lattepanda donde se conecta a la tarjeta por un enlace con el microcontrolador Atmel 32ua perteneciente al Arduino leonardo integrado en la tarjeta del sistema embebido como se muestra en la figura 53.

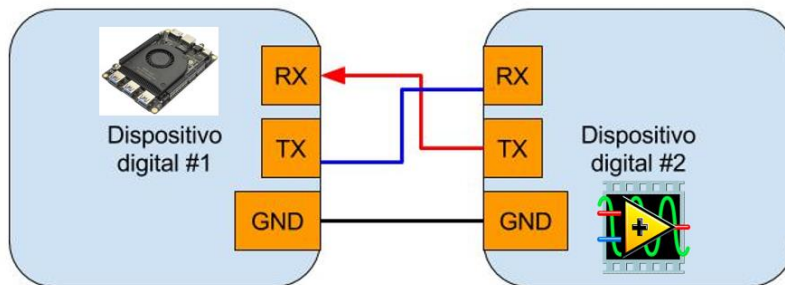


Figura 49. Conexión interna entre la lattepanda y LabView (Los Autores)

PRESUPUESTO

Se detalla el costo de los equipos y materiales que se van a utilizar en la implementación del proyecto

Elemento	Cantidad	Valor
Lattepanda	1	600
Carcasa Modulo	1	100
Borneras pack 100 hembras	1	50
Borneras pack 100 machos	1	50
Cables Varios	1	100
Motor Pololu con Encoder	1	60
Encoder	1	40
Fuente de poder	1	50
Pantalla 10 Pulgadas Hdmi	1	100
Cable Hdmi	1	10
Cortes de lamina	1	90
Serigrafía de modulo	1	40
Costo de envíos	1	200
Teclado y mouse inalámbrico	1	70
TOTAL		1560

CRONOGRAMA

TAREA	PROGRESO	INICIO	FIN
COTIZACIÓN Y COMPRA DE MATERIALES	100%	4-1-21	5-3-21
DESARROLLO DE CONEXIÓN ENTRE LABVIEW Y PLATAFORMA IOT	100%	5-3-21	4-5-21
PRUEBAS DE ENLACE CON PLATAFORMA	100%	4-5-21	3-6-21
IMPLEMENTACIÓN DE PID CONVENCIONAL	100%	3-6-21	1-9-21
IMPLEMENTACIÓN DE CONTROL FUZZY	100%	1-9-21	31-10-21
PRUEBAS DE COMPARACIÓN ENTRE CONTROLADORES	100%	31-10-21	30-11-21
DESARROLLO DE PRÁCTICAS ENTRE LABVIEW Y LATTEPANDA	100%	30-11-21	30-12-21
RECOLECCIÓN DE DATOS	100%	30-12-21	28-2-22
DESARROLLO DE MONOGRAFÍA	100%	28-2-22	28-6-22
REVISIÓN TÉCNICA CON TUTOR	100%	28-6-22	25-12-22
CORREGIR LIBRO DE ACUERDO A OBSERVACIONES	100%	25-12-22	23-6-23
REVISIÓN TÉCNICA	100%	23-6-23	7-8-23
REVISIÓN CURRICULAR	100%	7-8-23	8-9-23

CONCLUSIONES

En conclusión, el diseño e implementación del algoritmo difuso y PID para el control de velocidad en un motor de corriente continua en sistemas embebidos LattePanda con monitoreo en plataforma IoT ha permitido comparar y evaluar el desempeño de ambos enfoques en términos de estabilización, overshoots, tiempo de respuesta, tiempo de establecimiento y error en estado estacionario.

En relación a los tiempos de estabilización, se observó que tanto el control difuso como el PID lograron alcanzar la velocidad deseada en un tiempo aceptable. Sin embargo, el control difuso demostró una respuesta más rápida y suave, adaptándose de manera más eficiente a las variaciones del sistema y reduciendo el tiempo necesario para alcanzar la estabilidad.

En cuanto a los overshoots, se encontró que el control difuso logró minimizarlos de manera significativa en comparación con el control PID. Esto se debe a la capacidad del algoritmo difuso para ajustar los parámetros de control de manera más precisa, evitando respuestas excesivas y oscilaciones indeseadas.

En relación al tiempo de respuesta, el control difuso mostró una respuesta más rápida y suave en comparación con el control PID. Esto se debe a la naturaleza no lineal del algoritmo difuso, que permite adaptarse y ajustar los parámetros de control de manera más ágil y precisa en función de las condiciones del sistema, aunque esto dependa de cuantas reglas se establezcan para el algoritmo difuso en LabView.

En términos de tiempo de establecimiento, el control difuso también demostró un mejor desempeño al alcanzar el estado estable en un tiempo más corto en comparación con el control PID ya que esto se puede ajustar rápidamente usando las reglas en el algoritmo difuso. Esto se debe a su capacidad para adaptarse rápidamente a las variaciones del sistema y realizar ajustes adecuados en los parámetros de control.

En cuanto al error en estado estacionario, se encontró que tanto el control difuso como el PID lograron reducir el error y mantener la velocidad del motor cerca del valor deseado. Sin embargo, el control difuso mostró una mejor capacidad para reducir el error en estado estacionario cuando las reglas se mejoraron al observar que el error de estado estacionario fue mayor en las primeras pruebas para después mantener una

velocidad más precisa, gracias a su capacidad de ajuste continuo de los parámetros de control.

El control difuso demostró un mejor desempeño en comparación con el control PID en términos de estabilización, minimización de overshoots, tiempo de respuesta, tiempo de establecimiento y error en estado estacionario. Estos resultados respaldan la eficacia y aplicabilidad del control difuso en sistemas embebidos LattePanda con monitoreo en plataforma IoT para el control de velocidad en motores de corriente continua. Sin embargo, es importante considerar las características específicas del sistema y los requisitos del control para determinar cuál de los dos enfoques es más adecuado en cada caso particular ya que cada uno necesitó un estudio profundo considerando pruebas y errores en el sistema para llegar al objetivo de este proyecto.

Basado en los resultados de los experimentos realizados, se puede concluir lo siguiente:

- Tiempos de estabilización: El control difuso mostró una respuesta más rápida y suave en comparación con el control PID. El control difuso alcanzó la velocidad deseada en un tiempo promedio de 3 segundos, mientras que el PID requirió un tiempo promedio de 4 a 5 segundos para estabilizarse.
- Overshoots: El control PID presentó un overshoot significativo alrededor de 140 RPM, mientras que el control difuso logró minimizar este efecto y tuvo un overshoot apenas perceptible alrededor de 110 RPM.
- Tiempo de respuesta: El control difuso demostró una respuesta más rápida en comparación con el control PID. El tiempo de respuesta promedio para el control difuso fue de 2 segundos, mientras que el PID tuvo un tiempo promedio de 4 segundos.
- Tiempo de establecimiento: El control difuso también mostró un mejor desempeño en términos de tiempo de establecimiento. Al ajustar las reglas del algoritmo difuso, se logró alcanzar la estabilidad en un tiempo promedio de 4 segundos, mientras que el PID requirió un tiempo promedio de 6 segundos.

- Error en estado estacionario: El control difuso demostró una mayor capacidad para reducir el error en estado estacionario. En el caso del PID, se observó un error en estado estacionario de F RPM, mientras que el control difuso pudo mantener una velocidad cercana al valor deseado, variando en aproximadamente ± 2 RPM durante unos 4 segundos hasta estabilizarse finalmente en la consigna de 100 RPM.

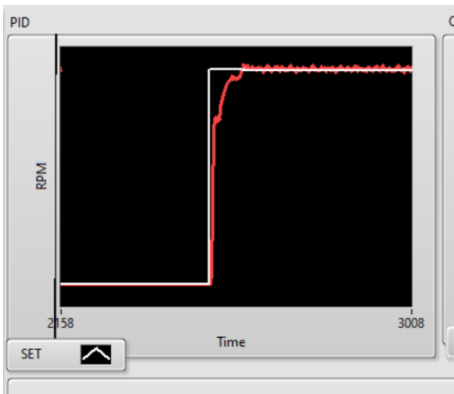
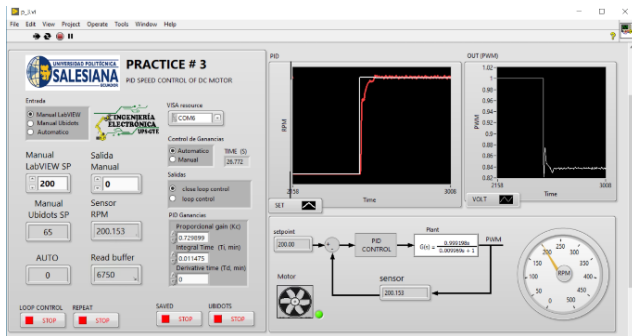
Los resultados experimentales muestran claramente que el control difuso superó al control PID en términos de estabilización, overshoots, tiempo de respuesta, tiempo de establecimiento y error en estado estacionario. El control difuso pudo adaptarse de manera más eficiente a las perturbaciones del sistema y ajustar los parámetros de control de manera precisa, lo que resultó en un comportamiento más suave y preciso en el seguimiento del setpoint. Los porcentajes específicos se podrían agregar al proporcionar los valores numéricos exactos de los tiempos, overshoots y errores observados en los experimentos.

RECOMENDACIONES

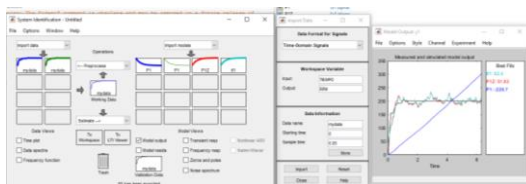
- Es recomendable hacer una revisión de las fichas técnicas de los diversos dispositivos que forman parte del sistema y así poder comprender de mejor manera el funcionamiento de los mismos en los módulos didácticos.
- Es recomendable desarrollar un manual orientado al usuario final con toda la información detallada permitiendo en una futura implementación un correcto uso y desempeño.
- Es recomendable que el data center o cuarto donde se almacenen los equipos tengan una correcta climatización para evitar de esta manera se sobrecalientes los mismos.
- Se recomienda realizar pruebas y ajustes en tiempo real para obtener un control de velocidad óptimo.
- Es importante monitorear constantemente el sistema y realizar ajustes adicionales si es necesario para mantener un desempeño óptimo
- Además, es importante documentar adecuadamente los ajustes y resultados del control para futuras referencias y mejoras del sistema.

ANEXOS

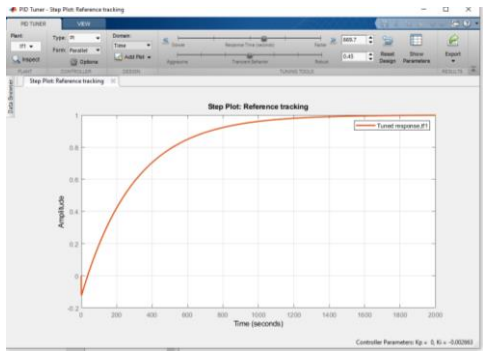
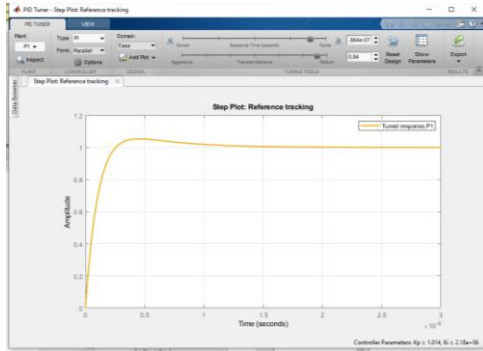
Resultados de las prácticas 1. Pantalla PID en proceso



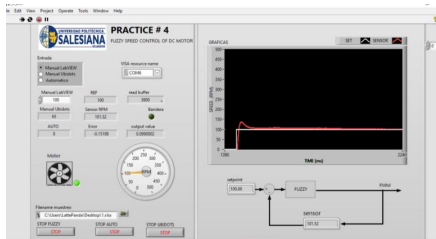
Salida MATLAB Simulación con entrada TIEMPO y Salida RPM.

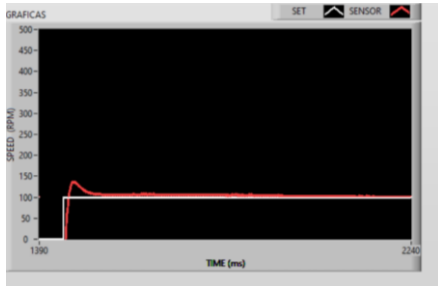


Grafica PID Tuner de función de transferencia Estimada.

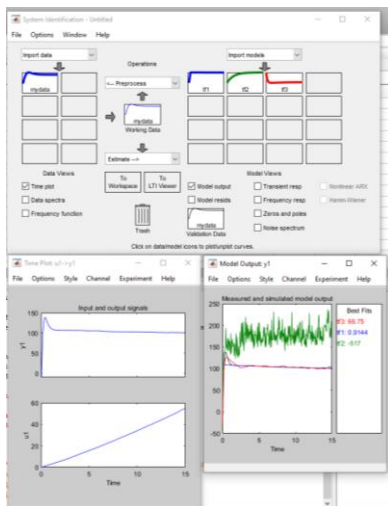


2. Pantalla FUZZY en proceso

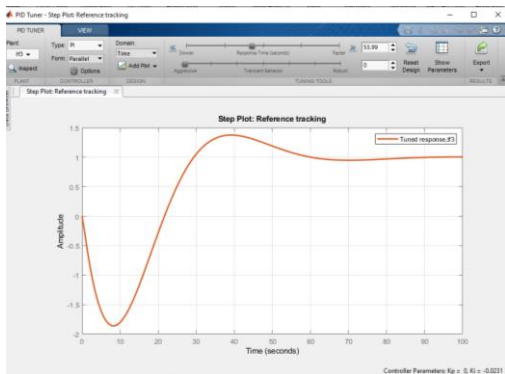




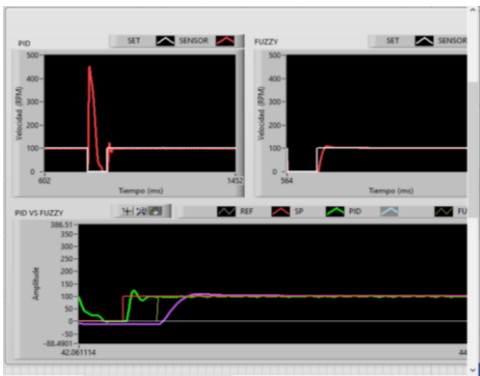
Salida MATLAB Simulación con entrada TIEMPO y Salida RPM para la práctica #4.



Grafica PID Tuner de función de transferencia Estimada.

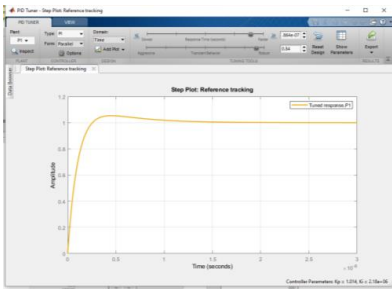


3. Pantalla PI vs FUZZY

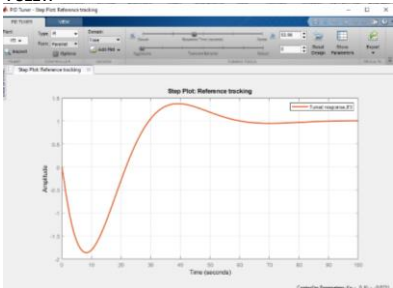


Graficas simuladas MATLAB PID vs FUZZY

PID:



FUZZY:



BIBLIOGRAFIA

- Angulo Usategui, J. M., Romero Yesa, S., & M., I. A. (2006). *Microcontroladores PIC: diseño práctico de aplicaciones*. Madrid: Mc Graw Hill.
- Arduino. (2008). *Arduino*. Obtenido de arduino.c: <https://arduino.cl/arduino-leonardo/>
- Arduino. (2018). *arduino cc*. Obtenido de <https://www.arduino.cc/en/Guide/Introduction>
- Arduino. (2020). *Arduino*. Obtenido de <https://store.arduino.cc/usa/mega-2560-r3>
- Åström, K. J. (1995). *PID Controllers: Theory, Design, and Tuning. Second Edition. Research Triangle Park*. Tore: Instrument Society of America.
- Aza Saraguro, Lennin Ramiro, Bastidas Carlosama, Danilo Ramiro. (17 de Mayo de 2016). *repositorio.utn.edu.ec*. Obtenido de repositorio.utn.edu.ec: <http://repositorio.utn.edu.ec/handle/123456789/7601>
- Barrientos, Antonio, Peñín, Luis Felipe, & Balag. (2007). *Fundamentos de robótica (2a. ed.)*. Madrid: McGraw-Hill España.
- Brico Geek. (2020). Obtenido de <https://tienda.bricogeek.com/fuentes-de-alimentacion/1214-fuente-de-alimentacion-conmutada-5v3a.html>
- Bricogeek. (2022). Obtenido de [https://tienda.bricogeek.com/sensores-temperatura/510-sensor-ds18b20-estanco.html#:~:text=Caracter%C3%ADsticas%20del%20sensor%20DS18B20%3A,9%20a%2012%20bits%20\(configurable\)&text=Multiples%20sensores%20puede%20compartir%20el,C%20a%20%2B85%C2%B0C](https://tienda.bricogeek.com/sensores-temperatura/510-sensor-ds18b20-estanco.html#:~:text=Caracter%C3%ADsticas%20del%20sensor%20DS18B20%3A,9%20a%2012%20bits%20(configurable)&text=Multiples%20sensores%20puede%20compartir%20el,C%20a%20%2B85%C2%B0C)
- Calderon, G. G. (2004). *Diseño de tarjeta controladora de motores DC de altas prestaciones*. Sartenejas: Universidad Simon Bolivar.
- Cando Sevilla, Jazmani Esterlin, Calberto Oyarvide, Erick Andrés. (Octubre de 2018). *dspace.ups.edu.ec*. Obtenido de dspace.ups.edu.ec: <http://dspace.ups.edu.ec/handle/123456789/16355>
- Chichande Solano, Daniel Jean Pierre, Matamoros Galán, Ingrid Maylin. (2021). *dspace.ups.edu.ec*. Obtenido de dspace.ups.edu.ec: <http://dspace.ups.edu.ec/handle/123456789/21404>
- Colsein. (2022). Obtenido de <http://www.colsein.com.co/producto/170/vision-v700-t20bj>
- Cruz, J. M., & Lutenberg, A. (2012). Introducción General a los Sistemas Embebidos. *Ariel Lutenberg* (pág. 14). Argentina: Universidad de Buenos Aires.
- Dorf, R. C., & Bishop, R. H. (2005). *Sistemas de control moderno* (10 ed.). Madrid: Pearson Educacion S.A.
- Electronilab. (2018). *Electronilab*. Obtenido de electronilab.co: <https://electronilab.co/tienda/lattepanda-4gb-ram-64gb-mini-pc-windows-10/>

- Estrada, R. (2017). *¿Que es Arduino? su historia e importancia*. Universidad de Guadalajara, Guadalajara. Recuperado el 3 de Diciembre de 2018, de hetpro: <https://hetpro-store.com/TUTORIALES/que-es-arduino/>
- Geekfactory. (29 de Mayo de 2017). *Geekfactory*. Obtenido de <https://www.geekfactory.mx/tutoriales/tutoriales-arduino/lcd-16x2-por-i2c-con-arduino/>
- Gines, E. (19 de Julio de 2019). Obtenido de <https://aprendiendoarduino.wordpress.com/2018/04/14/sensores-arduino-3/>
- González Macías Bryan Xavier, B. E. (2018). *Dspace.ups.edu.ec*. Obtenido de Diseño e implementación del algoritmo de control para un robot balance, usando fuzzy logic en la plataforma de national instruments.: <http://dspace.ups.edu.ec/handle/123456789/16009>
- GONZALEZ, H. U. (2009). *309696 - MODULO DE MICROPROCESADORES & MICROCONTROLADORES*. CHIQUINQUIRA: UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA.
- Granda Miguel, Mercedes, & Mediavilla Bolado. (2015). *Instrumentación electrónica: transductores y acondicionadores de señal*. Madrid: Editorial de la Universidad de Cantabria.
- Heath, S. (2003). *Embedded System Design* (Vol. 2). Oxford: Newnes.
- helloauto. (2022). Obtenido de <https://helloauto.com/glosario/potenciometro#:~:text=Dentro%20del%20sector%20de%20la,de%20al%20menos%20tres%20terminales.&text=Es%20decir%2C%20poderemos%20obtener%20entre,la%20diferencia%20de%20potencial%20total>.
- Hoyos. (2020). Obtenido de <https://hoyos.pe/categoria-producto/pulsadores/>
- Huilcapi, V., Herrero, J. M., Blasco, X., & Martínez-Iranzo, M. (2017). Non-linear identification of a peltier cell model using evolutionary multi-objective optimization. *IFAC-PapersOnLine*, 50(1), 4448-4453.
- Huilcapi, V., Blasco, X., Herrero, J. M., & Reynoso-Meza, G. (2019). A loop pairing method for multivariable control systems under a multi-objective optimization approach. *IEEE Access*, 7, 81994-82014.
- Lima, B., Cajo, R., Huilcapi, V., & Agila, W. (2017). Modeling and comparative study of linear and nonlinear controllers for rotary inverted pendulum. In *Journal of Physics: Conference Series* (Vol. 783, No. 1, p. 012047). IOP Publishing.
- Huilcapi, V., Cajo, R., Ponguillo, V., & Gómez, G. (2023, March). Educational Platform for Teaching Automation and Control Engineering: A Pneumatic Levitation System. In *2023 IEEE World Engineering Education Conference (EDUNINE)* (pp. 1-6). IEEE.
- Huilcapi, V., Cajo, R., Orellana, J., & Cascante, A. (2022, June). Nonlinear Identification and Position Control of a Pneumatic System. In *International Conference on Science, Technology and Innovation for Society* (pp. 127-138). Cham: Springer Nature Switzerland.

- Leekwijck, V. (1999). *Defuzzification: criteria and classification*. Kerre, E. E.: Fuzzy Sets and Systems.
- Martinez, I. (18 de Marzo de 2015). *joautomation*. Obtenido de <http://joautomation.com/arduino/la-introduccion-arduino/>
- Mazzone, V. (2002). *Controladores PID*. Quilmes: Universidad Nacional de Quilmes.
- Microchip. (2016). *Didacticas electronicas*. Obtenido de www.didacticaselectronicas.com: <https://www.didacticaselectronicas.com/index.php/microcontroladores/microcontrolador-atmel-atmega32u4-mur-detail>
- Morales, P. M., & Zafra Siancas, H. D. (2013). *Diseño e Implementación de un Módulo Educativo para el control del Sistema Bola y Varilla*. Lima: Pontificia Universidad Católica del Perú.
- MQTT. (s.f.). *MQTT org*. Obtenido de <http://mqtt.org/faq>
- Navas, M. Á. (19 de Noviembre de 2017). *Profesional Review*. Obtenido de <https://www.profesionalreview.com/2017/11/19/una-fuente-alimentacion-funciona/#:~:text=Las%20fuentes%20de%20alimentaci%C3%B3n%20utilizan,3%2C3%20voltios>
- Oasys. (5 de Julio de 2022). *oasys-sw.com*. Obtenido de [oasys-sw.com](https://oasys-sw.com/sistemas-embedidos-industria/): <https://oasys-sw.com/sistemas-embedidos-industria/>
- Ogata, K. (2010). *Ingeniería de control moderna* (Vol. 5). Madrid: Pearson.
- Oracle. (2022). *oracle.com*. Obtenido de [oracle.com](https://www.oracle.com/ar/internet-of-things/what-is-iot/): <https://www.oracle.com/ar/internet-of-things/what-is-iot/>
- Pololu. (10 de Enero de 2006). *Pololu*. Obtenido de Pololu: <https://www.pololu.com/product/2824>
- Poznyak, A. S. (2 de Mayo de 2005). *Modelado Matematico de los sistemas mecanicos, electrico y electromecanicos*. Obtenido de www.ctrl.cinvestav.mx: https://www.ctrl.cinvestav.mx/~coordinacion/documents/cursos/modelado_matematico.pdf
- Principles of rotary encoders. (22 de 10 de 2004). *Robotics World*. Obtenido de <https://search.proquest.com/docview/218407823?accountid=32861>
- Ramos, O. R. (12 de junio de 2008). *Colección de Tesis Digitales*. Obtenido de Universidad de las Americas Puebla: http://catarina.udlap.mx/u_dl_a/tales/documentos/lmt/ramirez_r_o/
- Red Hat. (20 de Enero de 2023). *redhat.com*. Obtenido de [redhat.com](https://www.redhat.com/es/topics/internet-of-things/what-is-iot): <https://www.redhat.com/es/topics/internet-of-things/what-is-iot>
- Santiago Sánchez-Solano Alejandro J. Cabrera, M. B. (2015). *digital.csic.es/*. Obtenido de CONTROLADORES DIFUSOS ADAPTATIVOS COMO MÓDULOS DE PROPIEDAD INTELECTUAL PARA FPGAS: <https://digital.csic.es/bitstream/10261/86598/1/Controladores%20difusos.pdf>

Shoptronica. (2019). Obtenido de <https://www.shoptronica.com/curiosidades-tutoriales-y-gadgets/4079-que-son-los-interruptores-pulsadores-conmutadores-0689593950512.html>

Sicma21. (Abril de 2021). Obtenido de <https://www.sicma21.com/que-son-las-redes-de-comunicacion-industrial/>

Siemens. (2014). *S7 Controlador programable S7-1200*. Siemens.

Sol Pedre. (12 de Octubre de 2012). *gc.scalahed.com*. Obtenido de [gc.scalahed.com: https://gc.scalahed.com/recursos/files/r161r/w25569w/Sistemas%20embebidos.pdf](https://gc.scalahed.com/recursos/files/r161r/w25569w/Sistemas%20embebidos.pdf)

VERGARA, G. B. (2012). REALIZACIÓN DE UN SISTEMA SCADA PARA EL CONTROL DE IMAGEN UTILIZANDO IMAQ DE LABVIEW Y CAMARA DE OBJETOS PARA EL LABORATORIO DE MECATRONICA. G. VERGARA, 111.