



UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

CARRERA INGENIERÍA ELECTROMECAÁNICA

PROYECTO DE TITULACIÓN

“DESARROLLO DE UN MÓDULO DE ENTRENAMIENTO PARA EL ESTUDIO DE SISTEMAS DE CONTROL, EMPLEANDO HARDWARE Y SOFTWARE LIBRE, PARA LOS LABORATORIOS DE LA CARRERA DE INGENIERÍA ELECTROMECAÁNICA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI”

Autores:

Juan Carlos Capa Cando,

Luis Alexander Pilamunga Sisa

Tutor de Titulación:

Ing. Byron Paúl Corrales Bastidas MsC.

Latacunga - Ecuador

Mayo 2020 – Septiembre 2020



DECLARACIÓN DE AUTORÍA

“Nosotros Capa Cando Juan Carlos y Pilamunga Sisa Luis Alexander declaráramos ser autores del presente proyecto de investigación “DESARROLLO DE UN MÓDULO DE ENTRENAMIENTO PARA EL ESTUDIO DE SISTEMAS DE CONTROL, EMPLEANDO HARDWARE Y SOFTWARE LIBRE, PARA LOS LABORATORIOS DE LA CARRERA DE INGENIERÍA ELECTROMECAÁNICA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI”, siendo Ing. Byron Paúl Corrales Bastidas MsC. tutor del presente trabajo; y eximo expresamente a la Universidad Técnica de Cotopaxi y a sus representantes legales de posibles reclamos o acciones legales.

Además, certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

.....

Pilamunga Sisa Luis Alexander

C.C: 1726161126

.....

Juan Carlos Capa Cando

C.C: 1724867278



AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN

En calidad de Tutor del Trabajo de Investigación sobre el título: “DESARROLLO DE UN MÓDULO DE ENTRENAMIENTO PARA EL ESTUDIO DE SISTEMAS DE CONTROL, EMPLEANDO HARDWARE Y SOFTWARE LIBRE, PARA LOS LABORATORIOS DE LA CARRERA DE INGENIERÍA ELECTROMECAÁNICA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI”, de **Capa Cando Juan Carlos y Pilamunga Sisa Luis Alexander**, de la Carrera de Ingeniería Electromecánica, considero que dicho Informe Investigativo cumple con los requerimientos metodológicos y aportes científico-técnicos suficientes para ser sometidos a la evaluación del Tribunal de Validación de Proyecto que el Consejo Directivo de la Facultad de Ciencias de la Ingeniería y Aplicadas de la Universidad Técnica de Cotopaxi designe, para su correspondiente estudio y calificación.

Latacunga, Septiembre del 2020

Tutor

.....

Ing. Byron Paúl Corrales Bastidas MsC.

CC: 0502347768



APROBACIÓN DEL TRIBUNAL DE TITULACIÓN

En calidad de Tribunal de Lectores, aprueban el presente Informe de Investigación de acuerdo a las disposiciones reglamentarias emitidas por la Universidad Técnica de Cotopaxi, y por la FACULTAD de CIENCIAS DE LA INGENIERÍA Y APLICADAS; por cuanto, el o los postulantes: **Capa Cando Juan Carlos y Pilamunga Sisa Luis Alexander** con el título de Proyecto de titulación: “DESARROLLO DE UN MÓDULO DE ENTRENAMIENTO PARA EL ESTUDIO DE SISTEMAS DE CONTROL, EMPLEANDO HARDWARE Y SOFTWARE LIBRE, PARA LOS LABORATORIOS DE LA CARRERA DE INGENIERÍA ELECTROMECAÁNICA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI” han considerado las recomendaciones emitidas oportunamente y reúne los méritos suficientes para ser sometido al acto de Sustentación de Proyecto. Por lo antes expuesto, se autoriza realizar los empastados correspondientes, según la normativa institucional.

Latacunga, Septiembre del 2020

Para constancia firman:

.....

Lector 1 (Presidente)

Ing. Edwin Homero Moreano Martínez

C.C: 0502607500

.....

Lector 2

Ing. Jefferson Alberto Porras Reyes

CC: 0704400449

.....

Lector 3

Ing. Héctor Raúl Reinoso Peñaherrera

C.C: 0502150899



AVAL DE IMPLEMENTACIÓN

Latacunga, Septiembre del 2020

Ing. Byron Paúl Corrales Bastidas MsC.

Docente de la Universidad Técnica de Cotopaxi

Presente. –

En calidad de docente de la Universidad Técnica de Cotopaxi, confirmo la realización del proyecto **“DESARROLLO DE UN MÓDULO DE ENTRENAMIENTO PARA EL ESTUDIO DE SISTEMAS DE CONTROL, EMPLEANDO HARDWARE Y SOFTWARE LIBRE, PARA LOS LABORATORIOS DE LA CARRERA DE INGENIERÍA ELECTROMECAÁNICA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI”** implementado por los señores estudiantes de la Universidad Técnica de Cotopaxi, de la Carrera de **Ingeniería Electromecánica**: Capa Cando Juan Carlos con CC: 1724867278 y Pilamunga Sisa Luis Alexander con CC: 1726161126.

Es cuanto puedo certificar en honor a la verdad, se expide el presente para que los interesados puedan hacer uso de los fines que crean convenientes.

Atentamente:

.....

Ing. Byron Paúl Corrales Bastidas MsC.

CC: 0502347768

AGRADECIMIENTO

A mi madre Gloria Esperanza y a mi padre Juan Antonio Capa quienes son mis pilares para alcanzar todas mis metas y objetivos propuestos, a mis hermanas quienes me alientan y apoyan siempre, para seguir adelante.

A la Universidad Técnica de Cotopaxi por abrirme las puertas y permitirme formar académicamente en sus instalaciones, y a sus docentes por compartirme sus conocimientos.

Juan C.

DEDICATORIA

A mis padres por haberme forjado en la persona que soy, muchos de mis logros y metas cumplidas se los debo a ellos. Me formaron con valores, me brindaron su apoyo y paciencia, y este trabajo es el resultado de ello.

A mis hermanas que me apoyan incondicionalmente y siempre está ahí para apoyarme y ayudarme en lo que necesite.

A toda mi familia que es lo más valioso que Dios me ha dado.

Juan C.

AGRADECIMIENTO

Primeramente, a Dios quien me ha ayudado desde los inicios de mi formación como profesional, a mis padres por ser un pilar fundamental en cada instante de mi vida, a mis hermanos quienes son un apoyo tremendo y amigos quienes me han ayudado a ser una mejor persona cada día.

A la Universidad Técnica de Cotopaxi y sus maestros quienes han hecho de mi un buen profesional en todos sus aspectos.

Alexander P.

DEDICATORIA

Este trabajo va dedicado primeramente a Dios pues ha sido la ayuda y fortaleza en todo este tiempo, también hacia mis padres que son el motivo por el cual he llegado hasta donde me encuentro hoy y hacia cada uno de mis amigos y compañeros que ha estado conmigo en tiempos duros durante mi formación como profesional.

Alexander P.

ÍNDICE

DECLARACIÓN DE AUTORÍA	i
AVAL DEL TUTOR DE PROYECTO DE TITULACIÓN	ii
APROBACIÓN DEL TRIBUNAL DE TITULACIÓN.....	iii
AVAL DE IMPLEMENTACIÓN.....	iv
RESUMEN.....	xii
ABSTRACT	xiii
AVAL DE TRADUCCIÓN.....	xiv
1. INFORMACIÓN BÁSICA.....	1
2. DISEÑO INVESTIGATIVO DE LA PROPUESTA TECNOLÓGICA	2
2.1 TÍTULO DE LA PROPUESTA TECNOLÓGICA	2
2.2 TIPO DE ALCANCE	2
2.3 ÁREA DEL CONOCIMIENTO:.....	2
2.4 SINOPSIS DE LA PROPUESTA TECNOLÓGICA	3
2.5 OBJETO DE ESTUDIO Y CAMPO DE ACCIÓN	3
2.5.1 Objeto de estudio.....	3
2.5.2 Campo de acción	3
2.6 SITUACIÓN PROBLÉMICA Y PROBLEMA	3
2.6.1 Situación Problemática.....	3
2.6.2 Problema.....	4
2.7 HIPÓTESIS	5
2.8 BENEFICIARIOS:	5
2.9 OBJETIVOS	5
2.9.1 Objetivo general	5
2.9.2 Objetivos específicos.....	5
3. MARCO TEÓRICO.....	7

3.1	Antecedentes	7
3.2	Sistemas de Control	8
3.2.1	Componentes de un sistema de control	9
3.2.2	Tipos de señales.....	11
3.3	Tipos de Controladores	12
3.3.1	De dos posiciones o de encendido y apagado (on/off)	13
3.3.2	Control Proporcional	13
3.3.3	Proporcional integral	14
3.3.4	Control proporcional derivativo	15
3.3.5	Proporcional integral derivativo	17
3.3.6	Control Predictivo	18
3.4	Programación gráfica.....	18
3.4.1	Qué es programación gráfica.....	19
3.5	Software de programación gráfico MyOpenLab	20
3.5.1	El Circuito	20
3.5.2	Panel Frontal.....	21
3.5.3	Formatos y tratamiento de Datos.....	21
3.5.4	Interfaces Implementadas en MyOpenLab.....	22
3.6	Adquisición de datos.....	22
3.6.1	Arduino.....	23
3.6.2	Librería Firmata.....	23
3.6.3	Comandos del protocolo Firmata	23
3.6.4	Raspberry PI.....	24
3.7	Variables de Control.....	25
3.8	Sensores	25
3.8.1	Sensores de temperatura	26
3.8.2	Sensores de intensidad luminosa	27

3.8.3	Sensores de Posición, Desplazamiento y Proximidad	27
3.8.4	Sensor de proximidad ópticos	27
4.	METODOLOGÍA Y DISEÑO EXPERIMENTAL	28
4.1	Definición de variables	28
4.2	Procedimiento para el diseño de los circuitos electrónicos	29
4.3	Diseño de la alimentación y filtrado de tensión para el módulo.....	29
4.4	Diseño del sensor y control de temperatura.....	30
4.4.1	Sensor básico de temperatura	31
4.4.2	Diseño del control de las lámparas incandescentes	31
4.4.3	Diseño del control del ventilador.....	32
4.5	Diseño del sensor y control de intensidad luminosa.....	33
4.6	Diseño del control del motor DC.....	34
4.7	Diseño del control de motor Step o motor paso a paso.....	35
4.8	Diseño de circuito del encordar óptico para los motores.....	35
4.9	Diseño del circuito para el control del relé	36
4.10	Disposición de los pines para el control de las variables.....	36
4.11	Ruteo de las pistas en la placa de circuito integrado	37
4.12	Impresión de la placa de circuito integrado y acople de elementos.....	38
4.13	Diseño y construcción de la caja del módulo	38
4.14	Diseño de los sistemas control para las variables en el Software libre.....	39
4.14.1	Control ON / OFF de temperatura.....	39
4.14.2	Control ON / OFF de intensidad luminosa.....	40
4.14.3	Control PID de temperatura.....	41
4.14.4	Control PID de intensidad luminosa.....	42
4.14.5	Control motor DC.....	43
4.14.6	Control motor STEP	43

5. DESARROLLO DE LA PROPUESTA (ANÁLISIS Y DISCUSIÓN DE RESULTADOS).....	43
5.1 Módulo para sistemas de control.....	44
5.2 Interface Firmata.....	44
5.3 Pruebas de comunicación entre Arduino y MyOpenLab.....	44
5.4 Prueba del control ON/OFF de temperatura.....	44
5.5 Prueba de control PID de temperatura y sus variantes	46
5.6 Prueba de control ON/OFF de Intensidad Luminosa	48
5.7 Pruebas de control PID de Intensidad Luminosa y sus variantes	49
5.8 Prueba de posición del motor Step	51
5.9 Pruebas del control de velocidad.....	52
5.10 Prueba de velocidad del motor DC.....	54
5.11 Prueba de activación del relé	55
6. IMPACTOS (TÉCNICOS SOCIALES, AMBIENTALES O ECONÓMICOS).....	56
7. VALORACIÓN ECONÓMICA	56
8. CONCLUSIONES Y RECOMENDACIONES.....	58
Conclusiones.....	58
Recomendaciones	59
9. REFERENCIA BIBLIOGRÁFICA	61
ANEXOS.....	66

ÍNDICE DE FIGURAS

Figura 3.1 Sistema de Control	9
Figura 3.2 Componentes de un sistema de control.....	9
Figura 3.3 Actuador Relé.....	10
Figura 3.4 Sensor de temperatura Lm 35	10
Figura 3.5 Señales analógicas y digital	11
Figura 3.6 Ejemplo de sistema de control de lazo abierto.....	12
Figura 3.7 Ejemplo de sistema de control de lazo cerrado.....	12
Figura 3.8 Circuito integrado L293N	12
Figura 3.9 Control ON - OFF	13
Figura 3.10 Señal de Control Proporcional	13
Figura 3.11 Control Proporcional.....	14
Figura 3.12 Proporcional Integral.....	15
Figura 3.13 Control proporcional derivativo.....	17
Figura 3.14 Curvas del control proporcional derivativo.....	17
Figura 3.15 Símbolos empleados para identificar los tres tipos de controles.....	18
Figura 3.17 Control Predictivo	18
Figura 3.18 Panel Circuito y Panel Frontal	21
Figura 3.19 Gráfica de operación de un sensor de temperatura	26
Figura 3.20 Gráfica de operación de un LDR, resistencia vs Lumen.....	27
Figura 3.21 Esquema del funcionamiento de un sensor óptico	28
Figura 4.1 Diagrama eléctrico de la alimentación de 5 V y 12 V	30
Figura 4.2 Sensor de temperatura	31
Figura 4.3 Diagrama control de lámparas	32
Figura 4.4 Diagrama control del ventilador	33
Figura 4.5 Diagrama sensor de intensidad luminosa.....	33
Figura 4.6 Diagrama de control del motor DC.....	35

Figura 4.7 Diagrama de control del motor a pasos bipolar	35
Figura 4.8 Diagrama del control del relé.....	36
Figura 4.9 Regulador PID de MyOpenLab	41
Figura 5.1 Panel frontal control On/Off de temperatura	45
Figura 5.2 Control On/Off de temperatura con Setpoint de 50 °C	45
Figura 5.3 Panel frontal control PID de temperatura.....	47
Figura 5.4 Control PID de Temperatura.....	47
Figura 5.5 Panel frontal control On/Off de intensidad luminosa	48
Figura 5.6 Control On/Off de Intensidad luminosa con setpoint de 3.5 (V) equivalente al 90 (%)	49
Figura 5.7 Panel frontal control PID Intensidad luminosa	50
Figura 5.8 Control PID de Intensidad luminosa – PID Tuner Matlab.....	51
Figura 5.9 Ángulo de 90° en la prueba de posición	52
Figura 5.10 Panel frontal control PID del motor DC	53
Figura 5.11 Control PID de velocidad con Setpoint de 4.400 (RPM).....	53
Figura 5.12 Panel frontal control de RPM del motor DC.....	54
Figura 5.13 Control RPM del motor DC e inversión de giro en la muestra N° 50	55
Figura 5.14 Activación del Relé.....	55

ÍNDICE DE TABLAS

Tabla 3.1 Software que permite una programación gráfica.....	19
Tabla 3.2 Tipos de datos en MyOpenLab.....	21
Tabla 3.3 Tabla de mensajes y bytes de datos utilizados	23
Tabla 3.4 Características modelos de Raspberry PI	24
Tabla 3.5 Variable de control	25
Tabla 3.6 Clasificación de los sensores	26
Tabla 4.1 Señalización de variables	29
Tabla 4.3. Disposición de los pines para el control de las variables	36
Tabla 4.4. Ruteo de las pistas en la placa de circuito integrado	37
Tabla 4.5. Impresión de la placa de circuito integrado y acople de elementos	38
Tabla 4.5. Diseño y construcción de la caja del módulo	38
Tabla 4.6. Diseño y construcción de la caja del módulo	39
Tabla 5.1 Constantes K_p , K_i , k_d calculadas para PID de Temperatura	46
Tabla 5.2 Constantes K_p , K_i , k_d calculadas para PID de Intensidad luminosa.....	50
Tabla 5.3 Constantes K_p , K_i , k_d calculadas para PID de Velocidad	52
Tabla 7.1 Presupuesto para el desarrollo de la protesta tecnológica	57

UNIVERSIDAD TÉCNICA DE COTOPAXI

FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

TEMA: “DESARROLLO DE UN MÓDULO DE ENTRENAMIENTO PARA EL ESTUDIO DE SISTEMAS DE CONTROL, EMPLEANDO HARDWARE Y SOFTWARE LIBRE, PARA LOS LABORATORIOS DE LA CARRERA DE INGENIERÍA ELECTROMECAÁNICA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI”

Autores: Juan Carlos Capa Cando

Luis Alexander Pilamunga Sisa

RESUMEN

El análisis y experimentación de los Sistemas de Control aplicados a diferentes procesos es uno de los aspectos importantes que se debe tomar en cuenta al momento de visualizar cada una de sus variables con sus respectivos controles y más aún que la tecnología avanza a pasos agigantados.

Es por ello que la presente propuesta tecnológica tiene como objetivo el desarrollo de un módulo de entrenamiento para el estudio de Sistemas de Control utilizando hardware y software de uso libre, a realizarse en el laboratorio de la Carrera de Ingeniería Electromecánica de la Universidad Técnica de Cotopaxi. El dispositivo tiene implementado procesos de temperatura, intensidad luminosa, velocidad y posición a fin de poder experimentar en cada una de esas variables el desarrollo de controladores y métodos de sintonización. El empleo de software libre como tarjeta de adquisición de datos sumado a ello MyOpenLab como plataforma de diseño libre de instrumentación virtual permite el análisis en tiempo real de los procesos, en donde se realiza la lógica de control para la estabilización de cada uno de los procesos y la experimentación de métodos de sintonía. Una de las potencialidades del módulo es que permite la conexión con diferentes controladores que pueden ir desde microcontroladores hasta PLCs sin importar el modelo y la marca, permitiendo con ello la implementación de controles clásicos como el control ON-OFF y PID e inclusive en futuras investigaciones controles modernos como Fuzzy y Predictivo.

Palabras claves: Software libre, hardware libre, Sistemas de Control, Control PID, Control ON-OFF.

UNIVERSIDAD TÉCNICA DE COTOPAXI
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS

THEME: “DEVELOPMENT OF A TRAINING MODULE FOR THE STUDY OF CONTROL SYSTEMS, USING HARDWARE AND FREE SOFTWARE, FOR THE LABORATORIES OF THE ELECTROMECHANICAL ENGINEERING CAREER OF THE TECHNICAL UNIVERSITY OF COTOPAXI”

Authors: Juan Carlos Capa Cando

Luis Alexander Pilamunga Sisa

ABSTRACT

The analysis and experimentation of Control Systems applied to different processes is one of the important aspects that must be consider in the academic training of the moment of visualizing each of its variables with its own controls and even more so that technology advances by leaps and bounds.

That is why the present technological proposal aims to develop a training module for the study of Control Systems using free-use hardware and software, to be carried out in the laboratory of the Electromechanical Engineering Degree of the Technical University of Cotopaxi. The device has implemented processes of temperature, light intensity, speed and position in order to be able to experiment in each of these variables the development of controllers and tuning methods. The use of free software as a data acquisition card, added to it, MyOpenLab as a free design platform for virtual instrumentation, allows real-time analysis of the processes, where the control logic is performed to stabilize each of the processes and the experimentation of tuning methods. One of the potentialities of the module is that it allows the connection with different controllers that can range from microcontrollers to PLCs regardless of the model and brand, thereby allowing the implementation of classic controls such as ON-OFF and PID control.

Keywords: Free Software, Free Hardware, Control Systems, PID Control, ON-OFF Control.



AVAL DE TRADUCCIÓN

En calidad de Docente del Idioma Inglés del Centro de Idiomas de la Universidad Técnica de Cotopaxi; en forma legal **CERTIFICO** que: La traducción del resumen del proyecto de investigación al Idioma Inglés presentado por los señores Egresados de la Carrera de **INGENIERÍA ELECTROMECAÁNICA** de la **FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS**, **JUAN CARLOS CAPA CANDO Y PILAMUNGA SISA LUIS ALEXANDER**, cuyo título versa "**DESARROLLO DE UN MÓDULO DE ENTRENAMIENTO PARA EL ESTUDIO DE SISTEMAS DE CONTROL, EMPLEANDO HARDWARE Y SOFTWARE LIBRE, PARA LOS LABORATORIOS DE LA CARRERA DE INGENIERÍA ELECTROMECAÁNICA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI**", lo realizaron bajo mi supervisión y cumple con una correcta estructura gramatical del Idioma.

Es todo cuanto puedo certificar en honor a la verdad y autorizo a los peticionarios hacer uso del presente certificado de la manera ética que estimaren conveniente.

Latacunga, septiembre del 2020

Atentamente,

Mg. BOLÍVAR MAXIMILIANO CEVALLOS GALARZA
DOCENTE CENTRO DE IDIOMAS
C.C. 0910821669



1. INFORMACIÓN BÁSICA

PROPUESTO POR:

- Juan Carlos Capa Cando
- Luis Alexander Pilamunga Sisa

TEMA APROBADO:

“DESARROLLO DE UN MÓDULO DE ENTRENAMIENTO PARA EL ESTUDIO DE SISTEMAS DE CONTROL, EMPLEANDO HARDWARE Y SOFTWARE LIBRE, PARA LOS LABORATORIOS DE LA CARRERA DE INGENIERÍA ELECTROMECAÁNICA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI”

CARRERA:

Ingeniería Electromecánica

DIRECTOR DE LA PROPUESTA TECNOLÓGICA:

Ing. Byron Paúl Corrales Bastidas MsC.

EQUIPO DE TRABAJO:

- Juan Carlos Capa Cando
- Luis Alexander Pilamunga Sisa

Y por el docente Tutor:

Ing. Byron Paúl Corrales Bastidas MsC.

LUGAR DE EJECUCIÓN:

Laboratorio de la Carrera de Ingeniería Electromecánica de la Universidad Técnica de Cotopaxi

TIEMPO DE DURACIÓN DE LA PROPUESTA:

Duración de proyecto de Mayo a Septiembre del 2020

FECHA DE ENTREGA:

Septiembre 2020

LÍNEA DE INVESTIGACIÓN:

La presente propuesta tecnológica se acoge a la línea de investigación de Procesos Industriales, estipuladas en el área de Investigación Científica y Desarrollo Tecnológico de la Universidad Técnica de Cotopaxi.

SUB-LÍNEA DE INVESTIGACIÓN:

De acuerdo a las sub-líneas- de la Carrera de Ingeniería Electromecánica, se relaciona a la segunda línea, que corresponde al área de automatización, control y protecciones de sistemas electromecánicos.

TIPO DE PROPUESTA TECNOLÓGICA:

Es una propuesta tecnológica, ya que se busca desarrollar un módulo para el estudio de las curvas de comportamiento de sintonización de los procesos de Control utilizando herramientas de uso libre para la adquisición de datos y tratamiento de señales digitales y analógicas.

2. DISEÑO INVESTIGATIVO DE LA PROPUESTA TECNOLÓGICA

2.1 TÍTULO DE LA PROPUESTA TECNOLÓGICA

“DESARROLLO DE UN MÓDULO DE ENTRENAMIENTO PARA EL ESTUDIO DE SISTEMAS DE CONTROL, EMPLEANDO HARDWARE Y SOFTWARE LIBRE, PARA LOS LABORATORIOS DE LA CARRERA DE INGENIERÍA ELECTROMECAÁNICA DE LA UNIVERSIDAD TÉCNICA DE COTOPAXI”

2.2 TIPO DE ALCANCE

Con el desarrollo del módulo planteado utilizando hardware y software libre, se logrará beneficiar a cada uno de los estudiantes de la Carrera de Ingeniería Electromecánica, ya que complementará el análisis de las curvas de comportamiento de sintonización de sistemas de control de temperatura, intensidad luminosa, velocidad y posición.

2.3 ÁREA DEL CONOCIMIENTO:

Según la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO) considera que el tema a tratar en esta propuesta tecnológica tiende a abarcar todo lo relacionado con la ingeniería, industria y construcción.

Según la UNESCO, en la sub área 52 Ingeniería y profesiones afines, el cual consta con: Dibujo técnico, mecánica, metalistería, electricidad, electrónica, telecomunicaciones, ingeniería energética y química, mantenimiento de vehículos, topografía.

2.4 SINOPSIS DE LA PROPUESTA TECNOLÓGICA

La presente propuesta tecnológica tiene como objetivo el desarrollo de un módulo de entrenamiento para el estudio de Sistemas de Control utilizando hardware y software de uso libre, para el laboratorio de la Carrera de Ingeniería Electromecánica de la Universidad Técnica de Cotopaxi; entendiéndose las aplicaciones de los Sistemas de Control, Sistemas Digitales, Control de Automatas Programables, Instrumentación, Electrónica y Electrónica de Potencia, es necesario el material de entrenamiento en los que se puede llevar a cabo prácticas de control on/off, control proporcional y control proporcional integral y derivativo de temperatura, intensidad luminosa, velocidad, cambio de giro de motores DC y control de posición de motores Step.

Todo esto enfocado hacia el estudio de las curvas de comportamiento de sintonización de los sistemas de control para las variables de temperatura, intensidad luminosa, velocidad y posición

2.5 OBJETO DE ESTUDIO Y CAMPO DE ACCIÓN

2.5.1 Objeto de estudio

Módulo de entrenamiento de Sistemas de Control

2.5.2 Campo de acción

Cada una de las áreas fundamentadas en los sistemas de control, su desempeño en cada proyecto a realizarse en los laboratorios de la Universidad Técnica de Cotopaxi, cada uno de los componentes internos en el módulo ayudarán a analizar las curvas de sintonización para el control de la variable de temperatura, intensidad luminosa, velocidad y posición.

2.6 SITUACIÓN PROBLÉMICA Y PROBLEMA

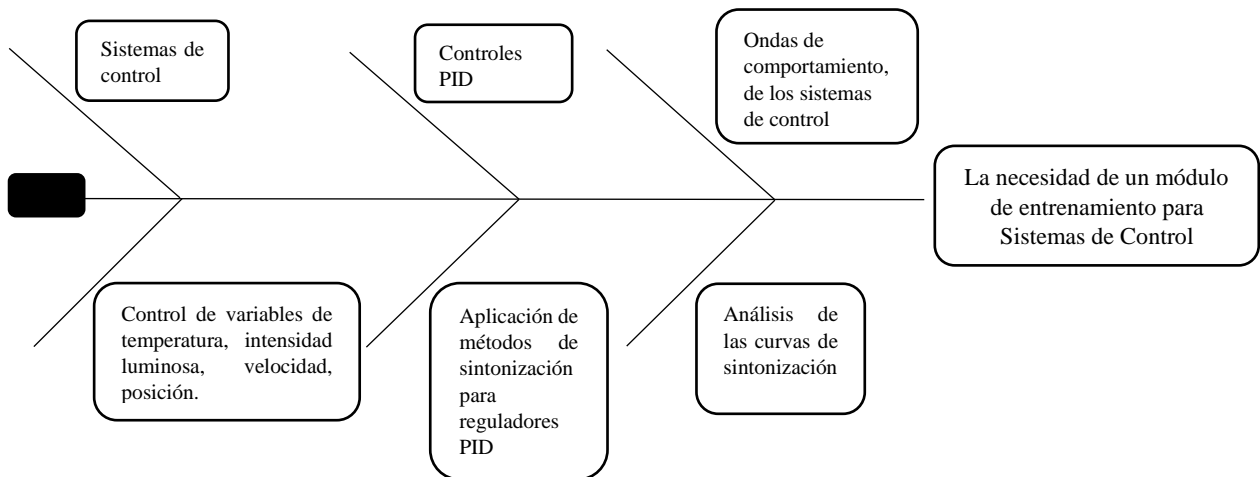
2.6.1 Situación Problemática

Dentro de las asignaturas que se imparten en la Carrera de Ingeniería Electromecánica de la Universidad Técnica de Cotopaxi se encuentran: Sistemas de Control, Sistemas Digitales, Control de Automatas Programables, Instrumentación, Electrónica y Electrónica de

Potencia; que van ligadas a un estudio teórico y práctico, es necesario el estudio de sistemas de control on – off, control proporcional, control proporcional integral y control proporcional integral derivativo, para las variables de temperatura, intensidad luminosa, velocidad, posición, ya que permite el análisis de las curvas, ondas amortiguadas de comportamiento y de sintonización de los sistemas.

La importancia que tiene el estudio de los Sistemas de Control para la aplicación en procesos de fabricación es tal que existen profesionales encargados del diseño y gestión del desarrollo de los equipamientos, utilizados para el monitoreo y control de maquinaria en procesos de producción a nivel mundial. El desarrollo de procesos de automatización, permiten optimizar tiempos de producción y gestión, recursos; alcanzando una adecuada gestión financiera y administrativa con ventajas obviamente significativas dentro del trabajo de este tema. Muchas ideas relevantes se dan a lo largo de esta demostración como sustituciones de diferentes lógicas tanto análogas y ahora digitales estos brindas efectividad en cada proceso que se disponga a realizarse, la investigación de nuevas interfaces de programación, sistemas de control en procesos industriales y las nuevas comunicaciones adaptadas crean la necesidad de optar por la construcción de un módulo de sistemas de control.

Matriz Causa Efecto



2.6.2 Problema

La necesidad de un módulo de entrenamiento para Sistemas de Control, Electrónica, Sistemas Digitales y Microcontroladores empleando software y hardware libre, en los laboratorios de Ingeniería Electromecánica de la Universidad Técnica de Cotopaxi, en los

que se pueda realizar prácticas de control de temperatura, intensidad luminosa, velocidad y posición enfocados al estudio de las curvas de comportamiento de cada sistema de control.

2.7 HIPÓTESIS

El diseño y construcción de un módulo de entrenamiento, empleando software y hardware libre, permitirá un análisis de las curvas de comportamiento de sintonización de los sistemas de control para las variables de temperatura, intensidad luminosa, velocidad y posición.

2.8 BENEFICIARIOS:

- Los beneficiados directos son, estudiantes y docentes que se vean interesados en la utilización del módulo, para la Carrera de Ingeniería Electromecánica y la Universidad Técnica de Cotopaxi.
- Los beneficiarios indirectos son, los lectores interesados en el desarrollo de la propuesta tecnológica.

2.9 OBJETIVOS

2.9.1 Objetivo general

- Desarrollar un módulo de entrenamiento para el estudio de Sistemas de Control, empleando hardware y software libre, para los laboratorios de la Carrera de Ingeniería Electromecánica de la Universidad Técnica de Cotopaxi.

2.9.2 Objetivos específicos

- Recopilar información de los elementos que conforman los sistemas de control de temperatura, intensidad luminosa, velocidad y posición.
- Diseñar el esquema electrónico de los elementos y la placa de circuito integrado que conforman el módulo, utilizando software de diseño de elementos finitos.
- Programar por medio de software libre, la adquisición e interpretaciones de datos obtenidos por los sensores y el accionamiento de actuadores.
- Validar el correcto funcionamiento del módulo de entrenamiento
- Desarrollar prácticas por medio de guías de laboratorio, manual de uso y mantenimiento.

2.10 DESCRIPCIÓN DE LAS ACTIVIDADES Y TAREAS PROPUESTAS CON LOS OBJETIVOS ESTABLECIDOS

Objetivo	Actividad	Resultado de la actividad	Descripción de la metodología
Recopilar información de los elementos que conforman los sistemas de control de temperatura, intensidad luminosa, velocidad y posición.	Obtención de la información sobre los elementos que conforman los	Selección de elementos y componentes que integran el módulo de entrenamiento.	Metodología inductiva, para deducir y entender la información recopilada
	sistemas de control de temperatura, intensidad luminosa, velocidad y posición.	Datos técnicos y especificación de los elementos de los diferentes Sistemas de Control.	
Diseñar el esquema electrónico de los elementos y la placa de circuito integrado que conforman el módulo, utilizado software de diseño de elementos finitos.	Diseño de la placa y distribución de los elementos en la placa de circuito integrado	placa de circuito integrado con los elementos electrónicos integrados.	Metodología experimental, al usar software de diseño de elementos finitos
	Construcción del módulo de entrenamiento.	El módulo de entrenamiento para Sistemas de Control	
Programar por medio de software libre, la adquisición e interpretaciones de datos obtenidos por los sensores y el	Adquisición de datos digitales y análogos por los sensores para la activación de los actuadores con hardware libre.	Tratamientos de los obtenidos por los sensores y activación de los actuadores.	Metodología deductiva, con el fin implementar los sistemas de control.

accionamiento de actuadores.	Diseño de los sistemas de control para temperatura, intensidad luminosa, velocidad y posición con software libre.	Obtención de la programación de los sistemas de control, capaces de interactuar con el módulo según se requiera.	
Validar el correcto funcionamiento del módulo de entrenamiento	Análisis de resultados, funcionamiento y comportamiento del módulo al realizar los sistemas de control	Recoger los datos del funcionamiento de los sistemas de control en el módulo	Por medio de un método analítico, se difiere el correcto funcionamiento del módulo
Desarrollar prácticas por medio de guías de laboratorio, manual de uso y mantenimiento.	Análisis de comunicación entre el hardware libre, módulo y la programación de los sistemas de control	Resultados de funcionamiento del módulo.	Desarrollo del manual y guías de laboratorio.
		Guías de laboratorio	
		Manual de uso y mantenimiento	

3. MARCO TEÓRICO

3.1 Antecedentes

Los profesionales Ing. Ángel Cárdenas León, Dr. Alejandro Israel Barranco Gutiérrez y Dr Francisco Javier Pérez Pinal, presentaron la implementación de un sistema difuso tipo Mamdani en una tarjeta Arduino Uno. También ellos proponen una forma condensada de calcular la defuzzicación por centroide para salidas con forma de triángulo isósceles [1].

García Martínez, José Román, Ramírez González, Luis David, Edison Cruz, Miguel y Martínez Sánchez. Parte de la Universidad Autónoma de Querétaro desarrollaron el artículo que trata sobre la Implementación de un controlador difuso de temperatura desarrollado en

un sistema embebido Raspberry, en este artículo, se presenta una aplicación práctica de un controlador difuso de temperatura desarrollado en un sistema embebido Raspberry Pi 3 [2].

Sebastián Herrera Aristizábal y Alejandro Hincapié Aorrea parte de la Universidad Tecnológica de Pereira, de Ingeniería Eléctrica previos a obtener su título de Ingeniero Eléctrico, desarrollaron la “Implementación de algoritmos de control de velocidad y torque de motores de corriente continua aplicados a un modelo de AGV multipropósito de potencia media utilizando Raspberry pi.” En este proyecto investigaron el tema de control de velocidad y torque de un motor DC utilizando un sistema de bajo costo, la cual se determinará mediante la comparación de acciones de control[3].

Por parte de la Facultad de Ingeniería Electrónica de la Universidad Politécnica Salesiana, los estudiantes Cando Sevilla Jazmani Esterlin y Calberto Oyarvide Erick Andrés, previo a la obtención del título de Ingeniero Electrónico, tomaron como tema a desarrollarse “Diseño e implementación de un prototipo de planta con un control de nivel por presión hidrostática, comunicación remota y monitoreado mediante el software MyOpenLab” El proyecto lo realizaron bajo la implementación y control, con el nivel por presión hidrostática de comunicación remota y monitoreado mediante el software MyOpenLab [4].

3.2 Sistemas de Control

Está definido como un conjunto de componentes funcionales que pueden regular su propia conducta. El propósito fundamental del sistema de control es conseguir mediante la manipulación de las variables de control, un dominio sobre las variables de salida, de modo que estas alcancen unos valores estables y requeridos a obtener, con pocas probabilidades de fallos [5].

- Garantizar la estabilidad, particularmente, ser robusto frente a perturbaciones y errores en los modelos.
- Llegar a la acción de control sobre las variables de entrada y estas que pueda ser realizable, evitando comportamientos bruscos e irreales.
- Ser fácil de implementar y cómodo de operar en tiempo real.

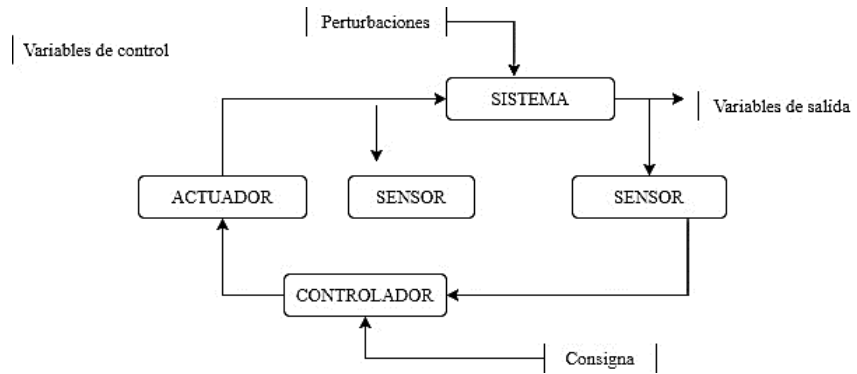


Figura 3.1 Sistema de Control

Fuente: Autores

3.2.1 Componentes de un sistema de control

A nivel industrial es de gran importancia el control automático en cada proceso que esté relacionado, con lo cual solamente se puede brindar ventajas en su utilización, podemos citar ejemplos de variables como temperatura, humedad, presión entre otras. El mantener todas estas variables estables es el objetivo de este campo [6].

Como componentes básicos tenemos los siguientes:

- Transductor (Sensor/Transmisor).
- Controlador.
- Actuador.

Operaciones correspondientes:

- Medición: la medición de la variable que se realiza con la combinación de sensor y transmisor.
- Decisión: con base en la medición, el controlador decide qué hacer para mantenerla variable en el valor que se desea.
- Acción: como resultado de la decisión del controlador se debe efectuar la acción.

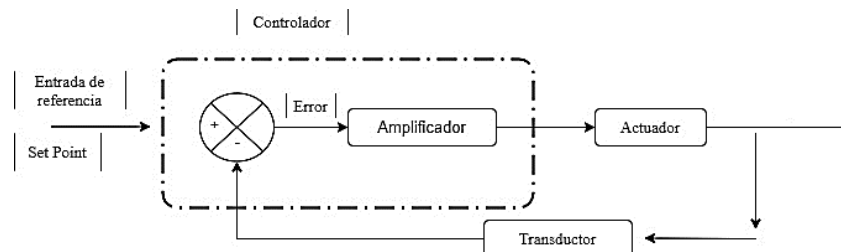


Figura 3.2 Componentes de un sistema de control

Fuente: Autores

La ilustración, muestra el diagrama de bloques de un sistema de control en lazo cerrado donde se tiene una entrada de referencia. La señal de entrada al sistema relaciona la señal de salida y el controlador la compara [6].

3.2.1.1 Actuador



Figura 3.3 Actuador Relé

Fuente: [7]

El actuador es un dispositivo que transforma la energía hidráulica, neumática o eléctrica para realizar una función que genera un efecto sobre un proceso. El actuador recibe la señal desde un controlador y en función a ella activa un elemento final de control; por ejemplo, una válvula [6]. Hay distintos tipos de actuadores cada uno según su señal de control, entre todos ellos:

- **Eléctricos:** En este actuador, su principal señal de control es la energía eléctrica.
- **Neumáticos:** Señal de control de este tipo de actuar es el aire.
- **Hidráulicos:** Señal de control en fluidos, normalmente algún tipo de aceite mineral.
- **Electrónicos:** En el empleo de dispositivos electrónicos.

3.2.1.2 Transductor

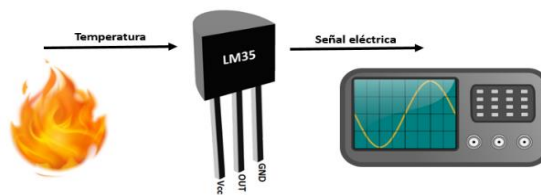


Figura 3.4 Sensor de temperatura Lm 35

Fuente: [8]

El transductor es un. dispositivo que transforma el efecto de una causa física, como la presión, la temperatura, la dilatación, la humedad, etc. En otro tipo de señal, es decir, este dispositivo toma una variable de entrada y produce una salida [6]. Este se clasifica en:

- **Directos:** Se colocan directamente en contacto con la variable se va a medir.

- **Indirectos:** Se sitúan alejados del punto de medición, pero se comunican con éste mediante una línea de transmisión.

Por su comportamiento:

- **Activos:** Son aquellos que no requieren suministro de energía para operar.
- **Pasivos:** Requieren suministro de energía para operar.

3.2.2 Tipos de señales

Las señales de un sistema de control pueden ser de dos tipos:

- **Señales analógicas:** Son señales que tienen una variación decimal dentro de un rango de valores y suelen ser eléctricos. Este rango es de 4 a 20 mA.
- **Señales digitales:** Son señales que solo pueden indicar dos valores (0 o 1). Por lo tanto, son binarias. Este tipo de señal indica al usuario si la variable está activada o desactivada con señales de 0 a 5V [9].

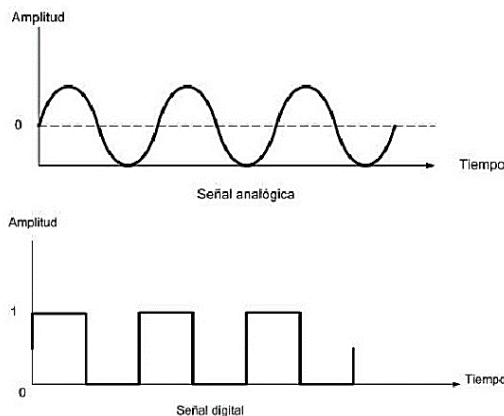


Figura 3.5 Señales analógicas y digital

Fuente: [10]

3.2.2.1 Sistema de control de lazo abierto

Son aquellos en los que la señal de salida no afecta al funcionamiento del sistema total [11].

- **Señal de mando:** indicación que da una persona haciendo que funcione el sistema
- **Unidad de control:** llamado transductor, capaz de traducir o interpretar la señal de mando establecer el valor de la señal de referencia es decir de la entrada.
- **Entrada:** señal producida por el transductor, para que el sistema opere.

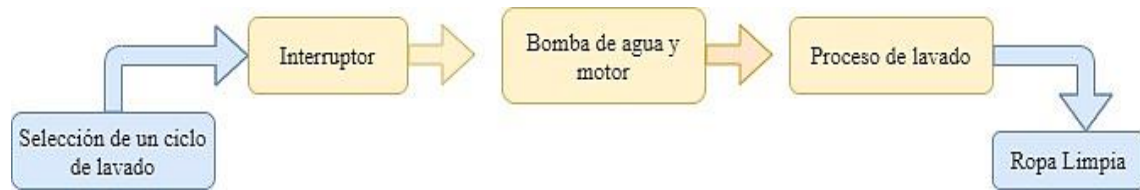


Figura 3.6 Ejemplo de sistema de control de lazo abierto

Fuente: Autores

3.2.2.2 Sistema de control de lazo cerrado

Son aquellos en los que se produce un proceso de realimentación, es decir que es capaz de modificar la señal de entrada en función de la señal de salida, la toma de decisiones no depende solo de la entrada sino también de la salida [12].

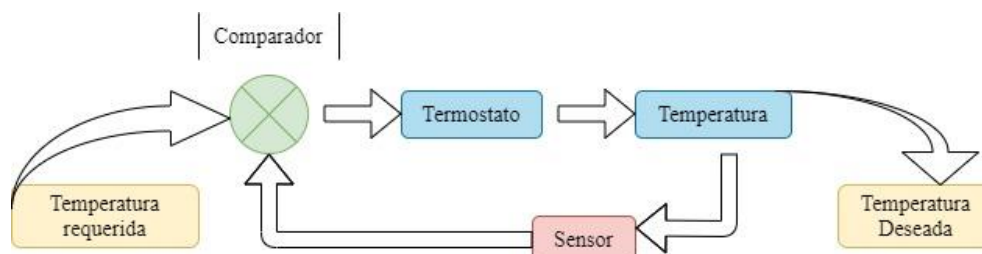


Figura 3.7 Ejemplo de sistema de control de lazo cerrado

Fuente: Autores

3.3 Tipos de Controladores

Tiene como fin comparar la variable de proceso medida de una causa física con un valor de referencia de entrada, determinar la desviación y produce una señal de control que reduce el error a un valor aproximado a cero en si detecta el error presente, por lo general muy bajo en su potencia y este a su vez lo puede intensificar [6].



Figura 3.8 Circuito integrado L293N

Fuente: [13]

3.3.1 De dos posiciones o de encendido y apagado (on/off)

También llamado “Todo o Nada”, usa un algoritmo simple que revisa si la variable de proceso está por encima o por debajo de un setpoint determinado. En términos prácticos, la variable manipulada o la señal de control del controlador cambia entre “totalmente ON” o “totalmente OFF”, sin estados alternos [14].

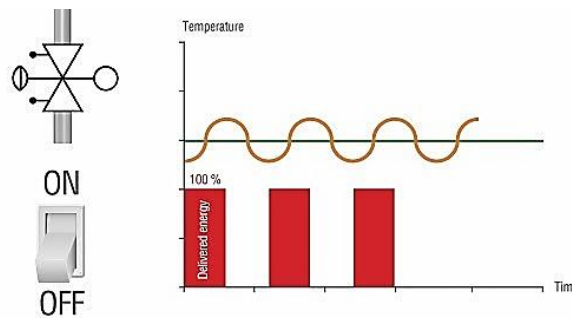


Figura 3.9 Control ON - OFF

Fuente: [7]

3.3.2 Control Proporcional

Es un tipo de sistema de control de realimentación lineal, es más complejo que un sistema de control encendido - apagado, el sistema de control tipo encendido/apagado será adecuado en situaciones donde el sistema en general tiene un tiempo de respuesta relativamente largo, pero dará lugar a un comportamiento inestable si el sistema que está siendo controlado tiene un tiempo de respuesta breve [15].

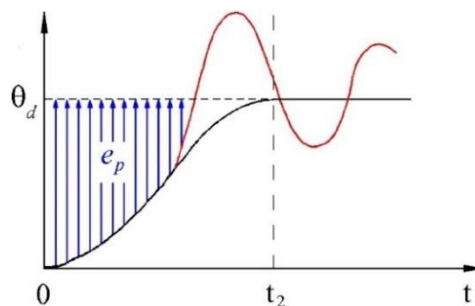


Figura 3.10 Señal de Control Proporcional

Fuente: [16]

Una analogía con el control de encendido/apagado es conducir un automóvil mediante la aplicación de potencia y variando el ciclo de trabajo para controlar la velocidad.

$$P_{out} = K_p e(t) \quad (3.1)$$

Donde:

P_{out} : Salida del controlador proporcional

K_p : Ganancia proporcional

$e(t)$: Error de proceso instantáneo en el tiempo t . $e(t) = SP - PV$

$m(t)$: salida del controlador

$$m(t) = K_p * e(t) \quad (3.2)$$

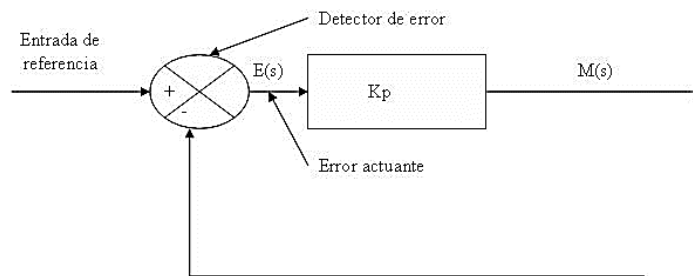


Figura 3.11 Control Proporcional

Fuente: [17]

3.3.3 Proporcional integral

En realidad, no existen controladores que actúen únicamente con acción integral, siempre actúan en combinación con reguladores de una acción proporcional, complementándose los dos tipos de reguladores, primero entra en acción el regulador proporcional mientras que el integral actúa durante un intervalo de tiempo [15].

$$G(s) = \frac{Y(s)}{E(s)} = K_p * \left(\frac{1}{T_i * S} + 1 \right) \quad (3.3)$$

Donde:

T_i : tiempo - integral.

K_p : Ganancia proporcional

La Función de transferencia del bloque de control PI responde a la ecuación, donde K_p y T_i son parámetros que se pueden modificar según las necesidades del sistema. Si T_i es grande la pendiente de la rampa, correspondiente al efecto integral será pequeña y su efecto será atenuado, y viceversa.

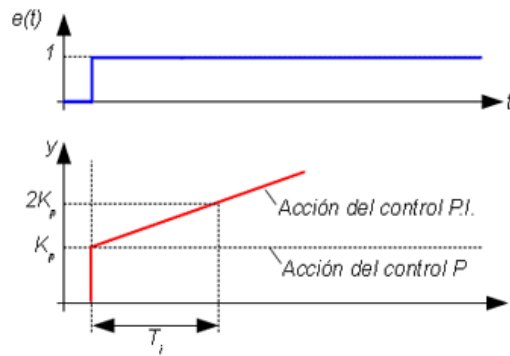


Figura 3.12 Proporcional Integral.

Fuente: [15]

El modo de control Integral tiene como propósito disminuir y eliminar el error en estado estacionario, provocado por perturbaciones exteriores y los cuales no pueden ser corregidos por el control proporcional, el control integral actúa cuando hay una desviación entre la variable y el punto de consigna, integrando esta desviación en el tiempo y sumándola a la acción proporcional. La fórmula de la integral está dada por:

$$I_{sal} = K_i \int_0^t e(t) dt \quad (3.4)$$

Donde:

K_i : es la ganancia integral, un parámetro de ajuste,

t : es la variable de integración (toma en cuenta el valor desde el instante 0 hasta el instante actual)

3.3.4 Control proporcional derivativo

El controlador derivativo se opone a desviaciones de la señal de entrada, con una respuesta que es proporcional a la rapidez con que se producen éstas. La salida de este regulador es:

$$Y(t) = t_d \frac{de(t)}{dt} \quad (3.5)$$

Que, en el dominio de Laplace, será:

$$Y(s) = T_d * s * E(s) \quad (3.6)$$

Por lo que su función de transferencia será:

$$G(s) = \frac{Y(s)}{E(s)} = Td * s \quad (3.7)$$

Donde:

$Y(t)$: Salida diferencial.

$e(t)$: Error (diferencia entre medición y punto de consigna [PC]. El PC no es otra cosa que el nivel deseado al que queremos que vuelva el sistema)

Td : Tiempo diferencial, se usa para dar mayor o menor trascendencia a la acción derivativa.

Si la variable de entrada es constante, no da lugar a respuesta del regulador diferencial, cuando las modificaciones de la entrada son instantáneas, la velocidad de variación será muy elevada, por lo que la respuesta del regulador diferencial será muy brusca, lo que haría desaconsejable su empleo [15].

$$y(t) = K p * t_d * \frac{de(t)}{dt} + K p * e(t) \quad (3.8)$$

Donde:

Kp y Td : son parámetros ajustables del sistema.

Td : es llamado tiempo derivativo y es una medida de la rapidez con que un controlador

PD : compensa un cambio en la variable regulada, comparado con un controlador P puro [15].

En el dominio de Laplace, será:

$$Y(s) = K p * T_d * s * E(s) + K p * E(s) \quad (3.9)$$

Y por tanto la función de transferencia del bloque de control PD será:

$$G(s) = \frac{E(s)}{Y(s)} = K p * (T_d * s + 1) \quad (3.10)$$

En los controladores diferenciales, al ser la derivada de una constante igual a cero, el control derivativo no ejerce ningún efecto, siendo únicamente práctico en aquellos casos en los que la señal de error varía en el tiempo de forma continua [15].

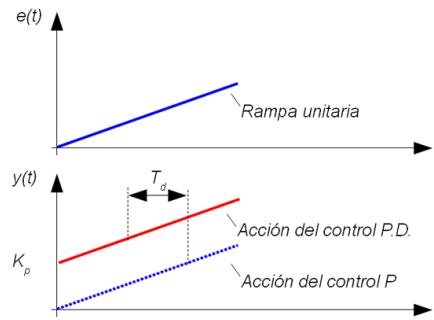


Figura 3.13 Control proporcional derivativo

Fuente: [18]

3.3.5 Proporcional integral derivativo

Es un sistema de regulación que trata de aprovechar las ventajas de cada uno de los controladores de acciones básicas, de manera que, si la señal de error varía lentamente en el tiempo, predomina la acción proporcional e integral y si la señal de error varía rápidamente, predomina la acción derivativa. Tiene la ventaja de ofrecer una respuesta muy rápida y una compensación de la señal de error inmediata en el caso de perturbaciones. Pero presenta el inconveniente de que este sistema es muy propenso a oscilar y los ajustes de los parámetros son mucho más difíciles de realizar [15]. La salida del regulador viene dada por:

$$y(t) = Kp * e(t) + Kp * td * \frac{de(t)}{dt} + Kp * \frac{1}{ti} * \int e(t) dt \quad (3.11)$$

Y por tanto la función de transferencia del bloque de control PID será:

$$G(s) = \frac{Y(s)}{E(s)} = Kp \left(a + Td * s + \frac{1}{Ti * s} \right) \quad (3.1)$$

Donde:

Kp , Ti y Td : son parámetros ajustables del sistema.

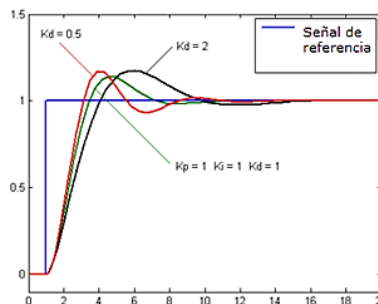


Figura 3.14 Curvas del control proporcional derivativo

Fuente: [18]

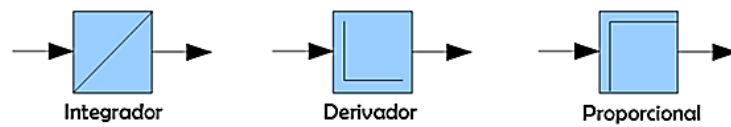


Figura 3.15 Símbolos empleados para identificar los tres tipos de controles

Fuente: [15]

3.3.6. Control Predictivo

El control predictivo tiene internamente un mecanismo de optimización el cual trata de encontrar la acción de control más óptima siempre respetando las restricciones establecidas por el ingeniero de control, dentro de una ventana de predicción, regular procesos muy complejos. Este controlador ha tenido una buena acogida en industrias petroleras y químicas dado que es un controlador avanzado [19].

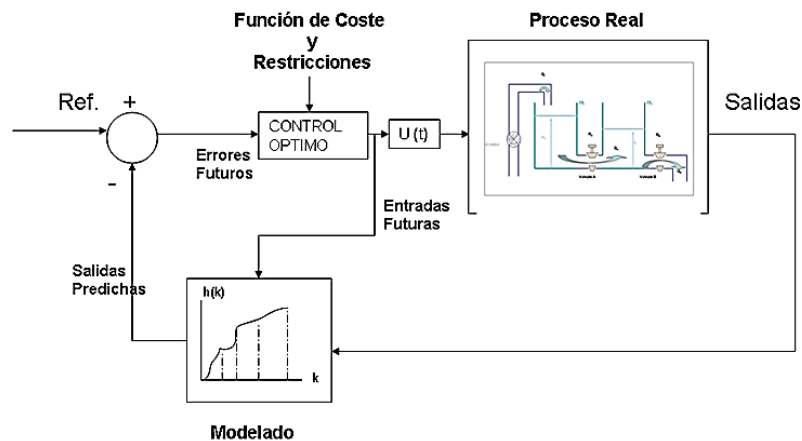


Figura 3.17 Control Predictivo

Fuente: [20]

3.4 Programación gráfica

En la programación del lenguaje visual, la interfaz gráfica se perfila como uno de los elementos más trascendentales, ya que de éste dependerá el resultado final que obtendrá cada proyecto. Es por ello que para muchos programadores la interfaz gráfica puede considerarse como un artefacto tecnológico que posibilita la representación visual e interactiva. En este caso como software de programación gráfica se verá reflejado en MyOpenLab [21].

3.4.1 Qué es programación gráfica

La programación gráfica va más allá de un lenguaje de programación, es toda una ciencia, que implica el conocimiento de los fenómenos físicos aplicados a diversos elementos, implica también mucha matemática, la iluminación, el sombreado para crear objetos, etc.

Tabla 3.1 Software que permite una programación gráfica

Visual Basic 6.0.	Es un software que permite el acceso de datos remotos de un proceso, utilizando las herramientas de comunicaciones necesarias, en el control del mismo, podemos deducir que no se trata de un sistema de control, sino de una herramienta para monitorización o supervisión que realiza la tarea de interface entre los niveles de control.
Java	Es el conjunto de componentes gráficos que posibilitan la interacción entre el usuario y la aplicación. (ventanas, botones, combos, listas, cajas de diálogo, campos de texto, etc). Se diseña la aplicación o programarla y los eventos que se generan a medida que el usuario interactúa con la Interfaz. Los componentes son objetos que heredan de la clase base/componente como: Button, List, TextField, TextArea, Label, etc.
LabView	Es un entorno de programación destinado al desarrollo de aplicaciones, similar a los sistemas de desarrollo que utilizan C o BASIC. LabVIEW se diferencia de dichos programas porque estos lenguajes de programación se basan en líneas de texto para crear el código fuente del programa, mientras que LabVIEW emplea la programación gráfica o lenguaje G para crear programas basados en diagramas de bloques.
VB NET	Es un lenguaje de programación multiplataforma y multiparadigma, de propósito general, se puede considerar una evolución de Visual Basic implementada sobre el framework .NET. Su simpleza, legibilidad y similitud con el idioma inglés lo convierten en un gran lenguaje ideal para principiantes.
Python	Es uno de los lenguajes de programación que utiliza una interfaz visual, permite programar en un entorno gráfico, realizar un gran número de tareas sin escribir código, solo se realiza operaciones con el ratón sobre la pantalla del computador.
C ++	Conocido por el nombre “C Plus”, orientado a objetos con una continuación y ampliación del C. Hay una gran cantidad de programas escritos en C++, como por ejemplo los paquetes de Adobe.
C #	También llamado “C Sharp”, orientado a objetos, desarrollado en el año 2000 por Microsoft para ser empleado en una amplia gama de aplicaciones empresariales ejecutadas en el framework .NET. C Sharp es una evolución del C y C++ que se destaca por su sencillez y modernidad.

Visual Basic.NET	Utilizado por una gran cantidad de personas que no cuentan con conocimientos profundos como desarrolladores, al contar con una sintaxis sencilla tiene la posibilidad de automatizar procesos y crear aplicaciones web.
JavaScript	No es igual a Java. Son lenguajes distintos. JavaScript es un lenguaje de programación que puede ser utilizado para crear programas que son acoplados a una página web o dentro de programas complejas. Sirve para crear efectos y realizar acciones interactivas.
PHP	Creado en 1994 por el programador canadiense Rasmus Lerdorf, cuenta con un conjunto de herramientas para el mantenimiento de las páginas web. Es de fácil acceso para nuevos programadores y ofrece grandes herramientas para gente experimentada.
SWIFT	Se trata de un lenguaje multi-paradigma creado por Apple y focalizado en el desarrollo de aplicaciones para iOS y macOS, se ha convertido en código abierto.
SQL	Este lenguaje de programación ha sido diseñado para administrar, proteger y recuperar los datos de sistemas de gestión de información y utilizado fuertemente en el desarrollo de la ciberseguridad.

Fuente: [20]

3.5 Software de programación gráfico MyOpenLab

MyOpenLab es una plataforma de desarrollo de aplicaciones basado en Java con interfaz gráfica, ejecutable en plataformas de bajo perfil, proporciona una versión estable y con conexión de diferentes protocolos de comunicación, incluido industrial como mod-BUS y OPC. Este Software surge en Alemania, iniciado por Carmelo Daniel Salafia como desarrollador, en el 2012 fue publicada sus primeras versiones y para el 2015 se desarrolla el acceso remoto en conjunto con la tarjeta Raspberry PI ejecutando en la plataforma Linux con arquitectura ARM por el Ing. Javier Velásquez y Carmelo Salafia. En Colombia se desarrollan elementos para el manejo de funciones básicas para GPIO de la Raspberry, y elementos para Arduino, control programático de las aplicaciones [22]. La programación realizada en MyOpenLab reciben el nombre de VM (Visual Modeling), el cual consta de dos partes: Circuito (panel circuito) y Panel de Visualización (Panel Frontal)

3.5.1 El Circuito

Es el conjunto de operaciones y funciones que enlazados responden a una funcionalidad relacionada con un circuito electrónico por medio de una comunicación serial. Este conjunto de funciones se edita en el correspondiente panel de edición de circuito, que aparece en la parte superior del área de trabajo del panel.

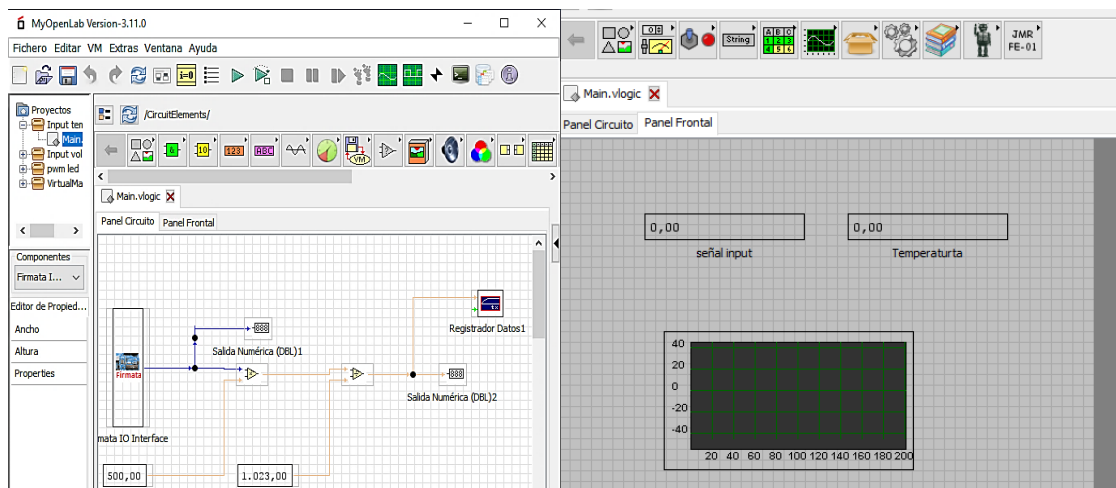


Figura 3.18 Panel Circuito y Panel Frontal

Fuente: Autores

3.5.2 Panel Frontal

En este se sitúa los objetos de visualización gráfica, asociados a las variables del circuito que permiten la visualización de la evolución de la simulación. En la figura 3.18 puede ver el aspecto de la pantalla cuando se inicia la aplicación y se carga la programación.

3.5.3 Formatos y tratamiento de Datos

MyOpenLab tiene la posibilidad de trabajar con varios datos, dependiendo de la naturaleza del dato y las conversiones en el entorno de la simulación; posee dos librerías en las que se abordan conversiones de datos y sus tratamientos, en el software podemos encontrar los siguientes tipos de datos:

Tabla 3.2 Tipos de datos en MyOpenLab

Double (dbl)	Considerados por defecto con dos decimales, al aportar datos con más decimales se producirá el redondeo a la hora de mostrar el dato.
Integer (int)	Son variables numéricas enteras, se emplea en casos donde no tiene sentido hablar de decimales como un contador del impulso.
String (str)	consiste en cualquier cadena de texto que incorpore letras, números y signos.
Booleano (bol)	son datos que pertenecen a los sistemas y operadores digitales, emplean como elementos “0” y “1”.
Var (var)	determinados bloques pueden aceptar distintos tipos de datos, produciéndose una adaptación del operador a las entradas, por ejemplo “menor que”.

Grupo (grp)	es un conjunto de datos que se transfiere de un bloque a otro y está formado por un número de datos individuales.
Fuente (font)	este dato indica el tipo de letra.
Color (col)	hacen referencia a la selección de color
Imagen (img)	Este tipo de datos en realidad responde a una imagen que se puede recuperar de un fichero de tipo (JPG, GIF, PNG, etc)

Fuente: [22]

3.5.4 Interfaces Implementadas en MyOpenLab

El software integra elementos para comunicarse con plataformas como Arduino, Raspberry PI, Android y dispositivos con Comunicación Serial, cuenta con una interface de entrada y salida, versión 1.2 para el acceso a los pines de Arduino Uno a través de una comunicación serial con protocolo Firmata, con posibilidad de ser modificada por el usuario por medio de Sketch; de igual forma permite el acceso a las funciones básicas de entradas y salidas digitales del GPIO de la Placa Raspberry PI. También permite la comunicación con la terminal del sistema Operativo de Linux, Mac y en el CMD de Windows para enviar y obtener respuesta de los comandos enviados [23] y [24]. Las características son las siguiente:

- Fácil de usar
- Amplia biblioteca de funciones para el manejo de señales análogas y digitales.
- Tratamiento y operación de diferentes tipos de datos.
- Diseño de aplicaciones mediante bloques de funciones.
- Puede crear pantallas de visualización para recoger el estado de los eventos y variables de la simulación.
- Tiene la posibilidad de ampliar las librerías de componentes.
- Crea sub-modelos que se pueden separar a su vez en otro subniveles [4].

3.6 Adquisición de datos

MyOpenLab es capaz de conectarse al mundo físico mediante Arduino UNO, Arduino MEGA, Arduino Mini, a través de la librería Firmata y cuenta con la posibilidad de acceder a las funciones básicas de salida y entrada digitales de los pines GPIO de las placas Raspberry PI.

3.6.1 Arduino

Es una placa electrónica y plataforma de desarrollo basado en hardware libre que incorpora microcontroladores reprogramables y una serie de pines que permiten establecer conexión entre el microcontrolador y diferentes sensores y actuadores, de una manera sencilla por medio de cables, Arduino nace en el 2005, en el Instituto de Diseño Interactivo de Ivrea en Italia, aparece por la necesidad de contar con un dispositivo para utilizar en aulas que fuera de bajo coste. La idea original era fabricar una placa para uso interno del instituto. Sus características son:

- Es de uso libre; cualquiera puede mejorar y ampliar el diseño del hardware, así como, el entorno de desarrollo.
- Entorno de programación en multiplataforma; es ejecutable en sistemas operativos de Windows, Mac OS y Linux, con lenguaje de programación basado en C ++.
- Bajo costo; tiene un costo de adquisición bajo e incluso, cualquiera puede construir la placa de circuito integrado.
- Es reutilizable y versátil; ya que el software de Arduino nos permite reprogramar el microcontrolador, la placa puede ser reutilizable en diversos proyectos.

3.6.2 Librería Firmata

Es un protocolo para la comunicación entre microcontroladores y un software instalado en un ordenador. Este protocolo se puede implementar en cualquier arquitectura de microcontroladores, así como en cualquier paquete de software. El objetivo de Firmata es permitir controlar completamente Arduino desde software instalado en un ordenador, sin escribir una sola línea de código en Arduino. Permitiendo instalar MyOpenLab en Linux, Mac, Windows; con soporte nativo de .NET para comunicarse con Arduino [26].

3.6.3 Comandos del protocolo Firmata

Tabla 3.3 Tabla de mensajes y bytes de datos utilizados

Tipo	Comando	Canal MIDI	Primer byte	Segundo byte
Dato analógico I/O	0xE0	pin #	LSB (bits 0-6)	MSB (bits 7-13)
Dato digital I/O	0x90	port	LSB (bits 0-6)	MSB (bits 7-13)
Reporte pin analógico	0xC0	pin #	disable/enable (0/1)	– n/a –

Reporte puerto digital	0xD0	port	disable/enable (0/1)	– n/a –
Inicio del Sistema exclusivo	0xF0			
Modo del pin establecido (I/O)	0xF4		pin # (0-127)	pin state(0=in)
Valor establecido pin digital	0xF5		pin # (0-127)	pin value (0/1)
Sistema exclusivo	0xF7			
Versión del protocolo	0xF9		Mayor versión	Menor versión

Fuente: [26]

3.6.4 Raspberry PI

Es una placa reducida que hace las funciones básicas de un ordenador, de bajo costo, implementada en Reino Unido por Raspberry Pi S.A.; con el objetivo de enseñar y estimular las Ciencias de la Computación. Su diseño incluye un Chip Systemon Broadcom BCM-2835, con unidad Central de procesamiento (CPU) con frecuencias de trabajo desde 700 MHz a 1 GHz, también posee una unidad de procesamiento Gráfica (GPU) y memoria RAM, pero no incluye unidad de almacenamiento de estado sólido o disco duro, en su lugar se utiliza tarjeta SD para el almacenamiento de información [4].

3.6.4.1 Modelos y características Raspberry Pi

A continuación, se detallará los diferentes modelos de Raspberry Pi y sus características.

Tabla 3.4 Características modelos de Raspberry PI

Raspberry Pi 1 modelo A	descatalogado, lanzado en 2012. No disponía de puerto ethernet y para su conexión necesitaba un adaptador Wi-Fi. Solo tenía 256 MB de RAM.
Raspberry Pi 1 modelo B y B+:	Es una evolución del modelo A, apareció en 2012, memoria RAM de 512 MB, con un puerto USB más y el modelo B+ tenía 4 puertos USB.
Raspberry Pi 2 modelo B	lanzado en 2014, se sustituye el procesador por otro de la misma marca, fue el Broadcom BCM2836, que contaba con cuatro núcleos a 900 MH, memoria RAM de 1 GB.
Raspberry Pi 3 modelo B	Salió en 2016, memoria RAM en 1 GB, pero el procesador pasó de 900 MHz a 1,2 GHz, con una GPU VideoCore IV a 400 MHz y por primera vez se incluye la conectividad Wi-Fi y bluetooth sin necesidad de adaptador externo
Raspberry Pi 3 modelo B+	Lanzada en 2018, las nuevas características de Raspberry Pi 3 B+ están en la mejora de la potencia del procesador de 1.2 GHz a 1.4 GHz, pero no contaba con GPU y memoria RAM de 1 Gb.

Raspberry Pi 3 modelo A+	Como procesador posee el Broadcom BCM28370B0 Cortex-A53, memoria RAM de 512 MB.
Raspberry Pi 4 modelo B:	Aparece en el año 2019, incluyen puertos USB 3.0. un procesador Broadcom BCM2711, de 4 núcleos a 1.5 GHz, con memoria RAM de 4 GB ddr4, y con puertos HDMI que soporta el 4K

Fuente: [4]

3.7 Variables de Control

Refiriéndonos textualmente al autor [8], “La variable de control es la señal que representa el nivel de control, requerido del controlador hacia el actuador del sistema físico; por ejemplo: en el caso de un autómatas, esta variable consiste en el voltaje de salida que corresponde al nivel de posición deseado por el controlador”

Tabla 3.5 Variable de control

Variable de actuación	Corresponde al nivel de actuación de carácter físico que el sistema requiere para moverse o; es la transformación de una variable de control a una variable dinámica que afecta al sistema; por ejemplo: en el caso de un autómatas, la variable de control es el nivel de voltaje de entrada y la variable de actuación es el torque del motor.
Variable de salida	Indica la señal correspondiente a la salida del sistema, es decir, la medición que queremos obtener; por ejemplo: variable de salida, la posición dinámica, y su referencia es una posición.
Variable de salida medida	Esta variable corresponde a la presencia de los elementos de sensado, en un sistema de “cero” de medición, la señal de la variable de salida va a ser la misma que la señal de la variable de salida medida.

Fuente: [27]

3.8 Sensores

Referenciado textualmente al autor [27]. “Un sensor es un dispositivo encargado de detectar y señalar una condición de cambio. Es una condición de cambio, se trata de la presencia o ausencia de un objeto o material con detención discreta, también se define como la capacidad de medir un cambio de distancia, tamaño, color, o detección de señales analógicas o digitales. Los sensores posibilitan la comunicación entre el mundo físico y el sistema de medición o control, tanto de forma eléctrica como electrónica, utilizados en todo tipo de procesos industriales y no industriales con el propósito de monitorear, medir, controlar y procesar”

Tabla 3.6 Clasificación de los sensores

<p>Según la energía</p>	<p>Activos: Emiten energía a partir de la transformación realizada; entre estos podemos mencionar las termocuplas, cristales, piezoeléctricos.</p> <p>Pasivos: Reciben energía para realizar la transformación, pueden ser las resistencias que varían según la temperatura, micrófono de condensadores, fotodiodos.</p>
<p>Según el principio de funcionamiento</p>	<p>Sensores primarios: permite obtener una señal traducible a partir de la magnitud física a medir.</p> <p>Sensores resistivos: donde varia la resistencia de un material conductor.</p> <p>Sensores de reactancia variable y electromagnética: mide la variación en el cambio del campo magnético.</p> <p>Sensores generadores: son aquellos que generan una señal eléctrica a partir de la magnitud que mide, sin necesidad de almacenar energía.</p>
<p>Según la magnitud a medir</p>	<p>Se clasifica con el fenómeno o estímulo físico que se desea, como: posición lineal o angular, desplazamiento o deformación, velocidad lineal o angular, aceleración, fuerza y par, nivel, presión, caudal, temperatura, presencia o proximidad y visión artificial.</p>

Fuente: [28] y [29]

3.8.1 Sensores de temperatura

Es un dispositivo capaz de medir las variaciones de temperatura del aire, del agua o un sólido; y transformarlas en señal eléctrica que llega hasta un sistema de control con el fin de controlar los cambios de temperatura dentro de un determinado proceso, algunos de los tipos de sensores de temperatura son los termopares, dispositivos de temperatura resistiva (RTD y Termistores, radiadores infrarrojos, dispositivos bimetálicos, dispositivos de dilatación de líquido y dispositivos de cambio de estrado.

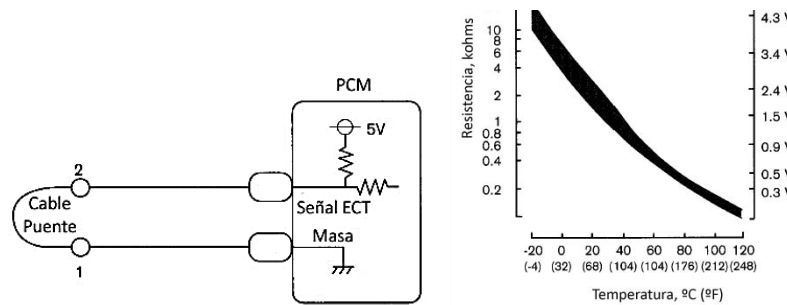


Figura 3.19 Gráfica de operación de un sensor de temperatura

Fuente: [30]

3.8.2 Sensores de intensidad luminosa

Un sensor de luminosidad o sensor crepuscular, es un dispositivo que ofrece la posibilidad de medir la variación de intensidad luminosa, permite censar la presencia de luz. Son utilizados para medir el nivel de iluminación. Al conocer el valor de la iluminación podemos corregir el nivel de iluminación de un poseso, como iluminar el área determinada [28] y [29].

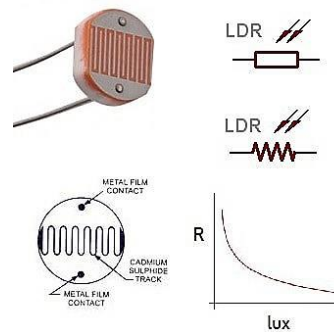


Figura 3.20 Gráfica de operación de un LDR, resistencia vs Lumen

Fuente: [31]

3.8.3 Sensores de Posición, Desplazamiento y Proximidad

Los sensores de desplazamiento miden la magnitud que se desplaza un objeto; mientras que los sensores de posición determinan la posición de un objeto en relación a una coordenada de referencia. Los sensores de proximidad son una modalidad de sensor de posición que determina en qué momento un objeto se mueve de una distancia crítica del sensor. Los sensores de desplazamiento y de posición se puede clasificar en dos tipos:

- **Sensores de contacto:** el objeto que se mide está en contacto mecánico con el sensor.
- **Sensores sin contacto:** en este no hay contacto físico entre el objetivo y el sensor.

3.8.4 Sensor de proximidad ópticos

En este tipo de sensor el dispositivo de detección de posición utiliza un haz de luz que puede ser interrumpido o reflejado por un objetivo. El sistema de control base, consiste en un emisor (fuente de luz), un receptor para detectar la luz emitida; y la asociación electrónica para evaluar y amplificar la señal causando un cambio de estado en el sensor fotoeléctrico (obstáculo) [28].

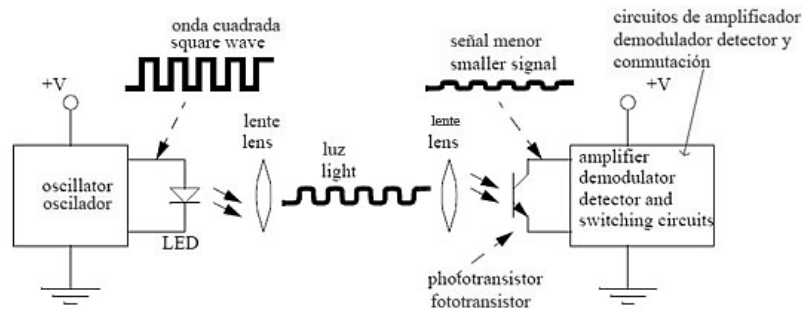


Figura 3.21 Esquema del funcionamiento de un sensor óptico

Fuente: [33]

4. METODOLOGÍA Y DISEÑO EXPERIMENTAL

En el presente capítulo se detalla los métodos utilizados para el desarrollo de la propuesta tecnológica, con el fin de cumplir los objetivos propuestos, se fundamenta los siguientes métodos:

El método inductivo: nos permitirá deducir y entender la información bibliográfica recopilada de libros, revistas, catálogos, sitios y páginas web por medio de la observación, análisis y el registro de información de los procesos industriales y de los sistemas de control, así como los diversos sensores y actuadores utilizados para el control de lazo cerrado; sustentará el trabajo teórico y práctico, también será la base para el cumplimiento de los objetivos planteados en la propuesta tecnológica.

Una vez recopilada la información, se procederá a diseñar los circuitos eléctricos en el software de diseño de elementos finitos para simular el funcionamiento y corregir errores del diseño en el control de las variables de temperatura, intensidad luminosa, velocidad y posición, todo esto ensamblado en una placa de circuito integrado

Con el método deductivo: se buscará implementar los sistemas de control para las variables temperatura, intensidad luminosa, velocidad y posición; así como el diseño definitivo de los componentes electrónicos y estructura del módulo didáctico, por medio de las conclusiones y el análisis de resultados obtenidas de las pruebas experimentales del funcionamiento del sistema de control y de los componentes eléctricos empleados en el control.

4.1 Definición de variables

En el desarrollo de la propuesta se emplean variables independientes y dependientes que se describirán a continuación:

Tabla 4.1 Señalización de variables

Variable independiente	Variable dependiente	Unidad
Módulo didáctico para sistemas de control	Temperatura	Grado Celsius (°C)
	Voltaje	Voltios (V)
	Velocidad	Revoluciones por minuto (RPM)
	Posición	Grados (°)

Fuente: Autores

Para lograr con el cumplimiento de los objetivos de la propuesta tecnológica se dividirá en tareas fundamentas, que se mencionan a continuación:

4.2 Procedimiento para el diseño de los circuitos electrónicos

Para lograr el diseño de los circuitos electrónicos que conformaran cada uno de los diferentes sistemas de control en el módulo didáctico, se establecerá una tabla de procedimientos a seguir.

Tabla 4.2 Procedimiento para el diseño de los circuitos electrónicos

Ítem	Procedimiento para el diseño de los circuitos electrónicos
Método	Diseño en software de elementos Finitos
Herramientas	Computadora y software
Procedimiento	
<ol style="list-style-type: none"> 1. Ejecutar el software y crear un nuevo proyecto. 2. Seleccionar los componentes a utilizar de las librerías del software 3. Distribuir los componentes electrónicos sobre la hoja de trabajo. 4. Conectar cada una los elementos, teniendo en cuenta la polaridad de cada elemento y los valores nominales de trabajo 5. Revisar las conexiones, nodos y puesta a tierra, con el fin de no marcar error al ejecutar la simulación 6. Por último, realizar pruebas de funcionamiento y guardar el trabajo. 	

Fuente: Autores

4.3 Diseño de la alimentación y filtrado de tensión para el módulo

Se trabajará con dos niveles de tensión, una a 5 voltios para la transmisión de la señal de los sensores de temperatura, LDR, encoders, hacia la tarjeta de adquisición de datos, así como los circuitos integrados y driver de control de los actuadores; y otro voltaje de 12 voltios

para potencia y la activación de los actuadores como lámpara, ventilador, motor DC, motor paso a paso, etc.

En la figura 4.1 se observa el diagrama electrónico de los componentes utilizados para regular el voltaje tanto de 5V. y 12V.; a continuación, se describen la característica de los componentes:

- LM-7805 es un regulador de voltaje a 5 V. y 1 A. a la salida [34]. Se utiliza un regulador por control para evitar ruido eléctrico en la señal generada por los sensores.
- Condensadores de 1000 uf. En la línea de 12 V. y 100 uf. En las líneas de 5 V.
- Diodo rectificador 1N4007 polarizado inversamente en los pines 1 y 3 de LM-7805.
- Resistencia de 220 Ω conectado a un diodo led indicador de encendido y apagado.

$$Resistencia (\Omega) = \frac{\text{tensión de alimentación} - \text{caída de tensión en el led}}{\text{corriente admisible en el led}} \quad (4.1)$$

Dónde:

Tensión de alimentación: voltaje aplicado al circuito

Caída de tensión en el led: voltaje nominal de funcionamiento del led, generalmente entre 1.7 y 3.3 voltios.

Corriente admisible en el led: corriente nominal del led, en miliamperios.

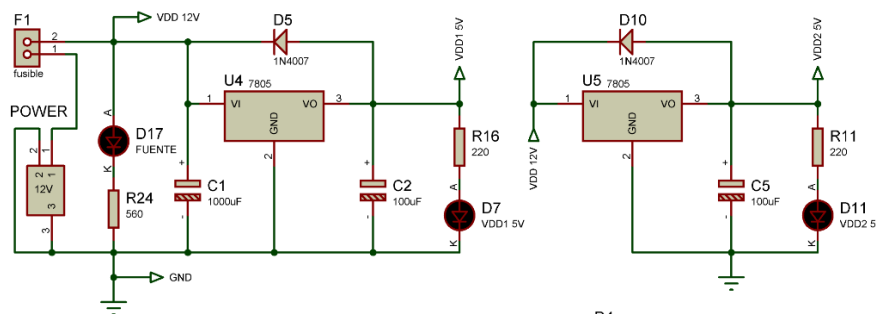


Figura 4.1 Diagrama eléctrico de la alimentación de 5 V y 12 V

Fuente: Autores

4.4 Diseño del sensor y control de temperatura

El LM-35 es un sensor de temperatura. Con rango de medición desde -55°C hasta 150°C esta calibrado para trabajar en grados Celsius, la tensión de salida es proporcional a la temperatura, con una precisión de 0.5°C a 25°C con una corriente baja de alimentación (60 μA .) y una alimentación de 4 V. hasta 20 V [35]. El LM-35 trabaja con dos configuraciones

como sensor de rango completo de temperatura (-55 °C hasta 150 °C) y sensor básico de temperatura (2 °C hasta 150 °C).

4.4.1 Sensor básico de temperatura

$$\text{Temperatura } (^{\circ}\text{C}) = \#mV \times 100 \quad (4.2)$$

Donde:

Alimentación: $V_s = 4 \text{ V. hasta } 20 \text{ V.}$

Señal de salida: $0 \text{ mV. } +10.0 \text{ mV}/^{\circ}\text{C}$

Al sensor en su pin de salida se le ha conectado un condensador de 1uf. Para estabilizar la señal de perturbaciones.

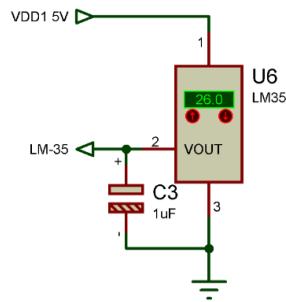


Figura 4.2 Sensor de temperatura

Fuente: Autores

4.4.2 Diseño del control de las lámparas incandescentes

Este control es implementado para aumenta la temperatura en el sensor LM-35 y variar el nivel de intensidad luminoso en el LDR, para el control de las lámparas se utiliza el circuito integrado PC-817 que es un opto-acoplador que permite aislar el circuito de control y el circuito de potencia, algunas características del PC-817 son:

- Voltaje Alto de aislamiento entre la entrada y salida igual a 5000 Vrms
- Velocidad de transferencia de corriente (CTR: MIN 50% SI = 5 mA, VCE = 5 V)
- Compacto dual-in-line package PC817: 1-canal.

Algunas de sus aplicaciones son en comunicaciones y red, administración de potencia [36]. Se emplea el transistor 2N222 como interruptor acoplado a la salida del opto-acoplador, el transistor es conector con la configuración de emisor común, la tensión de potencia entra por el colector y en la base se inyecta la corriente de saturación para abrir y cerrar contactos.

- El 2N222 es un transistor NPN
- Corriente DC Máxima del Colector 600 mA.
- Tensión Máxima Colector-Emisor 40 V.
- Disipación de Potencia Máxima 500 mW
- Tensión Base Máxima del Colector 75 V.
- Tensión Máxima Emisor-Base 6 V.
- Temperatura Máxima de Funcionamiento 175 °C.
- Tensión de Saturación Máxima Base-Emisor 2 V [37].

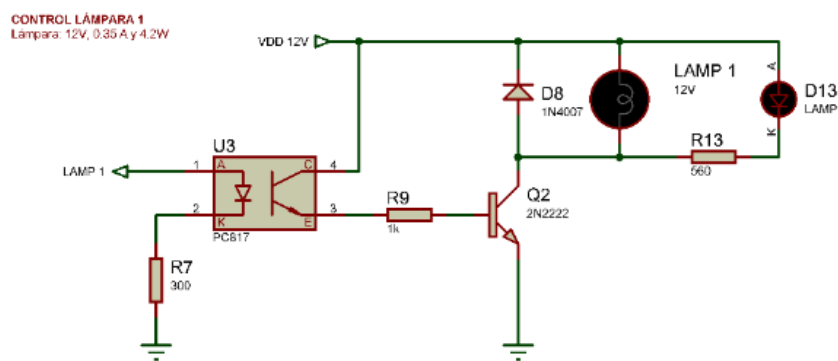


Figura 4.3 Diagrama control de lámparas

Fuente: Autores

El diagrama electrónico es similar para las dos lámparas ya que poseen las mismas características:

- Lámpara incandescente de 12 V.
- Corriente medida de 0.35 A.
- Y un consumo de 4.2 W.

4.4.3 Diseño del control del ventilador

Es necesario utilizar un actuador que acelere la caída de temperatura generada por las lámparas incandescentes.

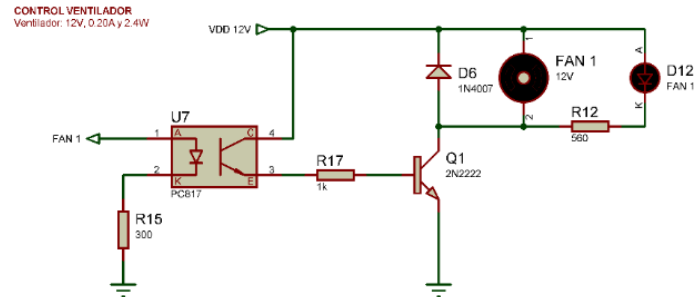


Figura 4.4 Diagrama control del ventilador

Fuente: Autores

En el diagrama se puede apreciar un ventilador de 30 x 30 mm. Con un voltaje de funcionamiento de 12 V y un consumo de 0.2 A.

4.5 Diseño del sensor y control de intensidad luminosa

El Light Dependent Resistor o LDR es un resistor que varía su valor de resistencia eléctrica dependiendo de la intensidad de luz que incide sobre su fotorresistencia, la resistencia del LDR es baja cuando la intensidad de luz es alta y en ausencia de luz la resistencia varía en los megohmios $M\Omega$.

- EL LDR tiene una tensión máxima a 25 °C igual a 150 VDC.
- Una potencia disipable de 90 mW.
- Resistencia de 50 Ω a 100 Ω K.
- Temperatura ambiente desde -25 a 75 °C [38].

Para que el LDR funcione como sensor de intensidad luminosa se debe emplear un divisor de voltaje que es un circuito de resistencias en serie el voltaje de salida es una fracción del voltaje de entrada.

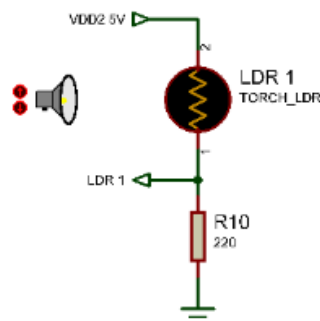


Figura 4.5 Diagrama sensor de intensidad luminosa

Fuente: Autores

Para determinar el voltaje de salida en el pin LDR 1 se emplea la siguiente ecuación:

$$V \text{ salida} = \left(V \text{ entrada} \times \frac{1}{R1 + R2} \right) R2 \quad (4.3)$$

Dónde:

V entrada: es el voltaje que alimenta el circuito.

V salida: es una fracción del voltaje de entrada.

R1 y R2: es el valor de las resistencias que conforman el divisor.

Reemplazando los datos del sensor de intensidad luminosa la ecuación final queda de la siguiente forma:

$$LDR 1 = \left(5 V \cdot \frac{1}{50 \Omega + 220 \Omega} \right) 220 \Omega$$

LDR 1 = 4.074 V; con una intensidad de luz alta

$$LDR 1 = \left(5 V \cdot \frac{1}{100 K\Omega + 0.22 K\Omega} \right) 0.22 K\Omega$$

LDR 1 = 0.0109 V; con una intensidad de luz baja.

El control de intensidad luminosa se utilizará el mismo circuito de las lámparas incandescentes, ya que nos brinda las mismas características necesarias de control.

4.6 Diseño del control del motor DC

El TB6612FNG nos permite manejar dos motores de corriente continua desde Arduino, variando tanto la velocidad como el sentido de giro. En este caso lo usaremos para controlar un solo motor y direccionar las aplicaciones antes mencionadas a este. También permite controlar intensidades de corriente altas, siendo capaz de suministrar 1.2A por canal de forma continua, y 3.2A de pico. Las dos salidas de motores A y B pueden ser controladas de manera separada (programaciones diferentes en asignación de pines), la velocidad del motor es controlada mediante una señal PWM con una frecuencia de hasta 100kHz como máximo, el pin STBY cuando es puesto en alto o high coloca al motor en modo de stand-by, también tenemos al driver con protección de diodos internos. En cada canal podemos controlar el sentido de giro y la velocidad para lo cual admite una señal PWM de frecuencia máxima de 100 kHz (muy por debajo del rango normal de PWM en Arduino). Con un voltaje de 5v y a una corriente de arranque de 1300 mA.

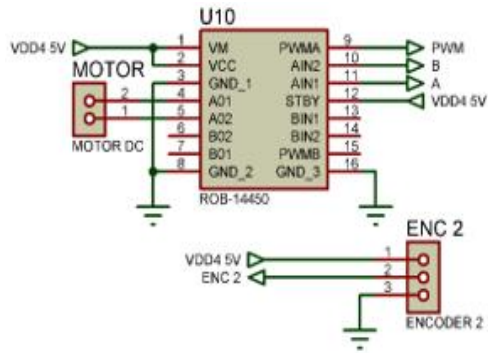


Figura 4.6 Diagrama de control del motor DC

Fuente: Autores

4.7 Diseño del control de motor Step o motor paso a paso

El L293D es un driver de 4 canales capaz de proporcionar una corriente de salida de hasta 600mA por canal y puede soportar picos de hasta 1.2 A. Cada canal es controlado por señales TTL (familia de circuitos de transistores bipolares) y cada pareja de canales dispone de una señal de habilitación para conectar o desconecta las salidas de los mismos. En esencia utilizamos este integrado para el control del motor a pasos siendo factible con sus características de brindadas en voltaje e intensidad.

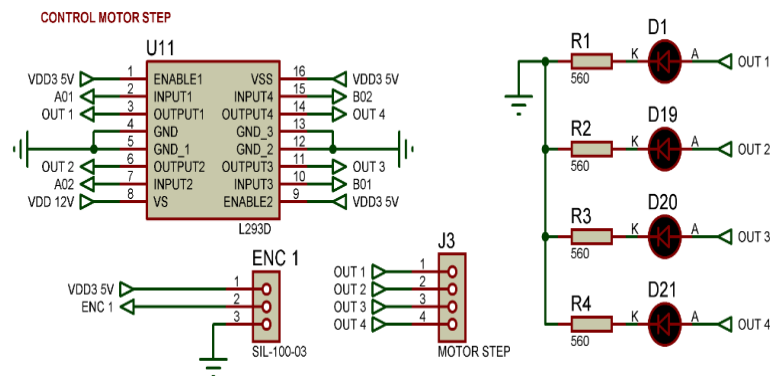


Figura 4.7 Diagrama de control del motor a pasos bipolar

Fuente: Autores

4.8 Diseño de circuito del encordar óptico para los motores

Se utiliza el sensor de velocidad IR con el driver LM393 para el control de la velocidad de rotación del motor Step y del motor DC. Se coloca una corona dentada que gira unida al eje se podría usar como un interruptor óptico, para medir las RPM en cada uno de los motores. Si se atraviesa cualquier obstáculo entre la ranura del sensor, este crea un pulso digital en el pin de salida, el pulso va de 0V a 5V y es una señal digital.

4.9 Diseño del circuito para el control del relé

La finalidad de este control es la de realizar el control de otros procesos que necesiten un sistema de control, el relé permitirá la activación y la desactivación de cualquier actuador, al igual que los controles anteriores se emplea el opto-acoplador PC-817 para el control de alta potencia. las especificaciones del relé se detallan en el anexo 7 donde se observa el diagrama final de todos los sistemas de control que conformaran el módulo, así como los pines de entrada y de salida detalladamente.

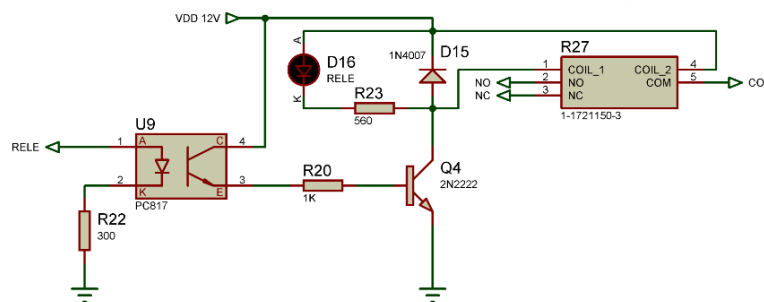


Figura 4.8 Diagrama del control del relé

Fuente: Autores

4.10 Disposición de los pines para el control de las variables

La disposición de los pines para el control de las variables de temperatura, intensidad luminosa, velocidad y posición, se encuentra enlistadas identificando las entradas y salidas de los conectores para el control como se muestra en la siguiente tabla.

Tabla 4.3. Disposición de los pines para el control de las variables

Disposición de pines para el control de las variables			
Control	Output	Input	Identificación
Temperatura	LM-35		Sensor de temperatura
		LAMP 1	Lámpara incandescente N° 1
		LAMP 2	Lámpara incandescente N° 2
		FAN 1	Ventilador
Intensidad luminosa	LDR 1		Light-Dependent Resistor (LDR)
		LAMP 1	Lámpara incandescente N° 1
Velocidad	ENC 2		Encoder motor DC
		PWM	Control de las RPM
		A	Conector positivo

		B	Conector negativo
Posición	ENC 1		Encoder Motor Step bipolar
		A01	Terminal de 1° bobina 1
		A02	Terminal de 1° bobina 2
		B01	Terminal de 2° bobina 1
		B02	Terminal de 2° bobina 2
Relé	Input	RELE	Activación del relé
	Contactos	NO	Normalmente abierto
		NC	Normalmente cerrado
		COM	Común

Fuente: Autores

4.11 Ruteo de las pistas en la placa de circuito integrado

Ya definido el diseño del circuito para el control de las variables, se debe diseñar la estructura de la placa de circuito integrado en donde se definirá la distribución y posición de los componentes, así como las pista y buses de material conductor, a continuación, el proceso:

Tabla 4.4. Ruteo de las pistas en la placa de circuito integrado

Ítem	Ruteo de las pistas en la placa de circuito integrado de los circuitos eléctricos
Método	Diseño en software de elementos Finitos
Herramientas o equipos	Computadora y software
Procedimiento	
<ol style="list-style-type: none"> 7. Crear un archivo para el diseño de la placa de circuito integrado o PCB. 8. Definir el tañado y límites de la PCB sobre la hoja de trabajo. 9. Distribuir y ordenar los componentes electrónicos sobre la placa PCB 10. Realizar el ruteo de las pistas sobre la PCB, tomando en cuenta el espesor, tamaño y características que deberá tener la pista 11. Colocar las conexiones de energía, tierra y señal, orientar los circuitos de potencia y control. 12. Revisión y corrección de la distribución de los componentes, pistas con el fin de que exista coherencia en el diseño de la PCB 13. Realizar pruebas de funcionamiento en el software. 	

Fuente: Autores

En el anexo 9.1 se puede observar la distribución de los componentes electrónicos sobre la placa de circuito integrado, así como el ruteo de las pistas.

4.12 Impresión de la placa de circuito integrado y acople de elementos

Finalizado en diseño de la placa de circuito integrado se procede con la impresión sobre la superficie de la baquelita, se realizar el acople de los componentes electrónicos y se procede con la soldadura de los mismo, a continuación, se enlista las tareas para completar este proceso:

Tabla 4.5. Impresión de la placa de circuito integrado y acople de elementos

Ítem	Impresión de la placa de circuito integrado y acople de elementos
Método	Transferencia por calor y soldadura de componentes por estaño
Herramientas o equipos	Pinzas, alicate, cortafrío, cautín
Procedimiento	
<p>14. Se imprime el diseño de la placa de circuito integrado sobre una hoja de papel utilizando tinta no corrosiva con cloruro férrico, se transfiere la impresión por medio de calor hacia la baquelita.</p> <p>15. Se realiza el tratamiento con el cloruro férrico y se obtiene las pistas ya definidas.</p> <p>16. En el reverso de la baquelita se imprime la nomenclatura de los componentes, nombres y señalización adicional.</p> <p>17. Se sueldan los componentes en la baquelita, respetando la polaridad de cada uno.</p> <p>18. Se revisa la correcta unión de los componentes y se corrigen errores.</p>	

Fuente: Autores

En el anexo 9.2 se poder ver la placa de circuito integrado impresa y con los componentes electrónicos.

4.13 Diseño y construcción de la caja del módulo

El diseño de la caja o gabinete del módulo debe cumplir con la necesidad para acoplar la placa de circuito integrado, se debe proporcionar los soportes para fijar los componentes adiciones como el ventilador, motor DC y freno mecánico; debe proporciones el espacio físico para el acople de los encoder de los motores, también debe permití una visualización, señalización adecuada con fines didácticos; el procedimiento a seguir en el siguiente:

Tabla 4.5. Diseño y construcción de la caja del módulo

Ítem	Diseño y construcción de la caja del módulo
Método	Diseño en software de elementos Finitos y corte láser
Herramientas o equipos	Computadora y software CAD

Procedimiento
<p>19. Crear un nuevo archivo CAD</p> <p>20. Teniendo previamente las dimensiones del módulo se crear la estructura de la caja, también se diseña los soportes para los motores DC, Step y para los encoders, el freno mecánico y los otros componentes que conforma el módulo</p> <p>21. Definido el diseño estructural del módulo se procede a corta con láser en acrílico.</p>

Fuente: Autores

4.14 Diseño de los sistemas control para las variables en el Software libre

Para la creación de los diferentes sistemas de control es necesario seguir un proceso que garantice la ejercitación correcta del mismo, a continuación, se detalla el procedimiento a seguir:

Tabla 4.6. Diseño y construcción de la caja del módulo

Ítem	Diseño de los sistemas control
Método	Diseño en software de elementos Finitos
Herramientas o equipos	Computadora y software de control
Procedimiento	
<p>22. Crear un nuevo proyecto dentro del software</p> <p>23. Se abre el archivo. vlogic en donde se encuentra las ventanas del panel circuito y panel frontal, donde se va a trabajar.</p> <p>24. Se realiza la comunicación por el puerto COM en donde se conectó la tarjeta de adquisición de datos (Arduino) y se configura los pines de entrada y salida para los diferentes sistemas de control.</p> <p>25. Se acondiciona la señal de entrada dependiendo del proceso, para trabajar en el sistema de unidades elegido: grados Celsius (°C), voltios (V), revoluciones sobre minuto (RPM) y grados (°).</p> <p>26. Se crea el control en la ventana de panel circuito y en la ventana del panel frontal se crea la visualización que tendrá la aplicación una vez arranque la máquina virtual VM.</p> <p>27. Se realizar pruebas de funcionamiento de cada aplicación, se corrigen errores.</p>	

Fuente: Autores

4.14.1 Control ON / OFF de temperatura

En la tabla 4.6 se detalló los pines para el control de temperatura, para los diferentes de sistemas de control de temperatura tanto ON/OFF como proporcional y PID:

- Input: sensor de temperatura (LM-35)
- Output: lámpara incandescente N°1, N°2 y ventilador

Las lámparas incandescentes se utilizan para aumentar la temperatura al nivel de Setpoint pre establecido y el ventilador se emplea para acelerar el descenso de la temperatura cuando esta sea mayor al setpoint.

- Temperatura > Setpoint = ventilador On y lámparas Off
- Temperatura < Setpoint = ventilador Off y lámparas On.

Par el control de temperatura primero se debe acondicionar la señal del sensor en la programación, se debe emplear la siguiente fórmula para transformar un voltaje de 5V:

$$Analog\ input\ (dbl) = 100 \times Analog\ input\ (int) \times \frac{5}{1023} \quad (4.4)$$

Dónde:

Analog input (dbl): señal del sensor en los pines análogos (dato flotante)

X 100: Temperatura (°C) = #mV x 100 hablada en la ecuación (4.2)

Analog input (int) x 5 / 1023: constante de transformación de una señal (dato entero), a un dato flotante en MyOpenLab.

Acondicionada la señal del sensor es comparada con un setpoint de temperatura y se establece las condiciones para el funcionamiento del control On/Off:

- If (Setpoint <= sensor) Out = True (bol) (dato booleano) en el ventilador
- If (Setpoint >= sensor) Out = True (bol); en las lámparas.

Si estas condiciones no se cumplen el dato booleano será un FALSE (bol) a la salida de la comparación, la programación se puede ver en el Anexo 15.

4.14.2 Control ON / OFF de intensidad luminosa

Pines para el control de intensidad luminosa:

- Input: señal del LDR
- Output: lámpara incandescente N°1

Acondicionamiento de la señal para el control, se emplea la siguiente fórmula:

$$\text{Analog input (dbl)} = \text{Analog input (int)} \times \frac{5}{1023} \quad (4.5)$$

Dónde:

Analog input (dbl): señal del sensor en los pines análogos (dato flotante)

Analog input (int) x 5 / 1023: constante de transformación de una señal (dato entero), a un dato flotante en MyOpenLab.

Condiciones para el funcionamiento del control On/Off de intensidad luminosa

- If (Setpoint > sensor) Out = True (bol) (dato booleano) en la lámpara N°1

Si esta condición no se cumple el dato booleano será un FALSE (bol) a la salida de la comparación. Esta condición se puede visualizar en el Anexo 16.

4.14.3 Control PID de temperatura

Al igual que el control On/Off de temperatura lo primero que se debe acondicionar la señal de entrada del sensor de temperatura, para esto se emplea la formula (4.4) ya mencionada, para este control se utiliza el componente Regulador PID que está dentro de las librerías de MyOpenLab, este componente posee 5 pines de entrada y un pin de salida como se puede ver en la figura 4.10, donde (e) es la desviación del control o error, (ta) es la ganancia, (kp) (ki) (kd) son las contantes proporcional, integra y derivativa, y el pin de salida (y) es la señal de salida del control PID.

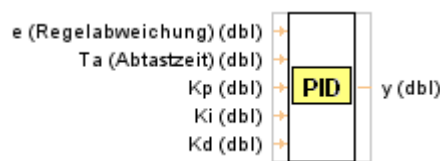


Figura 4.9 Regulador PID de MyOpenLab

Fuente: Autores

Este regulador PID trabaja con datos flotantes (dbl), es necesario acondicionar esta señal de salida; pasar de un dato flotante (dbl) a un dato entero (int), para que sea conectada a los pines de salida de la tarjeta de adquisición de datos Arduino; se establece las siguientes condiciones:

- If ((y) >= 255.00) out 1 = true (bol)
- If ((y) <= 0.00) out 2 = true (bol)

Esta señal booleana se conecta a dos comparadores para obtener una señal de dato entero (int), continuación se establece las condiciones:

- If (out 1) true = 255 (int), y If (out 1) false = 0 (int); esto en la salida del primer comparador (salida 1).
- If (out 2) true = 0 (int), y If (out 1) false = salida 1 (int): esto en la salida del segundo comparador (salida 2).
- La salida 2 va a variar de (0 a 255) (int); esto equivale a (0V – 5V) señal control y (0V – 12V) señal de potencia.

La programación de este control PID se puede observar en el anexo 17; esta programación será empleada para el control P, control PI, control PID de temperatura.

4.14.4 Control PID de intensidad luminosa.

Al igual que el control On/Off de intensidad luminosa se debe acondicionar la señal de entrada del LDR, para esto se emplea la formula (4.5) ya mencionada, para el control se utiliza el componente Regulador PID que está dentro de las librerías de MyOpenLab; y se puede observar en la figura 4.9 Al igual que el control PID de temperatura este trabaja con datos flotantes (dbl), es necesario acondicionar esta señal de salida, pasar de un dato flotante (dbl) a un dato entero (int); para que sea conectada a los pines de salida de la tarjeta de adquisición de datos Arduino; se establece las siguientes condiciones:

- If ((y) >= 255.00) out 1 = true (bol)
- If ((y) <= 0.00) out 2 = true (bol)

Esta señal booleana se conecta a dos comparadores para obtener una señal de dato entero (int), continuación se establece las condiciones:

- If (out 1) true = 255 (int), y If (out 1) false = 0 (int); esto en la salida del primer comparador (salida 1).
- If (out 2) true = 0 (int), y If (out 1) false = salida 1 (int): esto en la salida del segundo comparador (salida 2).
- La salida 2 va a variar de (0 a 255) (int); esto equivale a (0V – 5V) señal control y (0V – 12V) señal de potencia.

La programación de este control PID se puede observar en el anexo 19; esta programación será empleada para el control P, control PI, control PID de Intensidad Luminosa.

4.14.5 Control motor DC

Para lograr el funcionamiento y posteriormente el control del motor DC se utiliza un motor de 5 (V) muy amigable para realizar los controles mencionados, se aplica la comunicación Firmata IO Interface, consiste en comunicarnos a través de Arduino con la interfaz de MyOpenLab, adicional a esto se programa un controlador a base de Flip - Flop como recepción de datos de comunicación y salidas biestables para el debido funcionamiento de las solicitudes requeridas obviamente para el control. Establecemos condiciones de comunicación como lo es:

- In Para la variación de PWM
- In para entrada digital de carácter booleano
- In para entrada digital del complemento del dato booleano
- Una salida de comunicación de tipo entero para su lectura.

Todo esto lo podremos ampliar mediante el anexo 21 en donde se muestra las aplicaciones que se le dan en las prácticas correspondientes.

4.14.6 Control motor STEP

Para el control del motor a pasos se emplea un motor Step bipolar que posee 48 pasos y ángulo de paso de 7.5° a lo cual utilizamos una comunicación semejante a Firmata, pero con ciertas modificaciones en el código de comunicación. Ya que para el control de este tipo de motor necesitamos crear nuevas variables para que así se logre llegar al punto exacto de recepción de datos y simulación dentro de la plataforma de MyOpenLab. Podemos observar los diferentes datos a ser ingresados en nuestra tarjeta de adquisición Arduino detalladas en el anexo 23.

5. DESARROLLO DE LA PROPUESTA (ANÁLISIS Y DISCUSIÓN DE RESULTADOS)

En el presente capítulo se describe los resultados obtenidos en el desarrollo del módulo para Sistemas de Control y se analizará los resultados de las pruebas de los controles para la variable de temperatura, intensidad, velocidad y posición en el módulo por medio de gráficas y tablas obtenidos del software empleado.

5.1 Módulo para sistemas de control

Empleado la metodología descrita en el capítulo anterior, se logró obtener el módulo didáctico para sistemas de control, el cual permitirá realizar pruebas con el fin de verificar su correcto funcionamiento, en el anexo 9.2 y 10 se puede visualizar el módulo dictado y los terminales de entrada y salida para el control de las variables de temperatura, intensidad luminosa, velocidad y posición.

5.2 Interface Firmata

Para acceder a los pines y funciones de Arduino desde el software MyOpenLab se debe cargar el sketch “Firmata estándar” a la placa. Desde el programa Arduino se puede cargar el código del protocolo, en el anexo 31 se puede observar la ventana para acceder al código.

5.3 Pruebas de comunicación entre Arduino y MyOpenLab

MyOpenLab cuenta con una carpeta dentro del aparatado de Proyectos que permiten realizar pruebas de comunicación con Arduino por medio del protocolo Firmata, en una subcarpeta llamada VirtualMachines, hay prueba de comunicación tanto en español como en inglés, en el anexo 32 se puede visualizar los ejemplos mencionados. MyOpenLab identifica automáticamente si hay alguna tarjeta Arduino conectado en los puestos COM, en el caso de no hacerlo se seleccione Firmata IO Interface, se despliega el editor de propiedades y se selecciona manual el puerto de comunicación.

5.4 Prueba del control ON/OFF de temperatura.

En el anexo 15 se muestra la guía de laboratorio para el control ON/OFF de temperatura, en donde se detalla el procedimiento, la programación en el software y la forma de conexión.

El control está gobernado por un Setpoint en el que se introduce el rango de temperatura deseada, cuando en controlador alcanza la temperatura deseada las lámparas incandescentes se apaga y se enciende el ventilador según lo requiera el sistema. Para la prueba se establecer un setpoint de 50 °C, y se observa en la figura 5.2 el comportamiento del control.

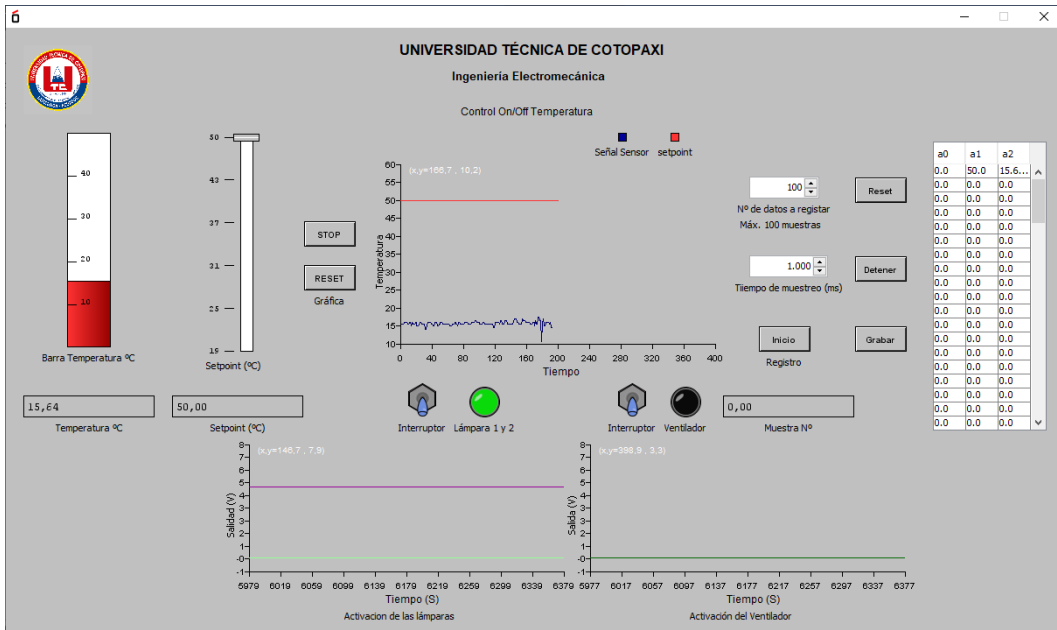


Figura 5.1 Panel frontal control On/Off de temperatura

Fuente: autores

En el anexo 24 se muestra algunos de los datos obtenidos del comportamiento del control al establecer un Setpoint de 50 °C, el tiempo de muestreo es 1 segundo.

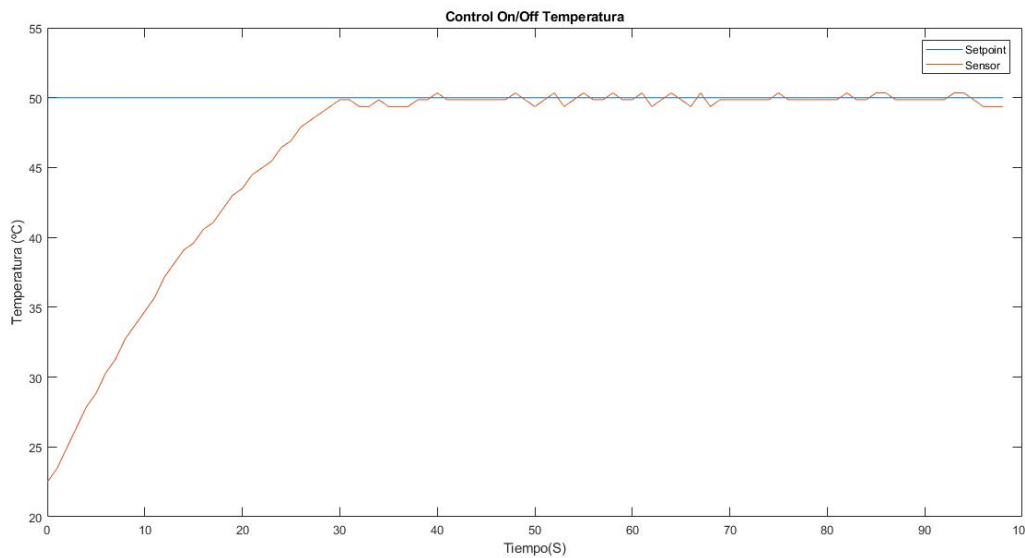


Figura 5.2 Control On/Off de temperatura con Setpoint de 50 °C

Fuente: autores

Para el análisis del error del control, se toma los datos del sensor cuando este alcanza el setpoint deseado (desde el dato N°29 hasta el dato N° 98), para el promedio.

$$\text{Promedio} = \frac{\sum_{i=29}^{98} \text{Datos del Sensor}}{69} = 49.85 \text{ (}^{\circ}\text{C)}$$

$$\text{Exactitud del control} = \frac{100\% * 49.85 \text{ }^{\circ}\text{C}}{50 \text{ }^{\circ}\text{C}} = 99.70 \%$$

$$\text{Error absoluto} = 100\% - 99.70\% = 0.29 \%$$

Se obtuvo un error absoluto entre el setpoint y el sensor de 0.2932 % con los datos obtenidos del control en MyOpenLab, como se visualiza en el anexo 24.

5.5 Prueba de control PID de temperatura y sus variantes

La programación del control PID se encuentra en el anexo 18, para definir las constantes del control se utilizó métodos de sintonización mencionados en el marco teórico, se analizó la forma de onda del control abierto antes una setpoint de 40.48 °C, obteniendo las siguientes funciones de transferencias, en la tabla 5.1 se define las constantes Kp, Ki, Kp calculadas.

Tabla 5.1 Constantes Kp, Ki, kd calculadas para PID de Temperatura

Método	Ecuación	Kp	Ki	Kd
Ziegler Nichols	$G(S) = \frac{0,159 * e^{-0,01s}}{42.2 s + 1}$	3.1704448872	0.075129	0.01585224
PID Tuner (Matlab)	$G(S) = \frac{0,272 * e^{-0 s}}{48.9328 s + 1}$	6.557	0.13441	0.017959
Tune your PID	$G(S) = \frac{0,1913 * e^{-0 s}}{50.9666 s + 1}$	6.52734033	0.1280070	0

Fuente: autores

Con la finalidad de analizar el comportamiento de la forma de onda se implementó en el panel frontal del control, un registrador de datos como se ve en la figura 5.3, en pruebas se trabajó con una ganancia de 0.1597 y como constantes Kp=3.1704, Ki=0.075129 y Kd=0.015852, en la figura 5.4 se puede observar el comportamiento del controlador PID de temperatura con un setpoint de 40.48 °C, los datos fueron recopilados con un tiempo de muestreo de un 1 segundo en el anexo 25 se visualiza algunos de los datos obtenidos.

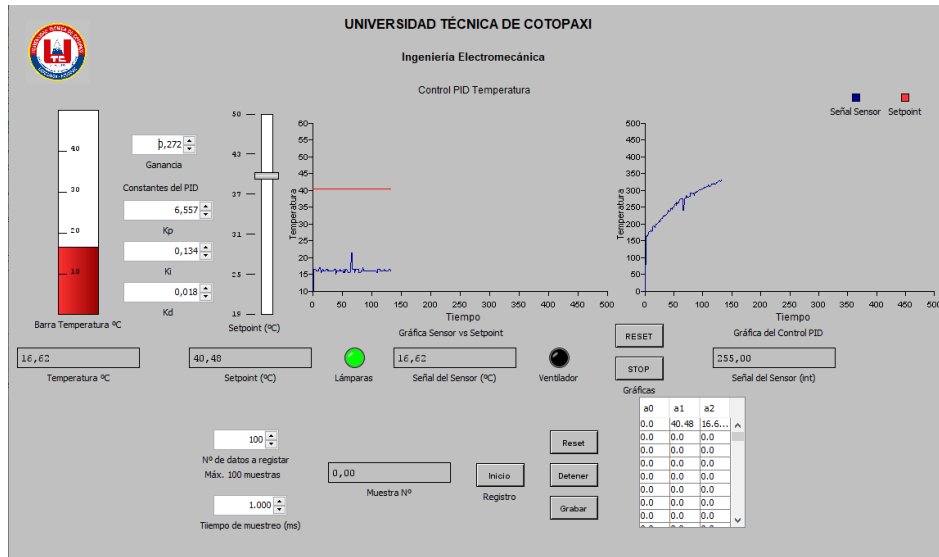


Figura 5.3 Panel frontal control PID de temperatura

Fuente: autores

En la figura 5.4 se observa el comportamiento del control PID de temperatura con un Setpoint de 40,48 °C y con constantes de Ziegler Nichols y PID Tuner (Matlab)

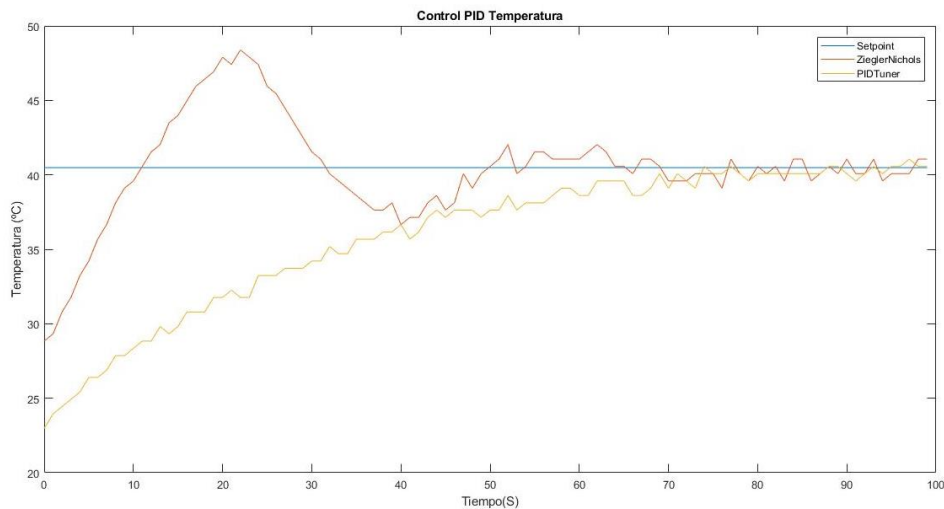


Figura 5.4 Control PID de Temperatura

Fuente: autores

Para el análisis del error de sintonización se trabajó con las constantes del PID Tuner, se toma los datos del sensor cuando este alcanza el setpoint deseado (desde el dato N°69 hasta el dato N° 99), para el promedio.

$$\text{Promedio} = \frac{\sum_{i=69}^{99} \text{Datos del Sensor}}{30} = 40.14 \text{ (}^{\circ}\text{C)}$$

$$\text{Exactitud del control} = \frac{100\% * 40.14 \text{ }^{\circ}\text{C}}{40.48 \text{ }^{\circ}\text{C}} = 99.16 \%$$

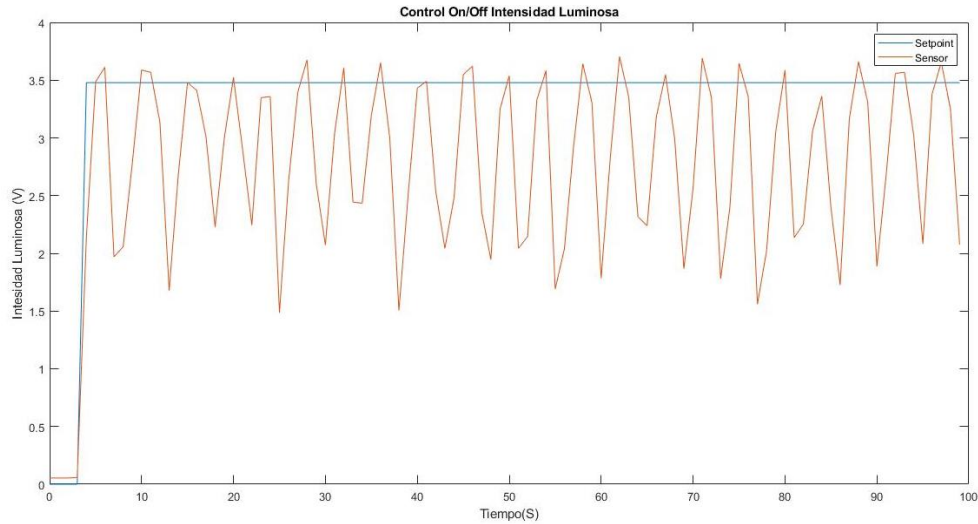


Figura 5.6 Control On/Off de Intensidad luminosa con setpoint de 3.5 (V) equivalente al 90 (%)

Fuente: autores

Para el análisis del error del control, se toma los datos del sensor cuando este alcanza el setpoint deseado (desde el dato N°5 hasta el dato N° 99), para el promedio.

$$\text{Promedio} = \frac{\sum_{i=5}^{99} \text{Datos del Sensor}}{94} = 2.84 \text{ (V)}$$

$$\text{Exactitud del control} = \frac{100 \% * 2.84 \text{ (V)}}{3.47 \text{ (V)}} = 81.80 \%$$

$$\text{Error absoluto} = 100\% - 81.80 \% = 18.19 \%$$

Se obtuvo un error absoluto entre el setpoint y el sensor de 18.19 % con los datos obtenidos del control en MyOpenLab, como se visualiza en el anexo 26.

5.7 Pruebas de control PID de Intensidad Luminosa y sus variantes

La programación del control PID se muestra en el anexo 27, para definir las constantes del control se utilizó métodos de sintonización mencionados en el marco teórico, se obtiene la forma de onda del control abierto antes una setpoint de 3,6591 (V), se establece la función de transferencias, y en la tabla 5.2 se define las constantes Kp, Ki, Kp calculadas.

Con la finalidad de analizar el comportamiento de la forma de onda se implementó en el panel frontal del control, un registrador de datos como se ve en la figura 5.7, en pruebas se trabajó con una ganancia de 68.61 y como constantes Kp=0.016, Ki=0.027 y Kd=0, en la figura 5.8 se puede observar el comportamiento del controlador PID de intensidad luminosa

con un Setpoint de 3,6591 (V) equivalente al 100 (%) de intensidad, los datos fueron recopilados con un tiempo de muestreo de un 1 segundo en el anexo 27 se visualiza los datos obtenidos.

Tabla 5.2 Constantes Kp, Ki, kd calculadas para PID de Intensidad luminosa

Método	Ecuación	Kp	Ki	Kd
Ziegler Nichols	$G(S) = \frac{68.61 * e^{-0,01s}}{0.5775s + 1}$	1.01	0.2	0
PID Tuner (Matlab)	$G(S) = \frac{68.61 * e^{-0,01s}}{0.5775s + 1}$	0.016	0.027	0
Tune your PID	$G(S) = \frac{67.6 * e^{-0,01s}}{0.5265s + 1}$	0.93	0.2	0

Fuente: autores

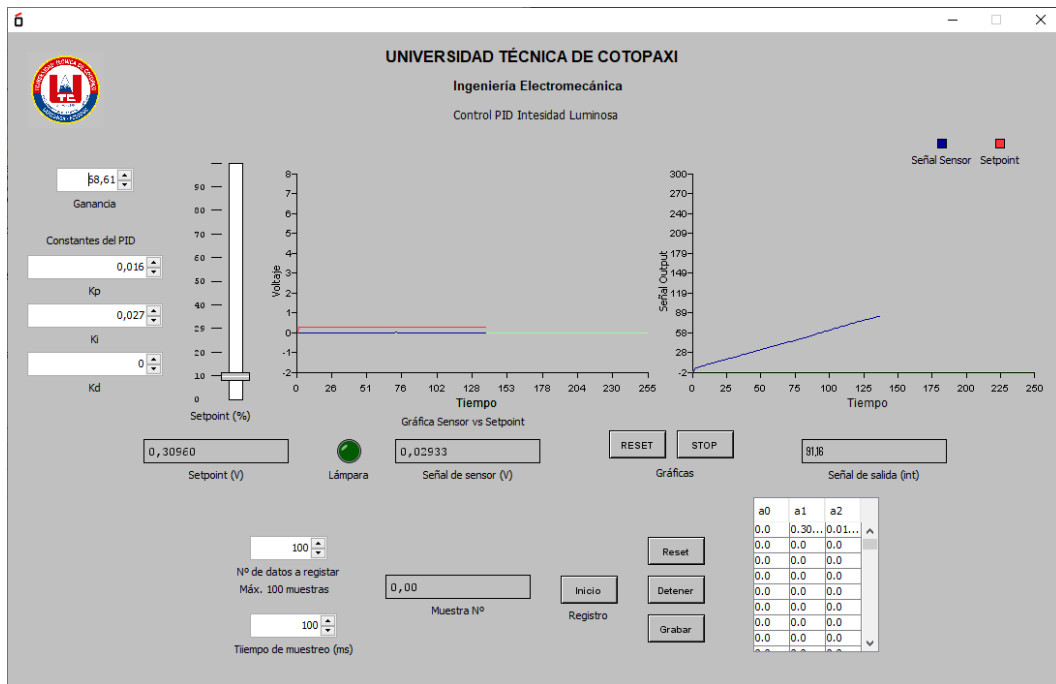


Figura 5.7 Panel frontal control PID Intensidad luminosa

Fuente: autores

En la figura 5.8 se observa el comportamiento del control PID de Intensidad luminosa, se definió un setpoint de 100 (%), el tiempo de respuesta es más dimano que el control PID de temperatura, el control es instantáneo, se presenta un pico en la sintonización del control.

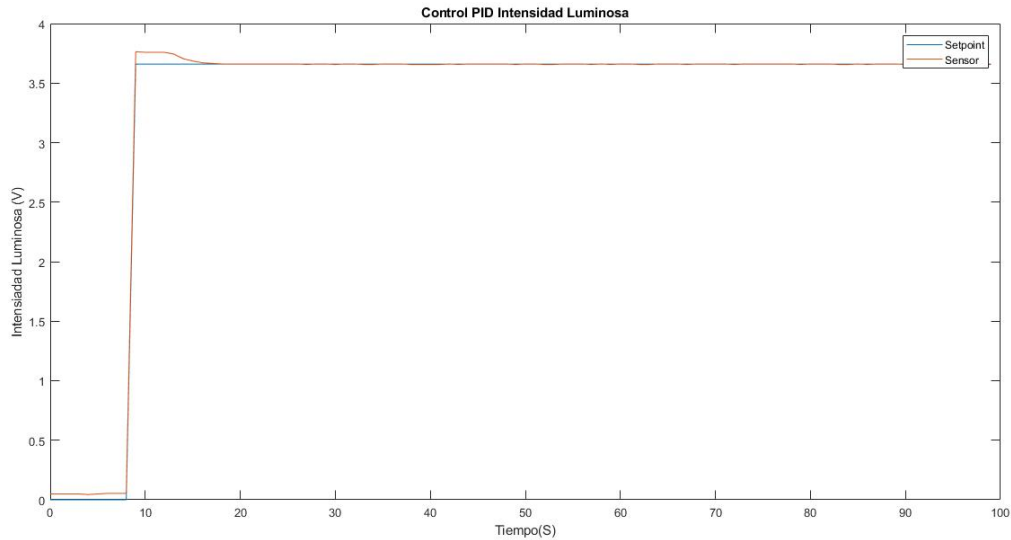


Figura 5.8 Control PID de Intensidad luminosa – PID Tuner Matlab

Fuente: autores

Para el análisis del error de sintonización, se toma los datos del sensor cuando este alcanza el setpoint deseado (desde el dato N°11 hasta el dato N° 99), para el promedio.

$$\text{Promedio} = \frac{\sum_{i=11}^{99} \text{Datos del Sensor}}{88} = 3.66 \text{ (V)}$$

$$\text{Exactitud del control} = \frac{100\% * 3.66 \text{ (V)}}{3.65 \text{ (V)}} = 100.17 \%$$

$$\text{Error absoluto} = 100 \% - 100.17 \% = - 0.17 \% \text{ (contantes PID Tuner)}$$

Se obtuvo un error absoluto entre el setpoint y el sensor de -0.17% (PID Tuner) con los datos obtenidos del control en MyOpenLab, como se visualiza en el anexo 27.

5.8 Prueba de posición del motor Step

La programación y el funcionamiento se detalla en el anexo 23, para esto se utiliza un motor a pasos bipolar de 48 pasos, se lleva a cabo la disposición de giro en grados de 45, 90, 180, 270, 360 utilizando como punto de referencia el ángulo cero (0°). El movimiento es realizado a través de matrices que se muestra en el anexo 23.1 que da la activación de cada una de las bobinas del motor y por ende la posición que se le requiera dar, se dispone de un encoder óptico para la referencia de ángulo cero (0°), en la figura 5.9 se muestra el panel frontal.

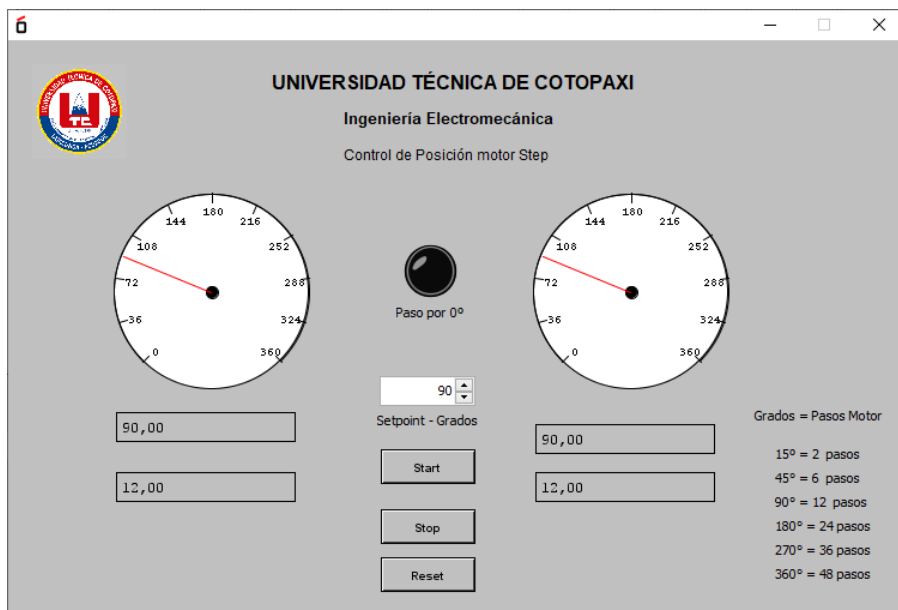


Figura 5.9 Ángulo de 90° en la prueba de posición

Fuente: autores

5.9 Pruebas del control de velocidad

La programación del control PID en el motor de velocidad se visualiza en el anexo 28, para determinar las constantes del control se utilizó métodos de sintonización mencionados en el marco teórico, se analizó la forma de onda del control abierto ante una Setpoint de 4400 (RPM), obteniendo las siguientes funciones de transferencias, en la tabla 5.3 se define las constantes K_p , K_i , K_d calculadas. Con la finalidad de analizar el comportamiento de la forma de onda se implementó en el panel frontal del control, un registrador de datos como se ve en la figura 5.8, en pruebas se trabajó con una ganancia de 0.01 y como constantes $K_p=0.017012$, $K_i=0.050374$ y $K_d=0$, en la figura 5.10 se puede observar el comportamiento del controlador PID de RPM con un Setpoint de 4400 (RPM), los datos fueron recopilados con un tiempo de muestreo de un 1 segundo en el anexo 21 se muestra estos valores mencionados

Tabla 5.3 Constantes K_p , K_i , k_d calculadas para PID de Velocidad

Método	Ecuación	K_p	K_i	K_d
Ziegler Nichols	$G(S) = \frac{15.51 * e^{-0,52s}}{0.183s + 1}$	0.03	1.04	0.26
PID tuner (Matlab)	$G(S) = \frac{18.71 * e^{-0,732s}}{0.1259s + 1}$	0.017012	0.050374	0
Tune your PID	$G(S) = \frac{16.2 * e^{-0,70s}}{0.132 s + 1}$	0.01	1.40	0

Fuente: autores

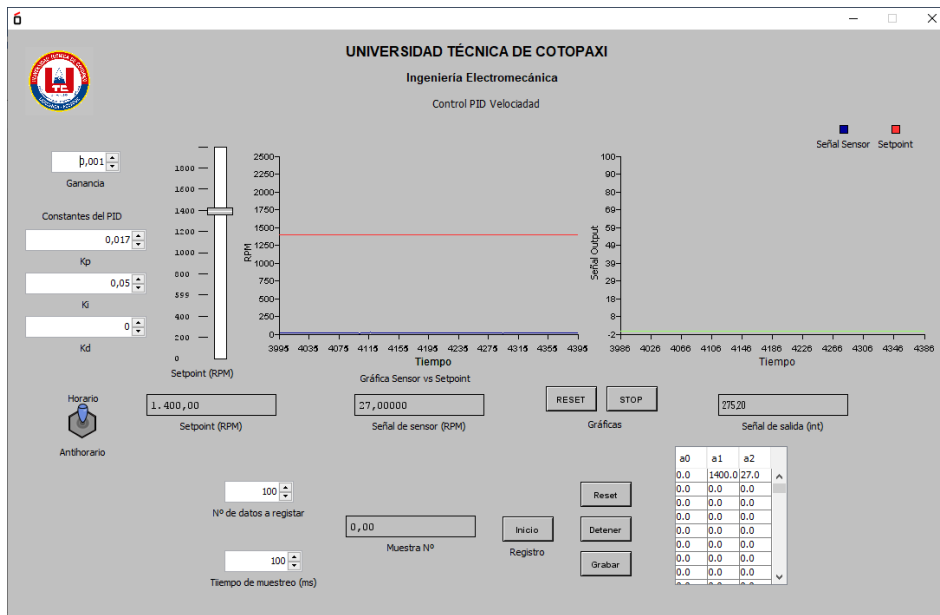


Figura 5.10 Panel frontal control PID del motor DC

Fuente: autores

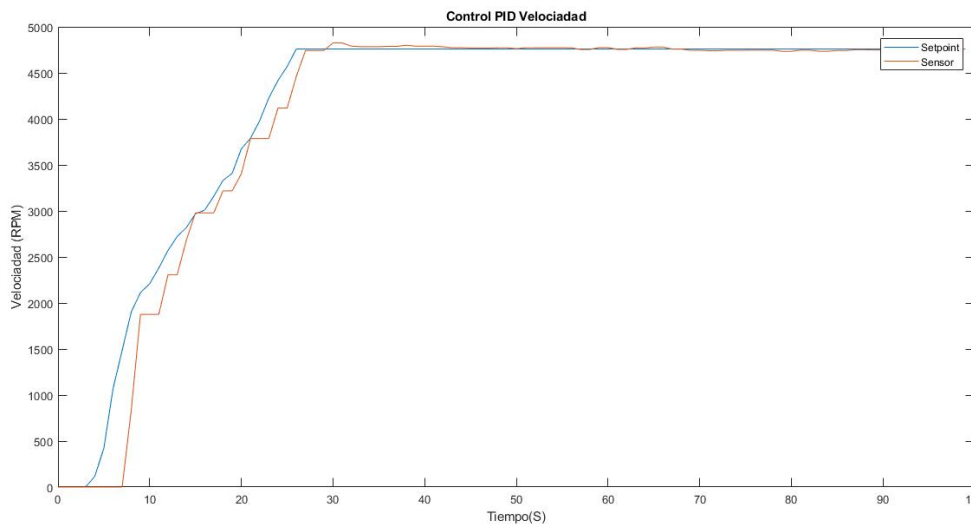


Figura 5.11 Control PID de velocidad con Setpoint de 4.400 (RPM)

Fuente: autores

Para el análisis del error de sintonización, se toma los datos del sensor cuando este alcanza el setpoint deseado (desde el dato N°16 hasta el dato N° 99), para el promedio.

$$\text{Promedio} = \frac{\sum_{i=16}^{99} \text{Datos del Sensor}}{83} = 4746.69 \text{ (RPM)}$$

$$\text{Exactitud del control} = \frac{100\% * 4746.69 \text{ (RPM)}}{4745 \text{ (RPM)}} = -0.03\%$$

$$\text{Error absoluto} = 100 \% - 100.17 \% = - 0.03 \% \text{ (contantes PID Tuner)}$$

Se obtuvo un error absoluto entre el setpoint y el sensor de -0.03% (PID Tuner) con los datos obtenidos del control en MyOpenLab, como se visualiza en el anexo 27.

5.10 Prueba de velocidad del motor DC

La guía para la práctica de control de velocidad se encuentra en el anexo 21, el control tiene un Setpoint que gobierna el porcentaje de velocidad deseado, en la figura 5.12 se muestra el panel frontal del control.

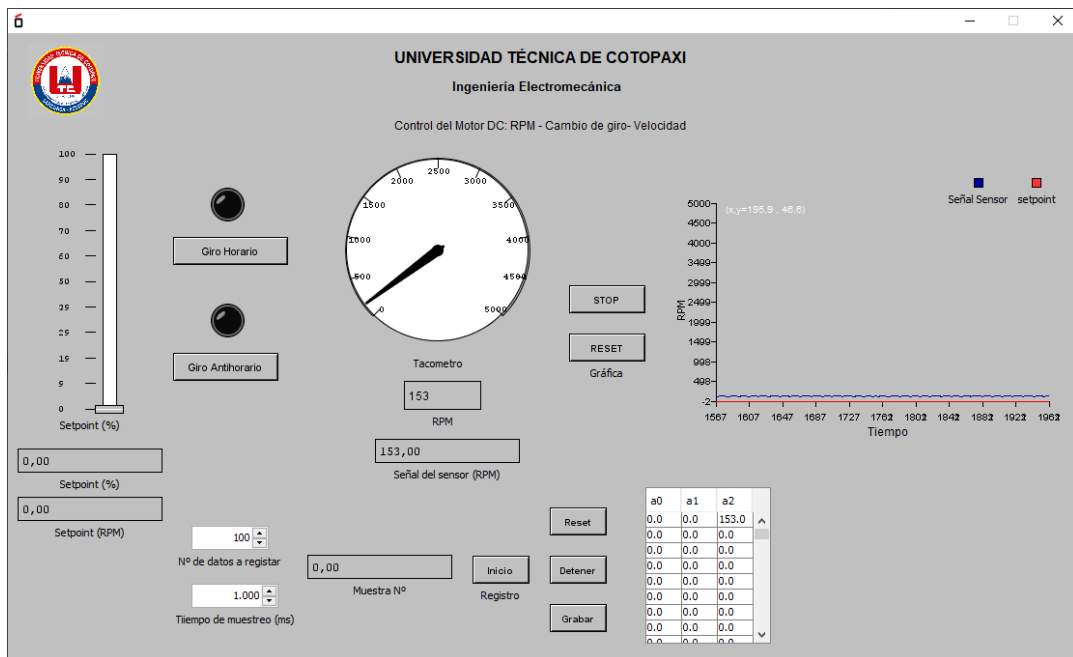


Figura 5.12 Panel frontal control de RPM del motor DC

Fuente: autores

Se define un Setpoint de 2803 (RPM) equivalente al 68% de la velocidad total, se muestra algunos de los datos obtenidos del control con un tiempo de muestreo de 1 Segundo. En la figura 5.13 se observa el comportamiento del control de RPM. La forma de onda que se obtiene que se obtiene del control de RPM, podemos observar que se presenta una caída de (RPM) que es el resultado del giro horario en el tiempo 0.7 (ms).

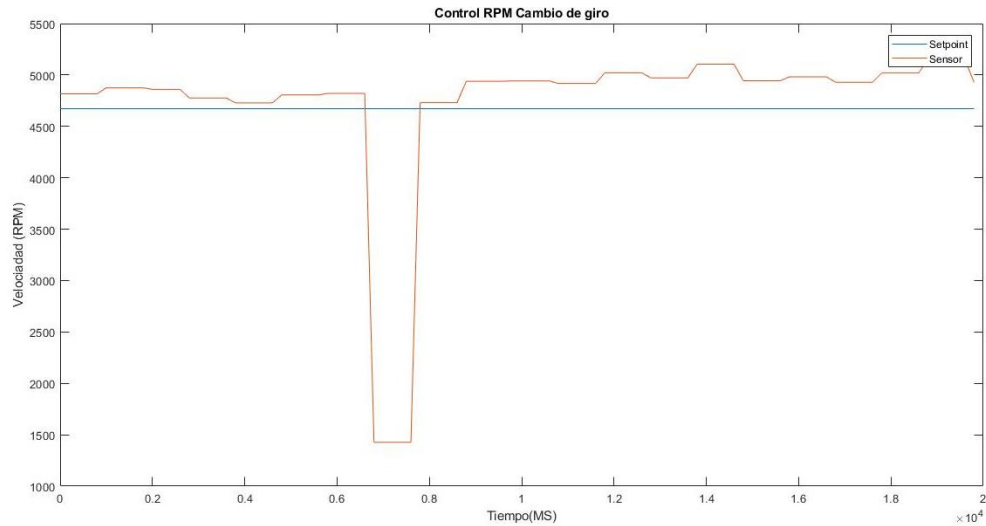


Figura 5.13 Control RPM del motor DC e inversión de giro en la muestra N° 50

Fuente: autores

5.11 Prueba de activación del relé

En el módulo se dispone de un relé mecánico de activación general, este contiene 3 contactos (normalmente cerrado, normalmente abierto y contacto común) que serán empleadas según sea la necesidad del programa, se emplea un accionamiento directo por medio de un interruptor de dos posiciones On / Off dispuestos en la programación de MyOpenLab, en la figura 5.14 podemos ver la interfaz de funcionamiento del relé.

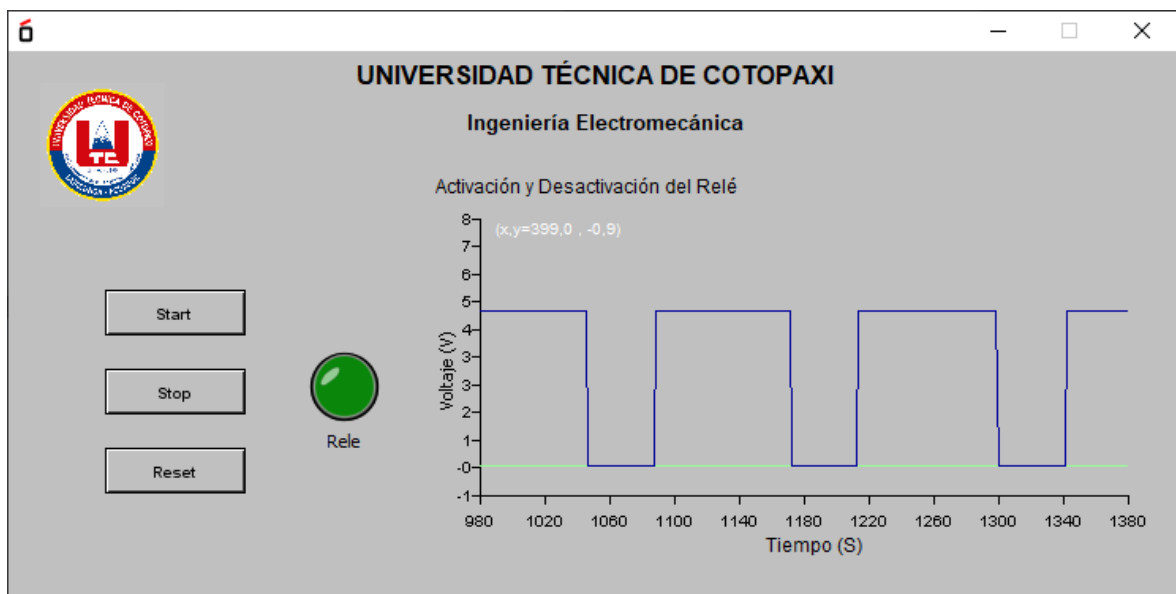


Figura 5.14 Activación del Relé

Fuente: autores

6. IMPACTOS (TÉCNICOS SOCIALES, AMBIENTALES O ECONÓMICOS)

La propuesta tecnológica se sustenta en un análisis técnico de los impactos en las diferentes áreas o ámbitos que involucra a las siguientes:

6.1 Impacto tecnológico

La finalidad del módulo es fomentar la utilización de nuevos equipos y dispositivos tecnológicos, en el que interviene la interacción entre el módulo, el software, el equipo de adquisición de datos y el estudiante, este se adapta a los requerimientos del docente y estudiante, pues por medio de diferentes componentes se puede recibir o emitir señales analógicas y digitales, emitiendo órdenes a actuadores o recibiendo señales de los sensores para transformarlas y procesarlas según el requerimiento del sistema de control, incentivando al estudiante el desarrollo de programas por medio de una interface virtual y visual como lo brinda el software gratuito MyOpenLab.

6.2 Impacto práctico

El módulo permite la realización de pruebas de sistemas de control en la Carrera de Ingeniería Electromecánica, este consta de sensores y actuadores con los que se pueden receptor la señal y transformarlas por medio de una tarjeta de adquisición de datos, con la finalidad de realizar prácticas de control de temperatura, velocidad, posición, luminosidad, para fortalecer la enseñanza teórica con lo práctico.

6.3 Impacto económico

El impacto económico de la propuesta tecnológica se ve reflejado en la disminución de costos comparados a productos similares de uso prepago, se evita costos elevados con la adquisición de materiales existentes en el mercado los cuales son accesibles y fáciles de adquirir, considerando que sus características técnicas sean las apropiadas para cada función a emplearse.

7. VALORACIÓN ECONÓMICA

En la tabla 7.1 se muestra a detalle los costos de componentes electrónicos, materiales y consto de construcción del módulo.

Tabla 7.1 Presupuesto para el desarrollo de la protesta tecnológica

Presupuesto de componentes y material electrónico					
Recursos		Cantidad	Unidad	V. Unitario	V. Total
Motor DC		1	c/u	\$3,00	\$3,00
Motor Step		1	c/u	\$4,00	\$4,00
Ventilador 12 V		1	c/u	\$2,00	\$2,00
Encoder Óptico		2	c/u	\$1,45	\$2,90
LDR		1	c/u	\$0,75	\$0,75
sensor LM 35		1	c/u	\$2,25	\$2,25
Resistencias	560 Ohm.	9	c/u	\$0,04	\$0,36
	300 Ohm	8	c/u	\$0,04	\$0,32
	1 Kohm.	4	c/u	\$0,04	\$0,16
lámparas 12v tipo uña		2	c/u	\$0,35	\$0,70
Octo-acoplador LM-817		4	c/u	\$0,20	\$0,80
Diodo 1N4007		8	c/u	\$0,09	\$0,72
Relé 12 V, 5 pines		1	c/u	\$0,89	\$0,89
Regulador LM 7805		4	c/u	\$0,56	\$2,24
Driver DC TB6612		1	c/u	\$4,00	\$4,00
Capacitores	1000 uf.	1	c/u	\$0,22	\$0,22
	100 uf.	4	c/u	\$0,08	\$0,32
	1 uf.	1	c/u	\$0,05	\$0,05
Transistor 2n222		4	c/u	\$0,15	\$0,60
Diodo led 5mm.	Amarillo	1	c/u	\$0,08	\$0,08
	Rojo	4	c/u	\$0,08	\$0,32
	verde	7	c/u	\$0,08	\$0,56
zócalos CI 8x8		1	c/u	\$0,08	\$0,08
estaño 4 metros.		1	c/u	\$2,14	\$2,14
termo-incogible 1mm		1	c/u	\$0,13	\$0,13
conector mólex 4 pines		1	c/u	\$0,53	\$0,53
conector mólex 3 pines		2	c/u	\$0,43	\$0,86
conector mólex 2 pines		3	c/u	\$0,35	\$1,05
conectores t block x 2		10	c/u	\$0,25	\$2,50
conectores t block x 3		1	c/u	\$0,35	\$0,35
conector hembra 12 V.		1	c/u	\$0,50	\$0,50

fusible	1	c/u	\$0,25	\$0,25
porta fusibles	1	c/u	\$0,75	\$0,75
transformados 12 V, 5A	1	c/u	\$7,75	\$7,75
Elaboración del módulo				
Pernos	15	c/u	\$0,11	\$1,65
Caja acrílica	1	c/u	\$20,00	\$20,00
Impresión 3D	1	c/u	\$5,00	\$5,00
Transporte y salida de campo				
Pasaje estudiante 1	10	c/u	\$0,25	\$2,50
Pasaje estudiante 2	10	c/u	\$0,25	\$2,50
Mano de obra				
Diseño de placa	1	c/u	\$30,00	\$30,00
Diseño de Caja, soportes y acoples	1	c/u	\$10,00	\$10,00
Ensamble del módulo	1	c/u	\$10,00	\$10,00
Material Bibliográfico y Fotocopias				
Internet	40	c/u	\$0,50	\$20,00
Impresiones	100	c/u	\$0,05	\$5,00
Subtotal				\$150,78
Imprevistos 10%				\$15,08
Total (\$)				\$165,86

8. CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- De la recopilación de información bibliográfica, se definió los elementos de los sistemas de control, como sensores Lm-35 para Temperatura, un LDR en el control de Intensidad Luminosa y el uso de encoder para la variables de velocidad y posición, como actuadores lámparas incandescentes y ventilador para el control de temperatura, en el control de intensidad Luminosa una de las lámparas del control de temperatura, en el control de velocidad se definió el uso de un motor DC y en el control de posición un motor Step bipolar.
- De pruebas realizadas se definió los componentes eléctricos para la lectura de señal de las variables de temperatura, Intensidad Luminosa, velocidad y posición; y

activación de los actuadores, se diseñó la placa de circuito integrado y caja de acrílico del módulo en el programa de elementos finitos, como resultado se obtuvo una placa de circuito integrado con los elementos electrónicos acoplados y que se montó sobre la caja de acrílico.

- En el programa MyOpenLab se diseñó la programación para el control temperatura, intensidad luminosa, velocidad y posición, logrando la adquisición e interpretación de datos del software, utilizando los sensores y actuadores del módulo para sistemas de control.
- Con la utilización del software libre MyOpenLab y módulo, se logra la visualización de las diferentes ondas de los sistemas de control, se puede realizar la comunicación de la interfaz gráfica, el circuito por medio de la tarjeta de adquisición de datos Arduino. MyOpenLab resulta ser muy útil en el desarrollo de Sistemas de control.
- De las pruebas realizadas en MyOpenLab utilizando el módulo, se obtuvo las curvas de sintonización para el control de las variables, obteniendo errores de sintonización en temperatura $e=0.83\%$, en Intensidad Luminosa $e=0.17\%$ y en Velocidad $e=0.03\%$, estos valores se encuentran dentro de los parámetros nominales de funcionamiento para reguladores PID, validando el correcto funcionamiento del módulo para Sistemas de Control.
- Con la utilización del software libre MyOpenLab y el correspondiente módulo, se desarrolló las guías de laboratorio, los manuales de uso y mantenimiento para complementar la guía de manipulación para el correcto funcionamiento, que permite el análisis de control y progreso de cada práctica a realizarse.

Recomendaciones

- En pruebas preliminares en la señal que entrega, el sensor de temperatura LM-35 presentaba ruido, lo que imposibilita el control, se hicieron pruebas para filtrar la señal, entre ellas un filtrado pasivo, que elimino el ruido eléctrico, utilizando carga capacitiva de $1\mu\text{f}$ entre la señal del sensor y tierra (GND), que mejoró notablemente la señal y permitió el control.
- Los sistemas de control de Temperatura e Intensidad luminosa están diseñados para trabajar con 5 (V) como voltaje de control y 12 (V) como voltaje de potencia y alimentación, esto se logró empleando el opto-acoplador PC-817 conectado a un

transistor 2N222 como interruptor, esto nos permite con una señal de 0 a 5 voltios controlar potencia de 0 a 12 (V) voltios y 0.5 (A) amperios.

- Para el motor DC y su respectivo control se debe optar por el uso de un driver Tb6612fng para obtener los datos en tiempo real ya que su capacidad de voltaje y corriente en la salida permite no tener pérdidas en la recepción de datos, se debe diferenciar su uso entre el motor DC y motor Step.
- Para el frenado del motor DC se debe tener en cuenta que el accionamiento no debe ser brusco ni retardado, ya que el encoder se encuentra censando en tiempo real y no se recibiría la curva de comportamiento, se debe realizar el frenado sutilmente en niveles de revolución para obtener el comportamiento adecuado.
- Es muy importante tener conectado todos los terminales GND a tierra común desde el módulo de control, la tarjeta de adquisición de datos Arduino y la fuente que alimenta a cada uno de estos circuitos de control ya que, sin esto se podría presentar ruido en la señal de salida de los sensores, no funcionará correctamente los controles y habrá una lectura errónea de temperatura, intensidades, velocidad y posición.
- Para el control de la variable de temperatura, intensidad, velocidad y posiciones en MyOpenLab se debe modificar el intervalo de actualización de salida, pasar de 200 (ms) a 10 (ms) como esta predefinido en las propiedades del Firmata IO Interface, esto mejora la respuesta del control, ya que la programación enviará las señales de control de carácter booleano o PWM en intervalos más cortos.
- En el software MyOpenLab, para graficar la señal del Sensor y del Setpoint en la misma gráfica en el panel frontal, se emplea dos elementos “Gráfico Temporal” X1 y X3 en el panel frontal, los dos gráficos deben tener las mismas unidades y dimensiones y no tener fondo (activar Background Transparent en las propiedades de la gráfica), para interpolar una gráfica sobre la otra.
- Dentro de los elementos que posee MyOpenLab, hay una herramienta que permite el registro de datos, esto recopila información del comportamiento de los sistemas de control de temperatura, intensidad luminosa, velocidad y posición, que pueden ser analizados, visualizados y modificados para obtener la forma de onda del sistema, desde programas como Matlab o Excel.

9. REFERENCIA BIBLIOGRÁFICA

- [1] A. Barranco Gutiérrez, Á. León Cárdenas, and F. Perez Pinal, “Implementación de sistema difuso en Arduino Uno,” no. November, pp. 1–7, 2016, [Online]. Available: https://www.researchgate.net/profile/Alejandro_Barranco_Gutierrez5/publication/309676195_Implementacion_de_sistema_difuso_en_Arduino_Uno/links/581cc82f08ae12715af20b4e/Implementacion-de-sistema-difuso-en-Arduino-Uno.pdf.
- [2] R. Pi, J. Román, and L. David, “Revista de Simulación Computacional Implementación de un controlador difuso de temperatura desarrollado en un sistema embebido Raspberry Pi Implementation of a fuzzy temperature controller developed in an embedded Raspberry Pi system,” no. December, pp. 8–15, 2018.
- [3] J. A. H. C. S. Herrera Aristizába, “Implementación de algoritmos de control de velocidad y torque de motores de corriente continua aplicados a un modelo de AGV multipropósito de potencia media utilizando Raspberry Pi,” *Psikol. Perkemb.*, no. October 2013, pp. 1–224, 2019, doi: 10.1017/CBO9781107415324.004.
- [4] J. Oliver, “TEMA: DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE PLANTA CON UN CONTROL DE NIVEL POR PRESIÓN HIDROSTÁTICA, COMUNICACIÓN REMOTA Y MONITOREADO MEDIANTE EL SOFTWARE,” *Hilos Tensados*, vol. 1, no., pp. 1–476, 2019, doi: 10.1017/CBO9781107415324.004.
- [5] Instituto de Tecnologías Educativas, “Sistema de control - Automatización y control,” *Automatización y control*, Jun. 07, 2015. <https://sites.google.com/site/automatizacionycontrol4/automatizacion/sistema-de-control> (accessed Aug. 04, 2020).
- [6] ISA México Sección Central - INTECH AUTOMATIZACIÓN, “Conceptos Básicos: Sistemas de Control ,” *Conceptos Básicos: Sistemas de Control*, Mar. 07, 2018. <https://www.isamex.org/intechmx/index.php/2018/12/24/conceptos-basicos-sistemas-de-control/> (accessed Aug. 04, 2020).
- [7] Murky Robot, “Equipos de Control,” *Relé: Control equipos de 220v con Arduino* , Feb. 01, 2015. <https://www.murkyrobot.com/guias/actuadores/rele> (accessed Aug. 12, 2020).
- [8] Geek Electrónica, “¿QUÉ SON LOS TRANSDUCTORES? - .”

- <https://geekelectronica.com/que-son-los-transductores/> (accessed Aug. 12, 2020).
- [9] J. Ignacio, “Conversores Análogo-Digital y Digital-Análogo : Conceptos Básicos,” *Juan Ignacio*, pp. 1–7, 2007.
- [10] Espinoza Ricardo, “Señales analógicas y Digitales,” *Señales analógicas y digitales*, May 03, 2017. <http://ricardoredesetac.blogspot.com/2012/08/senales-analogicas-y-digitales.html> (accessed Aug. 13, 2020).
- [11] Tecnología de Control, “SISTEMA DE CONTROL DE LAZO ABIERTO Y LAZO CERRADO ,” *SISTEMA DE CONTROL DE LAZO ABIERTO Y LAZO CERRADO*, May 07, 2015. <https://sites.google.com/site/tecnologiadecontrol2016/sistema-de-control-manual> (accessed Aug. 19, 2020).
- [12] S. Kouro and R. Musalem, “Control mediante lógica difusa,” *Técnicas Mod. Autom.*, pp. 1–7, 2002, [Online]. Available: <http://www2.elo.utfsm.cl/~elo377/documentos/Fuzzy.pdf>.
- [13] Tienda y Tutoriales Arduino, “Módulo controlador de motores L298N ,” *Módulo controlador de motores L298N* , May 05, 2017. <https://www.prometec.net/l298n/> (accessed Aug. 12, 2020).
- [14] Jose Carlos Villajulca, “Instrumentación y Automatización Industrial,” *Control ON/OFF o Todo/Nada -*, Jul. 01, 2017. <https://instrumentacionycontrol.net/control-on-off-o-todo-nada/> (accessed Aug. 04, 2020).
- [15] Marcelo Diez, “Tipos de control,” *Controlador de acción proporcional e integral (PI)*, Jul. 02, 2010. http://educativa.catedu.es/44700165/aula/archivos/repositorio/4750/4926/html/13_controlador_de_accin_proporcional_e_integral_pi.html (accessed Aug. 05, 2020).
- [16] Universidad de Santiago de Chile, “Control PID-Gráfica,” *Control Proporcional* , May 02, 2015. <http://www.udesantiagovirtual.cl/moodle2/mod/book/view.php?id=34408&chapterid=361> (accessed Aug. 23, 2020).
- [17] EcuRed, “Control proporcional ,” *Control proporcional Es una de las acciones de control empleadas en los lazos de regulación automática.*, Aug. 13, 2010.

- https://www.ecured.cu/Control_proporcional (accessed Aug. 13, 2020).
- [18] Wikipedia, “Controladores PID ,” *controlador PID controlador proporcional, integral y derivativo*, Jul. 10, 2020. https://es.wikipedia.org/wiki/Controlador_PID?fbclid=IwAR0DIPINad3a5j9aeFxEc kwJuip9JalFW5HI4SvWZZ54JqXE4txlrVKITrk (accessed Aug. 13, 2020).
- [19] SERGIO ANDRÉS CASTAÑO GIRALDO, “Control Predictivo Basado en Modelo - MPC,” *Ejemplos y Códigos MATLAB*, May 05, 2017. <https://controlautomaticoeducacion.com/control-predictivo/> (accessed Aug. 06, 2020).
- [20] “Sistema de Control Predictivo - Download Scientific Diagram,” *Aplicación a Tanques Acoplados.* , Jul. 01, 2015. https://www.researchgate.net/figure/Figura-1-Sistema-de-Control-Predictivo-aplicado-a-Tanques-Acoplados_fig1_43188842 (accessed Aug. 06, 2020).
- [21] Carlos Tamami, “Definición de programación gráfica ,” *QUE ES LENGUAJE DE PROGRAMACIÓN GRÁFICA*, Jan. 02, 2018. <https://prezi.com/155gwvzx0d00/definicion-de-programacion-grafica/> (accessed Aug. 06, 2020).
- [22] M. R. Guti, “MyOpenLab,” pp. 1–16.
- [23] P. José and M. Ruiz, “DIAGRAMAS DE FLUJO (guía rápida),” pp. 1–29, [Online]. Available: https://myopenlab.org/guia_diagramas_flujo_v2499.pdf.
- [24] T. Emporales, P. José, and M. Ruiz, “S m m t,” vol. 5, pp. 1–25, [Online]. Available: https://myopenlab.org/simulacion_modelos_mat_temporales.pdf.
- [25] Ingenieira MCI Ltda, “¿Que es Arduino? | Arduino.cl - Compra tu Arduino en Línea,” 2019. <https://arduino.cl/que-es-arduino/> (accessed Jul. 01, 2020).
- [26] “Firmata | Aprendiendo Arduino.” <https://aprendiendoarduino.wordpress.com/2016/03/06/firmata/> (accessed Jun. 24, 2020).
- [27] M. Gutiérrez and S. Iturralde, *Fundamentos Basicos De Instrumentacion y Control*. 2017.

- [28] J. H. Vanegas García, “3. Sensores y Transductores. 3.1. Sensores y Transductores. Un,” *Univ. Tarapacá*, p. 34, 2014, [Online]. Available: http://www.eudim.uta.cl/files/5813/2069/8949/fm_Ch03_mfuentesm.pdf.
- [29] F. R. Ramirez, Ed., *Sistemas de control automatico*, Septima. México.
- [30] “Sensores de Temperatura – Parte 2 – Encendido Electronico.” <https://www.encendidoelectronico.com/sensores-temperatura/sensores-temperatura-parte-2/> (accessed Aug. 10, 2020).
- [31] “LDR | Aprendiendo Arduino.” <https://aprendiendoarduino.wordpress.com/tag/ldr/> (accessed Aug. 10, 2020).
- [32] “Rotary Encoders | Tienda y Tutoriales Arduino.” <https://www.prometec.net/rotary-encoders/> (accessed Aug. 10, 2020).
- [33] “Sensores ópticos, capacitivos ultrasónicos efecto Hall.” http://www.sapiensman.com/tecnoficio/optica/sensores_opticos.php (accessed Aug. 10, 2020).
- [34] F. S. Corporation, “LM78XX/LM78XXA 3-Terminal 1A Positive Voltage Regulator,” 2006. <http://ee-classes.usc.edu/ee459/library/datasheets/LM7805.pdf> (accessed Aug. 06, 2020).
- [35] F. Peeters, M. Peetermans, and L. Indesteege, “Temperature Sensors,” *Mod. Sensors Handb.*, pp. 347–393, 2010, doi: 10.1002/9780470612231.ch8.
- [36] SHARP Corporation, “PC817X Series,” 2003. Accessed: Aug. 06, 2020. [Online]. Available: <https://www.farnell.com/datasheets/73758.pdf>.
- [37] STMicroelectronics GROUP OF COMPANIES, “2N2219A 2N2222A,” 2003. <https://docs.rs-online.com/3526/0900766b813647cf.pdf> (accessed Aug. 06, 2020).
- [38] CEBEK Componentes y accesorios, “FOTORRESISTENCIALDR 4,3mmxØ5,1mm.” Accessed: Aug. 06, 2020. [Online]. Available: www.fadisel.com/SOLICITEGRATUITAMENTE nuestroc.at.
- [39] Texas Instruments, “Datasheet Lm35,” no. November, pp. 1–13, 2013, [Online]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/517588/TI1/LM35.html>.

- [40] Caldas Electrónicos, “Motor DC 5V ,” *PPN7PA12C1*, Feb. 02, 2017. <https://www.electronicoscaldas.com/es/motores-y-servos/270-motor-dc-iman-permanente-escobillas-5v-ppn7pa12c1.html> (accessed Aug. 23, 2020).
- [41] T. B. Integrated and C. Silicon, “TB6612FNG,” pp. 1–11, 2014.
- [42] J. A. Melkebeek, “Stepping motors,” *Power Syst.*, no. 9783319727295, pp. 523–535, 2018, doi: 10.1007/978-3-319-72730-1_21.
- [43] SGS-Thomson Microelectronics, “L293D Datasheet,” no. June, pp. 1–7, 1996.
- [44] “(No Title).” [https://pdf.voron.ua/files/pdf/relay/JQC-3F\(T73\).pdf](https://pdf.voron.ua/files/pdf/relay/JQC-3F(T73).pdf) (accessed Aug. 30, 2020).
- [45] Bay Linear, “LM78XX 1.5A Positive Voltage Regulator,” vol. 220, no. D, 2010.

ANEXOS

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy (Note 7)	$T_A = +25^\circ\text{C}$	± 0.2	± 0.5		± 0.2	± 0.5		$^\circ\text{C}$
	$T_A = -10^\circ\text{C}$	± 0.3			± 0.3		± 1.0	$^\circ\text{C}$
	$T_A = T_{\text{MAX}}$	± 0.4	± 1.0		± 0.4	± 1.0		$^\circ\text{C}$
	$T_A = T_{\text{MIN}}$	± 0.4	± 1.0		± 0.4		± 1.5	$^\circ\text{C}$
Nonlinearity (Note 8)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	± 0.18		± 0.35	± 0.15		± 0.3	$^\circ\text{C}$
Sensor Gain (Average Slope)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	+10.0	+9.9, +10.1		+10.0		+9.9, +10.1	mV/ $^\circ\text{C}$
Load Regulation (Note 3) $0 \leq I_L \leq 1$ mA	$T_A = +25^\circ\text{C}$	± 0.4	± 1.0		± 0.4	± 1.0		mV/mA
	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	± 0.5		± 3.0	± 0.5		± 3.0	mV/mA
Line Regulation (Note 3)	$T_A = +25^\circ\text{C}$	± 0.01	± 0.05		± 0.01	± 0.05		mV/V
	$4\text{V} \leq V_S \leq 30\text{V}$	± 0.02		± 0.1	± 0.02		± 0.1	mV/V
Quiescent Current (Note 9)	$V_S = +5\text{V}, +25^\circ\text{C}$	56	67		56	67		μA
	$V_S = +5\text{V}$	105		131	91		114	μA
	$V_S = +30\text{V}, +25^\circ\text{C}$	56.2	68		56.2	68		μA
	$V_S = +30\text{V}$	105.5		133	91.5		116	μA
Change of Quiescent Current (Note 3)	$4\text{V} \leq V_S \leq 30\text{V}, +25^\circ\text{C}$	0.2	1.0		0.2	1.0		μA
	$4\text{V} \leq V_S \leq 30\text{V}$	0.5		2.0	0.5		2.0	μA
Temperature Coefficient of Quiescent Current		+0.39		+0.5	+0.39		+0.5	$\mu\text{A}/^\circ\text{C}$
Minimum Temperature for Rated Accuracy	In circuit of Figure 1, $I_L = 0$	+1.5		+2.0	+1.5		+2.0	$^\circ\text{C}$
Long Term Stability	$T_J = T_{\text{MAX}}$, for 1000 hours	± 0.08			± 0.08			$^\circ\text{C}$

Temperature Rise of LM35 Due To Self-heating (Thermal Resistance, θ_{JA})

	TO-46, no heat sink	TO-46*, small heat fin	TO-92, no heat sink	TO-92**, small heat fin	SO-8 no heat sink	SO-8**, small heat fin	TO-220 no heat sink
Still air	400 $^\circ\text{C/W}$	100 $^\circ\text{C/W}$	180 $^\circ\text{C/W}$	140 $^\circ\text{C/W}$	220 $^\circ\text{C/W}$	110 $^\circ\text{C/W}$	90 $^\circ\text{C/W}$
Moving air	100 $^\circ\text{C/W}$	40 $^\circ\text{C/W}$	90 $^\circ\text{C/W}$	70 $^\circ\text{C/W}$	105 $^\circ\text{C/W}$	90 $^\circ\text{C/W}$	26 $^\circ\text{C/W}$
Still oil	100 $^\circ\text{C/W}$	40 $^\circ\text{C/W}$	90 $^\circ\text{C/W}$	70 $^\circ\text{C/W}$			
Stirred oil	50 $^\circ\text{C/W}$	30 $^\circ\text{C/W}$	45 $^\circ\text{C/W}$	40 $^\circ\text{C/W}$			

(Clamped to metal,

Infinite heat sink)

(24 $^\circ\text{C/W}$)

(55 $^\circ\text{C/W}$)

*Wakefield type 201, or 1" disc of 0.020" sheet brass, soldered to case, or similar.

**TO-92 and SO-8 packages glued and leads soldered to 1" square of 1/16" printed circuit board with 2 oz. foil or similar.

Fuente: [39]

SHARP
PC817X Series
■ Absolute Maximum Ratings ($T_c=25^\circ\text{C}$)

	Parameter	Symbol	Rating	Unit
Input	Forward current	I_F	50	mA
	^{*1} Peak forward current	I_{FM}	1	A
	Reverse voltage	V_R	6	V
	Power dissipation	P	70	mW
Output	Collector-emitter voltage	V_{CEO}	^{*4} 80	V
	Emitter-collector voltage	V_{ECO}	6	V
	Collector current	I_C	50	mA
	Collector power dissipation	P_C	150	mW
	Total power dissipation	P_{MT}	200	mW
	^{*2} Isolation voltage	$V_{iso(mo)}$	5.0	kV
	Operating temperature	T_{opr}	-30 to +100	$^\circ\text{C}$
	Storage temperature	T_{stg}	-55 to +125	$^\circ\text{C}$
	^{*3} Soldering temperature	T_{sol}	260	$^\circ\text{C}$

^{*1} Pulse width \leq 100 μ s, Duty ratio : 0.001

^{*2} 40 to 60%RH, AC for 1minute, f=60Hz

^{*3} For 10s

^{*4} Up to Date code "P7" (July 2002) V_{CEO} : 35V.

■ Electro-optical Characteristics
($T_c=25^\circ\text{C}$)

	Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Input	Forward voltage	V_F	$I_F=20\text{mA}$	-	1.2	1.4	V	
	Peak forward voltage	V_{FM}	$I_{FM}=0.5\text{A}$	-	-	3.0	V	
	Reverse current	I_R	$V_R=4\text{V}$	-	-	10	μA	
	Terminal capacitance	C_t	$V=0, f=1\text{kHz}$	-	30	250	pF	
Output	Collector dark current	I_{CEO}	$V_{CE}=50\text{V}, I_F=0$	-	-	100	nA	
	Collector-emitter breakdown voltage	BV_{CEO}	$I_C=0.1\text{mA}, I_F=0$	^{*5} 80	-	-	V	
	Emitter-collector breakdown voltage	BV_{ECO}	$I_E=10\mu\text{A}, I_F=0$	6	-	-	V	
Transfer characteristics	Collector current	I_C	$I_F=5\text{mA}, V_{CE}=5\text{V}$	2.5	-	30.0	mA	
	Collector-emitter saturation voltage	$V_{CE(sat)}$	$I_F=20\text{mA}, I_C=1\text{mA}$	-	0.1	0.2	V	
	Isolation resistance	R_{ISO}	DC500V, 40 to 60%RH	5×10^{10}	1×10^{11}	-	Ω	
	Floating capacitance	C_f	$V=0, f=1\text{MHz}$	-	0.6	1.0	pF	
	Cut-off frequency	f_c	$V_{CE}=5\text{V}, I_C=2\text{mA}, R_L=100\Omega, -3\text{dB}$	-	80	-	kHz	
	Response time	Rise time	t_r	$V_{CE}=2\text{V}, I_C=2\text{mA}, R_L=100\Omega$	-	4	18	μs
		Fall time	t_f		-	3	18	μs

^{*5} From the production Date code "J5" (May 1997) to "P7" (July 2002), however the products were screened by $BV_{CEO} \geq 70\text{V}$.

Fuente: [36]

Datos técnicos	
Voltaje de operación nominal	5V
Rango de voltaje de operación	1 V a 5 V
Velocidad de operación	< 15000 rpm
Velocidad con carga	11600 rpm
Carga recomendada	0.49 mN·m
Rango de cargas de operación	0.1 mN·m a 0.98 mN·m
Corriente con carga	270 mA máx
Corriente sin carga	120 mA máx
Corriente de arranque	1300 mA (1.3 A) máx
Resistencia terminal	20 °C, 4.4 Ω \pm 10%
Diámetro	1.56 cm
Peso aproximado	10 g
Dirección de rotación	ambos sentidos

Fuente: [40]



TOSHIBA

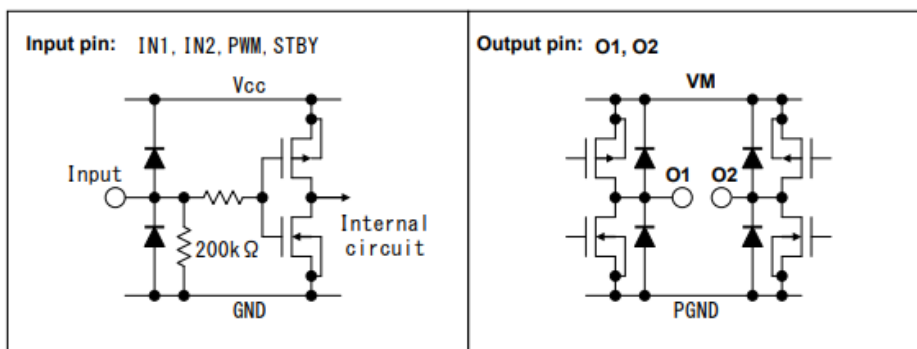
TB6612FNG

Absolute Maximum Ratings (Ta = 25°C)

Characteristics	Symbol	Rating	Unit	Remarks	
Supply voltage	VM	15	V		
	VCC	6			
Input voltage	VIN	-0.2 to 6	V	IN1,IN2,STBY,PWM pins	
Output voltage	VOUT	15	V	O1,O2 pins	
Output current	IOUT	IOUT	A	Per 1 ch	
		IOUT (peak)		2	tw = 20 ms Continuous pulse, Duty ≤ 20%
				3.2	tw = 10 ms Single pulse
Power dissipation	PD	0.78	W	IC only	
		0.89		50 mm × 50 mm t = 1.6 mm Cu ≥ 40% in PCB mounting	
		1.36		76.2 mm × 114.3 mm t = 1.6 mm Cu ≥ 30% in PCB mounting	
Operating temperature	Topr	-20 to 85	°C		
Storage temperature	Tstg	-55 to 150	°C		

Operating Range (Ta = -20 to 85°C)

Characteristics	Symbol	Min	Typ.	Max	Unit	Remarks
Supply voltage	VCC	2.7	3	5.5	V	
	VM	2.5	5	13.5	V	
Output current (H-SW)	IOUT	—	—	1.0	A	VM ≥ 4.5 V
		—	—	0.4		4.5 V > VM ≥ 2.5 V Without PWM Operation
Switching frequency	fPWM	—	—	100	kHz	

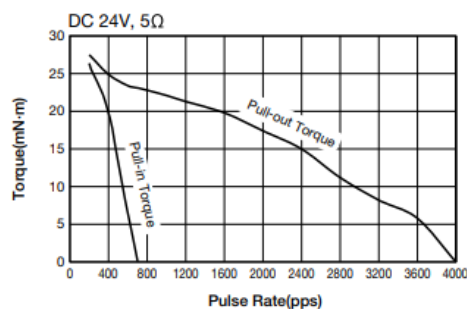
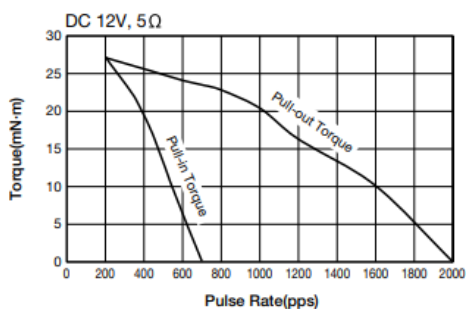


Fuente: [41]

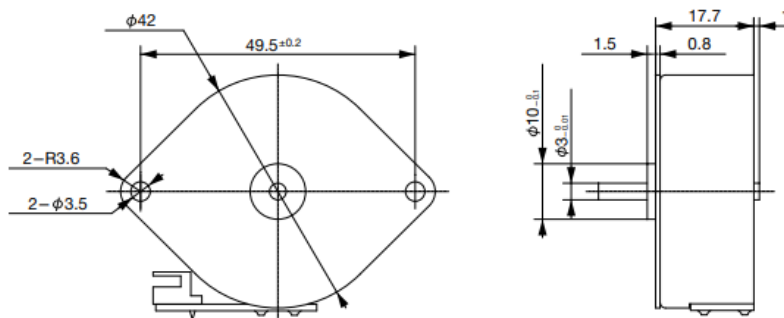
SPECIFICATIONS

Item	M42SP-6NK	
Rated Voltage	DC 12V	DC 24V
Working Voltage	DC 10.8-13.2V	DC 21.6-26.4V
Rated Current/Phase	400mA(PEAK)	
No. of Phase	2 Phase	
Coil DC Resistance	5Ω/phase±7%	
Step Angle	7.5°/step	
Excitation Method	2-2 Phase excitation (Bipolar driving)	
Insulation Class	Class E insulation	
Holding Torque	39.2mN·m	39.2mN·m
Pull-out Torque	23.2mN·m/800pps	17.4mN·m/2,000pps
Pull-in Torque	27.1mN·m/200pps	26.2mN·m/200pps
Max. Pull-out Pulse Rate	2,000pps	3,900pps
Max. Pull-in Pulse Rate	700pps	700pps

CHARACTERISTICS



DIMENSIONS



Unit : mm

Fuente: [42]



ELECTRICAL CHARACTERISTICS (for each channel, $V_S = 24\text{ V}$, $V_{SS} = 5\text{ V}$, $T_{amb} = 25\text{ }^\circ\text{C}$, unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 10)		V_{SS}		36	V
V_{SS}	Logic Supply Voltage (pin 20)		4.5		36	V
I_S	Total Quiescent Supply Current (pin 10)	$V_i = L; I_o = 0; V_{en} = H$		2	6	mA
		$V_i = H; I_o = 0; V_{en} = H$		16	24	mA
		$V_{en} = L$			4	mA
I_{SS}	Total Quiescent Logic Supply Current (pin 20)	$V_i = L; I_o = 0; V_{en} = H$		44	60	mA
		$V_i = H; I_o = 0; V_{en} = H$		16	22	mA
		$V_{en} = L$		16	24	mA
V_{IL}	Input Low Voltage (pin 2, 9, 12, 19)		-0.3		1.5	V
V_{IH}	Input High Voltage (pin 2, 9, 12, 19)	$V_{SS} \leq 7\text{ V}$	2.3		V_{SS}	V
		$V_{SS} > 7\text{ V}$	2.3		7	V
I_{IL}	Low Voltage Input Current (pin 2, 9, 12, 19)	$V_{IL} = 1.5\text{ V}$			-10	μA
I_{IH}	High Voltage Input Current (pin 2, 9, 12, 19)	$2.3\text{ V} \leq V_{IH} \leq V_{SS} - 0.6\text{ V}$		30	100	μA
V_{enL}	Enable Low Voltage (pin 1, 11)		-0.3		1.5	V
V_{enH}	Enable High Voltage (pin 1, 11)	$V_{SS} \leq 7\text{ V}$	2.3		V_{SS}	V
		$V_{SS} > 7\text{ V}$	2.3		7	V
I_{enL}	Low Voltage Enable Current (pin 1, 11)	$V_{enL} = 1.5\text{ V}$		-30	-100	μA
I_{enH}	High Voltage Enable Current (pin 1, 11)	$2.3\text{ V} \leq V_{enH} \leq V_{SS} - 0.6\text{ V}$			± 10	μA
$V_{CE(sat)H}$	Source Output Saturation Voltage (pins 3, 8, 13, 18)	$I_o = -0.6\text{ A}$		1.4	1.8	V
$V_{CE(sat)L}$	Sink Output Saturation Voltage (pins 3, 8, 13, 18)	$I_o = +0.6\text{ A}$		1.2	1.8	V
V_F	Clamp Diode Forward Voltage	$I_o = 600\text{ nA}$		1.3		V
t_r	Rise Time (*)	0.1 to 0.9 V_o		250		ns
t_f	Fall Time (*)	0.9 to 0.1 V_o		250		ns
t_{on}	Turn-on Delay (*)	0.5 V_i to 0.5 V_o		750		ns
t_{off}	Turn-off Delay (*)	0.5 V_i to 0.5 V_o		200		ns

TRUTH TABLE (one channel)

Input	Enable (*)	Output
H	H	H
L	H	L
H	L	Z
L	L	Z

Z = High output impedance
 (*) Relative to the considered channel

Figure 1: Switching Times

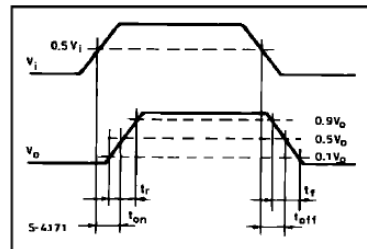
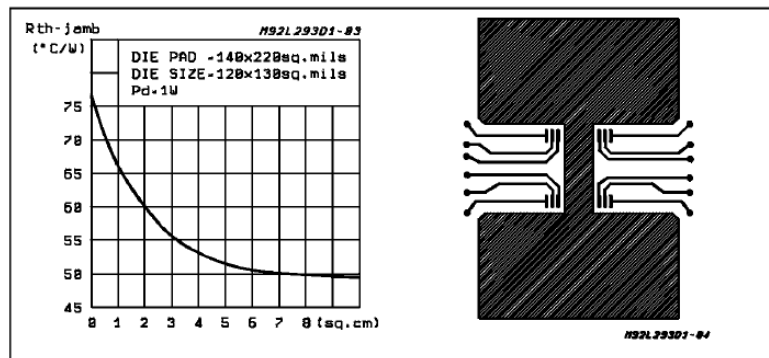


Figure 2: Junction to ambient thermal resistance vs. area on board heatsink (SO12+4+4 package)



Fuente: [43]

Voltaje de Operación	3.3V - 5V DC
Salidas	Analógica y Digital TTL
Sensor	MOCH22A
Modelo Placa	FC-03 / FZ0888
Tipo de emisor	Fotodiodo IR
Tipo de detector	fototransistor
Longitud de onda del emisor	950 nm (infrarrojo)
Peso	8 gramos
Dimensiones	3.2*1.4*0.7 cm
Ranura	5 mm
Led indicador de alimentación	SI
Led indicador de pulso	SI
Salida TTL ON	Sensor bloqueado
Salida TTL OFF	Sensor sin bloquear

Fuente: Autores

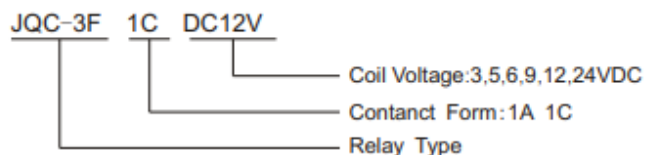


www.chinahwe.net
YUEQING HENGWEI ELECTRONICS CO.,LTD.



PCB Relay—JQC-3F(T73)

Order Model



Contact Data

Contact Form	1A,1C
Contact Material	AgCdO
Contact Resistance	$\leq 100M\Omega$ (1A 6VDC)
Switching Current	10A
Rated Load	10A 250VAC/30VDC

Coil Specification

Coil Voltage	3~24VDC
Coil Power	0.36W,0.45W

Technical Specification

Insulation Resistance	$\geq 100M\Omega$ (500VDC)
Dielectric Strength (Between coil and Contact)	$\geq 1000VAC$
Operate Time	$\leq 10ms$
Release Time	$\leq 5ms$
Mechanical Life	1×10^7
Electrical Life	1×10^5
Install Mode	PCB
Weight	10g

Fuente: [44]

Packaging Information

Ordering Information



Top View

1. Input
2. GND
3. Output

Device	Operating Voltage	Temp.
LM7805	7 to 20	0 to 125 °C
LM7806	8 to 20	0 to 125 °C
LM7808	10.5 to 23	0 to 125 °C
LM7809	11.5 to 24	0 to 125 °C
LM7810	12.5 to 25	0 to 125 °C
LM7812	14.5 to 27	0 to 125 °C
LM7815	17.5 to 30	0 to 125 °C
LM7818	20.5 to 33	0 to 125 °C
LM7824	26.5 to 39	0 to 125 °C

- TO-220 (T)
- TO-263 (S)
- TO-252 (D)

Absolute Maximum Rating

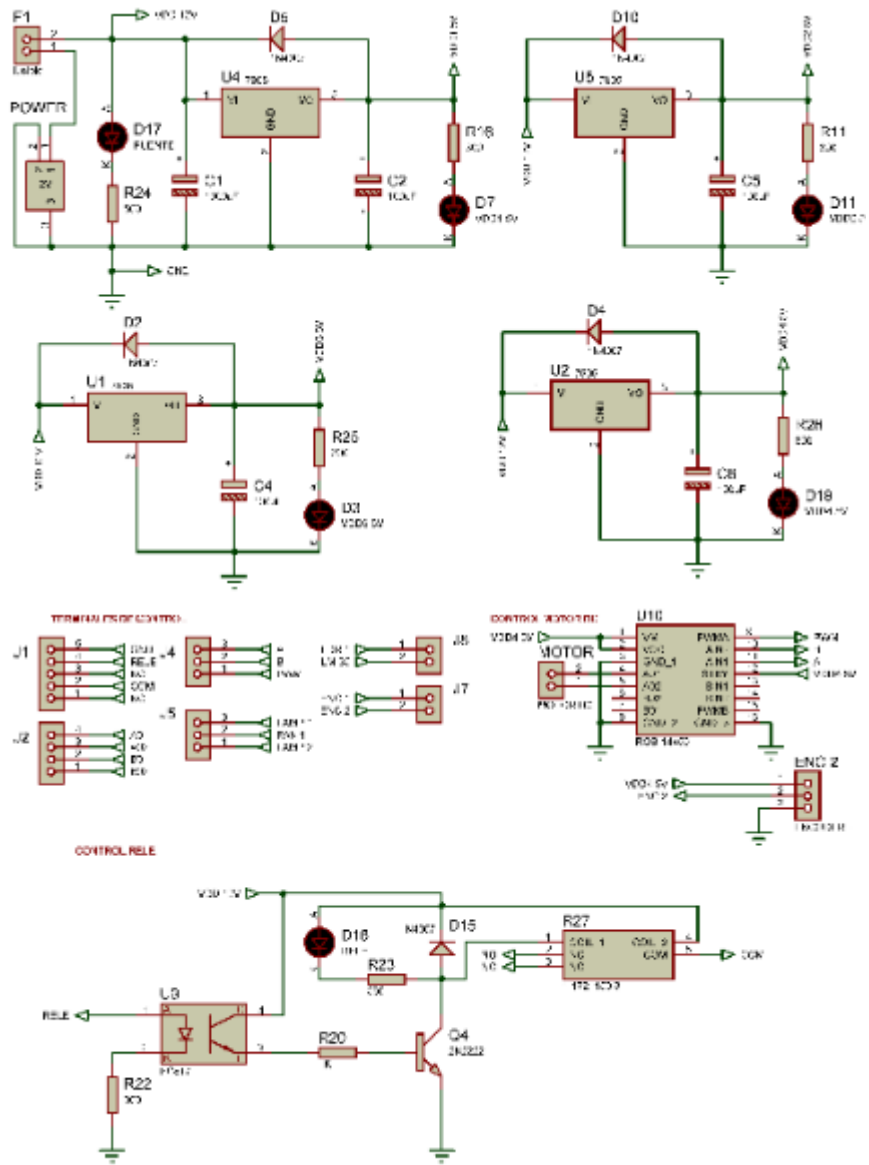
Parameter	LM78--	Unit
Input Voltage	LM7824, LM7827 All Others	40 35
Operating Free-Air, Case, Virtual Junction Temp.	0 to 150	°C
Storage Temperature Range	-65 to 150	
Lead temperature 1.6 mm from case for sec.	260	

Electrical Characteristics (LM7805)

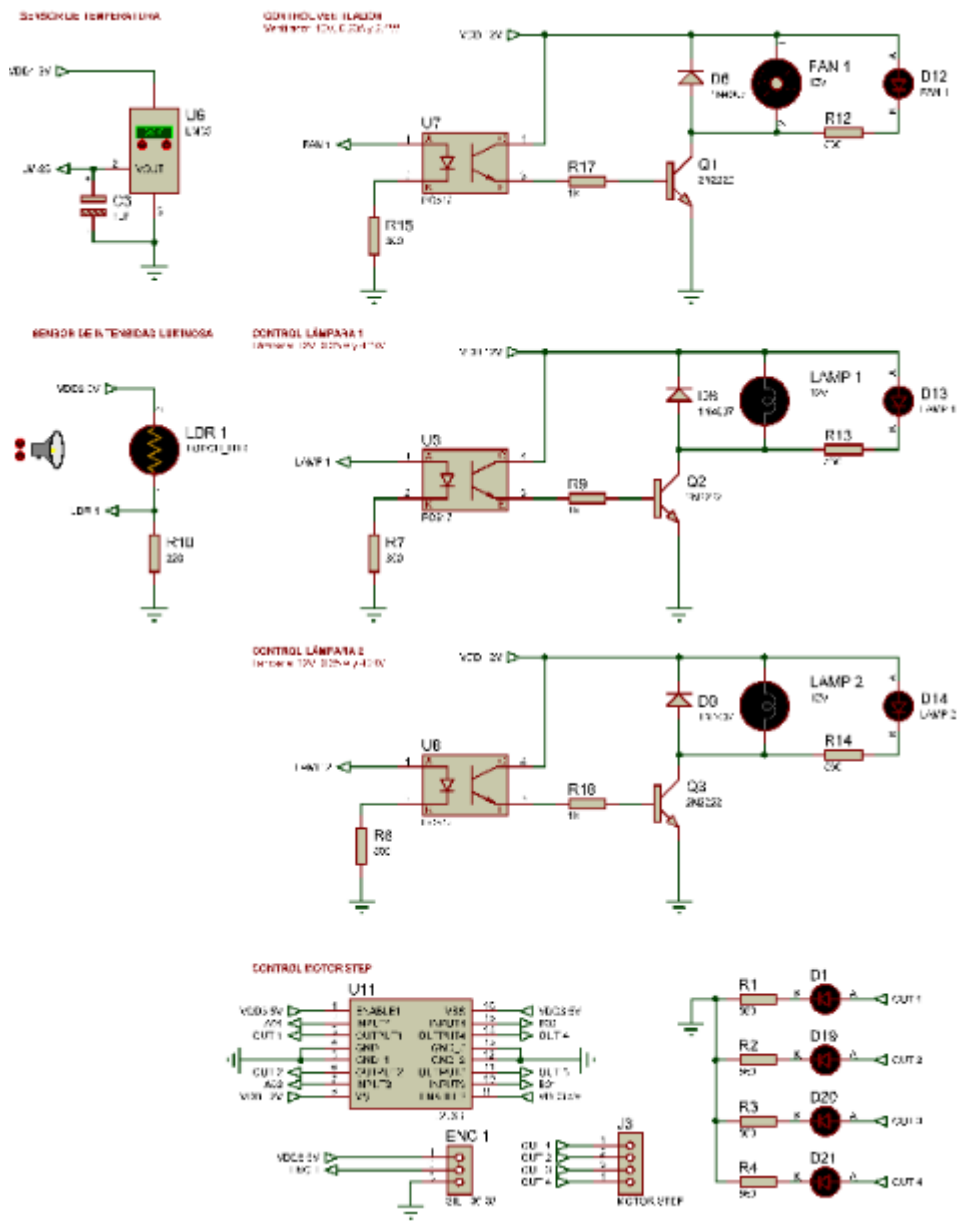
($V_i=10V$, $I_o=500mA$, $0^\circ C \leq T_j \leq 125^\circ C$, unless otherwise specified. (Note 1))

Parameter	Symbol	Conditions	MIN	TYP	MAX	UNIT
Output Voltage	V_o	$T_j = 25^\circ C$	4.8	5.0	5.2	V
Line Regulation	ΔV_o	$V_i = 7V$ to $25V$, $T_j = 25^\circ C$		3	100	mV
		$V_i = 8V$ to $12V$, $T_j = 25^\circ C$		1	50	
Load Regulation	ΔV_o	$I_o = 5mA$ to $1.5A$, $25^\circ C$		15	100	mV
		$I_o = 250mA$ to $750mA$, $25^\circ C$		5	50	
Ripple Rejection	RR	$V_i = 8V$ to $18V$, $f=120Hz$	62	78		dB
Output Noise Voltage	V_N	$F=10Hz$ to $100Hz$, $T_j = 25^\circ C$		40		μV
Dropout Voltage	V_D	$T_j = 25^\circ C$		2.0		V
Quiescent Current		$T_j = 25^\circ C$		4.2	8	mA
Quiescent Current Change	ΔI_Q	$V_i = 7V$ to $25V$, $T_j = 25^\circ C$			1.3	mA
		$I_o = 5mA$ to $1A$, $T_j = 25^\circ C$			0.5	

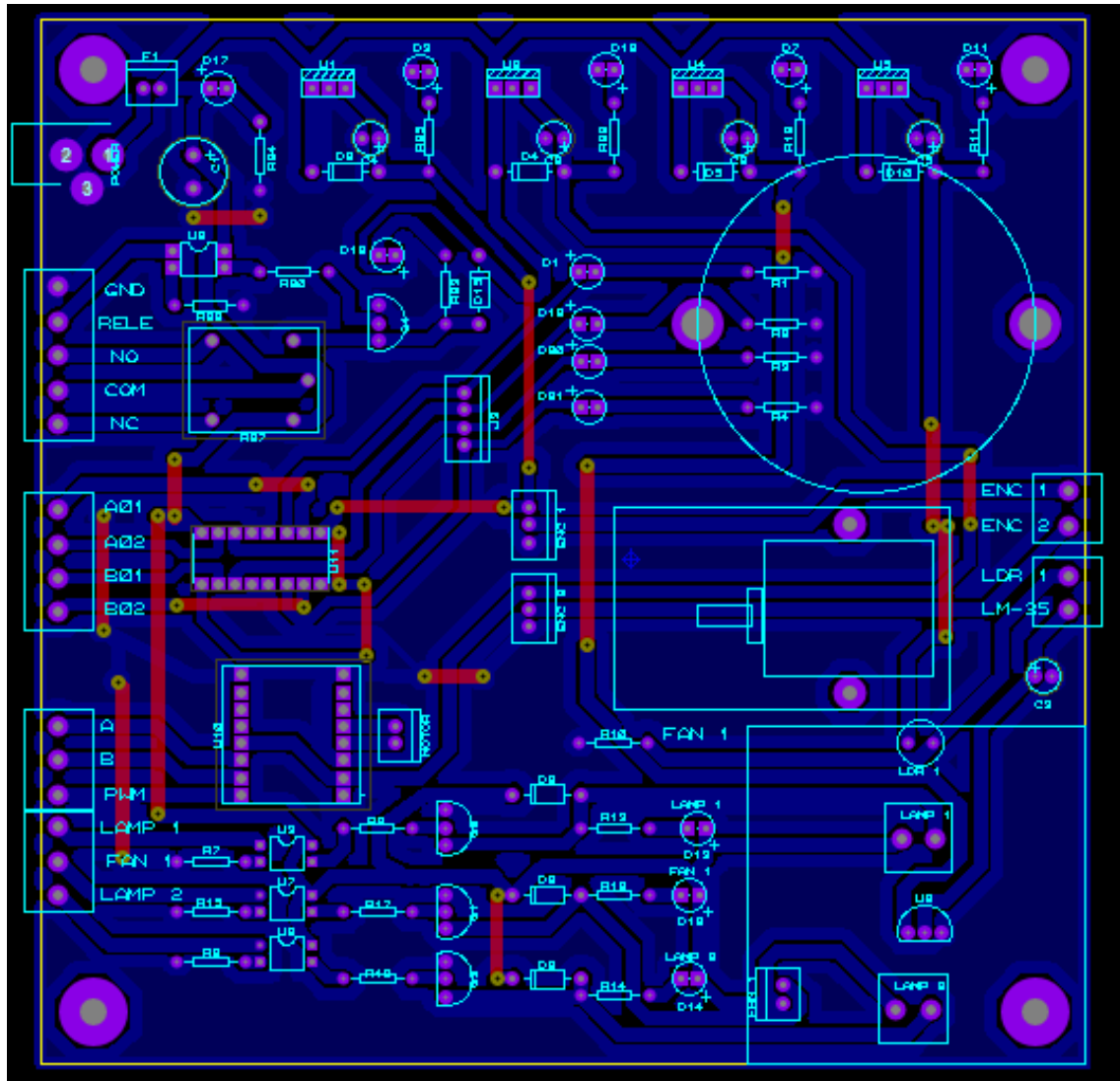
Fuente: [45]



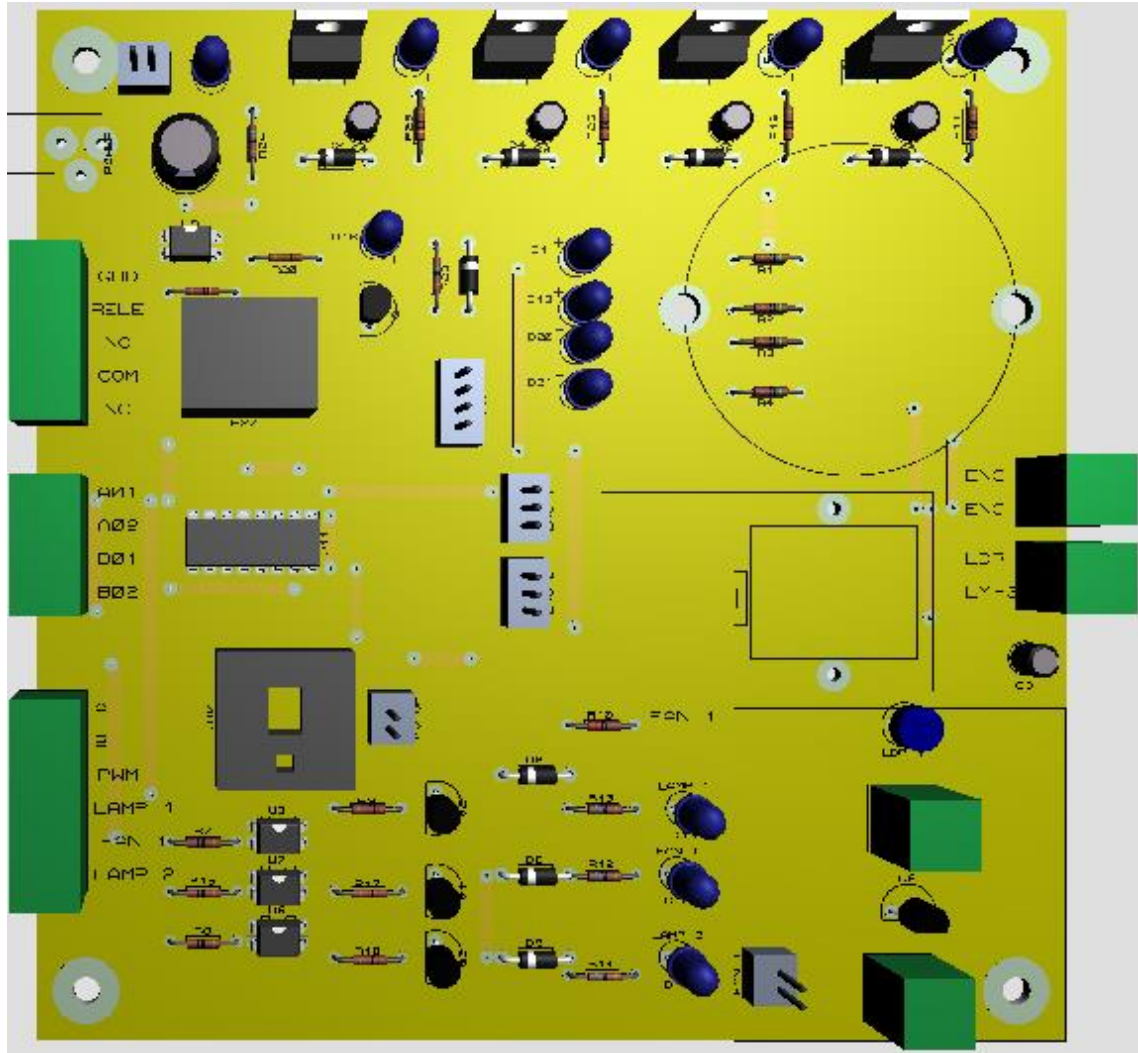
Fuente: Autores



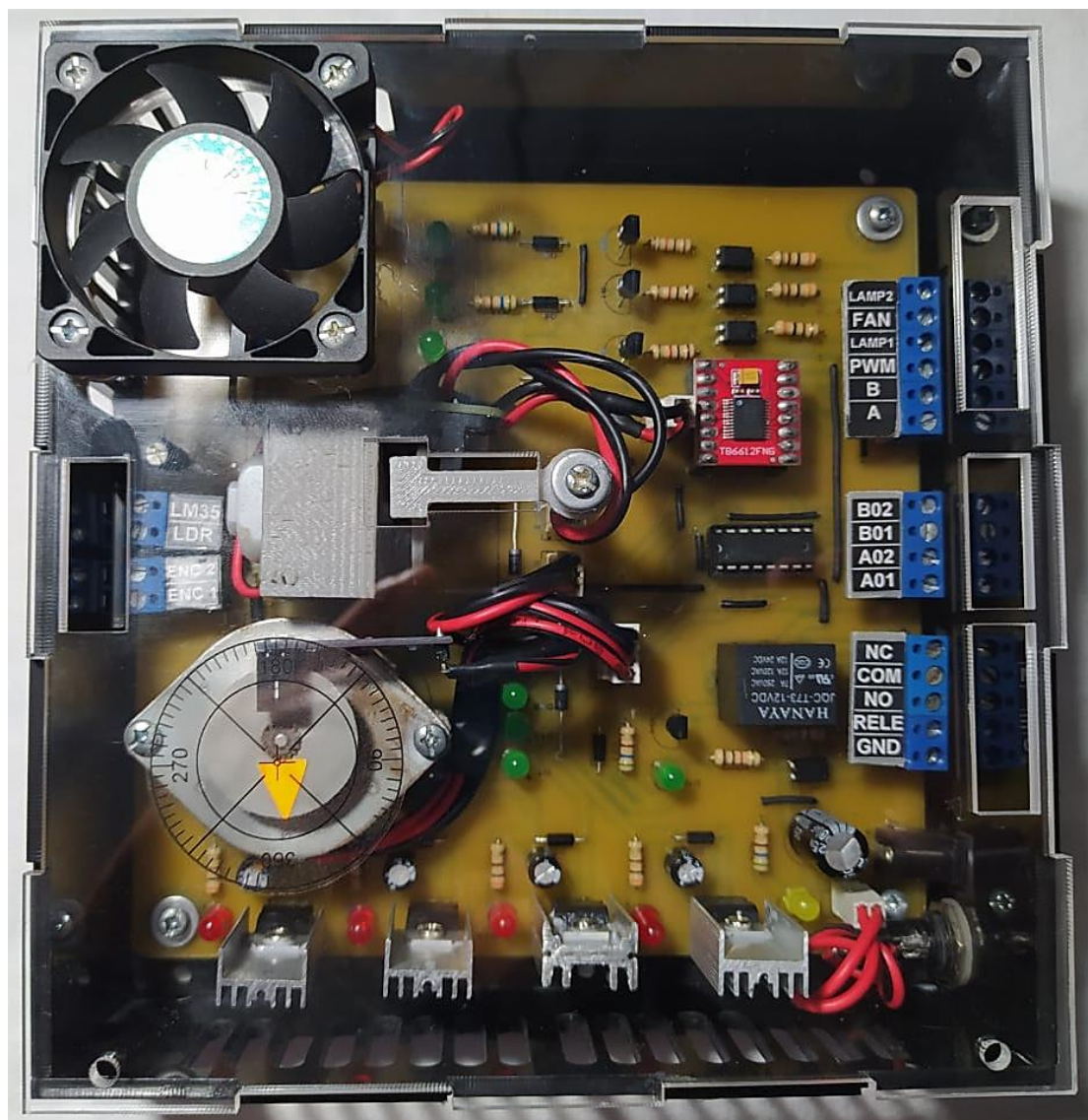
Fuente: Autores



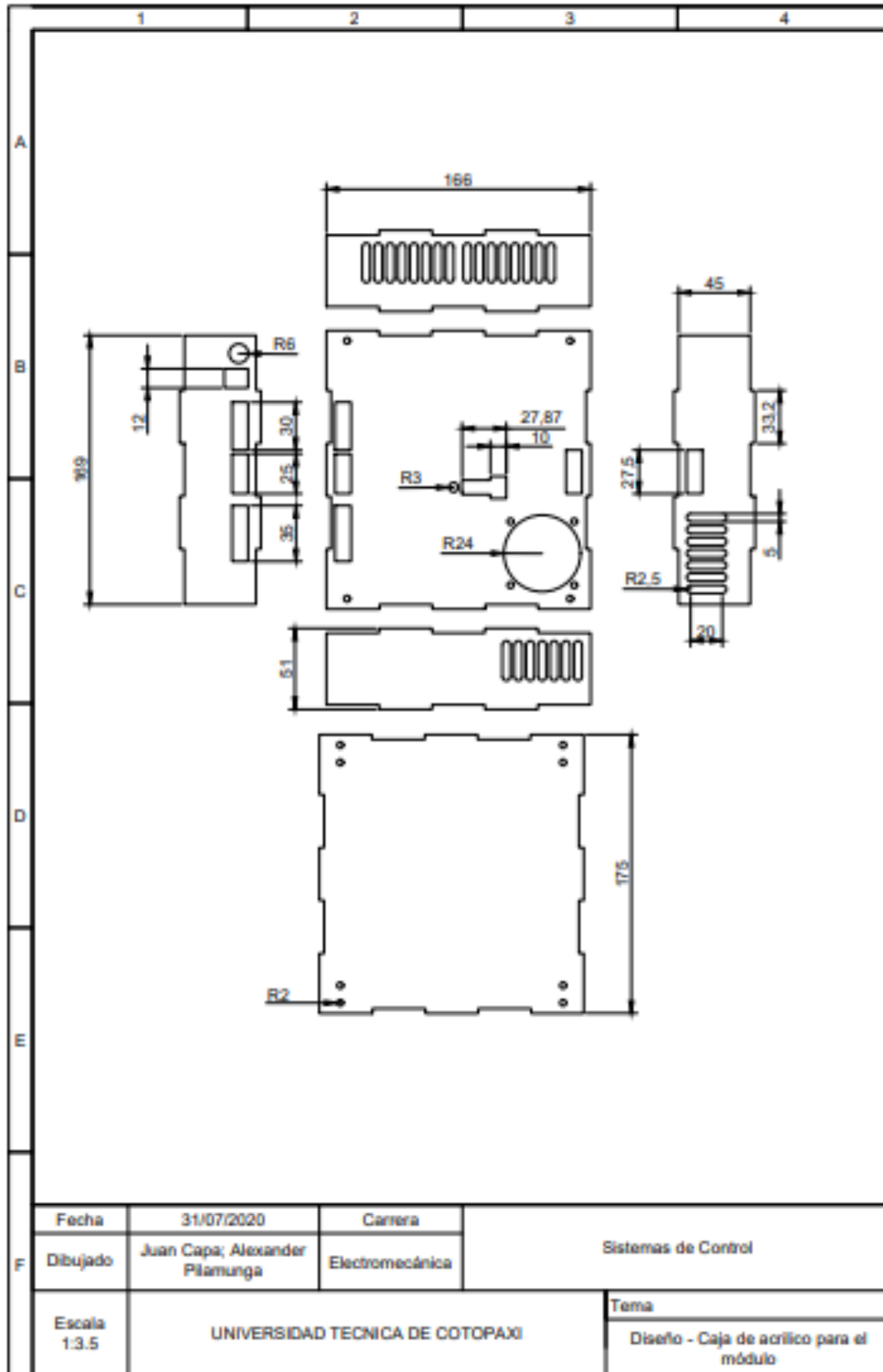
Fuente: Autores



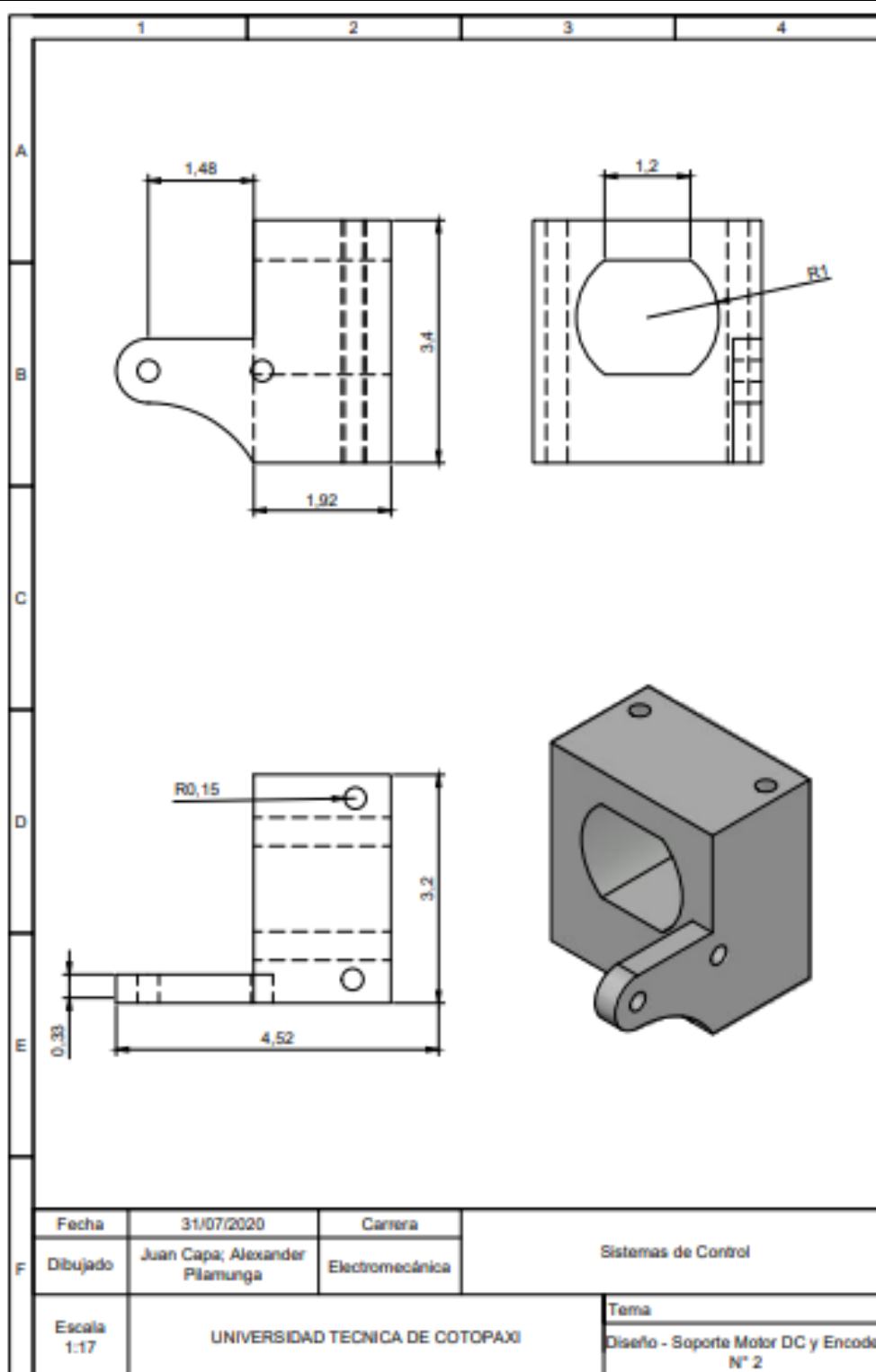
Fuente: Autores



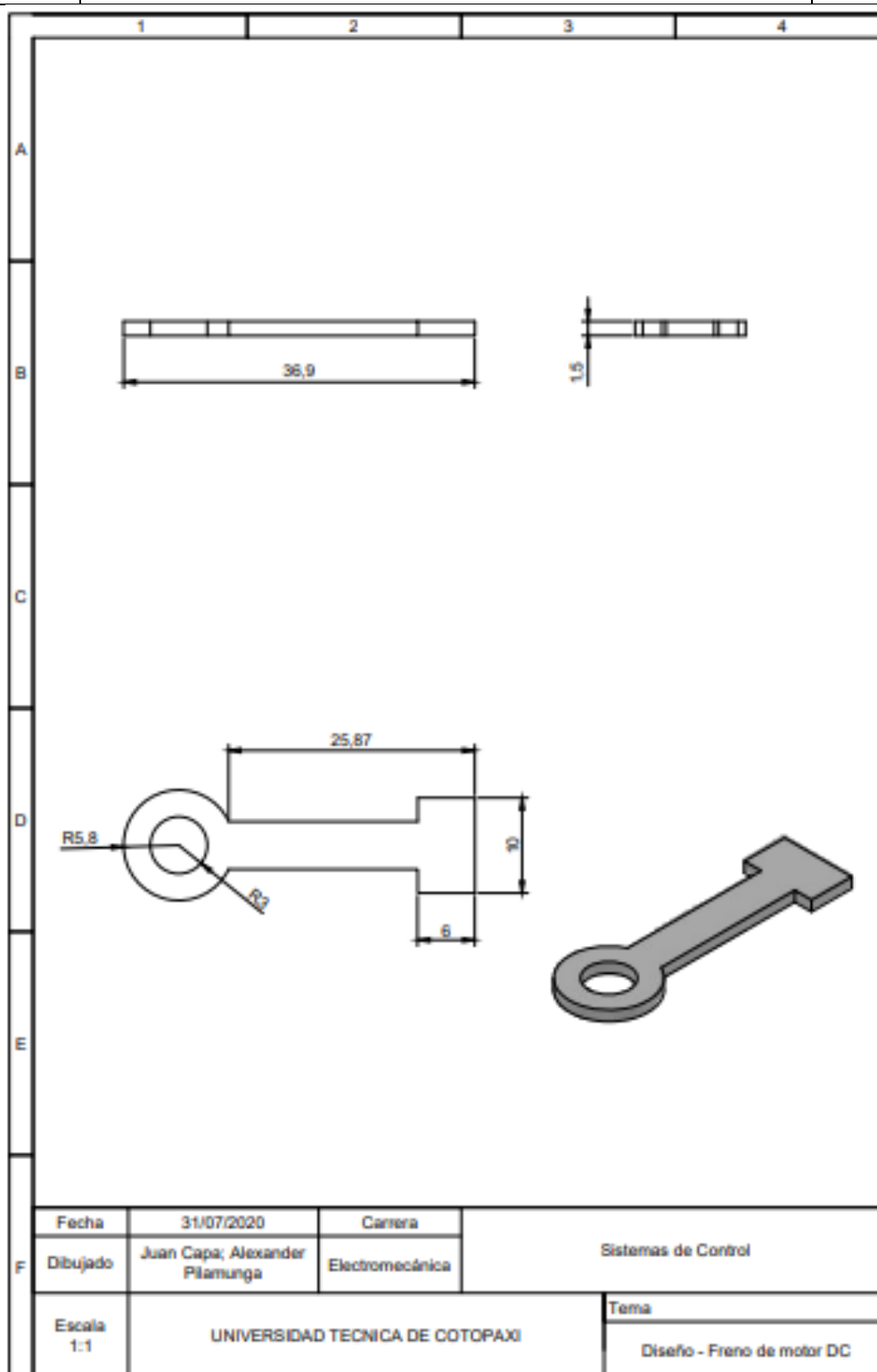
Fuente: autores



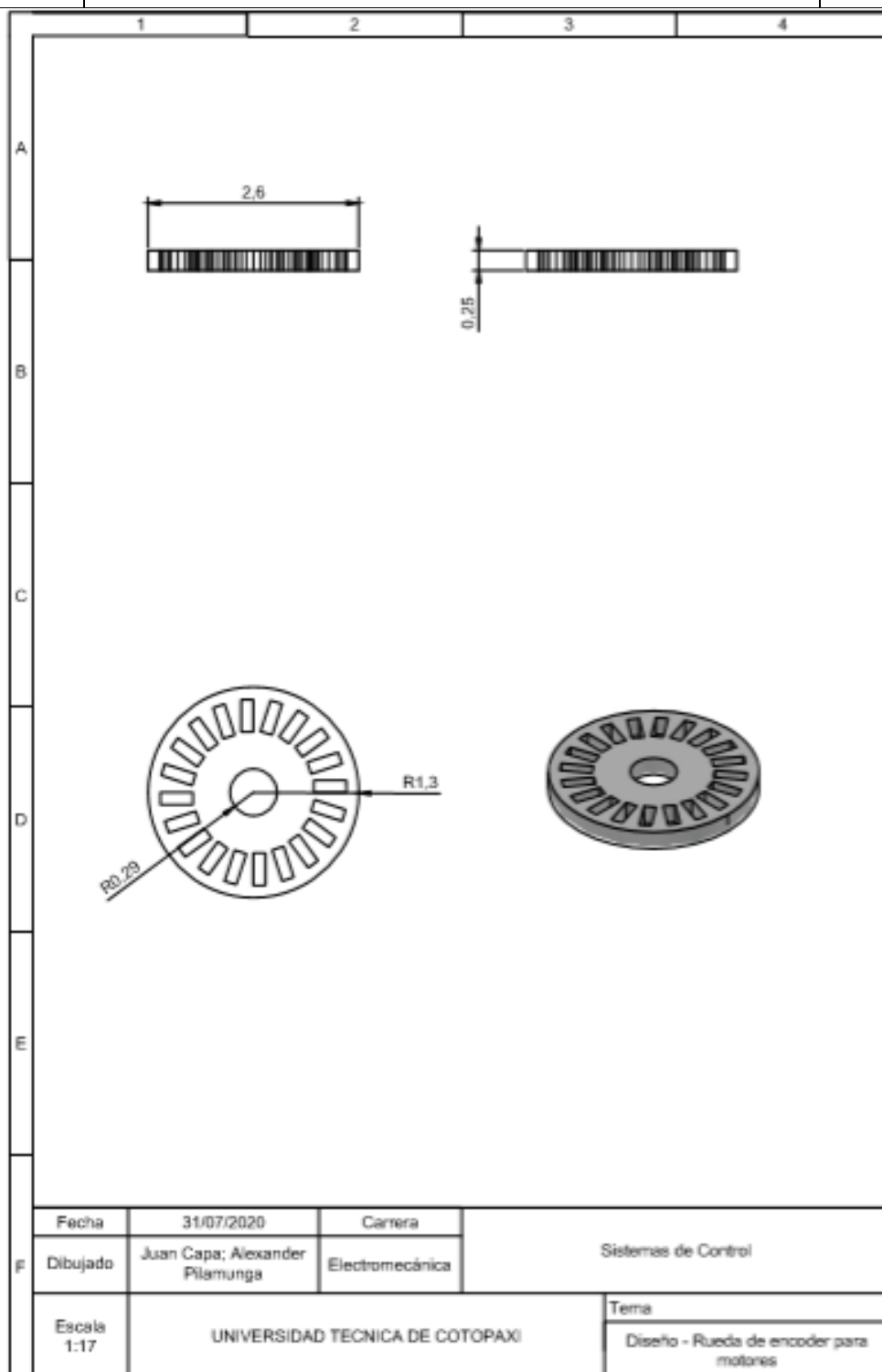
Fuente: autores



Fuente: autores



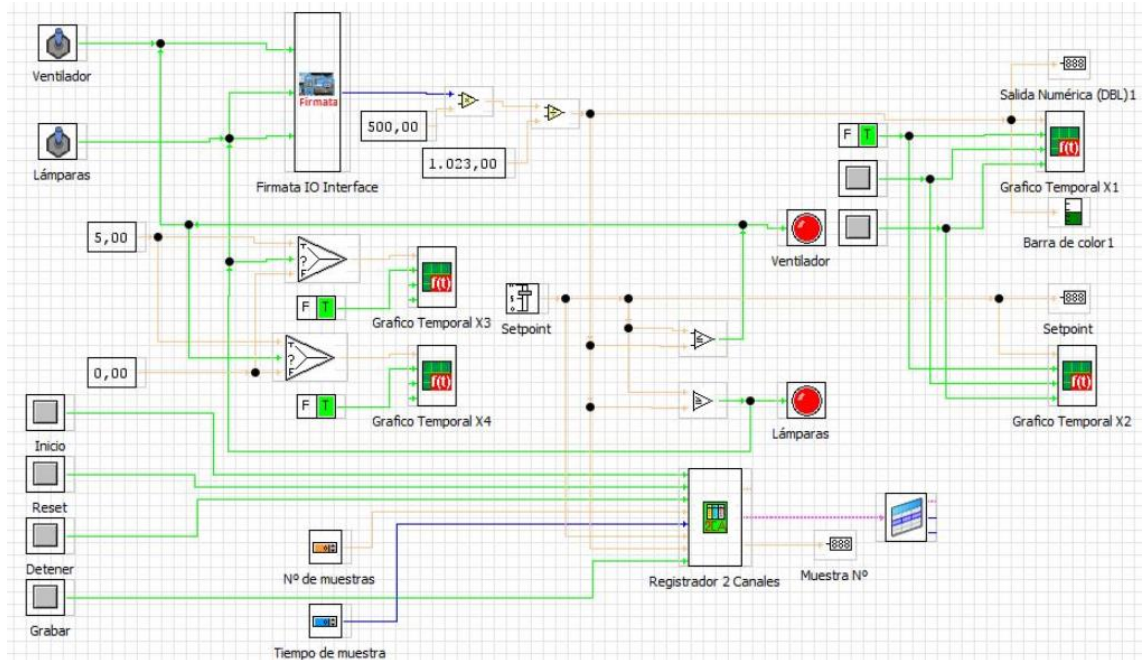
Fuente: autores



Fuente: Autores

Anexo 15

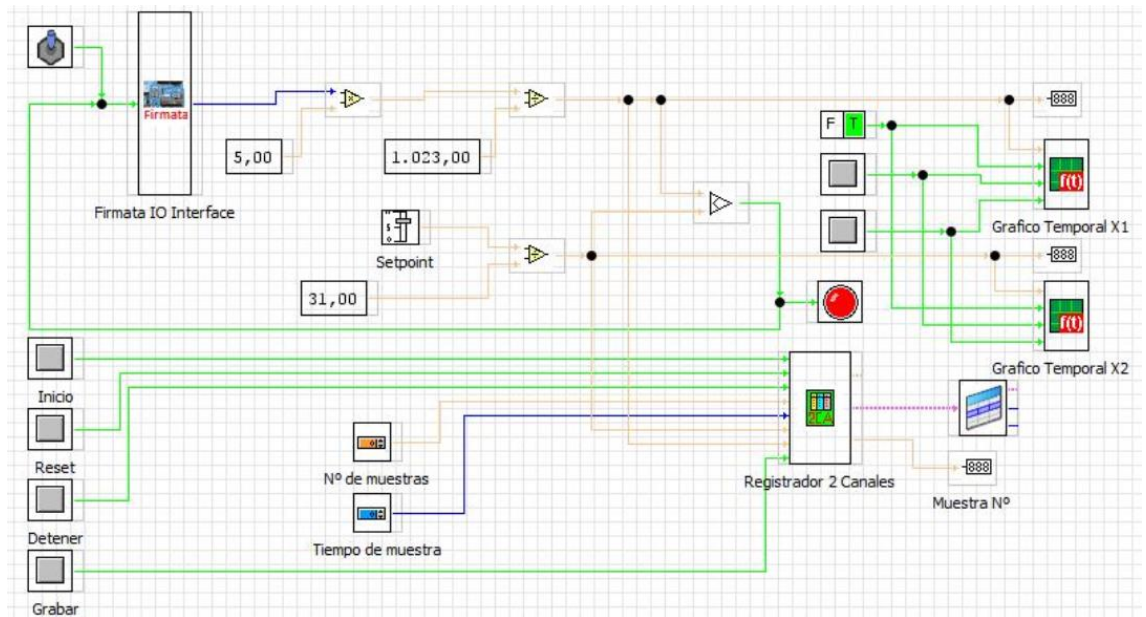
Control On/Off temperatura



Fuente: Autores

Anexo 16

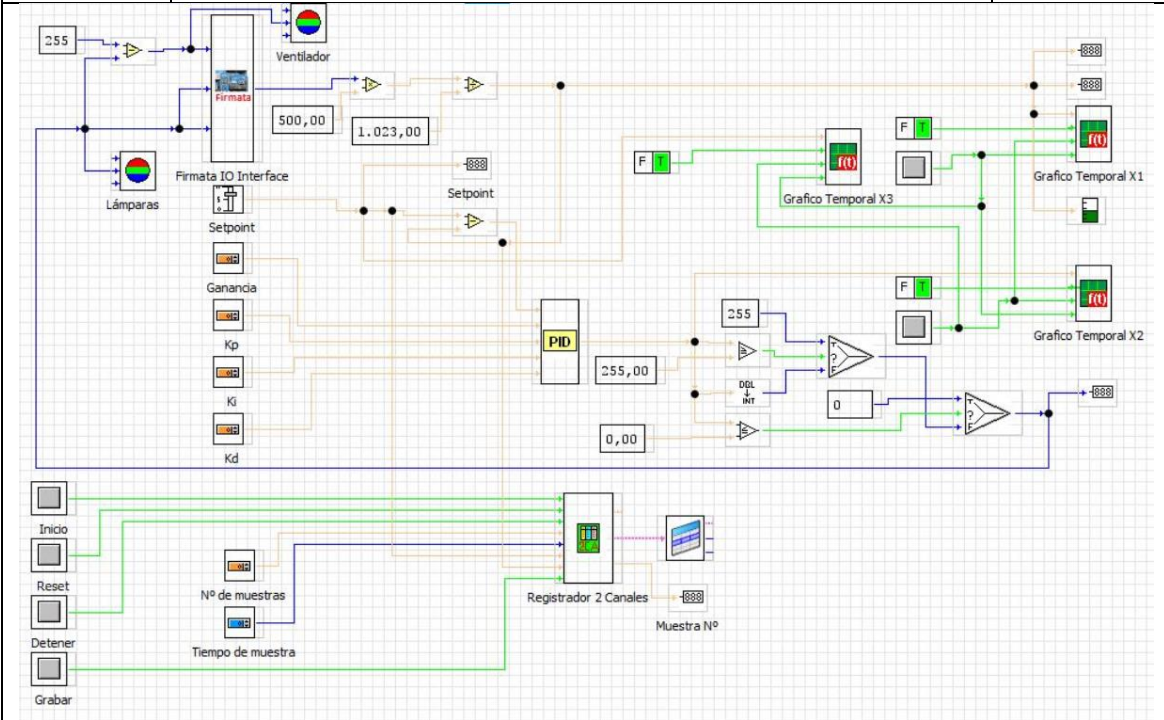
Control On/Off Intensidad Luminosa



Fuente: Autores

Anexo 17

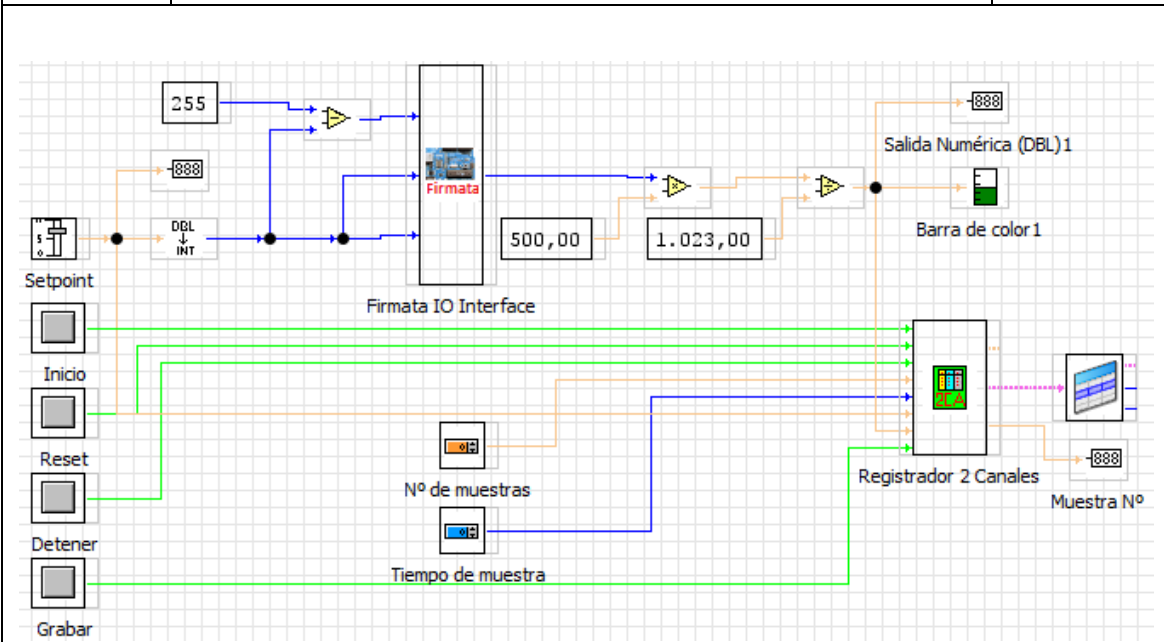
Control PID temperatura



Fuente: Autores

Anexo 18

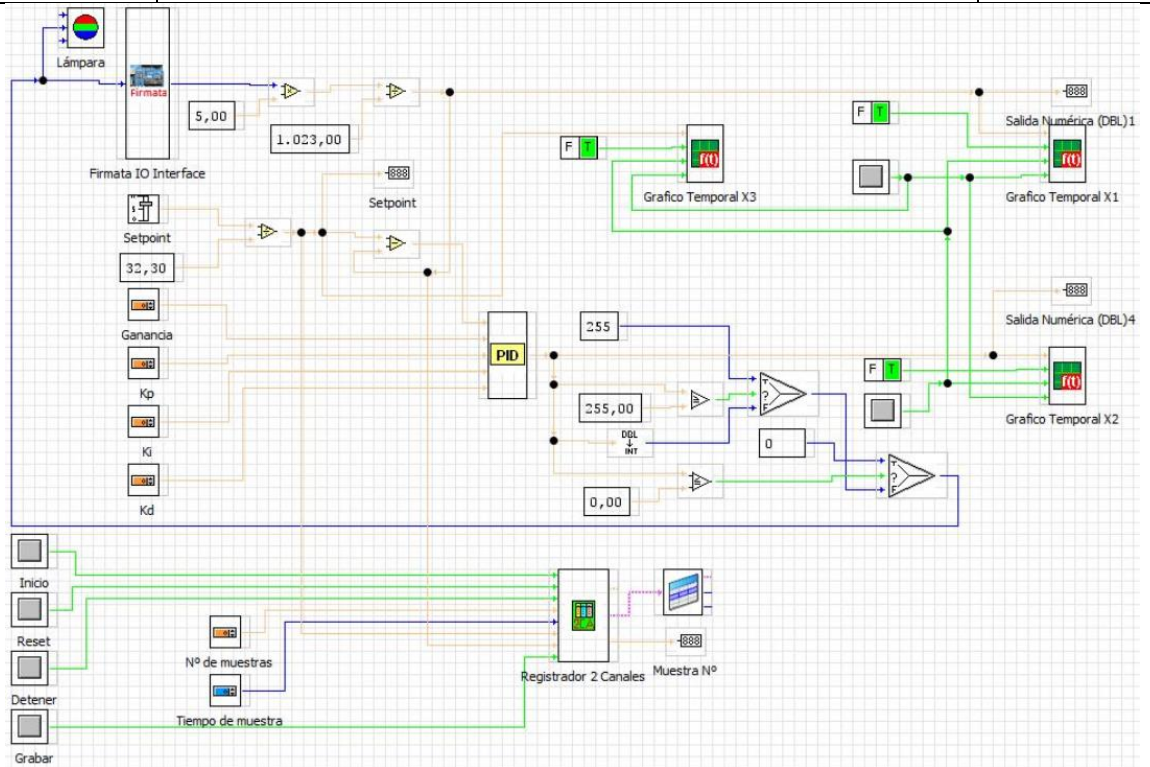
Registro de datos para la sintonización del control PID de temperatura



Fuente: Autores

Anexo 19

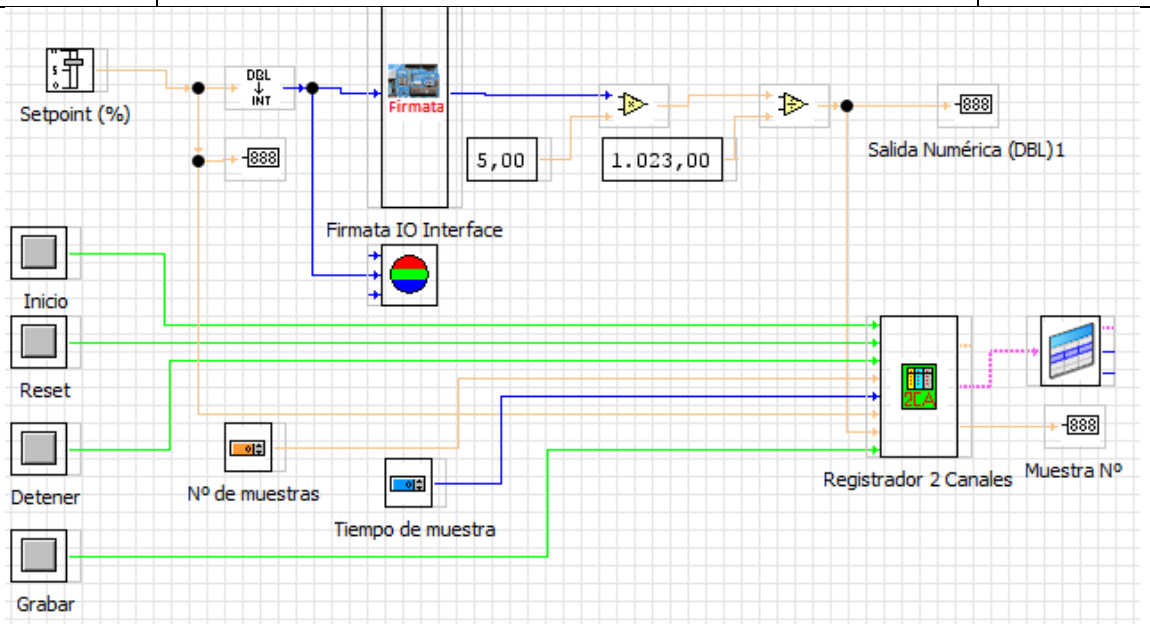
Control PID Intensidad Luminosa



Fuente: Autores

Anexo 20

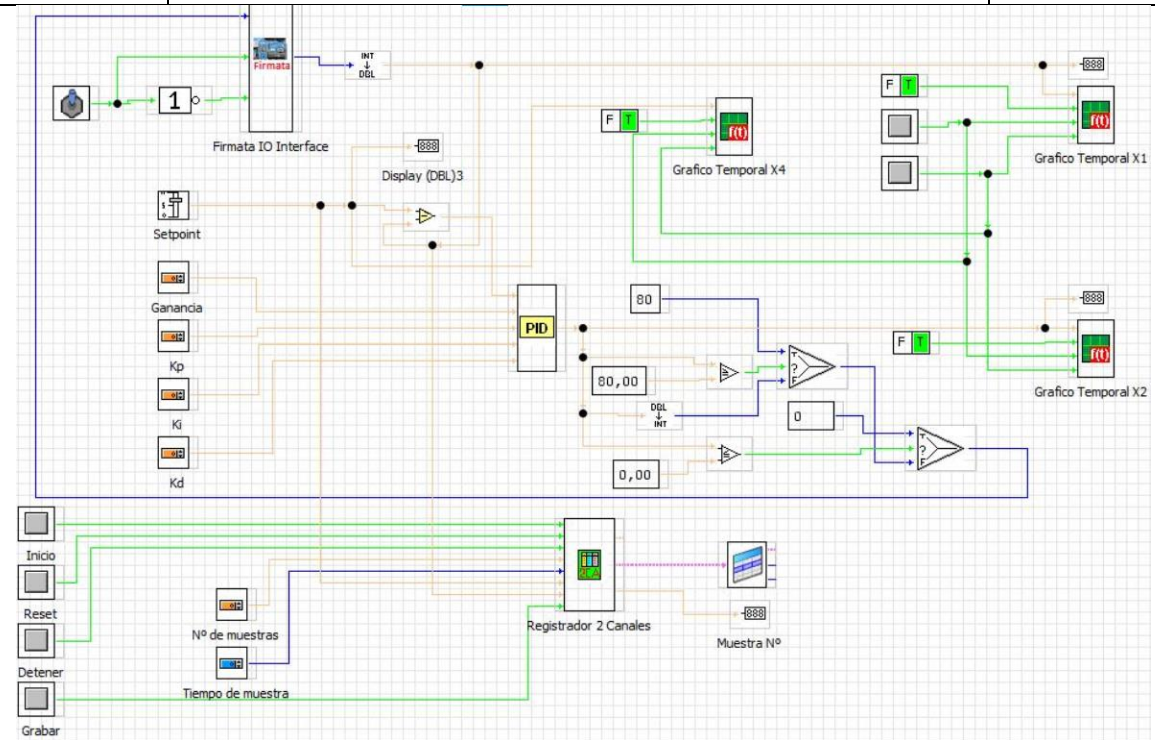
Registro de datos para la sintonización del control PID de Intensidad Luminosa



Fuente: Autores

Anexo 21

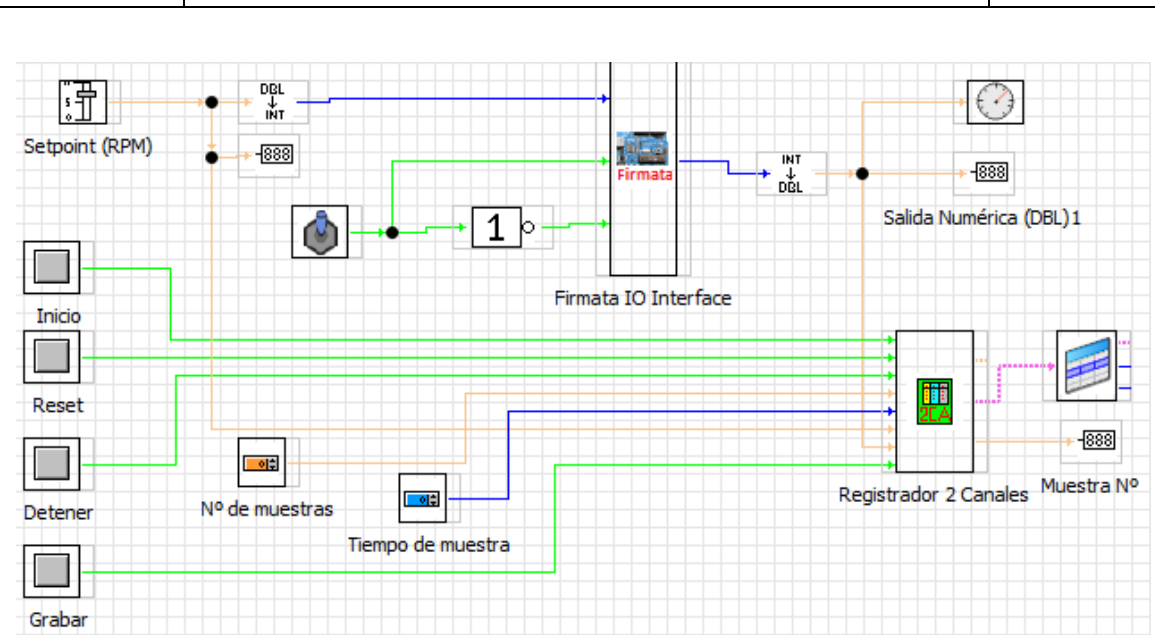
Control PID Velocidad



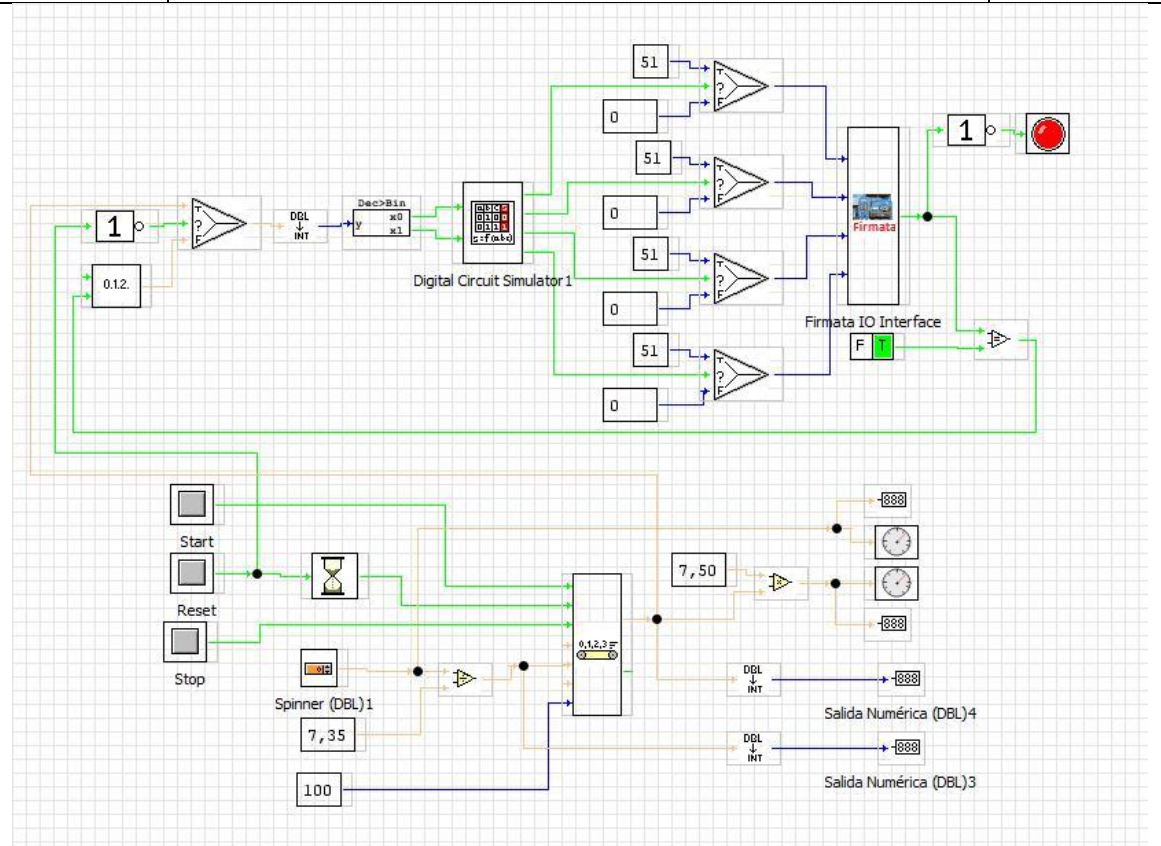
Fuente: Autores

Anexo 22

Registro de datos de velocidad PID



Fuente: Autores




Fuente: Autores

Data editor

Inputs		Outputs			
A	B	A1	A2	B1	B2
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

OK Cancel

Fuente: Autores

Anexo 24	Datos de control On/Off temperatura	
-----------------	--	---

Tiempo (S)	Setpoint (°C)	Temperatura(°C)
0	50	22,48289345
15	50	39,58944282
30	50	49,85337243
60	50	49,85337243
75	50	50,34213099
90	50	49,85337243
98	50	49,36461388

Fuente: autores

Anexo 25	Datos de control PID temperatura	
-----------------	---	---

Tiempo (S)	Setpoint (°C)	Temperatura (°C)
0	40,48031496	28,83675464
15	40,48031496	43,98826979
30	40,48031496	41,54447703
60	40,48031496	41,05571848
75	40,48031496	40,07820137
90	40,48031496	41,05571848
98	40,48031496	41,05571848

Fuente: autores

Anexo 26	Datos de control On/Off de intensidad luminosa	
-----------------	---	---

Tiempo (S)	Setpoint (V)	I. Luminosa (V)
0	0	0,053763441
15	3,47812871	3,479960899
30	3,47812871	2,072336266
60	3,47812871	1,783968719
75	3,47812871	3,646138807
98	3,47812871	3,250244379

Fuente: Autores

Anexo 27	Datos de control PID de intensidad luminosa	
----------	---	---


Tiempo (S)	Setpoint (V)	I. Luminosa (V)
0	0	0,048875855
15	3,6591	3,685239492
44	3,6591	3,660801564
60	3,6591	3,660801564
75	3,6591	3,660801564
90	3,6591	3,655913978
98	3,6591	3,655913978

Fuente: autores

Anexo 28	Datos de control de PID Velocidad	
----------	-----------------------------------	---


Tiempo (S)	Setpoint (RPM)	Encoder (RPM)
0	0	0
3	0	0
15	4745	2222
30	4745	4689
60	4745	4749
75	4745	4763
99	4745	4770

Fuente: Autores

Anexo 29	Datos de control de RPM y cambio de giro en el tiempo	
-----------------	--	---

Tiempo (S)	Setpoint (RPM)	Encoder (RPM)
0	3444,15160583941	0
5	4671,77	4562
10	4671,77	4696
20	4671,77	4735
30	4671,77	4646
40	4671,77	4550
Cambio de giro comportamiento		
50	4671,77	4579
60	4671,77	4662
70	4671,77	4658
80	4671,77	4614
90	4671,77	4598
99	4671,77	4553

Fuente: Autores

Anexo 30	Datos tabla motor a paso	
-----------------	---------------------------------	---

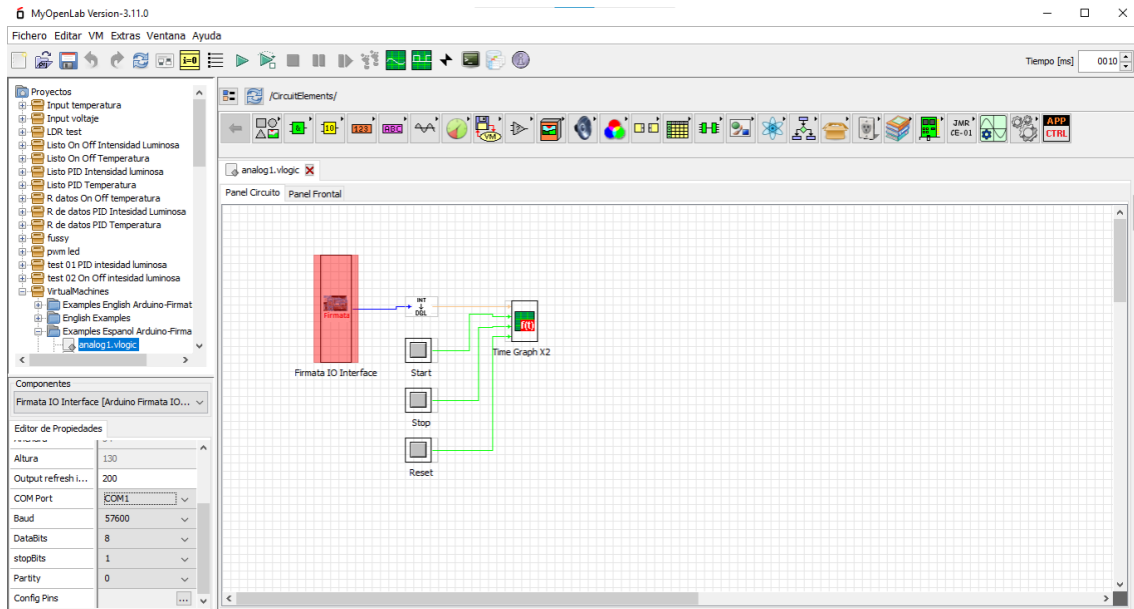
Equivalencias de números de grados con el número de pasos, esto basado que el motor posee 48 pasos en una revolución

7.5°	1 paso
45°	6 pasos
90°	12 pasos
180°	24 pasos
270°	36 pasos
360°	48 pasos

Fuente: Autores

Anexo 31

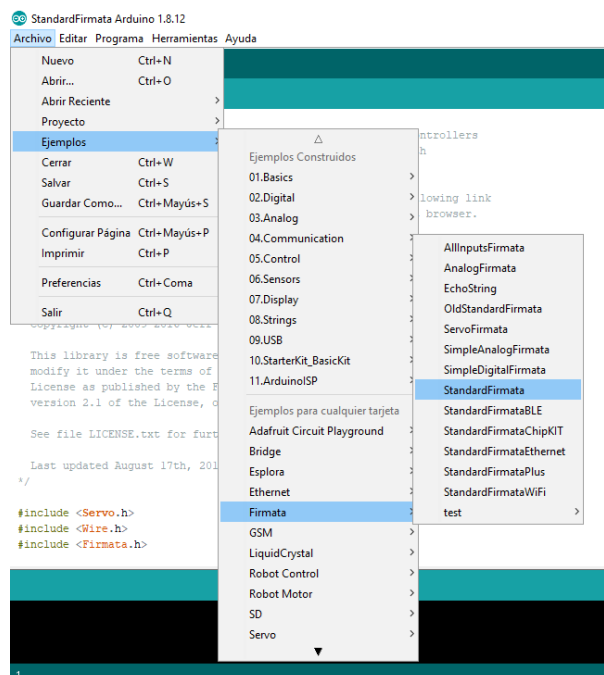
Ejemplo de programación para probar la comunicación con Arduino



Fuente: Autores

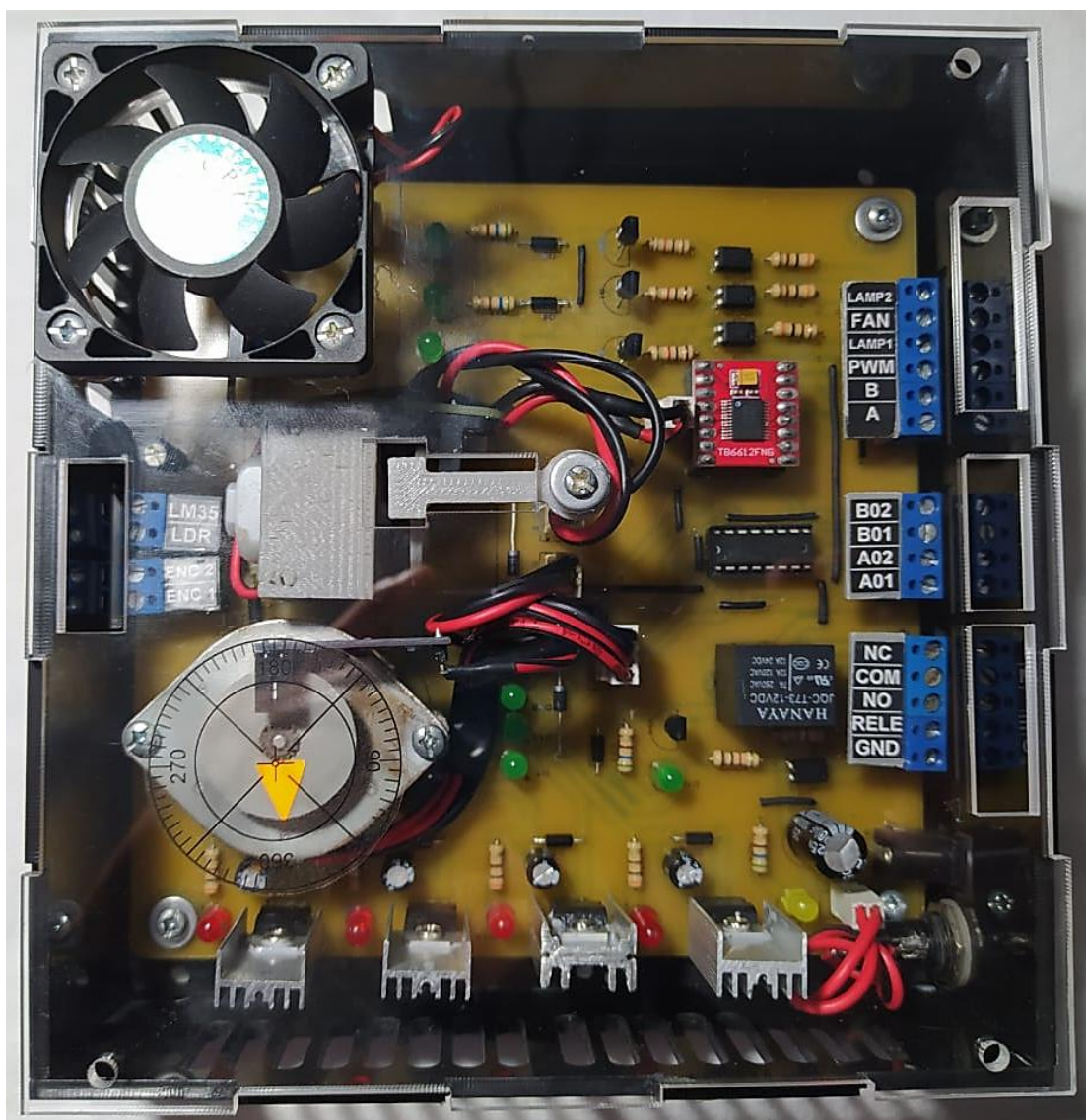
Anexo 32

Guía para cargar el “Standard Firmata”



Fuente: Autores

Manual de uso del módulo para Sistemas de Control



Antes de manipular el módulo se debe leer el presente manual

El presente manual tiene información de las características, componentes y uso, debe ser leído antes de manipular el modelo de Sistemas de control

1. Descripción

El módulo es un equipo electromecánico, diseñado para desarrollar prácticas y análisis de control de variables de Temperatura, Intensidad Luminosa, Velocidad y posición, utilizado como plataforma de programación gráfica el software MyOpenLab y como tarjeta de adquisición de datos una tarjeta Arduino, entre las prácticas de laboratorio que se pueden desarrollar están:

- Control On/Off de temperatura e intensidad luminosa
- Control PID de temperatura, intensidad luminosa y velocidad
- Control de RPM
- Control de posición de motor Step (paso a paso)
- Control de la activación de Relé mecánico

Como elementos sensores, el módulo dispone de un LM-35 para temperatura, LDR para censo de la intensidad luminosa, y encoder ópticos para el control de velocidad, posición alimentados a 5V, y como elementos actuadores se tiene lámparas incandescentes, ventilador, motor DC y motor bipolar tipo Step alimentación a 12 V.

2. Características

Módulo de Sistemas de Control		
	Características	Observación
Material	Caja de acrílico	
Dimensiones	166 x 169 x 51 (mm.)	
Alineación del módulo	12 (V), 5 (A)	
Componentes eléctricos		
Lámparas incandescentes	12 (V)	Evitar abrir y modificar los componentes del módulo
Ventilados de 50 x 50 mm		
Motor Step		
Relé		
Leds indicadores 5 (mm.)	5 (V)	No exceder el rango de voltaje de alimentación para evitar averías y perturbaciones de los componentes electrónicos y electromecánicos
Lm-35		
LDR		
Encoder 1 y 2		
ROB-14450		

Motor DC		
L293D	12 (V) y 5(V)	

3. Instalación de Software

Para la utilización del módulo para el estudio de sistemas de control se requiere instala MyOpenLab y Arduino, esto son los requisitos mínimos para la instalación del software:

Requisitos MyOpenLab

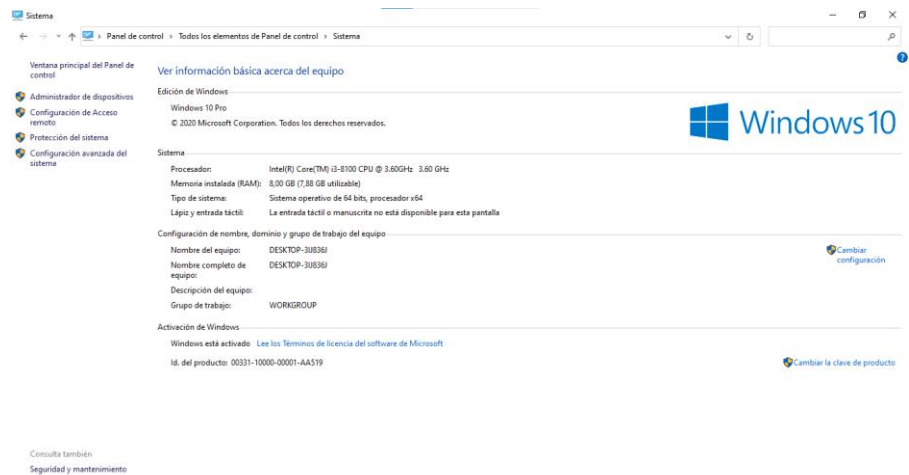
- Tener instalado Java de 32 o 64 bits

Instalación

1. Ingresar a <https://www.java.com/es/download/>
2. Según el Sistema Operativo que posea, descargue el correspondiente instalador:



Si usted no conoce cuál es el sistema operativo de su computador, puede consultar en la siguiente dirección: Panel de control/ sistema para Windows 10.



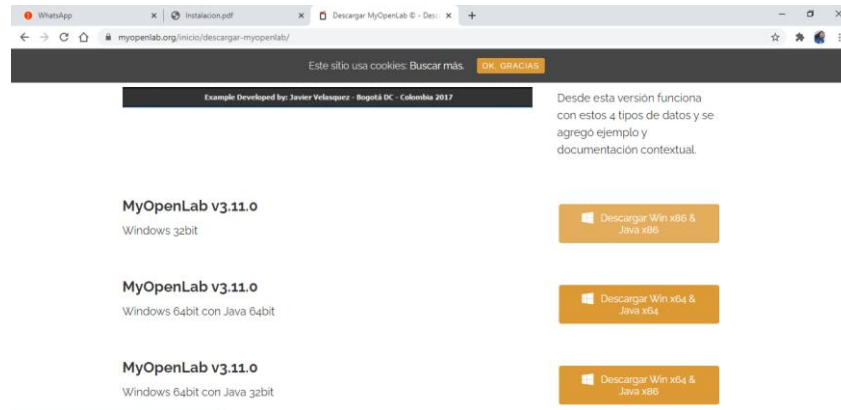
3. Al finalizar la descargar, proceda con la instalación
4. Finalizada la instalación se recomienda el reinicio de ordenador

Instalación MyOpenLab

1. Diríjase a la siguiente página y descarga la última versión:

<https://myopenlab.org/inicio/descargar-myopenlab/>

Recuerde descargar MyOpenLab para versión de su ordenador



2. Finalizada la descarga, proceda a descomprimir el archivo en la carpeta que desee alojar el programa, es un programa portable por lo que no requiere de una instalación.
3. Para ejecutar el programa se dirige a la carpeta donde se descomprimió el archivo y da doble clic sobre el icono Star_Windows
4. Por último, el programa se abrirá.

Instalar ambiente de desarrollo Arduino

1. Dirigirse a la siguiente página y descargar el archivo para su sistema operativo:

<https://www.arduino.cc/en/Main/Software>



1. Al finalizar la descarga, proceda a descomprimir el archivo en la carpeta que desee alojar el programa, es un programa portable por lo que no requiere de una instalación.

2. Para ejecutar el programa se dirige a la carpeta donde se descomprimió el archivo y da doble clic sobre el icono Arduino.

4. Distribución de componentes

En la siguiente figura se muestra los componentes que conforman el módulo y la distribución respectiva de cada uno.

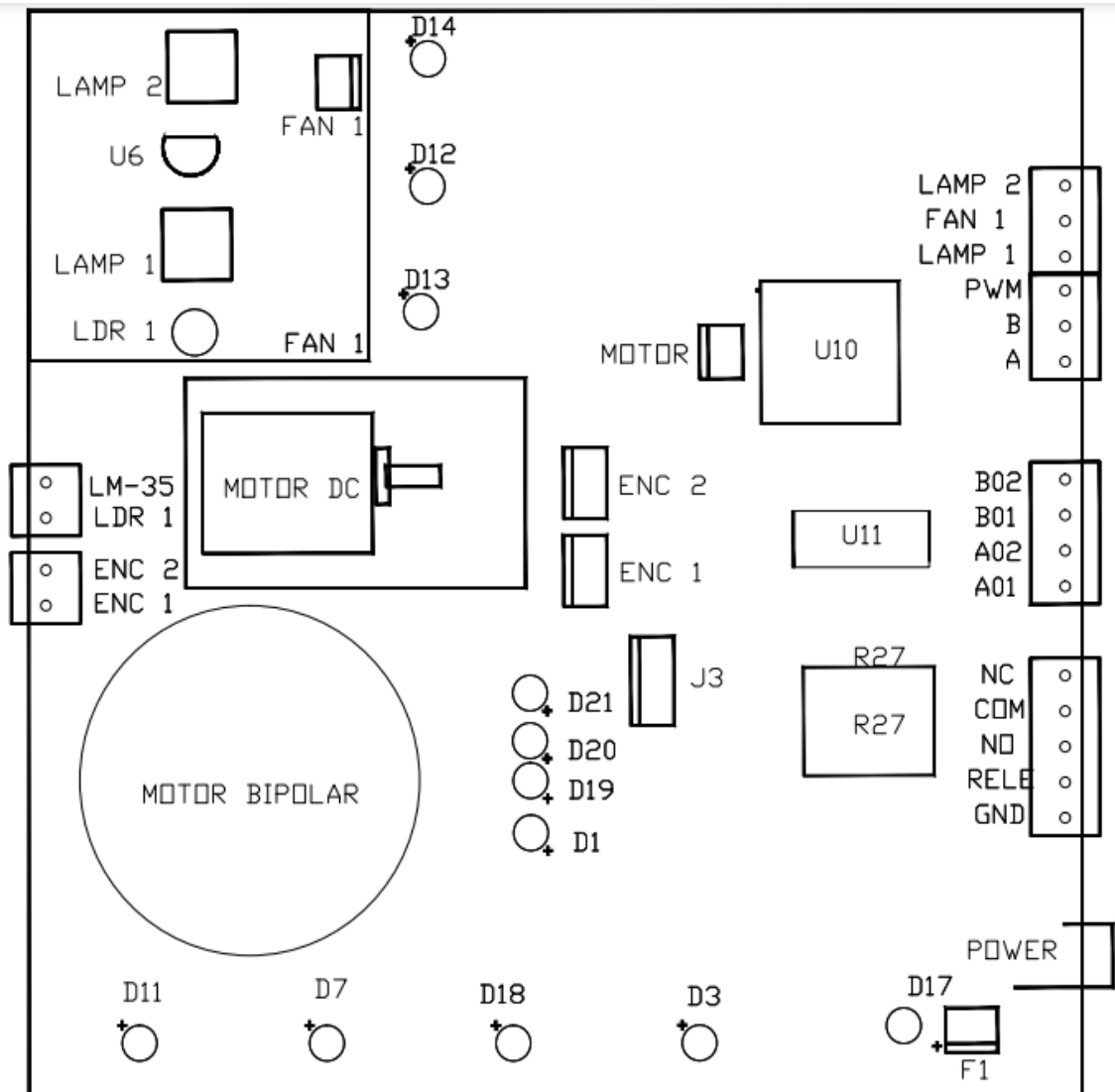


Figura 1. Disposición de los componentes e indicadores

Numeración de indicadores y componentes			
	Nomenclatura	componente	Indicador
Sensores	U6	LM-35	N/A
	N/A	LDR	N/A
	N/A	Encoder 1	N/A
	N/A	Encoder 2	N/A
Actuadores	LAMP 1	Lámpara 1	D13
	LAMP 2	Lámpara 2	D12
	FAN 1	Ventilador 1	D14
	MOTOR DC	Motor DC	N/A
	MOTOR BIPOLAR	Motor Step	N/A
	R27	Rele	N/A
Driver	U10	ROB-14450	N/A
	U11	L293d	N/A
Conectores Molex	FAN 1	Ventilador	N/A
	MOTOR	Motor	N/A
	ENC 1	Encoder 1	N/A
	ENC 2	Encoder 2	N/A
	J3	Motor Step	D21
			D20
			D19
D1			
F1	Fusible	N/A	
Alimentación	12 (V)	Power	N/A
	5 (V)	LM-35	D7
	5 (V)	LDR	D11
	5 (V)	Encoder 1	D3
	5 (V)	Encoder 2	D18

5. Pines de conexión

Conectores de salida (Sensores): estos conectores se encuentran en la parte derecha del módulo, son la Digital y Analog Out, y corresponde al LM-35 para censar temperatura,

LDR censa la intensidad luminosa, encoder 1 destinados al censo de posición y el encoder 2 mide las PRM de motor DC

Conectores de Entrada (Actuadores): se encuentra distribuidos a lo largo de la parte derecha de módulo, se dividen en:

Control	Nomenclatura	
Temperatura	LAMP 1	
	FAN 1	
	LAN 2	
Intensidad luminosa	LAMP 1	
Velocidad	A	
	B	
	PWM	
Posición	A01	
	A02	
	B01	
	B02	
Activación de Relé	RELE	
	Contactos	NC
		COM
		NO

Conectores Tierra: Con la nomenclatura (GND), esta se debe conectar al pin GND de la tarjeta de adquisición Arduino.

6. Requisitos para utilizar el módulo de control

Para completar el desarrollo de la práctica con éxito se debe completar los siguientes requisitos:

- Leer el manual de uso para el usuario antes de manipular el módulo
- Tener conocimientos básicos sobre sistemas de control, manipulación de componentes eléctricos y programación grafica en MyOpenLab.
- Poseer cualquier tarjeta Arduino que soporte el protocolo Standard Firmata.
- Ordenador o tarjeta Raspberry instalado los softwares MyOpenLab y Arduino
- Disponer de la fuente de alimentación de 12 (V)

- Destornillador plano tipo bornera.
- Poseer las guías de prácticas correspondientes.

Manipulación de módulo

- Colocar el módulo sobre una superficie plana y estable para su manipulación y conexión.
- No se puede trabajar con la caja de acrílico abierta, asegúrese que esta esté cerrada.
- La fuente de alimentación debe ser la adecuada y con el conector indicado, no hacer instalaciones provisionales, ni adaptaciones de alimentación.
- Utilizar cables cubierto con aislantes y de preferencia con terminales jumper que calcen sobre los terminales.

Conexión de módulo

- A parte de la alimentación de 12 (V), el módulo no requiere ningún tipo de alimentación extra ni de potencia 12 (V), ni de señal 5 (V), solo se debe conectar el terminal GND con el conector GND de la tarjeta Arduino
- La comunicación del módulo con el software MyOpenLab, se la realiza por medio de la tarjeta de adquisición de datos Arduino conectada al ordenador por un puerto tipo USB.
- Las conexiones para el control de las variables, se deber realizar con el módulo des-energizados y apoyándose en las instrucciones de las guías de laboratorio para cada sistema de control.

Almacenamiento

- Evitar húmedos y el contacto de equipo con sustancias liquidas esto puede ocasionar cortocircuito
- Evitar colocar objetos, cosas sobre la tapa de la caja, para evitar rallones, fisuras y trizaduras del acrílico por el peso.
- Evitar la manipulación de los tornillos colocados a sus esquinas ya que por la conste manipulación podríamos averiar la parte del acrílico.

Transporte

- Se debe tener cuidado con el trasporte de equipo, ya que posee elementos que se pueden romper o desmontarse de su socket como las lámparas incandescentes, los circuitos integrados o el driver.

Datos importantes

- No realizar conexiones ni modificaciones con el módulo encendido
- El módulo no requiere de alienación extra para funciona, a parte de los 12 (V) de la fuente con el conector indicado
- La temperatura máxima a la que llega las lámparas inductoras es de 90 °C en un tiempo de 5 minutos, evitar llegar a estas temperaturas, poder ocasionar problemas en los componentes eléctricos cercanos, para evitar esto definir un Setpoint máximo de 60 °C
- Evitar el frenado constante del motor DC, para evitar sobre cargas que pueden dañar el bobinado interno, el fin del freno es ver la permutación que se producen cuando se le introduce carga
- La señal para el control del motor Step es de 1 (V) equivalente a 51 (int) dato entero como se muestra en la guía práctica, para evitar la sobrecarga de tensión en los bobinados del motor bipolar (Step)

Anexo 34	UNIVERSIDAD TÉCNICA DE COTOPAXI	
----------	--	--

PRÁCTICA DE LABORATORIO

CARRERA	CÓDIGO DE LA ASIGNATURA	NOMBRE DE LA ASIGNATURA
INGENIERÍA ELECTROMECAÁNICA	ILM702M2	SISTEMAS DE CONTROL

PRÁCTICA N°	LABORATORIO:	Sistemas de Control	DURACIÓN (HORAS)
01	NOMBRE DE LA PRÁCTICA:	Control On/Off de Temperatura	2

1	OBJETIVOS
----------	------------------

- Cargar el código del protocolo “StandardFirmata” a la tarjeta Arduino
- Desarrollar la programación del control On/Off de temperatura en el software MyOpenLab, tanto del panel de circuito como el panel frontal
- Conectar el módulo a las salida y entradas; análogas y digitales de la tarjeta de adquisición (Arduino)
- Recopilar datos del comportamiento del Sistema de Control y realizar la gráfica.
- Analizar la forma de onda resultante del control.

2	FUNDAMENTACIÓN TEÓRICA
----------	-------------------------------

2.1. INTRODUCCIÓN:

El Laboratorio de Sistemas de control cuenta con equipos para realizar de prácticas, a través de las cuales se pretende, complementar el estudio teórico con el práctico y crear en el estudiante habilidades para el diseño e implementación de Sistemas de control, generalmente utilizados en plantas industriales, procesos de producción, en la automatización de procesos, etc. El término control hace referencia a los métodos y maneras de gobernar el comportamiento de un aparato, máquina o proceso. En un sistema de control, hay varios parámetros de entrada que actúan sobre otros parámetros de salida, organizados en un orden lógico de control.

Un sistema de control ON/OFF también llamado control todo o nada o control abierto/cerrado, es el sistema más simple de controlar una variable, es un control que usa un algoritmo simple para comparar la variable de proceso, si este está por encima o por debajo del Setpoint. La señal de control cambia entre totalmente apagado a totalmente encendido sin estados intermedios. La adquisición de datos y señales consiste en la toma de muestras de sistemas análogos (mundo real) con el fin de generar datos que puede ser manipulados, modificados e interpretados por un ordenador o sistema digital. También poder definirse como: tomar un conjunto de señales físicas y convertirlas en tensión y corriente eléctrica y digitalizarlas para ser procesados por un computador. Dentro del proceso de adquisición de datos se requiere una etapa de acondicionamiento de datos y señales con el fin de adecuar la señal a niveles compatibles con el elemento digital.

2.2 EQUIPO, INSTRUMENTOS Y MATERIALES NECESARIOS

- Módulo de Sistemas de Control
- Fuente de 12 voltios.
- Destornillador plano tipo bornera
- Cables Jumper
- Computadora

2.3 SOFTWARE REQUERIDO

- MyOpenLab
- Arduino
(revisar manual de uso, para su instalación)

2.4 MEDIDAS DE SEGURIDAD

- Revisar el manual de uso del módulo de sistemas de control
- Realizar la conexión de la tarjeta Arduino con el módulo des-energizado y cuando los leds indicadores no estén encendidos.

2.5 ACTIVIDADES A DESARROLLA

2.5.1 Cargar el código del protocolo “StandardFirmata” a la tarjeta Arduino, siguiendo el siguiente enlace: [Abrir Arduino/Archivo/Ejemplos/Firmata/Estándar Firmata](#), como se muestra en la figura 2.1

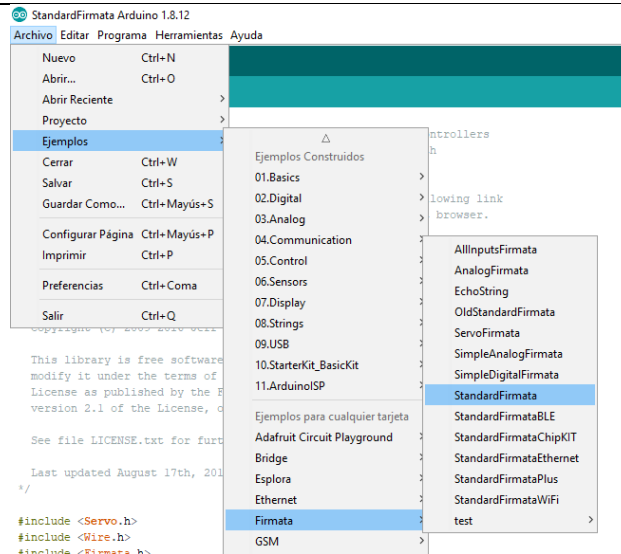


Figura 2.1. Guía para cargar el protocolo “StandardFirmata”

2.5.1.1. Se desplegará el código del protocolo StandardFirmata, sobre la pantalla, la tarjeta debe estar conectada previamente, se da clic en verificar, para revisar que no exista errores en el código y a continuación se da clic en Subir, si la ventana de comandos inferior cambia a naranja (error), revisar el puerto de comunicación de la tarjeta Arduino.

2.5.2 Abrir el programa MyOpenLab y crear un nuevo proyecto.

En la parte superior de la ventana de MyOpenLab, se debe seguir el siguiente enlace para crear un nuevo proyecto: Fichero/Nuevo Proyecto, dentro de la ventana “Nuevo Proyecto” se puede cambiar el nombre del proyecto y seleccionar la ubicación; como se muestra en la siguiente figura.

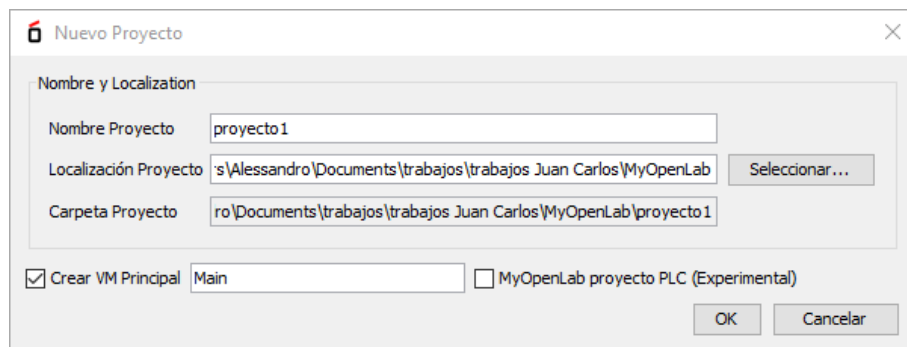


Figura 2.2 Ventana “Nuevo Proyecto”

2.5.3 Dentro del “Panel de circuito”, replicar el código de programación que se muestra en el **Anexo 1**, los pines de entrada y salida del Firmata IO Interface son:

- **1 Analog_Input:** LM-35
- **3 Digital_Output:** LAMP 1. LAMP 2, FAN 1.

Para configurar los pines se debe dar clic sobre el “Firmata IO Interface”, aparecerá una ventana en la esquina inferior “Componentes”, al final de la ventana se debe dar clic sobre los 3 puntos (...) en “Config Pins” a continuación, se desplegará una ventana donde se poder habilitar cada una de las entradas y salidas deseadas como se muestra en la siguiente imagen.

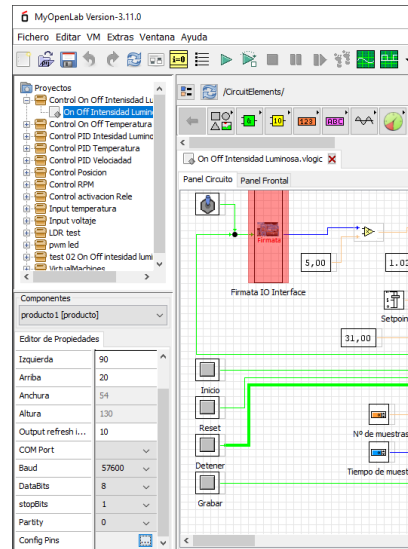


Figura 2.3 configuración de los pines de Firmata IO Interface

Antes de programar, la tarjeta Arduino debe estar conectada, para que MyOpenLab la reconozca automáticamente. La señal de entrada analógica debe ser acondicionada, para ser visualizada en grados Celsius, se multiplica 500 y se divide para 1023, a continuación, se muestra un ejemplo:

$$\text{Analog_input} = 61.38$$

$$^{\circ}\text{C} = 61.38 * \frac{500}{1023} = 30^{\circ}\text{C}$$

Condiciones para el funcionamiento de control On/Off de temperatura:

- Setpoint <= Sensor; las lámparas deben encenderse
- Setpoint >= Sensor; el ventilados deben encenderse.

En el **Anexo 2** se visualiza la apariencia del control On/Off de temperatura del panel frontal

2.5.4 Después de programar, se recomienda guardar el archivo, antes de conectar al módulo de Sistemas de Control.

2.5.5. Se debe conectar con el módulo des-energizado, al lado izquierdo de módulo se encuentran los sensores que deben conectarse a las entradas analógicas de la tarjeta Arduino, y al lado derecho se encuentran dispuestos los actuadores; en el **Anexo 3** se visualiza un ejemplo de conexión del control On/Off de temperatura.

- LM-35, puede ser conectado en: A0, A1, A2, A3, A4.

- LAMP 1, LAMP 2 y FAN 1, se conecta: entre los pines 13 hasta el 2.

2.5.6. Realizada las conexiones respectivas, se puede energizar el módulo y ejecutar la programación.

2.6. METODOLOGÍA Y TÉCNICA EXPERIMENTAL

2.6.1. Control On/Off

Este tipo de controlador, también llamado Todo o Nada, usa un algoritmo simple para solamente revisa si la variable de proceso está por encima o por debajo de un setpoint determinado. En términos prácticos, la variable manipulada o la señal de control del controlador cambia entre “totalmente ON” o totalmente OFF, sin estados intermedios. Este tipo de accionamiento provoca un control muy impreciso de la variable de proceso, un ejemplo muy común es el control de temperatura con termostatos en aires acondicionados. El termostato activa el aire frío si (ON) la temperatura es mayor a la de referencia o setpoint y lo desactivan (OFF) cuando la temperatura es menor (o igual) al setpoint. [27]

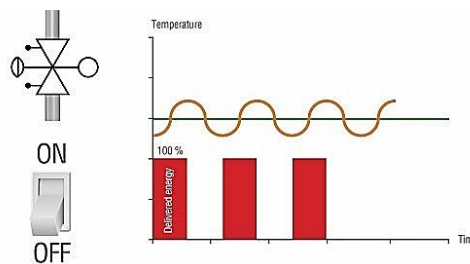


Figura 2.4 Control ON - OFF

Fuente: [14]

INFORME DE PRÁCTICA

3 RESULTADOS Y DISCUSIÓN

Para realizar el análisis del comportamiento del sistema de control, se debe establecer un Setpoint diferente a la temperatura actual y se empieza a registrar los datos, el “Tiempo de Muestreo” es el tiempo entre muestra tomada. En la figura 3.1 se visualiza la gráfica de los datos tomado, con el siguiente comportamiento:

- Temperatura inicial = 23 °C
- Setpoint = 50 °C
- Tiempo de muestro = 1000 milisegundo.

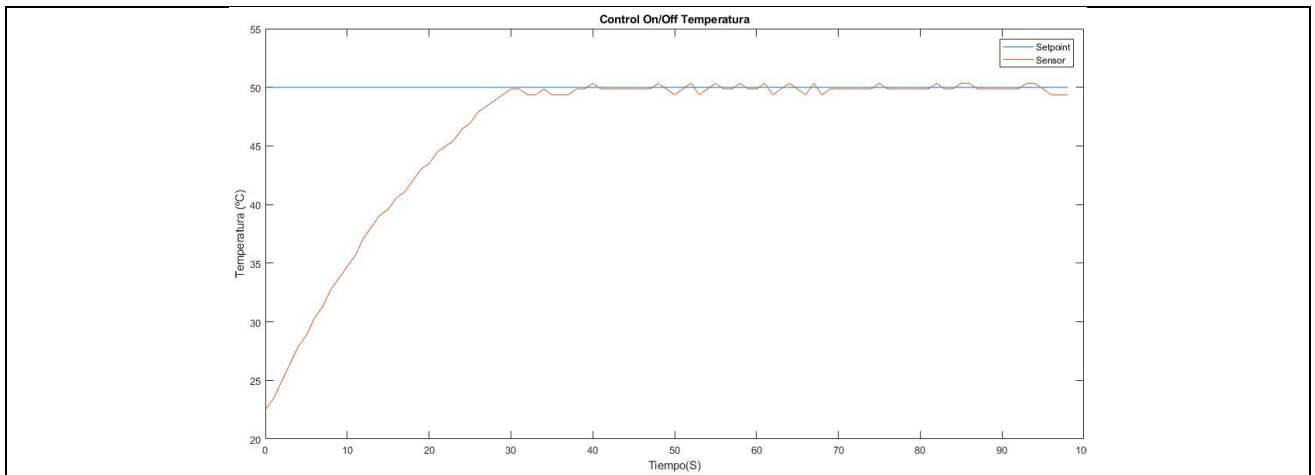


Figura 3.1 Control On/Off de temperatura con Setpoint de 50 °C

3.1 ANÁLISIS DE RESULTADOS:

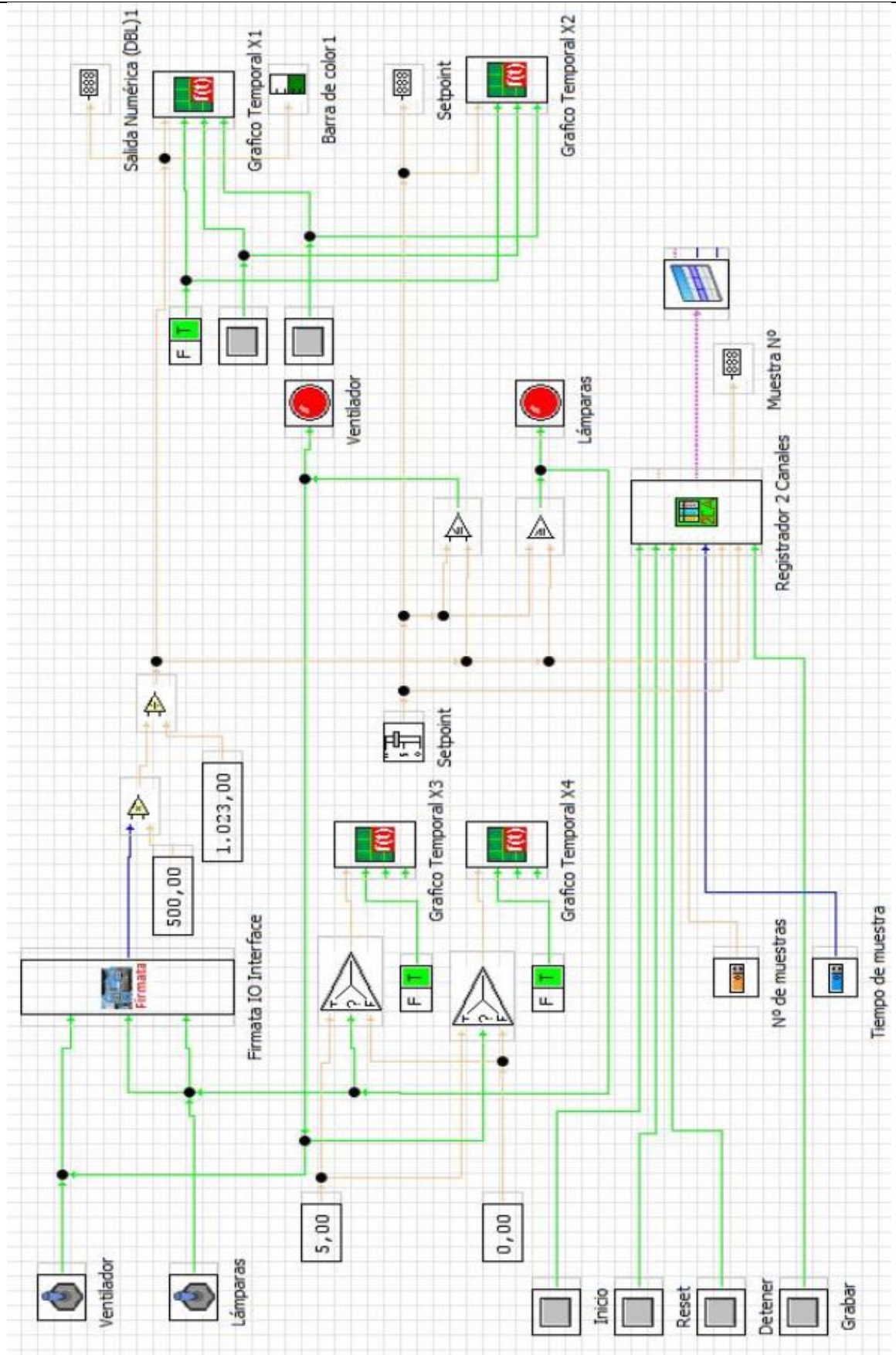
-

5	CONCLUSIONES
	•

6	RECOMENDACIONES
	<ul style="list-style-type: none"> • Para graficar la señal del Sensor y del Setpoint en la misma gráfica como se muestra en la figura 2.3 del panel frontal, se emplea dos elementos “Gráfico Temporal” X1 y X2 en el panel frontal, los dos gráficos deben tener las mismas unidades y dimensiones y no tener fondo (activar Background Transparent en las propiedades de la gráfica), para interpolar una gráfica sobre la otra.

7	REFERENCIAS BIBLIOGRÁFICAS (Utilizar las normas IEEE)
	<p>METODOLOGÍA</p> <p>[1] M. Gutiérrez and S. Iturralde, <i>Fundamentos Basicos De Instrumentacion y Control</i>. 2017.</p> <p>[2] Jose Carlos Villajulca, “Instrumentación y Automatización Industrial,” <i>Control ON/OFF o Todo/Nada</i> -, Jul. 01, 2017. https://instrumentacionycontrol.net/control-on-off-o-todo-nada/ (accessed Aug. 04, 2020).</p>

8	ANEXOS



Anexo 1 Control On/Off de temperatura, panel de circuito

Anexo 35	UNIVERSIDAD TÉCNICA DE COTOPAXI	
PRÁCTICA DE LABORATORIO		

CARRERA	CÓDIGO DE LA ASIGNATURA	NOMBRE DE LA ASIGNATURA
INGENIERÍA ELECTROMECAÁNICA	ILM702M2	SISTEMAS DE CONTROL

PRÁCTICA N°	LABORATORIO:	Sistemas de Control	DURACIÓN (HORAS)
02	NOMBRE DE LA PRÁCTICA:	Control On/Off de Intensidad Luminosa	2

1	OBJETIVOS
	<ul style="list-style-type: none"> • Cargar el código del protocolo “StandardFirmata” a la tarjeta Arduino • Desarrollar la programación del control On/Off de Intensidad Luminosa en el software MyOpenLab, tanto del panel de circuito como el panel frontal • Conectar el módulo a las salida y entradas; análogas y digitales de la tarjeta de adquisición (Arduino) • Recopilar datos del comportamiento del Sistema de Control y realizar la gráfica. • Analizar la forma de onda resultante del control.

2	FUNDAMENTACIÓN TEÓRICA
	<p>2.1. INTRODUCCIÓN:</p> <p>El Laboratorio de Sistemas de control cuenta con equipos para realizar de prácticas, a través de las cuales se pretende, complementar el estudio teórico con el práctico y crear en el estudiante habilidades para el diseño e implementación de Sistemas de control, generalmente utilizados en plantas industriales, procesos de producción, en la automatización de procesos, etc. El término control hace referencia a los métodos y maneras de gobernar el comportamiento de un aparato, máquina o proceso. En un sistema de control, hay varios parámetros de entrada que actúan sobre otros parámetros de salida, organizados en un orden lógico de control.</p> <p>Un sistema de control ON/OFF también llamado control todo o nada o control abierto/cerrado, es el sistema más simple de controlar una variable, es un control que usa un algoritmo simple</p>

para comparar la variable de proceso, si este está por encima o por debajo del Setpoint. La señal de control cambia entre totalmente apagado a totalmente encendido sin estados intermedios. La adquisición de datos y señales consiste en la toma de muestras de sistemas análogos (mundo real) con el fin de generar datos que puede ser manipulados, modificados e interpretados por un ordenador o sistema digital. También poder definirse como: tomar un conjunto de señales físicas y convertirlas en tensión y corriente eléctrica y digitalizarlas para ser procesados por un computador. Dentro del proceso de adquisición de datos se requiere una etapa de acondicionamiento de datos y señales con el fin de adecuar la señal a niveles compatibles con el elemento digital.

2.2 EQUIPO, INSTRUMENTOS Y MATERIALES NECESARIOS

- Módulo de Sistemas de Control
- Fuente de 12 voltios.
- Destornillador plano tipo bornera
- Cables Jumper
- Computadora

2.3 SOFTWARE REQUERIDO

- MyOpenLab
- Arduino
(revisar manual de uso, para su instalación)

2.4 MEDIDAS DE SEGURIDAD

- Revisar el manual de uso del módulo de sistemas de control
- Realizar la conexión de la tarjeta Arduino con el módulo des-energizado y cuando los leds indicadores no estén encendidos

2.5 ACTIVIDADES A DESARROLLAR:

2.5.1 Cargar el código del protocolo “StandardFirmata” a la tarjeta Arduino, siguiendo el siguiente enlace: [Abrir Arduino/Archivo/Ejemplos/Firmata/Estándar Firmata](#), como se muestra en la figura 2.1

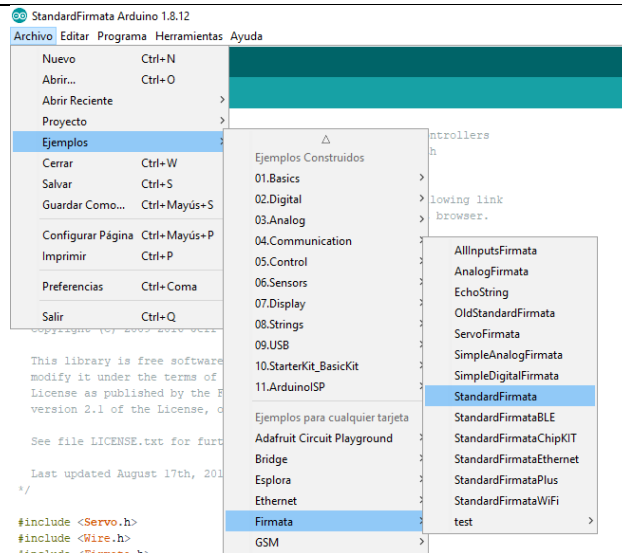


Figura 2.1. Guía para cargar el protocolo “StandardFirmata”

2.5.1.1 Se desplegará el código del protocolo StandardFirmata, sobre la pantalla, la tarjeta debe estar conectada previamente, se da clic en verificar, para revisar que no exista errores en el código y a continuación se da clic en Subir, si la ventana de comandos inferior cambia a naranja (error) revisar el puerto de comunicación de la tarjeta Arduino.

2.5.2 Abrir el programa MyOpenLab y crear un nuevo proyecto.

En la parte superior de la ventana de MyOpenLab, se debe seguir el siguiente enlace para crear un nuevo proyecto: Fichero/Nuevo Proyecto, dentro de la ventana “Nuevo Proyecto” se puede cambiar el nombre del proyecto y seleccionar la ubicación; como se muestra en la siguiente figura.

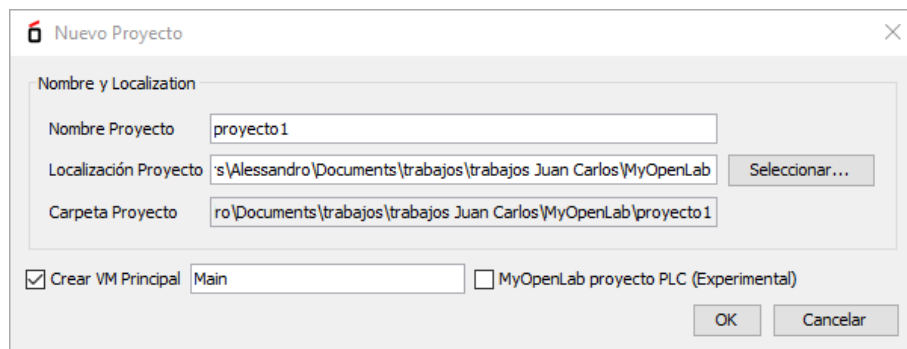


Figura 2.2 Ventana “Nuevo Proyecto”

2.5.3 Dentro del “Panel de circuito”, Replicar el código de programación que se muestra en el **Anexo 1**, los pines de entrada y salida del Firmata IO Interface son:

- **1 Analog_Input:** LDR 1.
- **1 Digital_Output:** LAMP 1.

Para configurar los pines se debe dar clic sobre el “Firmata IO Interface”, aparecerá una ventana en la esquina inferior “Componentes”, al final de la ventana se debe dar clic sobre los

3 puntos (...) en “Config Pins” a continuación, se desplegará una ventana donde se poder habilitar cada una de las entradas y salidas deseadas como se muestra en la siguiente imagen.

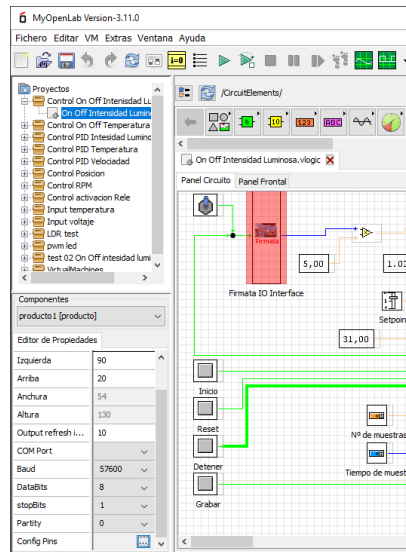


Figura 2.3 configuración de los pines de Firmata IO Interface

Antes de programar, la tarjeta Arduino debe estar conectada, para que MyOpenLab la reconozca automáticamente. La señal de entrada analógica debe ser acondicionada, para ser visualizada en voltios (V), por lo que se multiplica 5 y se divide para 1023, a continuación, se muestra un ejemplo:

$$\text{Analog_input} = 763.158 \approx \text{Lámpara encendida}$$

$$V = 763.158 * \frac{5}{1023} = 3.73 \text{ voltios (V)}$$

Condiciones para el funcionamiento de control On/Off de temperatura:

- Setpoint > Sensor; la lámpara se encenderá.

Para trabajar con un setpoint en porcentajes se debe dividir 100% para el voltaje de entrada cuando la lámpara incandescente este encendido:

$$\text{Setpoint (\%)} = \frac{100}{V} = \frac{100}{3.7119} = 26.94$$

En el **Anexo 2** se visualiza la apariencia del control On/Off de Intensidad luminosa del panel frontal

2.5.4 Después de programar, se recomienda guardar el archivo, antes de conectar al módulo de Sistemas de Control.

2.6.5. Se debe conectar con el módulo des-energizado, al lado izquierdo de módulo se encuentran los sensores que deben conectarse a las entradas analógicas de la tarjeta Arduino, y al lado derecho se encuentran dispuestos los actuadores; en el **Anexo 3** se visualiza un ejemplo de conexión del control On/Off de temperatura.

- LM-35, puede ser conectado en: A0, A1, A2, A3, A4.
- LAMP 1, se conecta entre los pines 13 hasta el 2.

2.6.6. Realizada las conexiones respectivas, se puede energizar el módulo y ejecutar la programación.

2.7. METODOLOGÍA Y TÉCNICA EXPERIMENTAL

Control On/Off

Este tipo de controlador, también llamado Todo o Nada, usa un algoritmo simple para solamente revisa si la variable de proceso está por encima o por debajo de un setpoint determinado. En términos prácticos, la variable manipulada o la señal de control del controlador cambia entre “totalmente ON” o totalmente OFF, sin estados intermedios. Este tipo de accionamiento provoca un control muy impreciso de la variable de proceso, un ejemplo muy común es el control de temperatura con termostatos en aires acondicionados. El termostato activa el aire frío si (ON) la temperatura es mayor a la de referencia o Setpoint y lo desactivan (OFF) cuando la temperatura es menor (o igual) al Setpoint. [27]

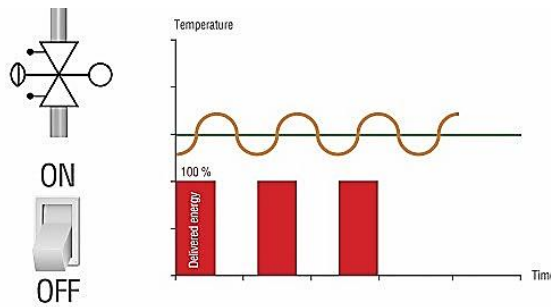


Figura 2.5 Control ON - OFF

Fuente: [14]

INFORME DE PRÁCTICA

3 RESULTADOS Y DISCUSIÓN

Para realizar el análisis del comportamiento del sistema de control, se debe establecer un Setpoint diferente a la temperatura actual y se empiece a registrar los datos, el “Tiempo de Muestreo” es el tiempo entre muestra tomada. En la figura 3.1 se visualiza la gráfica de los datos tomado, con el siguiente comportamiento:

- Intensidad luminosa inicial = 0 (V) = 0 (%)
- Setpoint = 3.47 (V) = 93.48 (%)
- Tiempo de muestro = 1000 milisegundo.

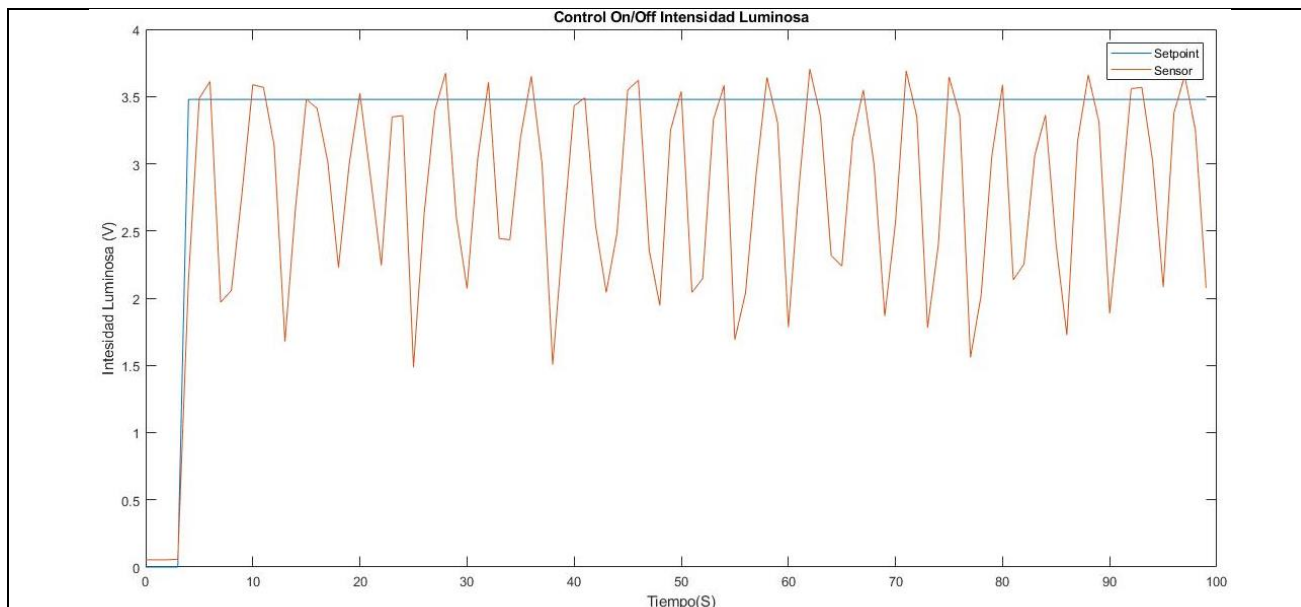


Figura 3.1 Control On/Off de Intensidad Luminosa con Setpoint de 3.47 (V)

3.2 ANÁLISIS DE RESULTADOS:

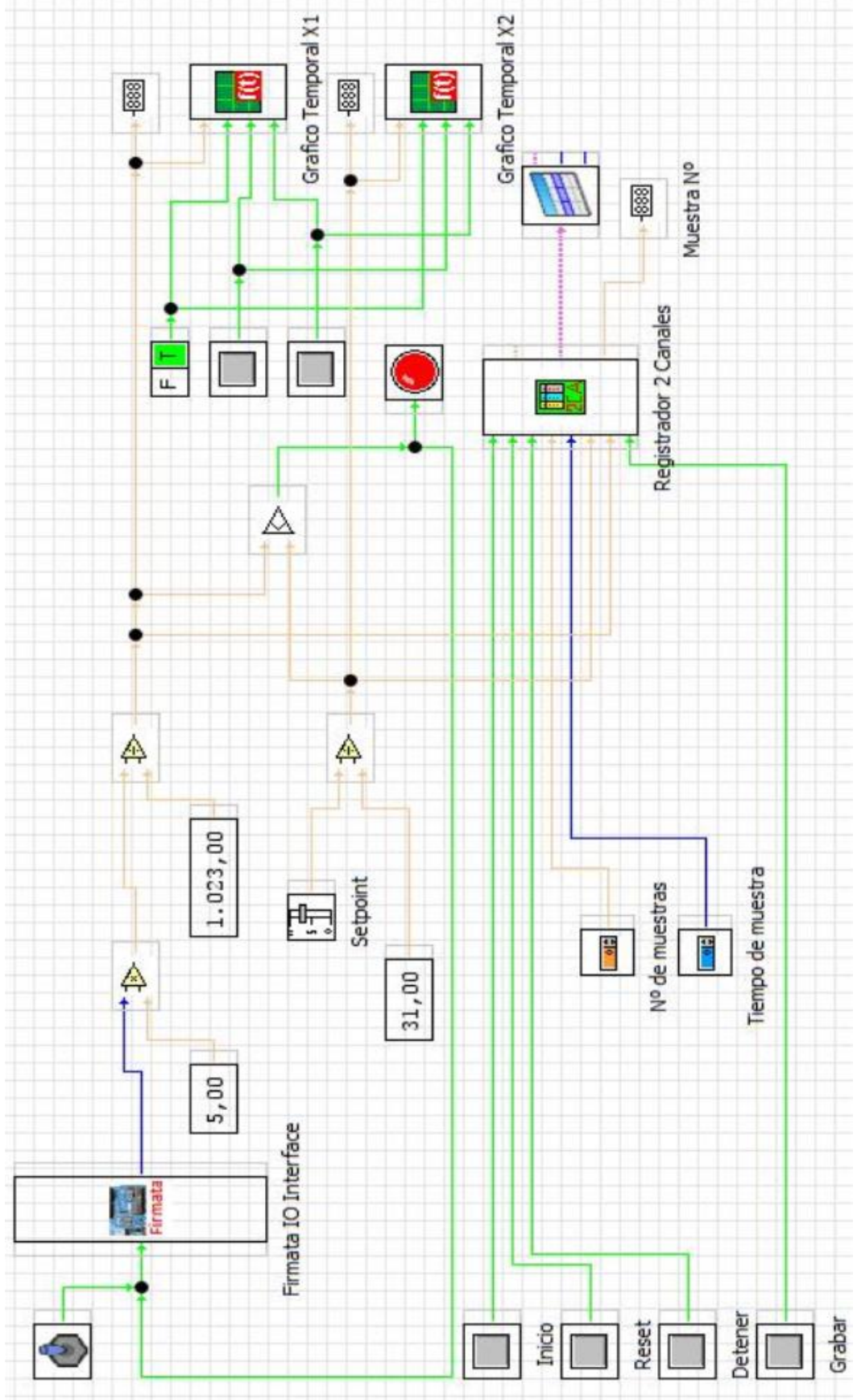
-

5	CONCLUSIONES
	•

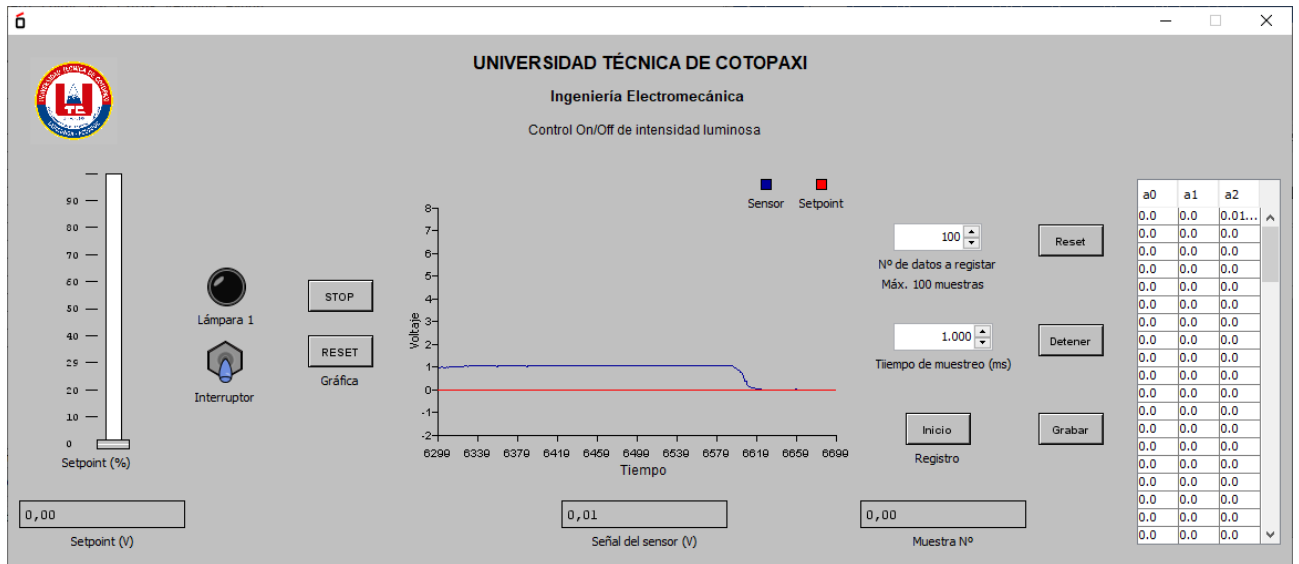
6	RECOMENDACIONES
	<ul style="list-style-type: none"> • Para graficar la señal del Sensor y del Setpoint en la misma gráfica como se muestra en la figura 2.3 del panel frontal, se emplea dos elementos “Gráfico Temporal” X1 y X2 en el panel frontal, los dos gráficos deben tener las mismas unidades y dimensiones y no tener fondo (activar Background Transparent en las propiedades de la gráfica), para interpolar una gráfica sobre la otra.

7	REFERENCIAS BIBLIOGRÁFICAS (Utilizar las normas IEEE)
	<p>METODOLOGÍA</p> <p>[1] M. Gutiérrez and S. Iturralde, <i>Fundamentos Basicos De Instrumentacion y Control</i>. 2017.</p> <p>[2] Jose Carlos Villajulca, “Instrumentación y Automatización Industrial,” <i>Control ON/OFF o Todo/Nada</i> -, Jul. 01, 2017. https://instrumentacionycontrol.net/control-on-off-o-todo-nada/ (accessed Aug. 04, 2020).</p>

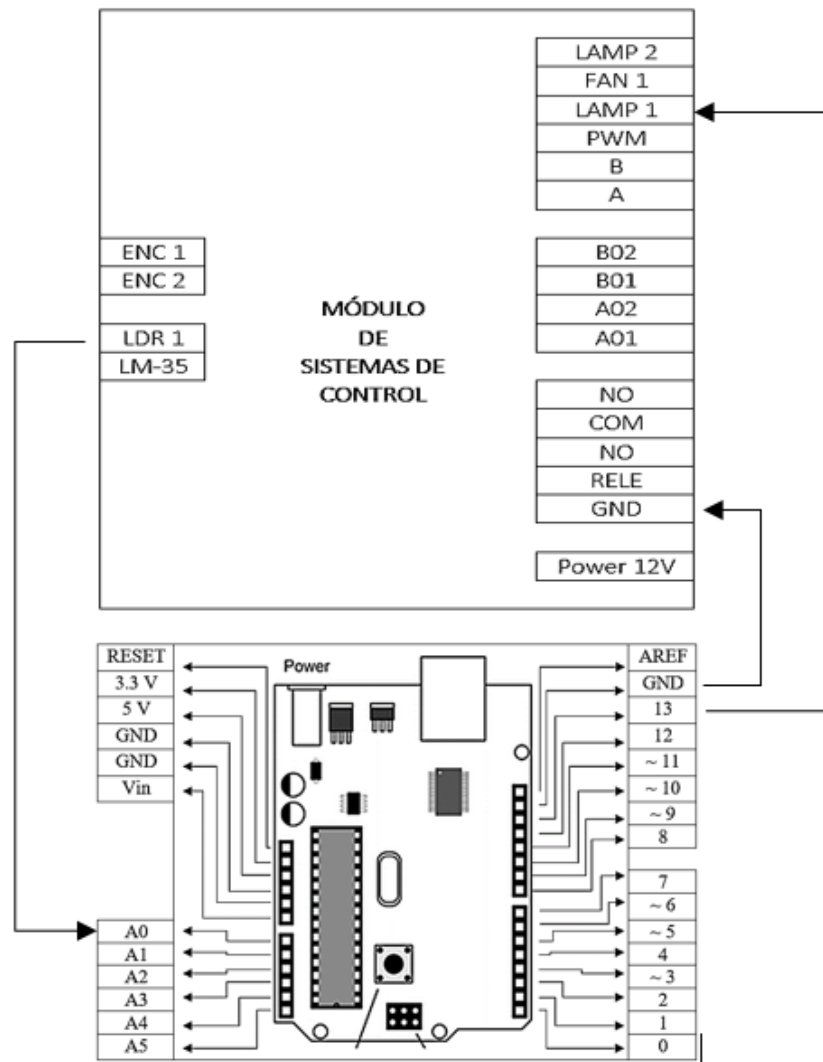
8	ANEXOS
----------	---------------



Anexo 1. Control On/Off de Intensidad Luminosa, panel de circuito



Anexo 2 Control On/Off de Intensidad Luminosa, panel Frontal.



Anexo 3 Diagrama de conexión Módulo/Arduino, Control On/Off de Intensidad Luminosa

Anexo 36	UNIVERSIDAD TÉCNICA DE COTOPAXI	
PRÁCTICA DE LABORATORIO		

CARRERA	CÓDIGO DE LA ASIGNATURA	NOMBRE DE LA ASIGNATURA
INGENIERÍA ELECTROMECAÁNICA	ILM702M2	SISTEMAS DE CONTROL

PRÁCTICA N°	LABORATORIO:	Sistemas de Control	DURACIÓN (HORAS)
03	NOMBRE DE LA PRÁCTICA:	Control PID de Temperatura	3

1	OBJETIVOS
	<ul style="list-style-type: none"> • Cargar el código del protocolo “StandardFirmata” a la tarjeta Arduino • Desarrollar la programación del control PID de temperatura en el software MyOpenLab, tanto del panel de circuito como el panel frontal • Obtener la función de transferencia y definir las constantes KP, Ki y KD del Sistema de Control. • Conectar el módulo a las salida y entradas; análogas y digitales de la tarjeta de adquisición (Arduino) • Recopilar datos del comportamiento del Sistema de Control y realizar la gráfica. • Analizar la forma de onda resultante del control.

2	FUNDAMENTACIÓN TEÓRICA
	<p>2.1. INTRODUCCIÓN:</p> <p>El Laboratorio de Sistemas de control cuenta con equipos para realizar de prácticas, a través de las cuales se pretende, complementar el estudio teórico con el práctico y crear en el estudiante habilidades para el diseño e implementación de Sistemas de control, generalmente utilizados en plantas industriales, procesos de producción, en la automatización de procesos, etc. El término control hace referencia a los métodos y maneras de gobernar el comportamiento de un aparato, máquina o proceso. En un sistema de control, hay varios parámetros de entrada que actúan sobre otros parámetros de salida, organizados en un orden lógico de control.</p>

Un regulador o control PID es un algoritmo que permite controlar un sistema en lazo cerrado, con la única finalidad de alcanzar un salida deseado que es gobernado por un Setpoint. El control PID está compuesto de tres parámetros que proporcionan una acción Proporcional, un tiempo Integral y un tiempo Derivativo, estas tres constantes son las que dan el nombre al regulador o controlador PID.

La adquisición de datos y señales consiste en la toma de muestras de sistemas análogos (mundo real) con el fin de generar datos que puede ser manipulados, modificados e interpretados por un ordenador o sistema digital. También poder definirse como: tomar un conjunto de señales físicas y convertirlas en tensión y corriente eléctrica y digitalizarlas para ser procesados por un computador. Dentro del proceso de adquisición de datos se requiere una etapa de acondicionamiento de datos y señales con el fin de adecuar la señal a niveles compatibles con el elemento digital.

2.2 EQUIPO, INSTRUMENTOS Y MATERIALES NECESARIOS

- Módulo de Sistemas de Control
- Fuente de 12 voltios.
- Destornillador plano tipo bornera
- Cables Jumper
- Computadora

2.3 SOFTWARE REQUERIDO

- MyOpenLab
- Arduino
(revisar manual de uso, para su instalación)

2.4 MEDIDAS DE SEGURIDAD

- Revisar el manual de uso del módulo de sistemas de control
- Realizar la conexión de la tarjeta Arduino con el módulo des-energizado y cuando los leds indicadores no estén encendidos

2.5 ACTIVIDADES A DESARROLLAR:

2.5.1 Cargar el código del protocolo “StandardFirmata” a la tarjeta Arduino, siguiendo el siguiente enlace: Abrir Arduino/Archivo/Ejemplos/Firmata/Estándar Firmata, como se muestra en la figura 2.1

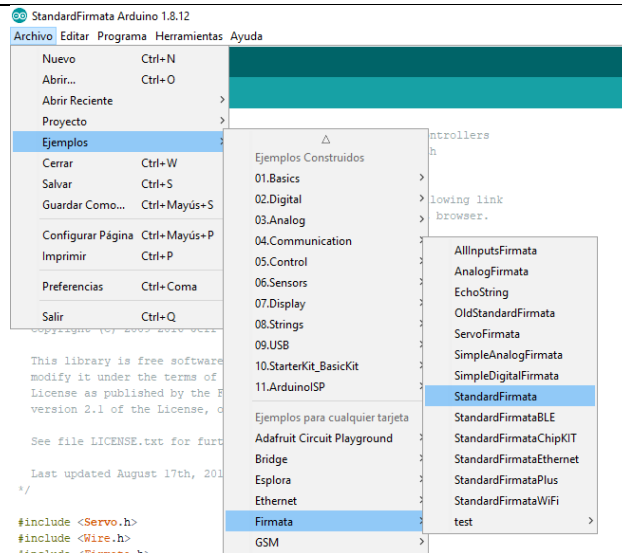


Figura 2.1. Guía para cargar el protocolo “StandardFirmata”

2.5.1.1 Se desplegará el código del protocolo StandardFirmata, sobre la pantalla, la tarjeta debe estar conectada previamente, se da clic en verificar, para revisar que no exista errores en el código y a continuación se da clic en Subir, si la ventana de comandos inferior cambia a naranja (error) revisar el puerto de comunicación de la tarjeta Arduino.

2.5.2 Abrir el programa MyOpenLab y crear un nuevo proyecto.

En la parte superior de la ventana de MyOpenLab, se debe seguir el siguiente enlace para crear un nuevo proyecto: Fichero/Nuevo Proyecto, dentro de la ventana “Nuevo Proyecto” se puede cambiar el nombre del proyecto y seleccionar la ubicación; como se muestra en la siguiente figura.

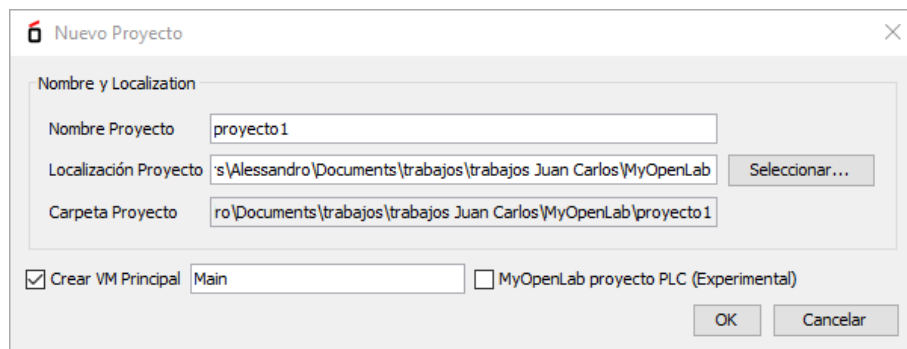


Figura 2.2 Ventana “Nuevo Proyecto”

2.5.3 Dentro del “Panel de circuito”, replicar código de programación que se muestra en el **Anexo 1**, los pines de entrada y salida del Firmata IO Interface son:

- **1 Analog_Input:** LM-35
- **3 PWM_Ouput:** LAMP 1. LAMP 2, FAN 1.

Para configurar los pines se debe dar clic sobre el “Firmata IO Interface”, aparecerá una ventana en la esquina inferior “Componentes”, al final de la ventana se debe dar clic sobre los 3 puntos

(...) en “Config Pins” a continuación, se desplegará una ventana donde se poder habilitar cada una de las entradas y salidas deseadas como se muestra en la siguiente imagen.

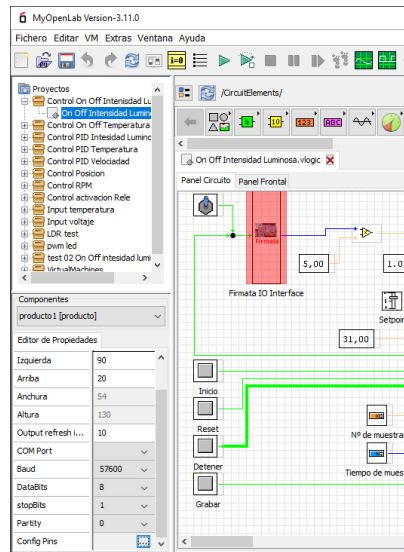


Figura 2.3 configuración de los pines de Firmata IO Interface

Antes de programar, la tarjeta Arduino debe estar conectada, para que MyOpenLab la reconozca automáticamente. La señal de entrada análoga debe ser acondicionada, para ser visualizada en grados Celsius, por lo que se multiplica 500 y se divide para 1023, a continuación, se muestra un ejemplo:

$$\text{Analog_input} = 61.38$$

$$^{\circ}\text{C} = 61.38 * \frac{500}{1023} = 30^{\circ}\text{C}$$

Para este control se utiliza el componente Regulador PID que está dentro de las librerías de MyOpenLab, este componente posee 5 pines de entrada y un pin de salida como se puede ver en la figura 2.4, donde (e) es la desviación del control o error, (ta) es la ganancia, (kp) (ki) (kd) son las contantes proporcional, integra y derivativa, y el pin de salida (y) es la señal de salida del control PID.

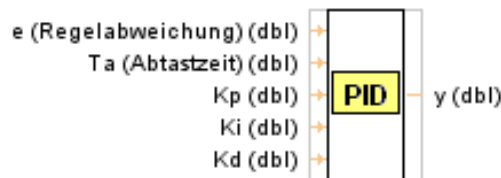


Figura 2.4 Regulador PID de MyOpenLab

Este regulador PID trabaja con datos flotantes (dbl), es necesario acondicionar esta señal de salida; pasar de un dato flotante (dbl) a un dato entero (int), para que sea conectada a los pines de salida de la tarjeta de adquisición de datos Arduino, como se muestra en el **Anexo 1** a la salida del regulador PID.

En el **Anexo 3** se visualiza la apariencia del control PID de temperatura del panel frontal

2.5.4 Para realizar la sintonización del control PID se debe crear un nuevo proyecto, como se muestra en el **Anexo 2**, es un control abierto de temperatura, esta programación permitirá recopilar datos para generar la función de transferencia y las constantes Kp, Ki y Kd. Los métodos que se puede utilizar se mencionarán en el siguiente capítulo de Metodología.

Después de programar, se recomienda guardar el archivo, antes de conectar al módulo de Sistemas de Control.

2.5.6. Se debe conectar con el módulo des-energizado, al lado izquierdo de módulo se encuentran los sensores que deben conectarse a las entradas análogas de la tarjeta Arduino, y al lado derecho se encuentran dispuestos los actuadores; en el **Anexo 4** se visualiza un ejemplo de conexión del control PID de temperatura.

- LM-35, puede ser conectado en: A0, A1, A2, A3, A4.
- LAMP 1, LAMP 2 y FAN 1, se conecta entre los pines 13 hasta el 2 con PWM (~N°).

2.6. METODOLOGÍA Y TÉCNICA EXPERIMENTAL

Para realizar la curva de transferencia se debe registrar los datos desde que el Setpoint está en 0 (int) y subir al 100 % = 255 (int) se debe registrar el comportamiento del sistema, se recomienda un tiempo de muestreo igual a 1 segundo, como se visualiza en la siguiente imagen.

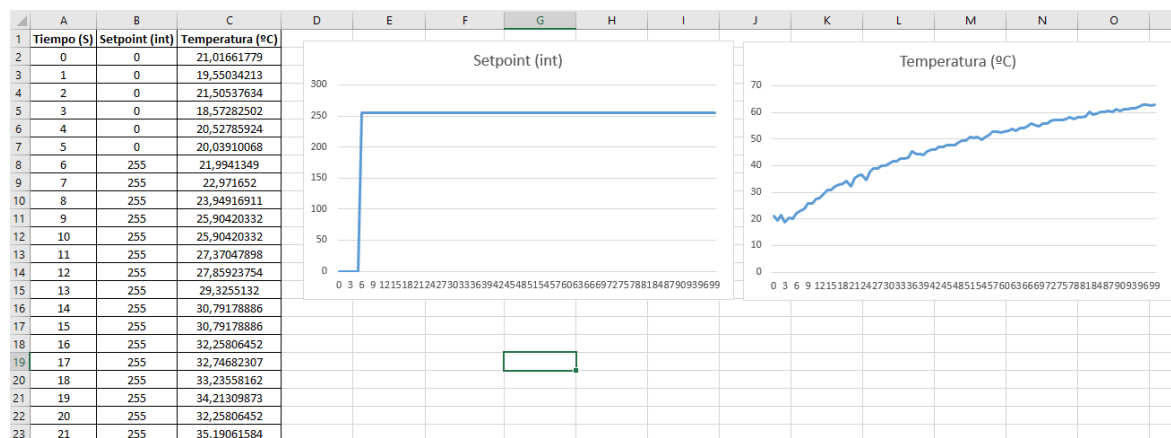


Figura 2.5. Curva de Reacción del sistema en lazo abierto

Con los datos recopilados en MyOpenLab, se crea la curva de ración en lazo abierto del sistema, se debe convertir los datos del Setpoint (int) en datos de Setpoint (°C) para obtener el comportamiento del setpoint y del sensor en la misma gráfica como se muestra en la figura 2.5.1. realizado esto se puede aplicar cualquiera de los siguientes métodos.

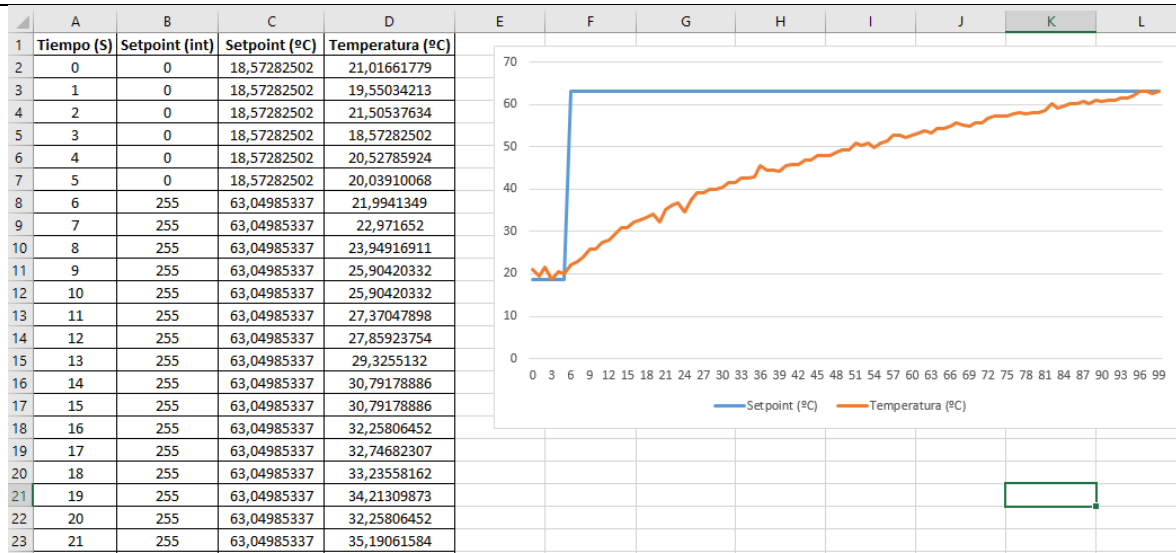


Figura 2.5.1 Curva de Reacción del sistema en lazo abierto

2.6.1. Sintonización por el método de curva de Reacción de Ziegler-Nichols

Se lleva manualmente el sistema o planta a lazo abierto, se establece un Setpoint y se registra el comportamiento de la planta, para calcular los parámetros del modelo se emplea las siguientes formula:

$$Ganancia (K) = \frac{Temperatura\ 2 - temperatura\ 1}{Setpoint\ 2 - Setpoint\ 1} \left(\frac{^{\circ}C}{int} \right)$$

Se establece un Setpoint equivalente al 28.30% y se registra la temperatura que se obtiene después de un tiempo; a continuación, se establece otro Setpoint equivalente al 63.20 % y se registra el valor de la temperatura [1], para facilitar el registro se define los Setpoint equivalentes a 28.30% y 63.20% en un tiempo de muestreo de 1 segundo, registrado estos datos se promedia, obtenido los valores equivalentes en temperatura; como se muestra en la figura 2.6. Estos valores se reemplazan en la siguiente formula:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		28,3 (%)				63,20 (%)							
2	Tiempo (s)	Setpoint (int)	Temperatura (°C)		Tiempo (s)	Setpoint (int)	Temperatura (°C)			Setpoint (int)	Sensor (°C)		
3	0	72,42	25,41544477		0	161,16	39,10068426		Promedio 1	72,42	27,06744868		
4	1	72,42	24,92668622		1	161,16	39,10068426		Promedio 2	161,16	41,24144673		
5	2	72,42	25,41544477		2	161,16	39,58944282						
6	3	72,42	25,90420332		3	161,16	40,07820137		Ganancia	0,159725017			
7	4	72,42	25,41544477		4	161,16	38,61192571						
8	5	72,42	25,41544477		5	161,16	39,58944282						
9	6	72,42	25,41544477		6	161,16	40,07820137						
10	7	72,42	25,90420332		7	161,16	39,58944282						
11	8	72,42	25,90420332		8	161,16	39,10068426						
12	9	72,42	26,39296188		9	161,16	37,14565005						
13	10	72,42	25,41544477		10	161,16	40,56695992						
14	11	72,42	25,41544477		11	161,16	39,58944282						
15	12	72,42	25,90420332		12	161,16	41,05571848						
16	13	72,42	26,39296188		13	161,16	40,07820137						
17	14	72,42	26,39296188		14	161,16	40,07820137						
18	15	72,42	25,90420332		15	161,16	39,58944282						
19	16	72,42	26,39296188		16	161,16	40,07820137						
20	17	72,42	25,90420332		17	161,16	40,56695992						
21	18	72,42	26,39296188		18	161,16	40,07820137						
22	19	72,42	26,39296188		19	161,16	39,10068426						
23	20	72,42	26,39296188		20	161,16	40,07820137						

$$Ganancia (K) = \frac{Temperatura\ 2 - temperatura\ 1}{Setpoint\ 2 - Setpoint\ 1} \left(\frac{^{\circ}C}{int} \right)$$

Figura 2.6 Cálculo de la ganancia del sistema

En la curva de reacción del sistema en lazo abierto se traza un tangente en el punto de inflexión de la curva y obtiene el tiempo de retador del sistema (L) y la constante del tiempo (T) como se muestra en la figura 2.7.

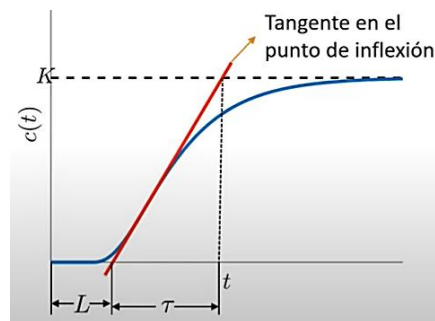


Figura 2.7 (L) y (T) en la curva de Reacción del sistema

Fuente [1]

Calculado la Ganancia del sistema, el tiempo de retador del sistema y la constante del tiempo se establece la función de transferencia equivalente del sistema de control.

$$G(S) = \frac{K * e^{-Ls}}{Ts + 1}$$

Tabla 2.1 Método Ziegler y Nichols

Control	Kp	Ti	Td
P	$T/(K*L)$	n/a	n/a
PI	$0.9*T/(K*L)$	$L/ (0.3)$	n/a
PID	$1.2*T/(K*L)$	$2L$	$0.5L$

Fuente: [3]

Los valores obtenidos del método de lazo abierto de Ziegler y Nichols esta expresados el modelo estándar de control PID, MyOpenLab emplea el modelo paralelo, para esto se emplea las siguientes fórmulas, para transformar esto valores:

$$Kp = Kp$$

$$Ki = Kp / Ti$$

$$Kd = Kp * Td$$

2.6.2. Sintonización utilizando PID Tuner, herramienta de Matlab

Para logra la sintonización del control con esta herramienta se debe seguir los siguientes pasos:

1. Se deben cargar los datos de Temperatura (°C) y del Setpoint (int) en la ventana “Workspace”, Para carga los datos, se dan clic derecho sobre la ventana, se desplegará una ventana donde se selecciona “NEW”, a continuación, se define un nombre “Setpoint” y otra con el nombre “Temperatura”, creada estas dos variables se da doble clic en cualquiera de ellas, se desplegará

una hoja de cálculo (paseada a Excel), en se debe pegar los datos obtenidos de MyOpenLab del Setpoint y de la Temperatura.

- Una vez cargado los datos, se emplea el comando (`>>ident`) en la ventana “Command Window” se presiona la tecla “Entre”, se desplegará una ventana con el nombre “System identification” en donde se importarán los datos del sistema de control en función del tiempo, como se puede ver en la figura 2.8.

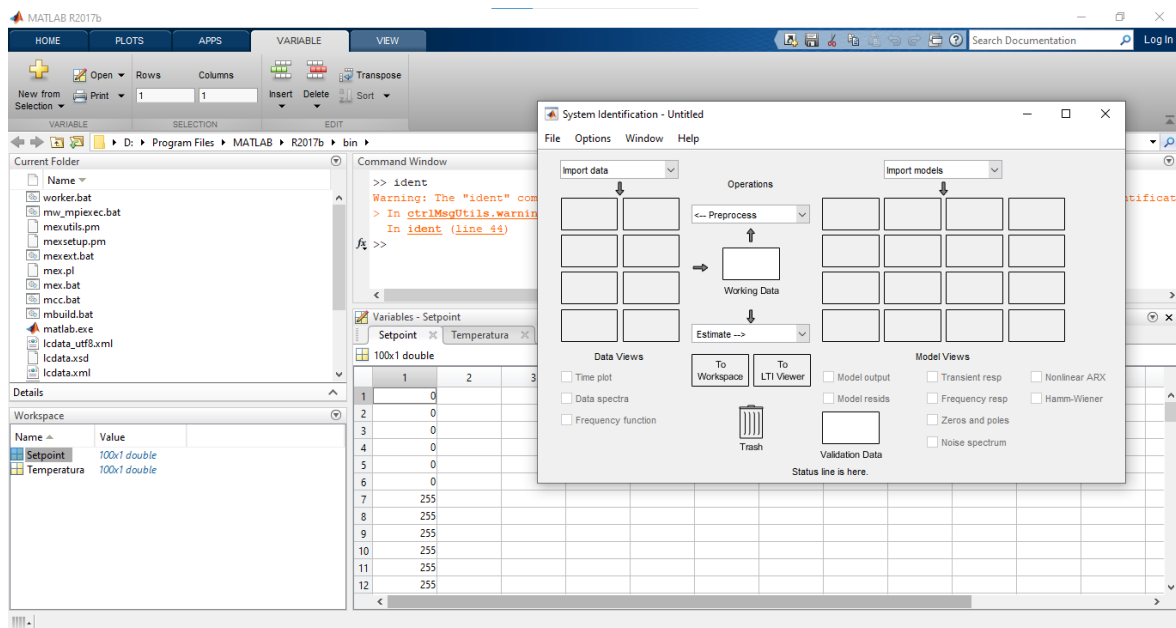


Figura 2.8 Vista de datos de Setpoint y Temperatura en el Workspace

- Dentro de la ventana “System Identification” se da clic en “Import data” y se selecciona “Time Dimain Data”, se desplegará una ventana con el nombre “Import Data”, se debe llenar con los siguientes datos:

Workspace Variable

- Input:** “Setpoint”
- Output:** “Temperatura”

Se debe ingresar el nombre de las variables ingresadas en la ventana “Workspace”.

Nota: Revisar las mayúsculas y minúsculas

Data Information

- Starting time: “0” (es en tiempo desde donde inicia el registro de datos, MyOpenLab registra los datos desde $t = 0s$)
- Sample time: “1” (Tiempo entre Muestras)

En la figura 2.9 se puede visualizar la ventana “Import Data” a continuación, se da clic en “Import”, obtenido así la función de transferencia de lazo abierto del sistema.

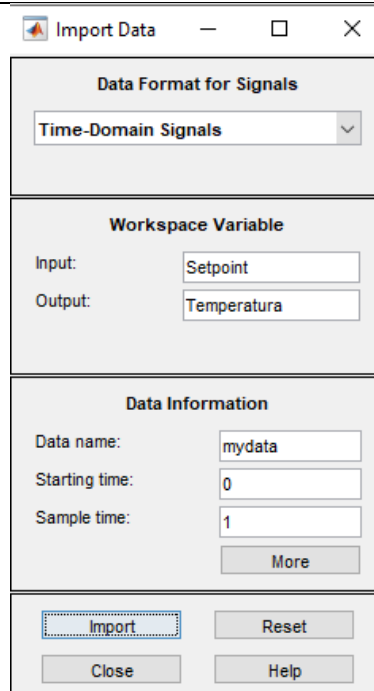


Figura 2.9 Ventana “Import Data”

Nota: Para visualizar las funciones de transferencia del Setpoint y Temperatura, se debe dar clic “Time plot” en la ventana “System Identification”.

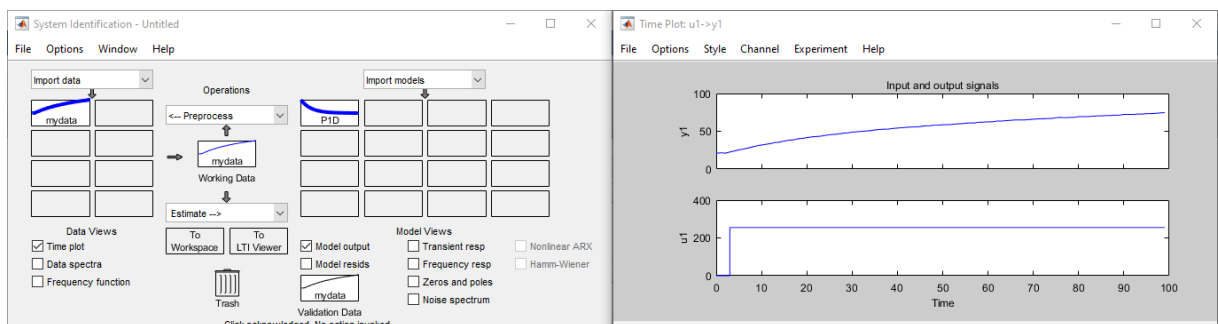


Figura 2.10 Ventana System Identification y gráfica de datos resultante

- En “Estimate -->” se selecciona “Process Model”, se desplegará la ventana “Process Model” al final de la venta se debe dar clic en estimar, para obtener las constantes K, Ti y Td, de este paso se obtiene el “Model output” final, como se ve en la figura 2.11

Nota: para visualizar el modelo de salida se debe dar clic en “Model output”

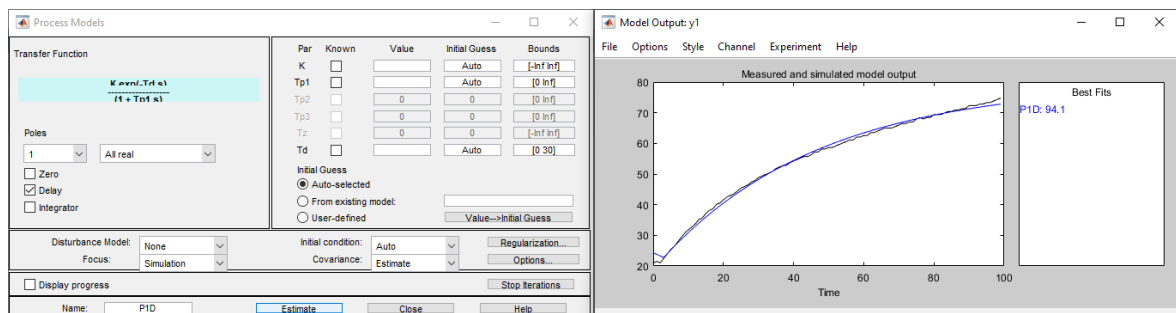


Figura 2.11 Ventana Process Model y gráfica del modelo PID de salida

Se selecciona la ventana del “PID” y se arrastra hasta la opción “To workspace” en la ventana “System Identification”.

- Transferido el (PID) a “To workspace” se cierra la ventana “System Identification” (no es necesario guardar nada). En la ventana principal de Matlab en “Command Window” se escribe el siguiente comando “ $Control = tf(PID)$ ” y se obtiene la función de transferencia del control como se ve en la figura 2.12.

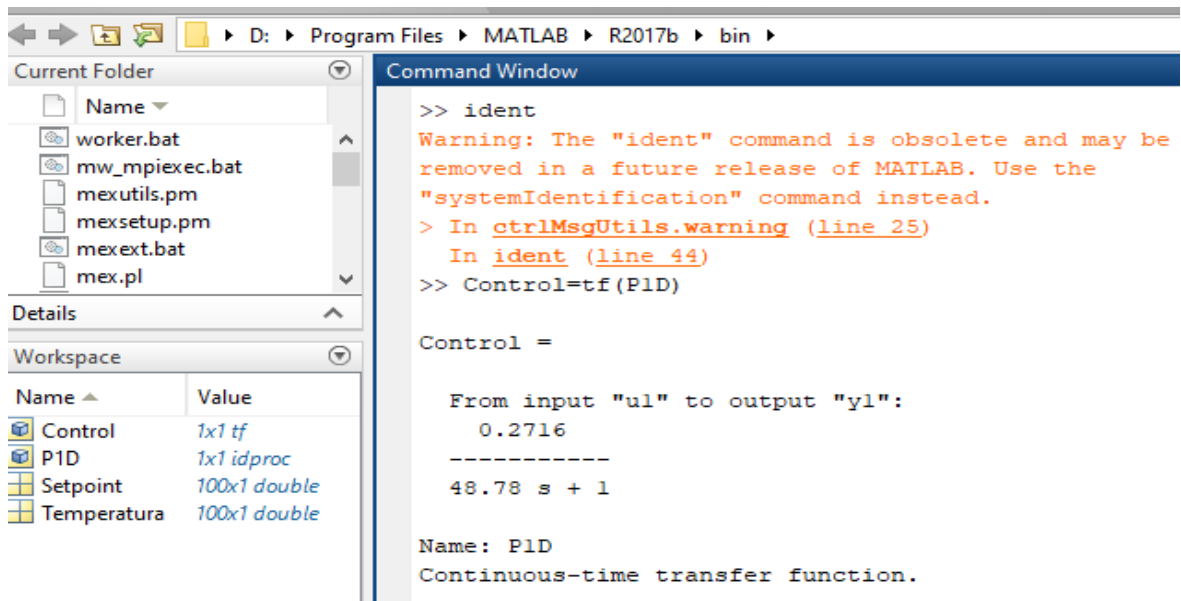


Figura 2.12 función de transferencia obtenida del Matlab

- En la parte superior de Matlab se selecciona “APPS”, y dentro de este se elige la herramienta “PID Tuner”, a continuación, se desplegará una ventana “PID Tune”, en esta ventana en la parte superior izquierda en “Plant” se puede importar el modelo del control, como se ve en la Figura 2.13.

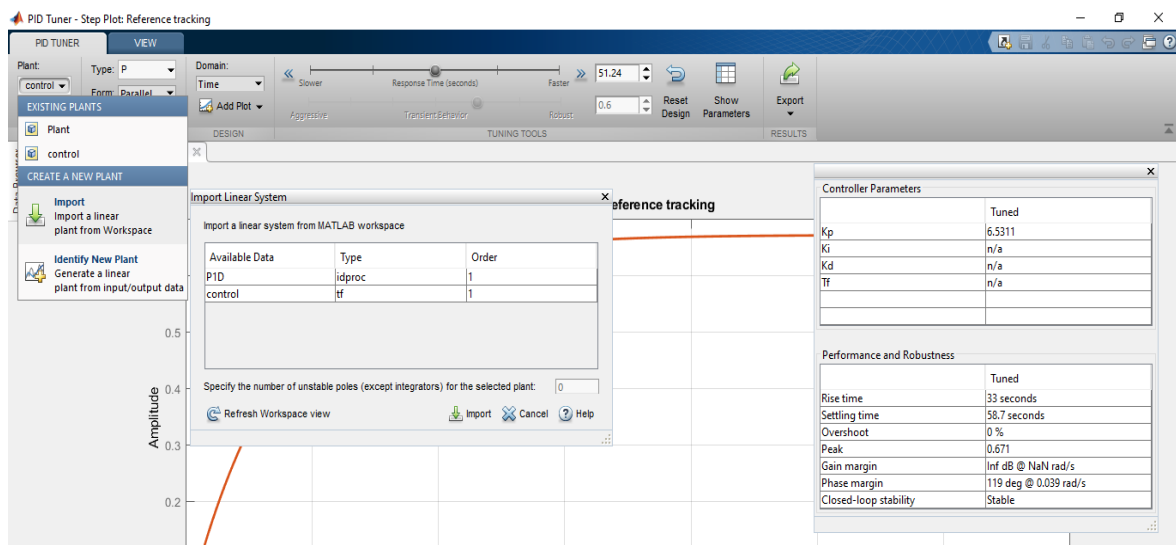


Figura 2.13 Ventana PID Tuner

En PID Tuner tenemos las opciones de elegir el tipo de proceso (P, PI, PD o PID) y elegir entre el modelo estándar (K , T_i , T_d) y el modelo paralelo (K_p , K_i , y K_d).

2.6.3. Sintonización con método alternativo a PID Tuner, herramienta de Matlab

Una alternativa de uso gratuito y en líneas es Tune your PID (<https://pidtuner.com/>), los pasos son similares a Matlab, se deber cargar los datos del Setpoint y Sensor en función del tiempo y se obtienen las gráficas de Relación de la curva.

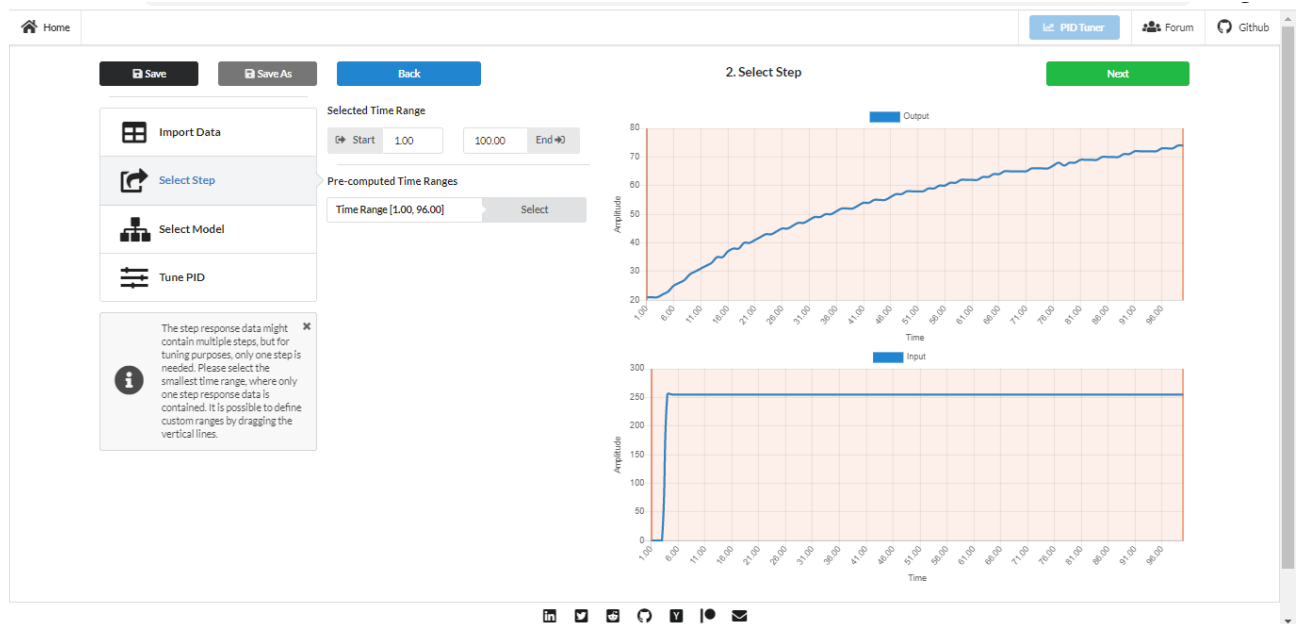


Figura 2.14 Ventana Tune your PID

Esta página también nos entrega la función de transferencia de proceso o control, como se ve en la figura 2.12

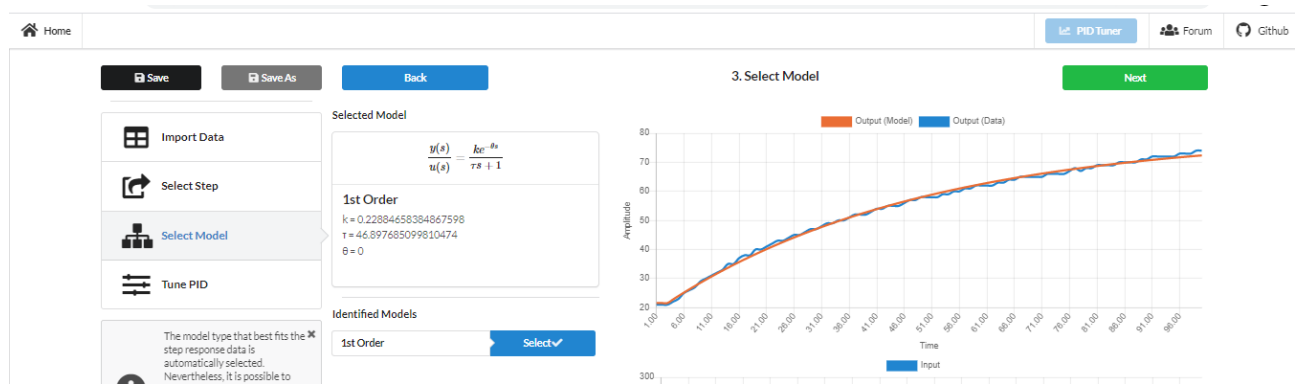


Figura 2.15 Valores de la función de transferencia en Tune your PID

Las constantes que entre Tune your PID, se encuentra en función del modelo estándar que son: (K , T_i y T_d). MyOpenLab emplea el modelo paralelo, para esto se emplea las siguientes fórmulas, para transformar esto valores:

$$K_p = K_p$$

$$K_i = K_p / T_i$$

$$K_d = K_p * T_d$$

INFORME DE PRÁCTICA

3 RESULTADOS Y DISCUSIÓN

Para realizar el análisis del comportamiento del sistema de control, se debe establecer un Setpoint diferente a la temperatura actual y se empieza a registrar los datos, el “Tiempo de Muestreo” es el tiempo entre muestra tomada. En la figura 3.1 se visualiza la gráfica de los datos tomados, con el siguiente comportamiento:

- Temperatura inicial = 28.83 °C
- Setpoint = 40.48 °C
- Tiempo de muestro = 1000 milisegundo.

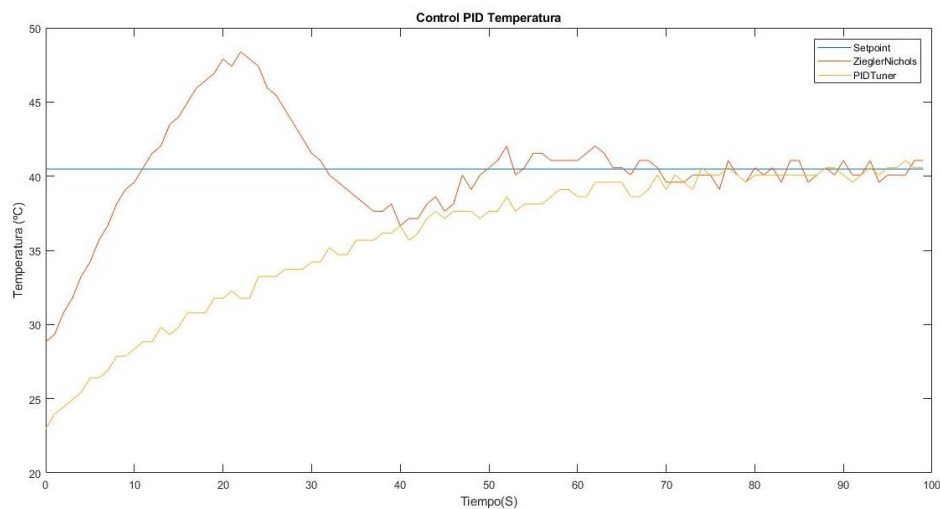


Figura 3.1 Comportamiento del PID de temperatura con un Setpoint de 40.48 °C

3.3 ANÁLISIS DE RESULTADOS:

-

5 CONCLUSIONES

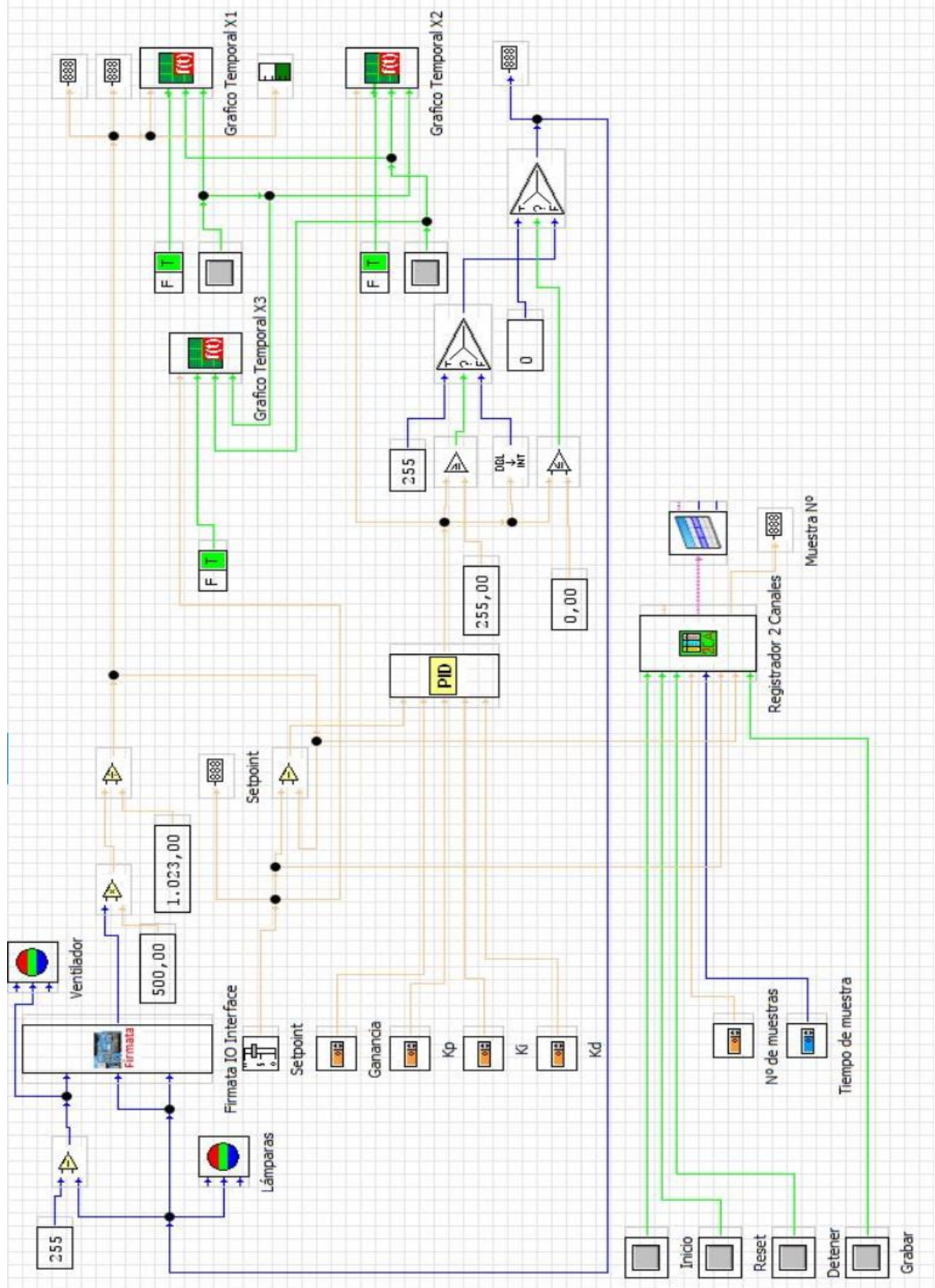
-

6 RECOMENDACIONES

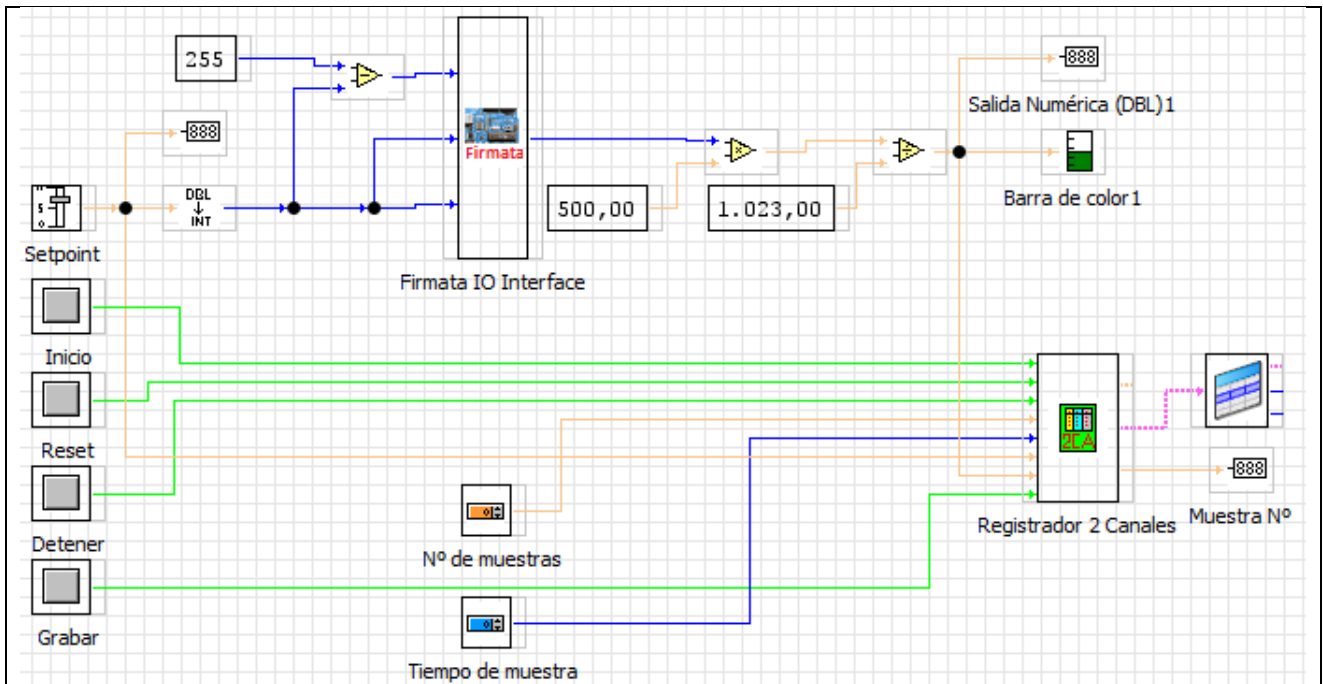
- Para graficar la señal del Sensor y del Setpoint en la misma gráfica como se muestra en la figura 2.3 del panel frontal, se emplea dos elementos “Gráfico Temporal” X1 y X2 en el panel frontal, los dos gráficos deben tener las mismas unidades y dimensiones y no tener fondo (activar Background Transparent en las propiedades de la gráfica), para interpolar una gráfica sobre la otra.

7 REFERENCIAS BIBLIOGRÁFICAS (Utilizar las normas IEEE)

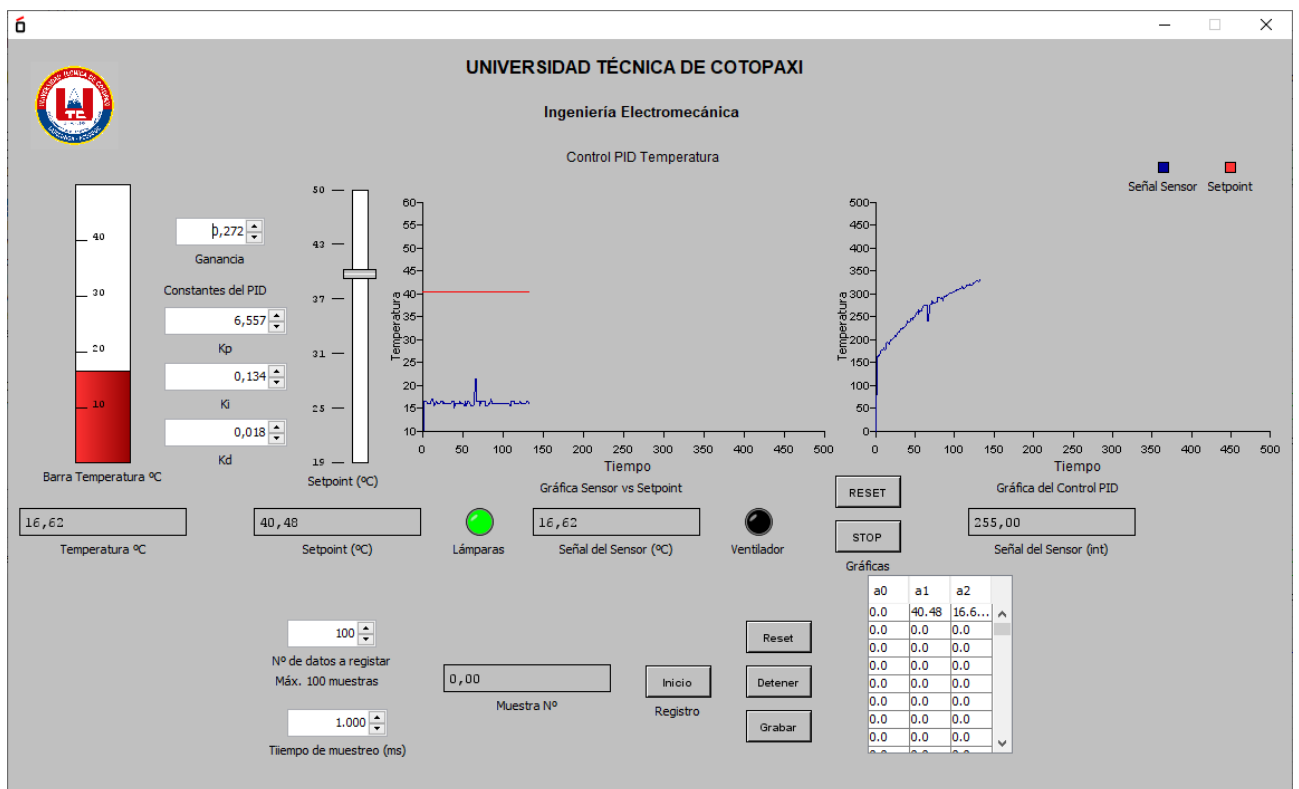
-



Anexo 1. Control PID de temperatura, panel de circuito



Anexo 2 Diagrama de lazo abierto del Control PID de Temperatura



Anexo 3 Control PID de temperatura, panel Frontal.

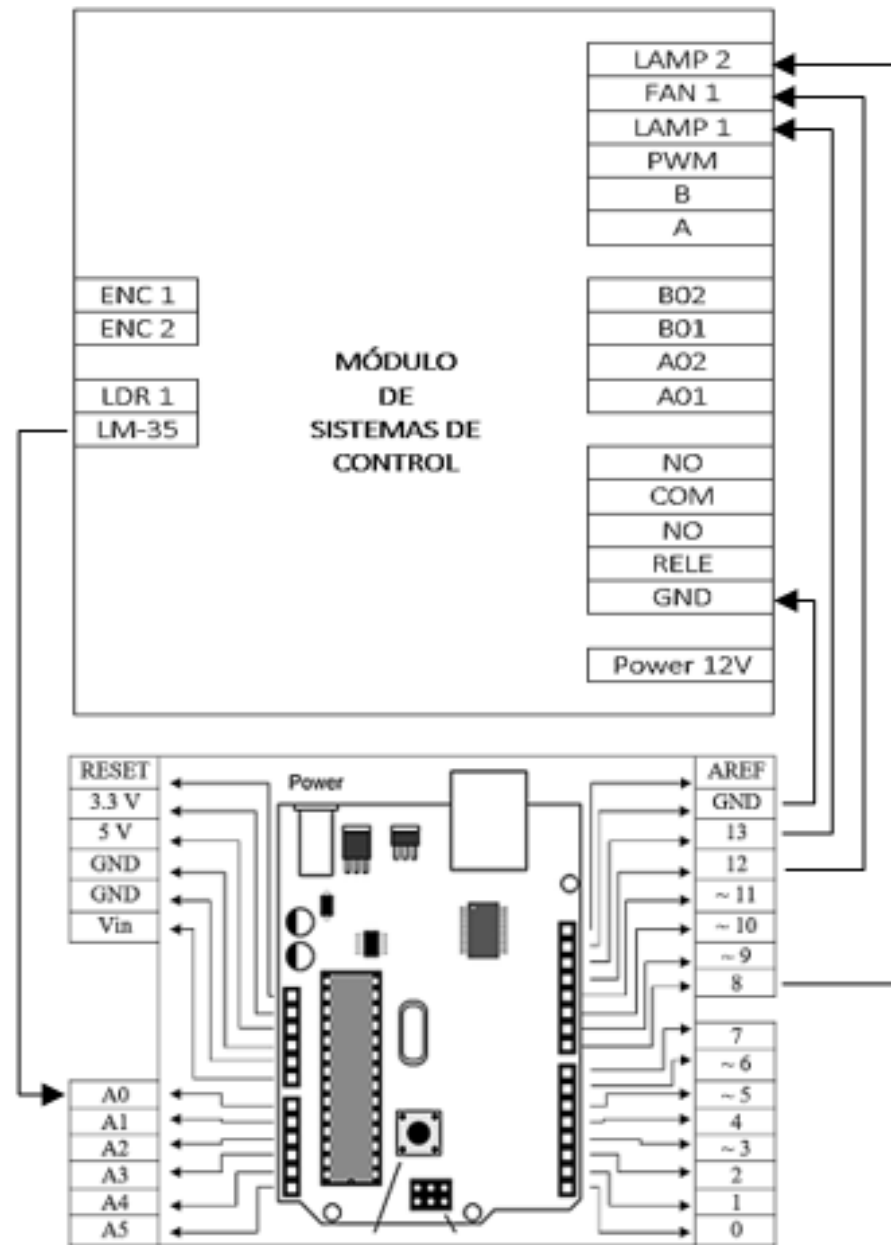


Figura 4 Diagrama de conexión Módulo/Arduino, Control PID de temperatura

Anexo 37	UNIVERSIDAD TÉCNICA DE COTOPAXI	
PRÁCTICA DE LABORATORIO		

CARRERA	CÓDIGO DE LA ASIGNATURA	NOMBRE DE LA ASIGNATURA
INGENIERÍA ELECTROMECAÁNICA	ILM702M2	SISTEMAS DE CONTROL

PRÁCTICA N°	LABORATORIO:	Sistemas de Control	DURACIÓN (HORAS)
03	NOMBRE DE LA PRÁCTICA:	Control PID de Intensidad Luminosa	3

1	OBJETIVOS
	<ul style="list-style-type: none"> • Cargar el código del protocolo “StandardFirmata” a la tarjeta Arduino • Desarrollar la programación del control PID de Intensidad Luminosa en el software MyOpenLab, tanto del panel de circuito como el panel frontal • Obtener la función de transferencia y definir las constantes KP, Ki y KD del Sistema de Control. • Conectar el módulo a las salida y entradas; análogas y digitales de la tarjeta de adquisición (Arduino) • Recopilar datos del comportamiento del Sistema de Control y realizar la gráfica. • Analizar la forma de onda resultante del control.

2	FUNDAMENTACIÓN TEÓRICA
	<p>2.1. INTRODUCCIÓN:</p> <p>El Laboratorio de Sistemas de control cuenta con equipos para realizar de prácticas, a través de las cuales se pretende, complementar el estudio teórico con el práctico y crear en el estudiante habilidades para el diseño e implementación de Sistemas de control, generalmente utilizados en plantas industriales, procesos de producción, en la automatización de procesos, etc. El término control hace referencia a los métodos y maneras de gobernar el comportamiento de un aparato, máquina o proceso. En un sistema de control, hay varios parámetros de entrada que actúan sobre otros parámetros de salida, organizados en un orden lógico de control.</p>

Un regulador o control PID es un algoritmo que permite controlar un sistema en lazo cerrado, con la única finalidad de alcanzar un salida deseado que es gobernado por un Setpoint. El control PID está compuesto de tres parámetros que proporcionan una acción Proporcional, un tiempo Integral y un tiempo Derivativo, estas tres constantes son las que dan el nombre al regulador o controlador PID.

La adquisición de datos y señales consiste en la toma de muestras de sistemas análogos (mundo real) con el fin de generar datos que puede ser manipulados, modificados e interpretados por un ordenador o sistema digital. También poder definirse como: tomar un conjunto de señales físicas y convertirlas en tensión y corriente eléctrica y digitalizarlas para ser procesados por un computador. Dentro del proceso de adquisición de datos se requiere una etapa de acondicionamiento de datos y señales con el fin de adecuar la señal a niveles compatibles con el elemento digital.

2.2 EQUIPO, INSTRUMENTOS Y MATERIALES NECESARIOS

Módulo de Sistemas de Control

Fuente de 12 voltios.

Destornillador plano tipo bornera

Cables Jumper

Computadora

2.3 SOFTWARE REQUERIDO

MyOpenLab

Arduino

(revisar manual de uso, para su instalación)

2.4 MEDIDAS DE SEGURIDAD

Revisar el manual de uso del módulo de sistemas de control

Realizar la conexión de la tarjeta Arduino con el módulo des-energizado y cuando los leds indicadores no estén encendidos

2.5 ACTIVIDADES A DESARROLLAR:

2.5.1 Cargar el código del protocolo “StandardFirmata” a la tarjeta Arduino, siguiendo el siguiente enlace: [Abrir Arduino/Archivo/Ejemplos/Firmata/Estándar Firmata](#), como se muestra en la figura 2.1

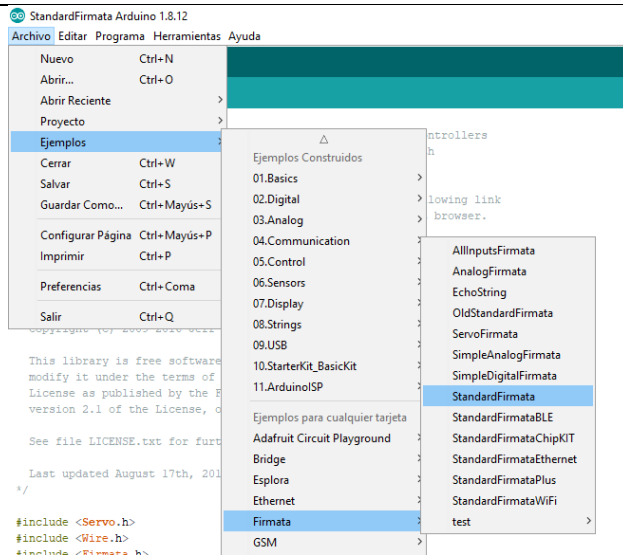


Figura 2.1 Guía para cargar el protocolo “StandardFirmata”

2.5.1.1 Se desplegará el código del protocolo StandardFirmata, sobre la pantalla, la tarjeta debe estar conectada previamente, se da clic en verificar, para revisar que no exista errores en el código y a continuación se da clic en Subir, si la ventana de comandos inferior cambia a naranja (error) revisar el puerto de comunicación de la tarjeta Arduino.

2.5.1 Abrir el programa MyOpenLab y crear un nuevo proyecto.

En la parte superior de la ventana de MyOpenLab, se debe seguir el siguiente enlace para crear un nuevo proyecto: Fichero/Nuevo Proyecto, dentro de la ventana “Nuevo Proyecto” se puede cambiar el nombre del proyecto y seleccionar la ubicación; como se muestra en la siguiente figura.

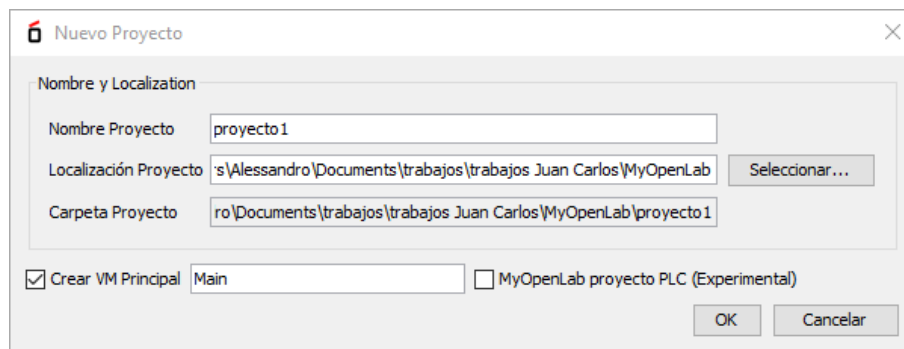


Figura 2.2 Ventana “Nuevo Proyecto”

2.5.2 Dentro del “Panel de circuito”, replicar código de programación que se muestra en el **Anexo 1**, los pines de entrada y salida del Firmata IO Interface son:

- **1 Analog_Input:** LDR 1
- **1 PWM_Ouput:** LAMP 1.

Para configurar los pines se debe dar clic sobre el “Firmata IO Interface”, aparecerá una ventana en la esquina inferior “Componentes”, al final de la ventana se debe dar clic sobre los 3 puntos

(...) en “Config Pins” a continuación, se desplegará una ventana donde se poder habilitar cada una de las entradas y salidas deseadas como se muestra en la siguiente imagen.

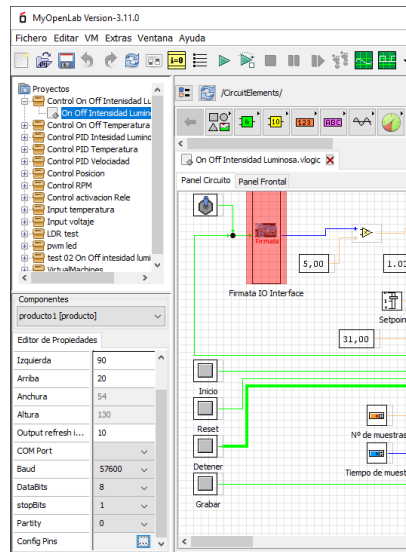


Figura 2.3 configuración de los pines de Firmata IO Interface

Antes de programar, la tarjeta Arduino debe estar conectada, para que MyOpenLab la reconozca automáticamente. La señal de entrada análoga debe ser acondicionada, para ser visualizada en voltios (V), por lo que se multiplica 5 y se divide para 1023, a continuación, se muestra un ejemplo:

$$\text{Analog_input} = 763.158 \approx \text{Lámpara encendida}$$

$$V = 763.158 * \frac{5}{1023} = 3.73 \text{ voltios (V)}$$

Para trabajar con un Setpoint en porcentajes se debe dividir 100% para el voltaje de entrada cuando la lámpara incandescente este encendido:

$$\text{Setpoint (\%)} = \frac{100}{V} = \frac{100}{3.7119} = 26.94$$

Para este control se utiliza el componente Regulador PID que está dentro de las librerías de MyOpenLab, este componente posee 5 pines de entrada y un pin de salida como se puede ver en la figura 2.3, donde (e) es la desviación del control o error, (ta) es la ganancia, (kp) (ki) (kd) son las constantes proporcional, integra y derivativa, y el pin de salida (y) es la señal de salida del control PID.

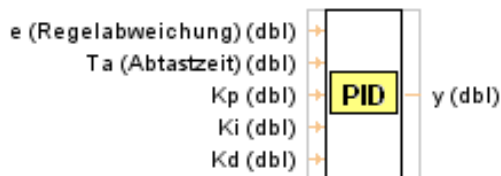


Figura 2.4 Regulador PID de MyOpenLab

Este regulador PID trabaja con datos flotantes (dbl), es necesario acondicionar esta señal de salida; pasar de un dato flotante (dbl) a un dato entero (int), para que sea conectada a los pines

de salida de la tarjeta de adquisición de datos Arduino, como se muestra en el **Anexo 1** a la salida del regulador PID.

En el **Anexo 3** se visualiza la apariencia del control PID de Intensidad luminosa del panel frontal

2.5.3 Para realizar la sintonización del control PID se debe crear un nuevo proyecto, como se muestra en el anexo 2, es un control abierto de Intensidad Luminosa, esta programación permitirá recopilar datos para generar la función de transferencia y las constantes K_p , K_i y K_d . Los métodos que se puede utilizar se mencionarán en el siguiente capítulo de Metodología. Después de programar, se recomienda guardar el archivo, antes de conectar al módulo de Sistemas de Control.

2.5.4 Se debe conectar con el módulo des-energizado, al lado izquierdo de módulo se encuentran los sensores que deben conectarse a las entradas análogas de la tarjeta Arduino, y al lado derecho se encuentran dispuestos los actuadores; en el **Anexo 4** se visualiza un ejemplo de conexión del control PID de Intensidad Luminosa.

- LDR 1, puede ser conectado en: A0, A1, A2, A3, A4.
- LAMP 1, se conecta entre los pines 13 hasta el 2 con PWM ($\sim N^\circ$).

2.6 METODOLOGÍA Y TÉCNICA EXPERIMENTAL

Para realizar la curva de transferencia se debe registrar los datos desde que el Setpoint está en 0 (int) y subir al 100 % = 255 (int) se debe registrar el comportamiento del sistema, se recomienda un tiempo de muestreo igual a 50 (ms) mili-segundo, como se visualiza en la siguiente imagen.

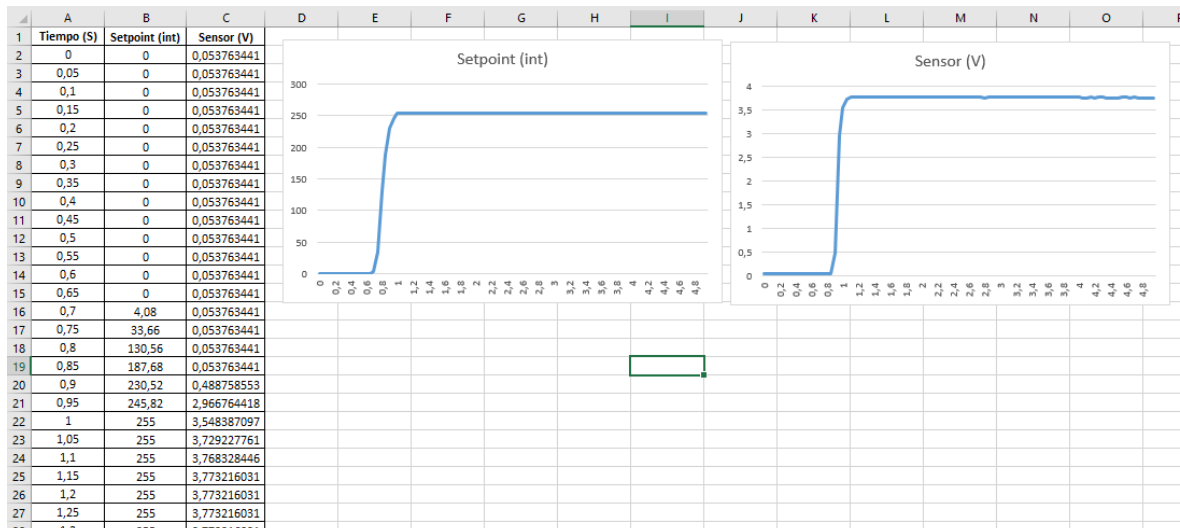


Figura 2.5 Curva de Reacción del sistema en lazo abierto

Con los datos recopilados en MyOpenLab, se crea la curva de ración en lazo abierto del sistema, se debe convertir los datos del Setpoint (int) en datos de Setpoint (V) para obtener el comportamiento del setpoint y del sensor en la misma gráfica como se muestra en la figura 2.5.1. realizado esto se puede aplicar cualquiera de los siguientes métodos.

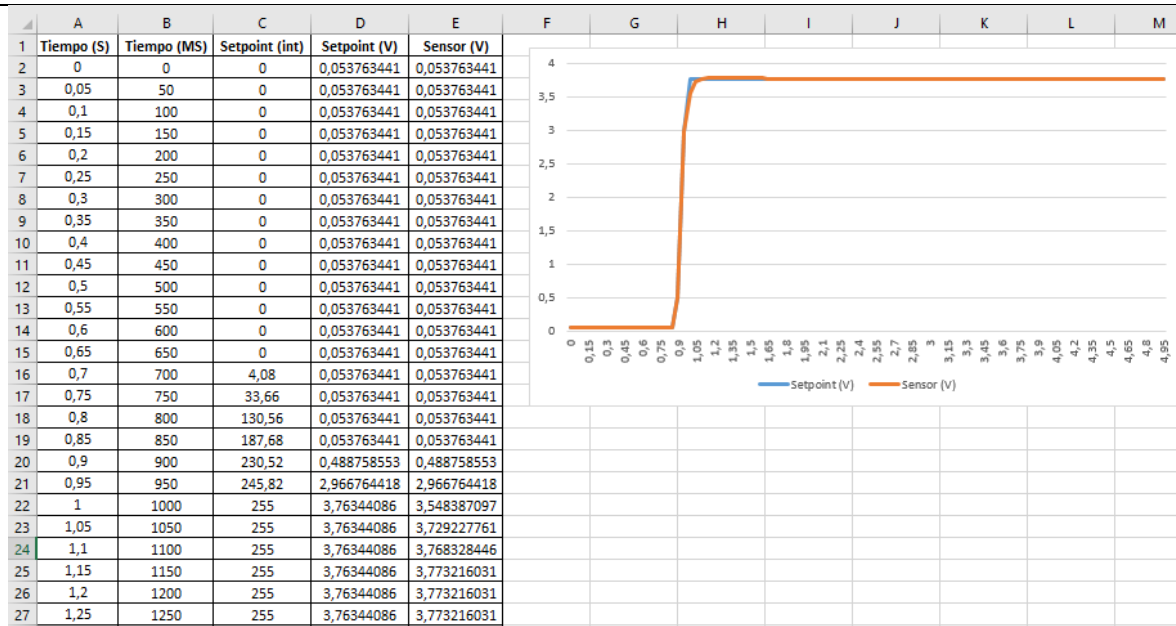


Figura 2.5.1 Curva de Reacción del sistema en lazo abierto

2.6.1 Sintonización por el método de curva de Reacción de Ziegler-Nichols

Se lleva manualmente el sistema o planta a lazo abierto, se establece un Setpoint y se registra el comportamiento de la planta, para calcular los parámetros del modelo se emplea las siguientes formula:

$$Ganancia (K) = \frac{Intensidad\ luminosa\ 2 - Intensidad\ luminosa\ 1}{Setpoint\ 2 - Setpoint\ 1} \left(\frac{V}{int} \right)$$

Se establece un Setpoint equivalente al 28.30% y se registra la Intensidad Luminosa que se obtiene después de un tiempo; a continuación, se establece otro Setpoint equivalente al 63.20 % y se registra el valor de la Intensidad luminosa, para facilitar el registro se define los Setpoint equivalentes a 28.30% y 63.20% en un tiempo de muestreo de 50 mili-segundo, registrado estos datos se promedia, obtenido los valores equivalentes en Intensidad Luminosa; como se muestra en la figura 2.6. Estos valores se reemplazan en la siguiente formula:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Tiempo (ms)	28,3 (%)			Tiempo (S)	63,20 (%)								
2		Setpoint (int)	I. Luminosa (V)			Setpoint (int)	I. Luminosa (V)			Setpoint (int)	I. Luminosa (V)			
3	0	72,42	1,068817204		0	161,16	2,378494624		Promedio 1	72,42	1,068817204			
4	0,05	72,42	1,068817204		0,05	161,16	2,378494624		Promedio 2	161,16	2,378494624			
5	0,1	72,42	1,068817204		0,1	161,16	2,378494624							
6	0,15	72,42	1,068817204		0,15	161,16	2,378494624							
7	0,2	72,42	1,068817204		0,2	161,16	2,378494624		Ganancia	0,014758592				
8	0,25	72,42	1,068817204		0,25	161,16	2,378494624							
9	0,3	72,42	1,068817204		0,3	161,16	2,378494624							
10	0,35	72,42	1,068817204		0,35	161,16	2,378494624							
11	0,4	72,42	1,068817204		0,4	161,16	2,378494624							
12	0,45	72,42	1,068817204		0,45	161,16	2,378494624							
13	0,5	72,42	1,068817204		0,5	161,16	2,378494624							
14	0,55	72,42	1,068817204		0,55	161,16	2,378494624							
15	0,6	72,42	1,068817204		0,6	161,16	2,378494624							
16	0,65	72,42	1,068817204		0,65	161,16	2,378494624							
17	0,7	72,42	1,068817204		0,7	161,16	2,378494624							
18	0,75	72,42	1,068817204		0,75	161,16	2,378494624							
19	0,8	72,42	1,068817204		0,8	161,16	2,378494624							
20	0,85	72,42	1,068817204		0,85	161,16	2,378494624							
21	0,9	72,42	1,068817204		0,9	161,16	2,378494624							
22	0,95	72,42	1,068817204		0,95	161,16	2,378494624							
23	1	72,42	1,068817204		1	161,16	2,378494624							

$$Ganancia (K) = \frac{Intensidad\ luminosa\ 2 - Intensidad\ luminosa\ 1}{Setpoint\ 2 - Setpoint\ 1} \left(\frac{V}{int} \right)$$

Figura 2.6 Cálculo de la ganancia del sistema

En la curva de reacción se traza un tangente en el punto de inflexión de la curva y obtiene el tiempo de retador del sistema (L) y la constante del tiempo (T) como se muestra en la figura 2.7.

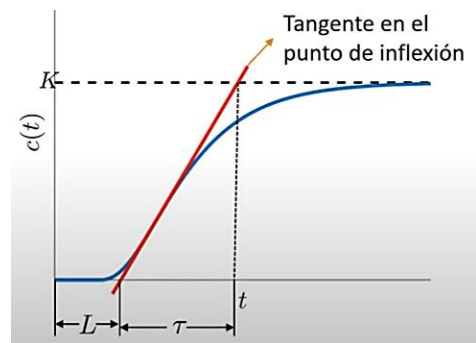


Figura 2.7. (L) y (T) en la curva de Reacción del sistema

Fuente [1]

Calculado la Ganancia del sistema, el tiempo de retador del sistema y la constante del tiempo se establece la función de transferencia equivalente del sistema de control.

$$G(S) = \frac{K * e^{-Ls}}{Ts + 1}$$

Tabla 2.1 Método Ziegler y Nichols

Control	Kp	Ti	Td
P	$T/(K*L)$	n/a	n/a
PI	$0.9*T/(K*L)$	$L/ (0.3)$	n/a
PID	$1.2*T/(K*L)$	$2L$	$0.5L$

Fuente: [3]

Los valores obtenidos del método de lazo abierto de Ziegler y Nichols están expresados en el modelo estándar de control PID, MyOpenLab emplea el modelo paralelo, para esto se emplean las siguientes fórmulas, para transformar estos valores:

$$Kp = Kp$$

$$Ki = Kp / Ti$$

$$Kd = Kp * Td$$

2.6.2 Sintonización utilizando PID Tuner, herramienta de Matlab

Para lograr la sintonización del control con esta herramienta se debe seguir los siguientes pasos:

1. Se deben cargar los datos de Intensidad Luminosa (V) y del Setpoint (int) en la ventana "Workspace".

Para cargar los datos, se hace clic derecho sobre la ventana, se desplegará una ventana donde se selecciona "NEW", a continuación, se define un nombre "Setpoint" y otra con el nombre

“Intensidad”, creada estas dos variables se da doble clic en cualquiera de ellas, se desplegará una hoja de cálculo (padecida a Excel), en se debe pegar los datos obtenidos de MyOpenLab del Setpoint e Intensidad.

- Una vez cargado los datos, se emplea el comando (`>>ident`) en la ventana “Command Window” se presiona la tecla “Entre”, se desplegará una ventana con el nombre “System identification” en donde se importarán los datos del sistema de control en función del tiempo, como se puede ver en la figura 2.8.

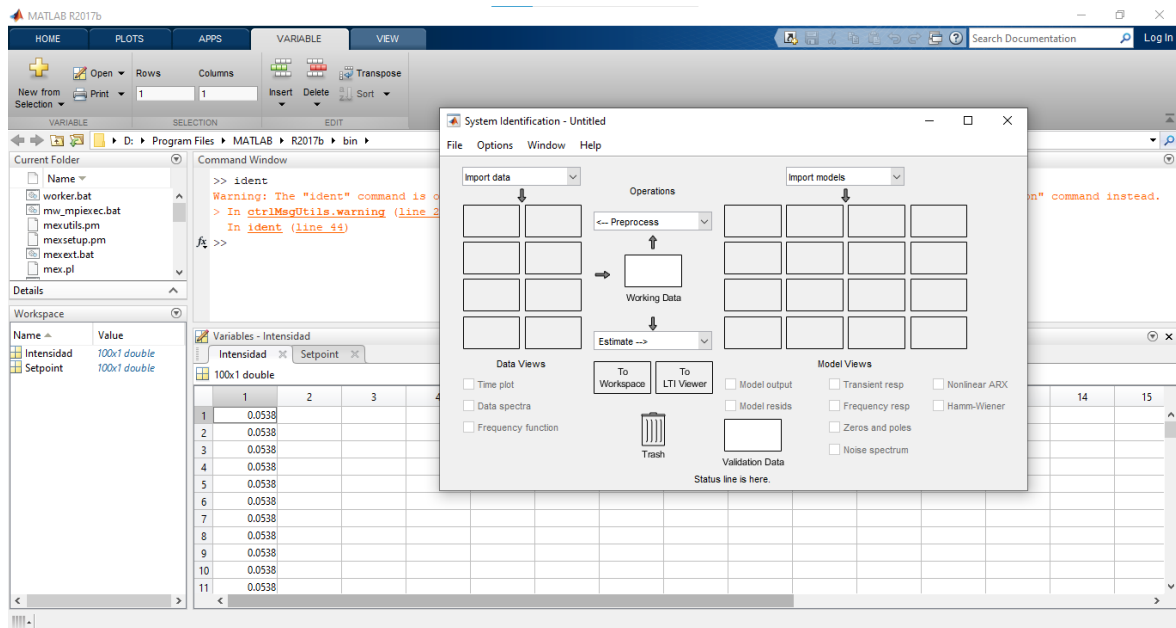


Figura 2.8 Vista de datos de Setpoint e Intensidad en el Workspace

- Dentro de la ventana “System Identification” se da clic en “Import data” y se selecciona “Time Dimain Data”, se desplegará una ventana con el nombre “Import Data”, se debe llenar con los siguientes datos:

Workspace Variable

- Input:** “Setpoint”
- Output:** “Intensidad”

Se debe ingresar el nombre de las variables ingresadas en la ventana “Workspace”.

Nota: Revisar las mayúsculas y minúsculas

Data Information

- Starting time: “0” (es en tiempo desde donde inicia el registro de datos, MyOpenLab registra los datos desde $t = 0s$)
- Sample time: “0.05” (Tiempo entre Muestras)

En la figura 2.9 se puede visualizar la ventana “Import Data” a continuación, se da clic en “Import”, obtenido así la función de transferencia de lazo abierto del sistema.

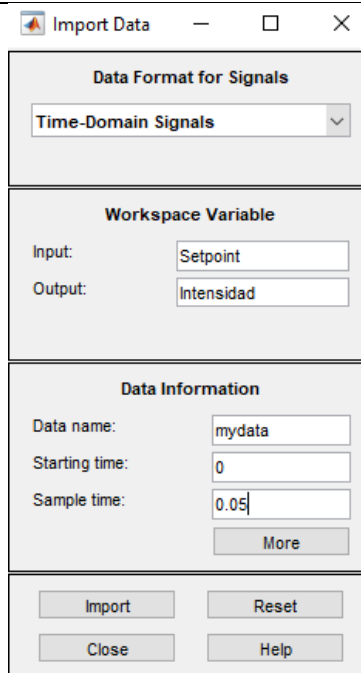


Figura 2.9 Ventana “Import Data”

Nota: Para visualizar las funciones de transferencia del Setpoint e Intensidad, se debe dar clic “Time plot” en la ventana “System Identification”.

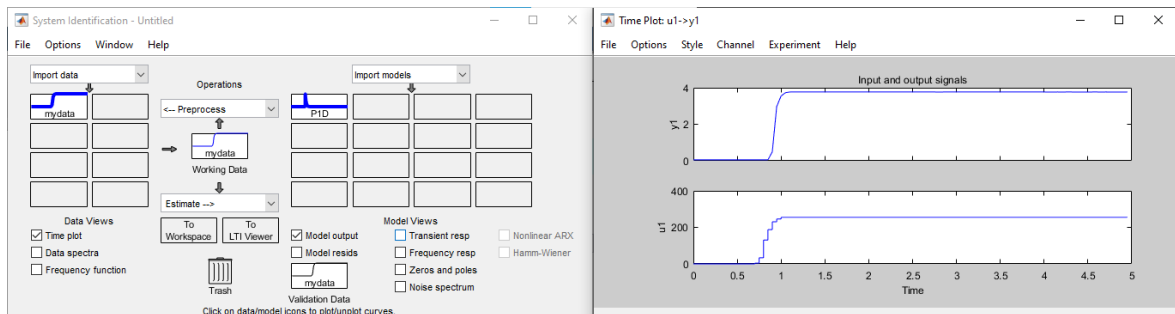


Figura 2.10 Ventana System Identification y gráfica de datos resultante

- En “Estimate -->” se selecciona “Process Model”, se desplegará la ventana “Process Model” al final de la venta se debe dar clic en estimar, para obtener las constantes K, Ti y Td, de este paso se obtiene el “Model output” final, como se ve en la figura 2.11

Nota: para visualizar el modelo de salida se debe dar clic en “Model output”

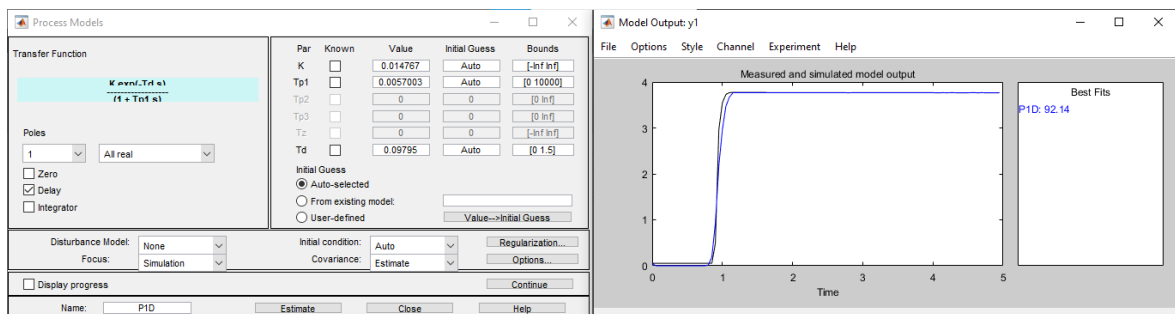


Figura 2.11 Ventana Process Model y gráfica del modelo PID de salida

Se selecciona la ventana del “PID” y se arrastra hasta la opción “To workspace” en la ventana “System Identification”.

- Transferido el (PID) a “To workspace” se cierra la ventana “System Identification” (no es necesario guardar nada). En la ventana principal de Matlab en “Command Window” se escribe el siguiente comando “ $Control = tf(PID)$ ” y se obtiene la función de transferencia del control como se ve en la figura 2.12.

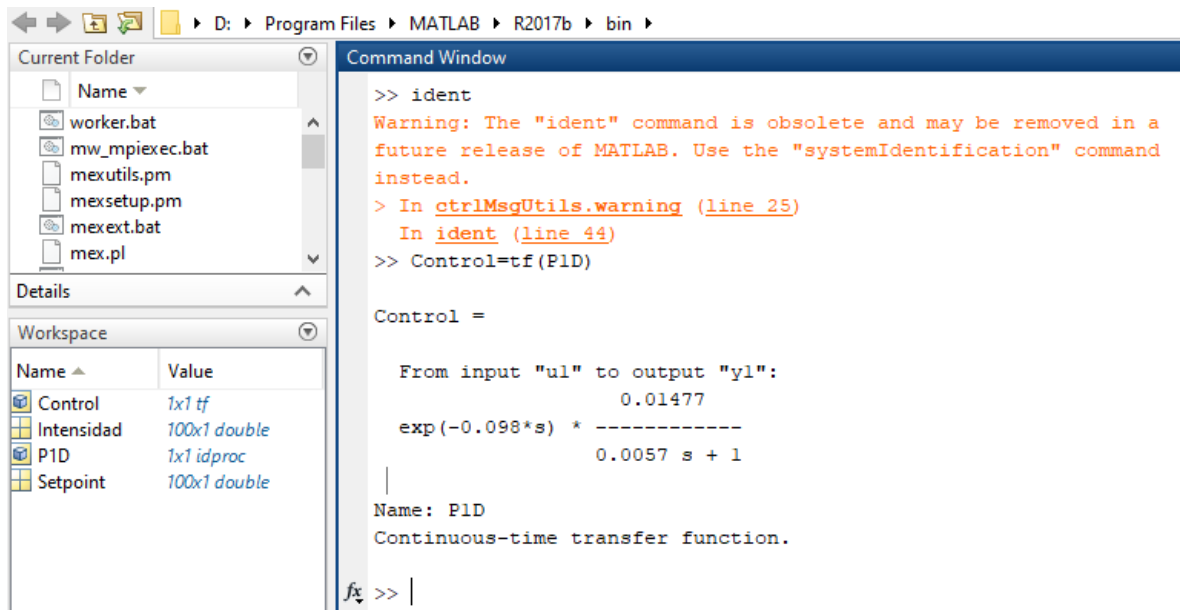


Figura 2.12 función de transferencia obtenida del Matlab

- En la parte superior de Matlab se selecciona “APPS”, y dentro de este se elige la herramienta “PID Tuner”, a continuación, se desplegará una ventana “PID Tune”, en esta ventana en la parte superior izquierda en “Plant” se puede importar el modelo del control, como se ve en la Figura 2.13.

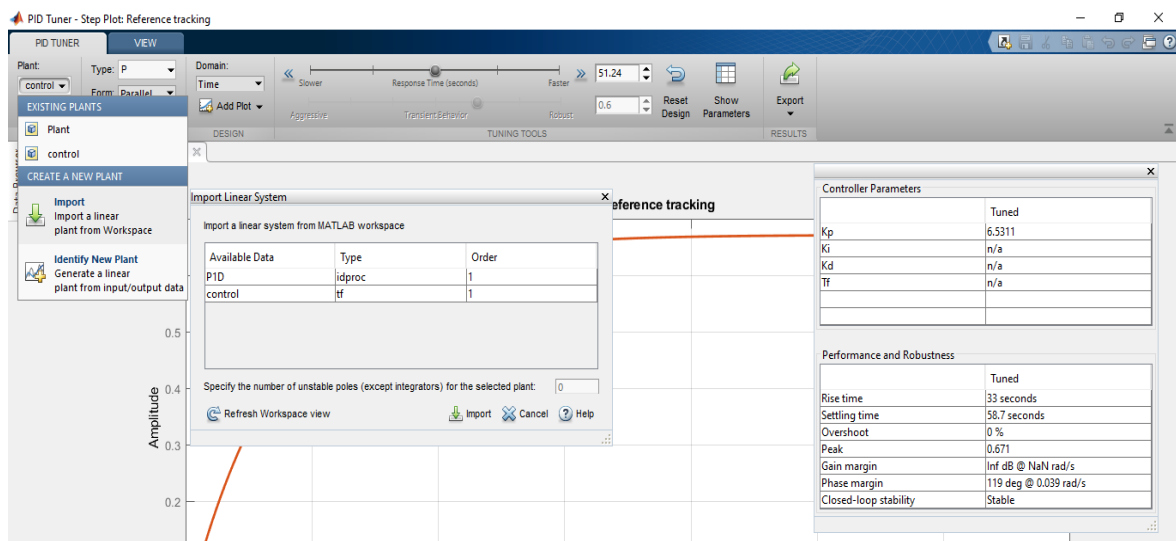


Figura 2.13 Ventana PID Tuner

En PID Tuner tenemos las opciones de elegir el tipo de proceso (P, PI, PD o PID) y elegir entre el modelo estándar (K, T_i, T_d) y el modelo paralelo (K_p, K_i, K_d).

2.6.4. Sintonización con método alternativo a PID Tuner, herramienta de Matlab

Una alternativa de uso gratuito y en líneas es Tune your PID (<https://pidtuner.com/>), los pasos son similares a Matlab, se deber cargar los datos del Setpoint y Sensor en función del tiempo y se obtienen las gráficas de Relación de la curva.

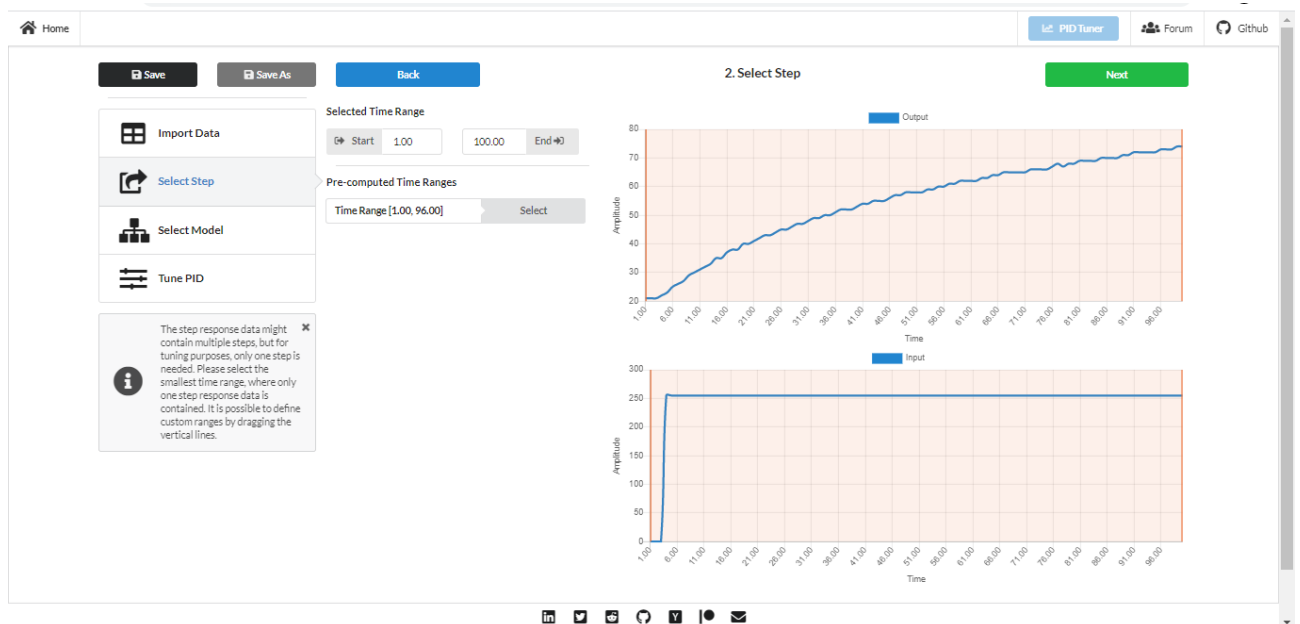


Figura 2.14 Ventana Tune your PID

Esta página también nos entrega la función de transferencia de proceso o control, como se ve en la figura 2.12

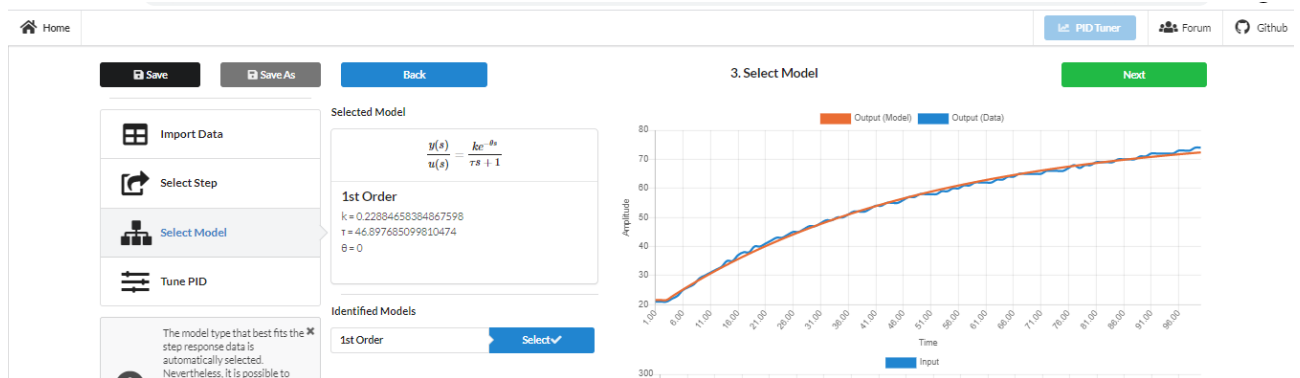


Figura 2.15 Valores de la función de transferencia en Tune your PID

Las contantes que entre Tune your PID, se encuentra en función del modelo estándar que son: (K , T_i y T_d). MyOpenLab emplea el modelo paralelo, para esto se emplea las siguientes fórmulas, para transformar esto valores:

$$K_p = K_p$$

$$K_i = K_p / T_i$$

$$K_d = K_p * T_d$$

INFORME DE PRÁCTICA

3 RESULTADOS Y DISCUSIÓN

Para realizar el análisis del comportamiento del sistema de control, se debe establecer un Setpoint diferente a la Intensidad Luminosa Actual y se empiece a registrar los datos, el “Tiempo de Muestreo” es el tiempo entre muestra tomada. En la figura 3.1 se visualiza la gráfica de los datos tomado, con el siguiente comportamiento:

- Intensidad Luminosa inicial = 0.0488 (V)
- Setpoint = 3.6591 (V)
- Tiempo de muestro = 1000 milisegundo.

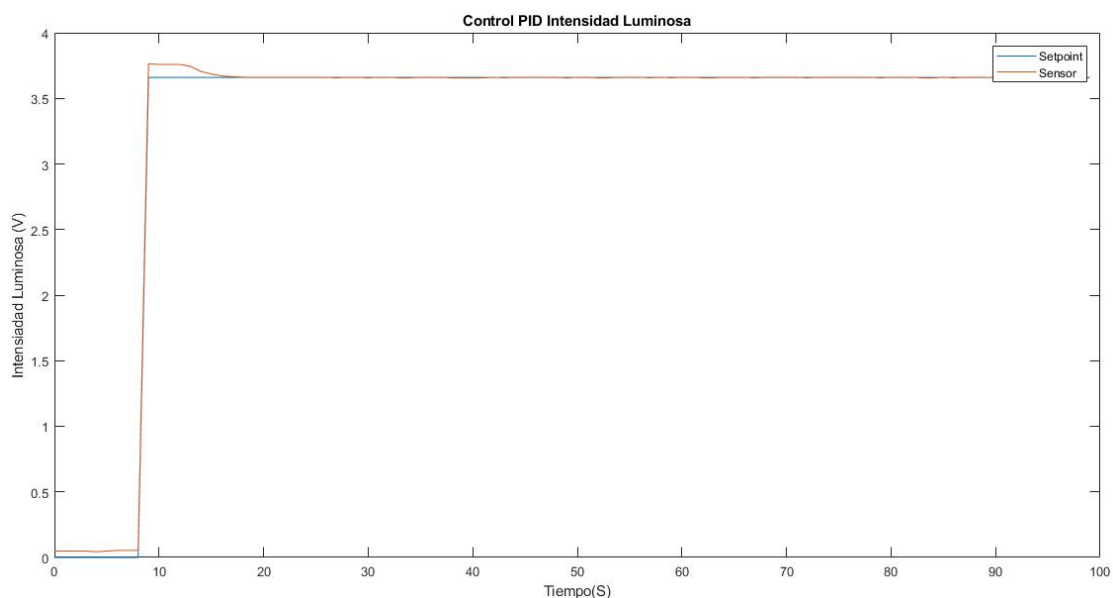


Figura 3.1 Comportamiento del PID de Intensidad Luminosa con un Setpoint de 3.6591 (V) equivalente al 100(%) de Intensidad Luminosa

3.4 ANÁLISIS DE RESULTADOS:

-

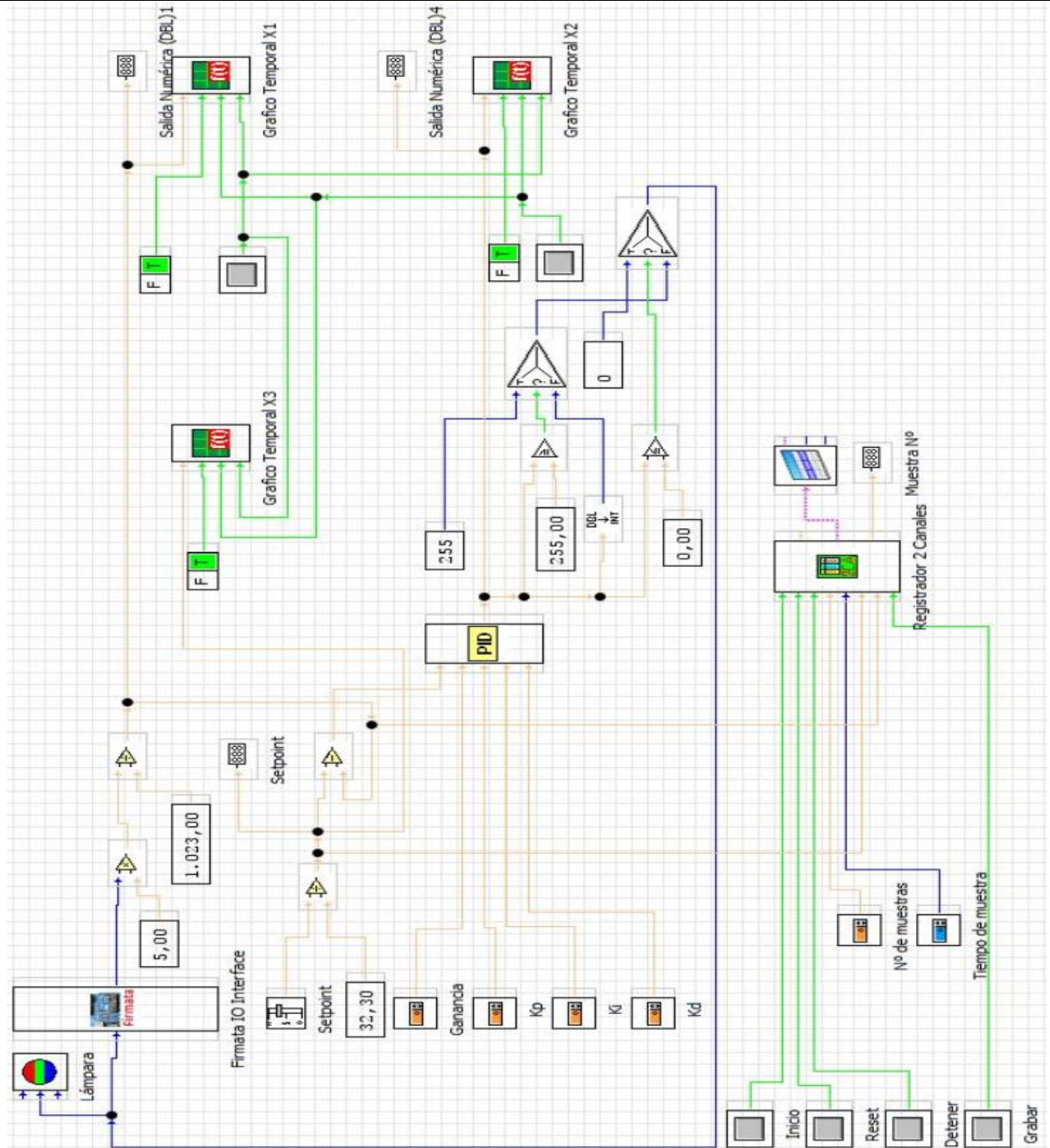
5 CONCLUSIONES

-

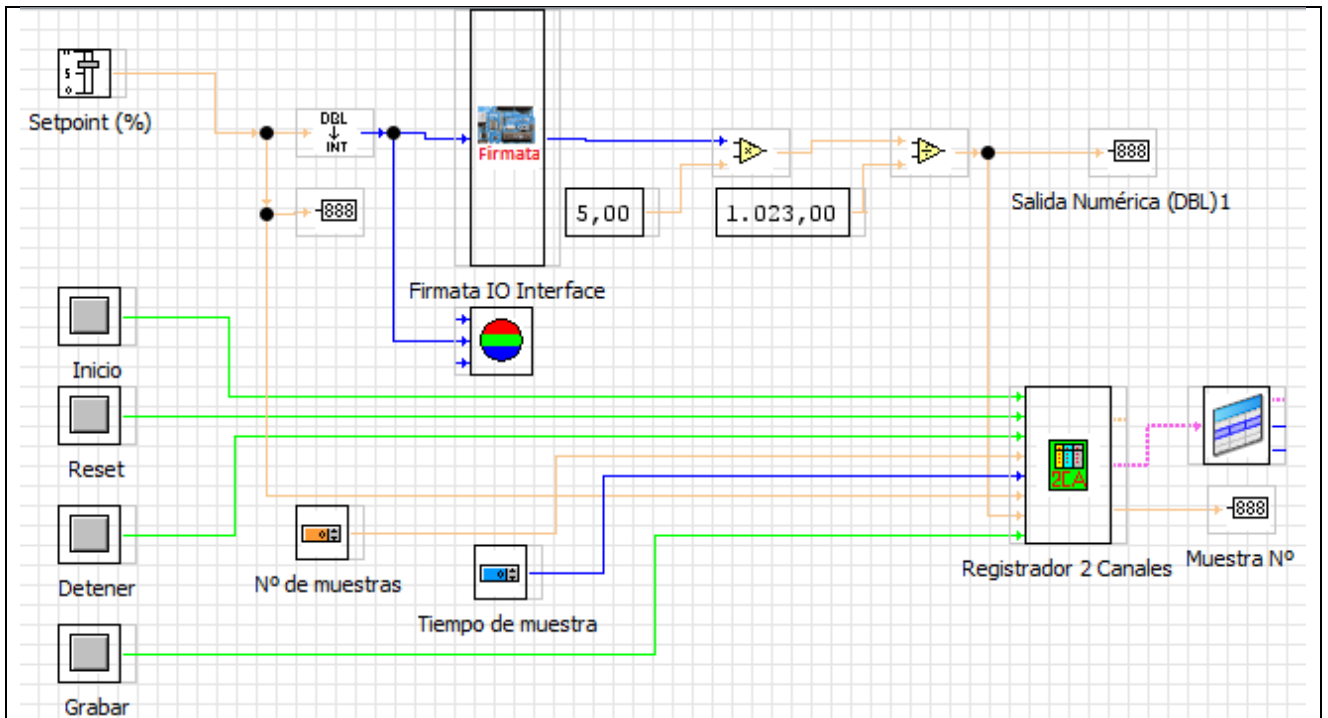
6 RECOMENDACIONES

- Para graficar la señal del Sensor y del Setpoint en la misma gráfica como se muestra en la figura 2.3 del panel frontal, se emplea dos elementos “Gráfico Temporal” X1 y X2 en el panel frontal, los dos gráficos deben tener las mismas unidades y dimensiones y no tener fondo (activar Background Transparent en las propiedades de la gráfica), para interpolar una gráfica sobre la otra.
-

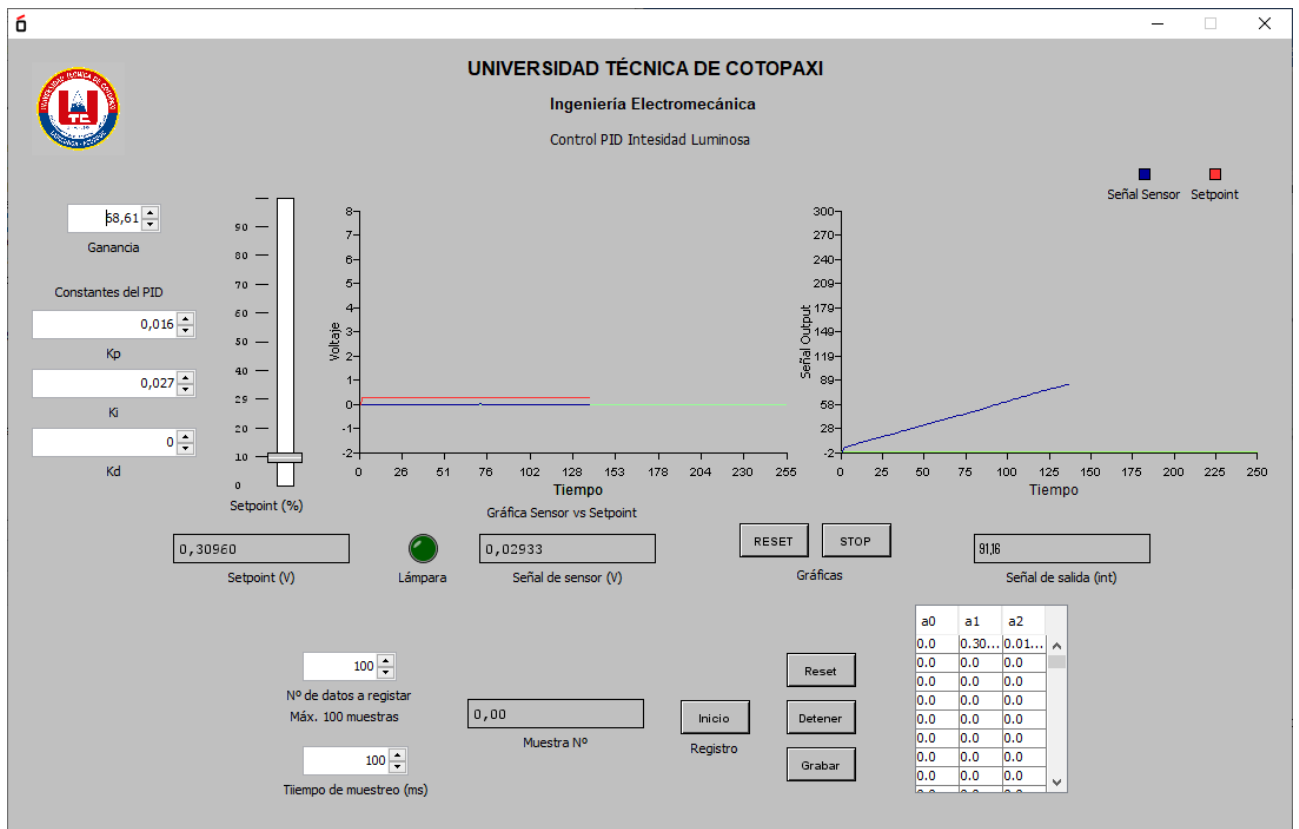
•



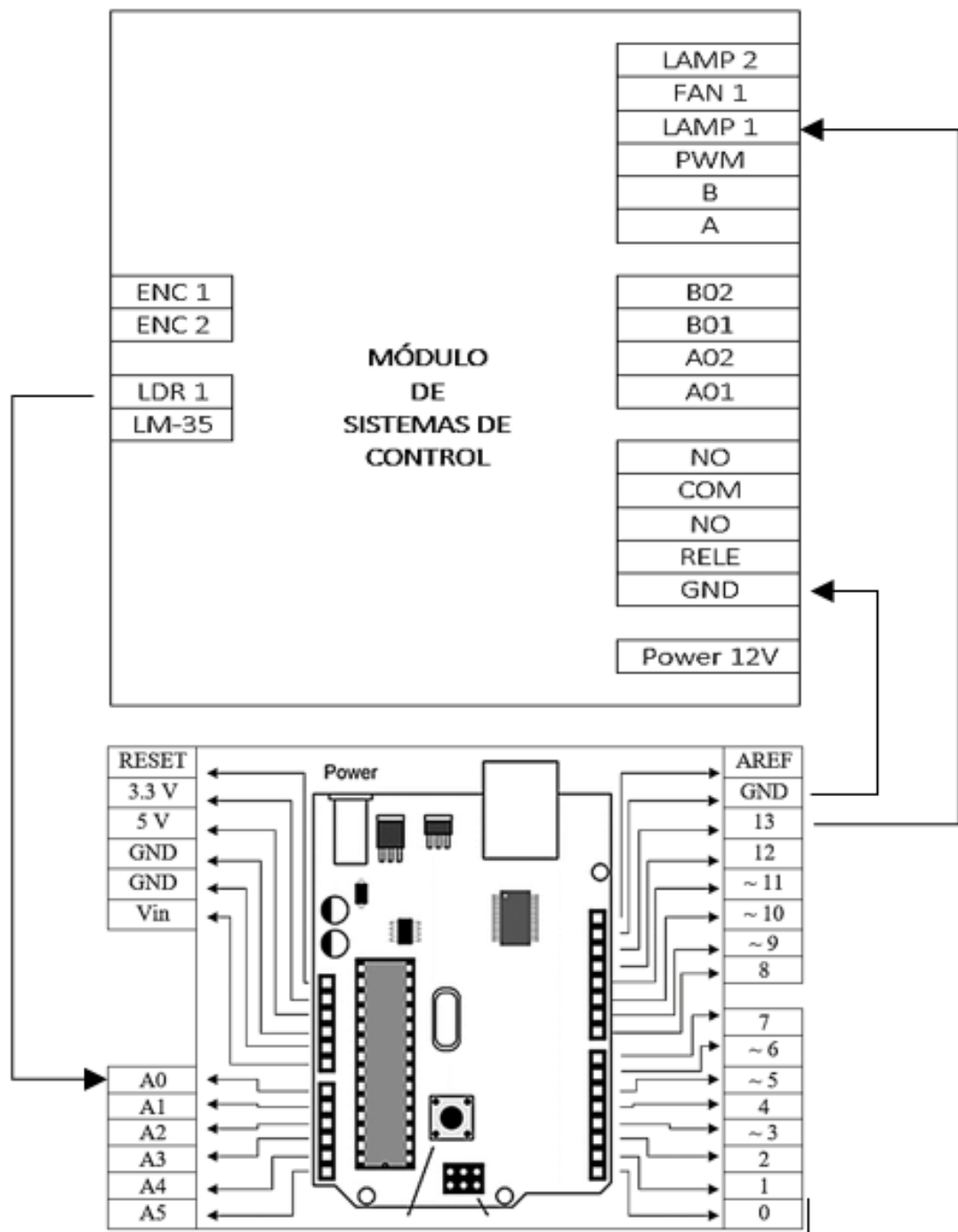
Anexo 1 Control PID de Intensidad Luminosa, panel de circuito




Anexo 2 Diagrama lazo abierto del Control PID de Intensidad Luminosa



Anexo 3 Control PID de Intensidad Luminosa, panel Frontal.



Anexo 4 Diagrama de conexión Módulo/Arduino, Control PID de Intensidad Luminosa

Anexo 38	UNIVERSIDAD TÉCNICA DE COTOPAXI	
PRÁCTICA DE LABORATORIO		

CARRERA	CÓDIGO DE LA ASIGNATURA	NOMBRE DE LA ASIGNATURA
INGENIERÍA ELECTROMECAÁNICA	ILM702M2	SISTEMAS DE CONTROL

PRÁCTICA N°	LABORATORIO:	Sistemas de Control	DURACIÓN (HORAS)
04	NOMBRE DE LA PRÁCTICA:	Control PID de Velocidad	3

1	OBJETIVOS
	<ul style="list-style-type: none"> • Cargar el código del protocolo “StandardFirmata” a la tarjeta Arduino. • Desarrollar la programación del control PID de control de velocidad y corrección en el software MyOpenLab, tanto del panel de circuito como el panel frontal • Obtener la función de transferencia y definir las constantes KP, Ki y KD del Sistema de Control. • Conectar el módulo a las salida y entradas; análogas y digitales de la tarjeta de adquisición (Arduino) • Recopilar datos del comportamiento del Sistema de Control y realizar la gráfica. • Analizar la forma de onda resultante del control.

2	FUNDAMENTACIÓN TEÓRICA
	<p>2.1. INTRODUCCIÓN:</p> <p>El Laboratorio de Sistemas de control cuenta con equipos para realizar de prácticas, a través de las cuales se pretende, complementar el estudio teórico con el práctico y crear en el estudiante habilidades para el diseño e implementación de Sistemas de control, generalmente utilizados en plantas industriales, procesos de producción, en la automatización de procesos, etc. El término control hace referencia a los métodos y maneras de gobernar el comportamiento de un aparato, máquina o proceso. En un sistema de control, hay varios</p>

parámetros de entrada que actúan sobre otros parámetros de salida, organizados en un orden lógico de control.

Un regulador o control PID es un algoritmo que permite controlar un sistema en lazo cerrado, con la única finalidad de alcanzar un salida deseado que es gobernado por un Setpoint. El control PID está compuesto de tres parámetros que proporcionan una acción Proporcional, un tiempo Integral y un tiempo Derivativo, estas tres constantes son las que dan el nombre al regulador o controlador PID.

La adquisición de datos y señales consiste en la toma de muestras de sistemas análogos (mundo real) con el fin de generar datos que puede ser manipulados, modificados e interpretados por un ordenador o sistema digital. También poder definirse como: tomar un conjunto de señales físicas y convertirlas en tensión y corriente eléctrica y digitalizarlas para ser procesados por un computador. Dentro del proceso de adquisición de datos se requiere una etapa de acondicionamiento de datos y señales con el fin de adecuar la señal a niveles compatibles con el elemento digital.

2.2 EQUIPO, INSTRUMENTOS Y MATERIALES NECESARIOS

- Módulo de Sistemas de Control
- Fuente de 12 voltios.
- Destornillador plano tipo bornera
- Cables Jumper
- Computadora

2.3 SOFTWARE REQUERIDO

- MyOpenLab
 - Arduino
- (revisar manual de uso, para su instalación)

2.4 MEDIDAS DE SEGURIDAD

- Revisar el manual de uso del módulo de sistemas de control
- Realizar la conexión de la tarjeta Arduino con el módulo des-energizado y cuando los leds indicadores no estén encendidos

2.5 ACTIVIDADES A DESARROLLAR:

2.5.1 Cargar el código del protocolo “StandardFirmata” a la tarjeta Arduino, siguiendo el siguiente enlace: [Abrir Arduino/Archivo/Ejemplos/Firmata/Estándar Firmata](#), como se muestra en la figura 2.1

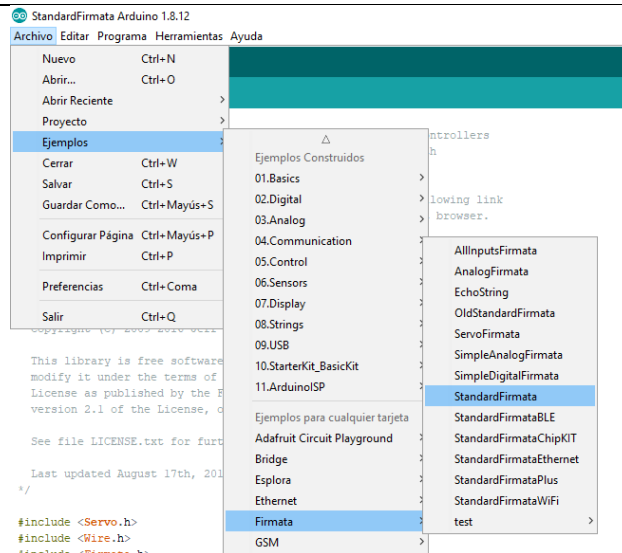


Figura 2.1. Guía para cargar el protocolo “StandardFirmata”

2.5.1.1 Se desplegará el código del protocolo StandardFirmata, sobre la pantalla, la tarjeta debe estar conectada previamente, se da clic en verificar, para revisar que no exista errores en el código y a continuación se da clic en Subir, si la ventana de comandos inferior cambia a naranja (error) revisar el puerto de comunicación de la tarjeta Arduino.

2.5.2 Abrir el programa MyOpenLab y crear un nuevo proyecto.

En la parte superior de la ventana de MyOpenLab, se debe seguir el siguiente enlace para crear un nuevo proyecto: Fichero/Nuevo Proyecto, dentro de la ventana “Nuevo Proyecto” se puede cambiar el nombre del proyecto y seleccionar la ubicación; como se muestra en la siguiente figura.

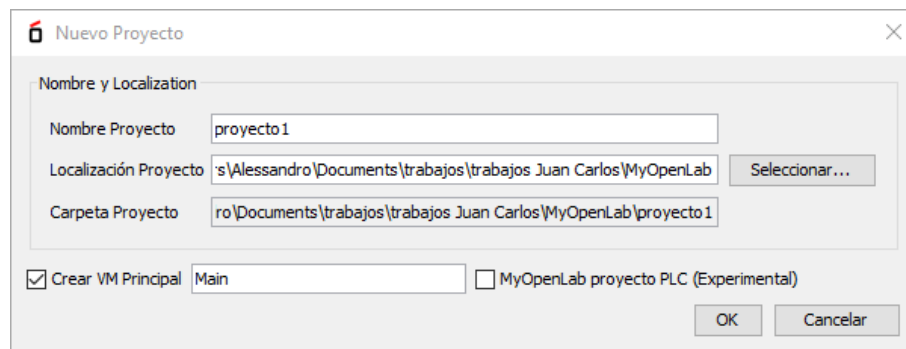


Figura 2.2 Ventana “Nuevo Proyecto”

2.5.3 Repicar el código de programación que se muestra en el **Anexo 1**, los pines de entrada y salida del Firmata IO Interface son:

- **1 Analog_Input:** ENC 2 (pin 14, es necesario habilitar este por la programación, ya que la señal del encoder 2 (señal 0,1) entra por el pin 2, el StandardFirmata_Editado acondiciona la señal y la entrega en el Pin 14 como señal (int) dentro de MyOpenLab).
- **2 Digital_Output:** A, B (cambio de giro) pines 4 y 5
- **1 PWM_Output:** PWM, pin 3

Para configurar los pines se debe dar clic sobre el “Firmata IO Interface”, aparecerá una ventana en la esquina inferior “Componentes”, al final de la ventana se debe dar clic sobre los 3 puntos (...) en “Config Pins” a continuación, se desplegará una ventana donde se poder habilitar cada una de las entradas y salidas deseadas como se muestra en la siguiente imagen.

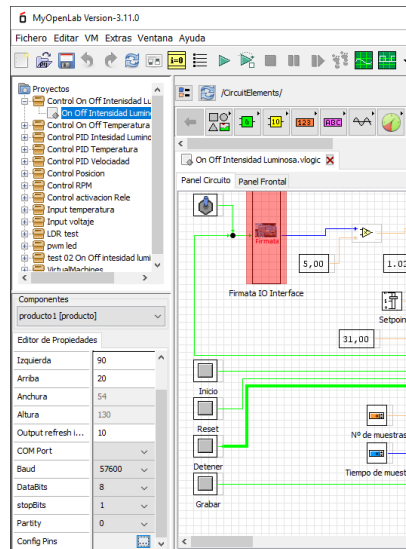


Figura 2.3 configuración de los pines de Firmata IO Interface

Antes de programar, la tarjeta Arduino debe estar conectada, para que MyOpenLab la reconozca automáticamente. Se debe editar el código del StandardFirmata, para transforma la señal del encoder 2 (dato booleano) a dato entero (int), esta son las modificaciones del código:

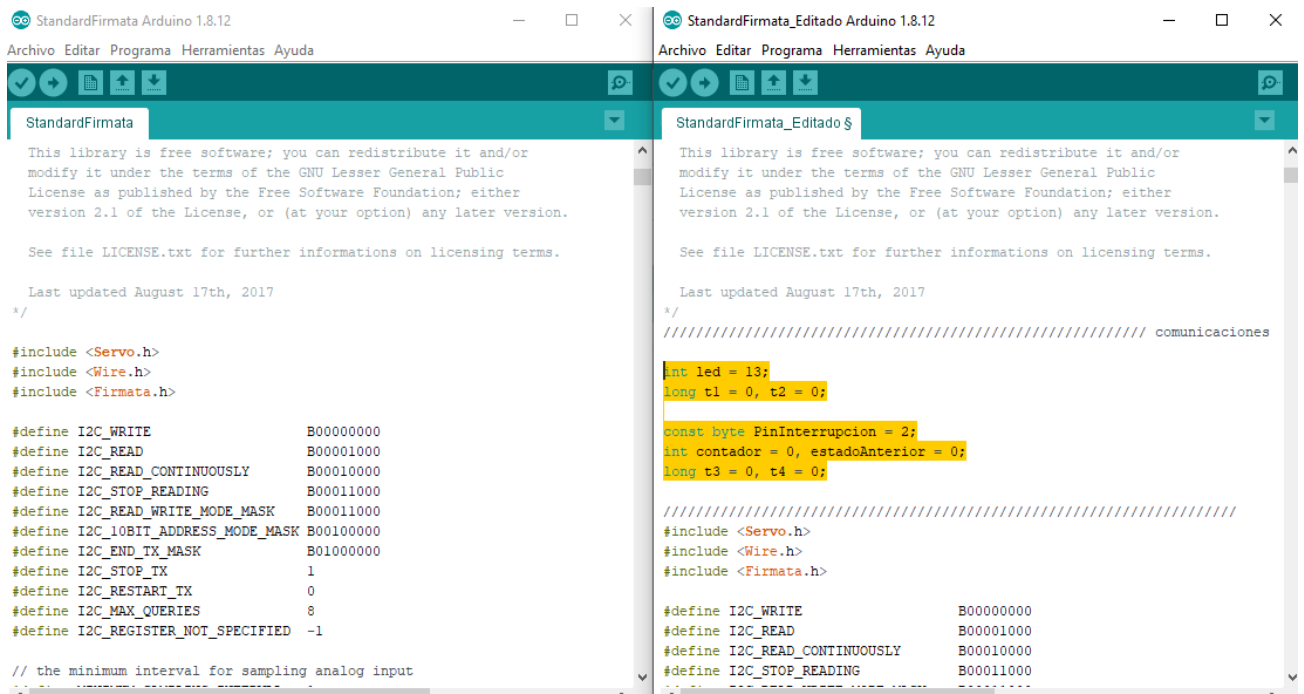


Figura 2.4 StandardFirmata Original - StandardFirmata Modificado; declaración de variables

Figura 2.5 StandardFirmata Original - StandardFirmata Modificado; (modificación 2)

Figura 2.5 StandardFirmata Original - StandardFirmata Modificado; (modificación 3)

Si no se logra modificar el código del protocolo StandardFirmata, se puede copiar el código del siguiente link:

https://create.arduino.cc/editor/juan_carlos_capa/331e9f20-6d20-4686-ba00-83134384e82a/preview

Para este control se utiliza el componente Regulador PID que está dentro de las librerías de MyOpenLab, este componente posee 5 pines de entrada y un pin de salida como se puede ver en la figura 2.6, donde (e) es la desviación del control o error, (ta) es la ganancia, (kp) (ki) (kd) son las contantes proporcional, integra y derivativa, y el pin de salida (y) es la señal de salida del control PID.

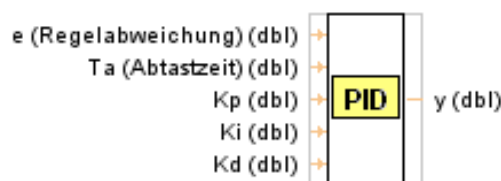


Figura 2.6 Regulador PID de MyOpenLab

Este regulador PID trabaja con datos flotantes (dbl), es necesario acondicionar esta señal de salida; pasar de un dato flotante (dbl) a un dato entero (int), para que sea conectada a los pines de salida de la tarjeta de adquisición de datos Arduino, como se muestra en la figura 2.2 a la salida del regulador PID.

En el **Anexo 3** se visualiza la apariencia del control PID de velocidad del panel frontal

2.5.5 Para realizar la sintonización del control PID se debe crear un nuevo proyecto, como se muestra en el anexo 2, es un control abierto de Intensidad Luminosa, esta programación permitirá recopilar datos para generar la función de transferencia y las constantes K_p , K_i y K_d . Los métodos que se puede utilizar se mencionarán en el siguiente capítulo de Metodología. Después de programar, se recomienda guardar el archivo, antes de conectar al módulo de Sistemas de Control.

2.5.6 Se debe conectar con el módulo des-energizado, al lado izquierdo de módulo se encuentran los sensores que deben conectarse a las entradas análogas de la tarjeta Arduino, y al lado derecho se encuentran dispuestos los actuadores; en el **Anexo 4** se visualiza un ejemplo de conexión del control PID de velocidad

- ENC 2, se debe conetera en el pin 2 (por la programación)
- PWM, se conecta en el pin 3
- A y B, se conecta en el pin 4 y 5

3. METODOLOGÍA Y TÉCNICA EXPERIMENTAL

Para realizar la curva de transferencia se debe registrar los datos desde que el Setpoint está en 0 (int) y subir al 100 % = 255 (int) se debe registrar el comportamiento del sistema, se recomienda un tiempo de muestreo igual a 200 (ms) mili-segundo, como se visualiza en la siguiente imagen.

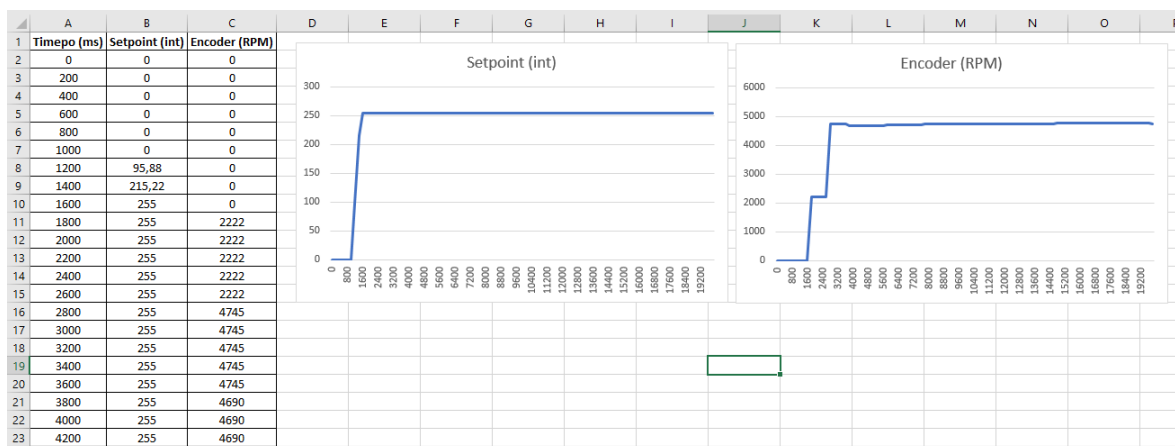


Figura 2.7 Curva de Reacción del sistema en lazo abierto

Con los datos recopilados en MyOpenLab, se crea la curva de ración en lazo abierto del sistema, se debe convertir los datos del Setpoint (int) en datos de Setpoint (RPM) para obtener el comportamiento del setpoint y del Encoder en la misma gráfica como se muestra en la figura 2.7.1. realizado esto se puede aplicar cualquiera de los siguientes métodos.

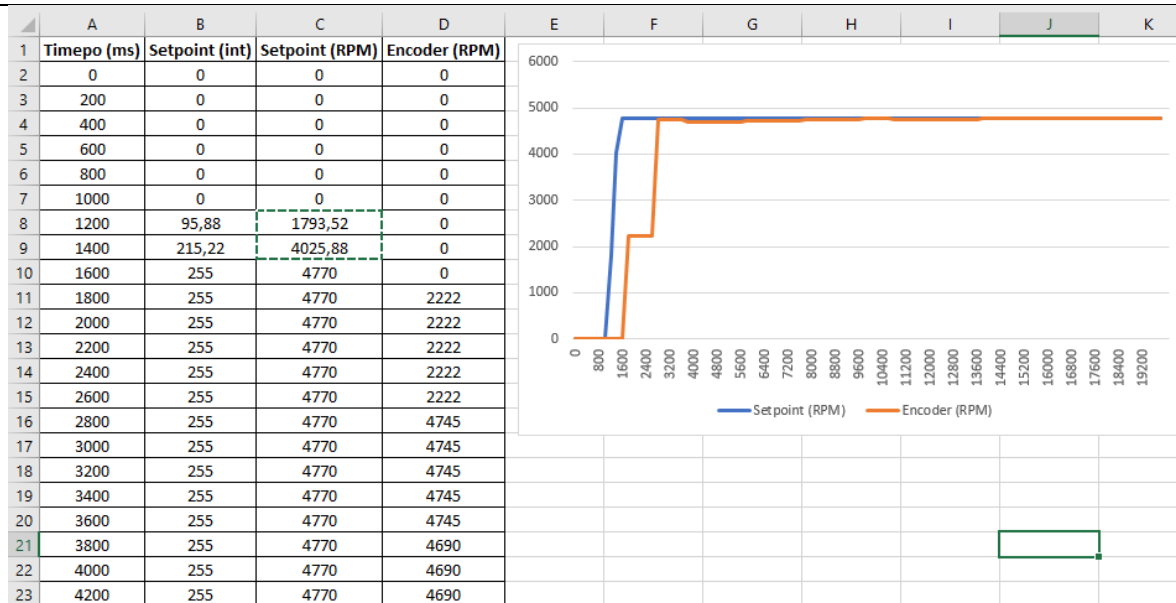


Figura 2.7.1 Curva de Reacción del sistema en lazo abierto

2.6.3 Sintonización por el método de curva de Reacción de Ziegler-Nichols

Se lleva manualmente el sistema o planta a lazo abierto, se establece un Setpoint y se registra el comportamiento de la planta, para calcular los parámetros del modelo se emplea las siguientes formula:

$$Ganancia (K) = \frac{RPM\ 2 - RPM\ 1}{Setpoint\ 2 - Setpoint\ 1} \left(\frac{RPM}{int} \right)$$

Se establece un Setpoint equivalente al 28.30% y se registra las RPM que se obtiene después de un tiempo; a continuación, se establece otro Setpoint equivalente al 63.20 % y se registra el valor de las RPM, para facilitar el registro se define los Setpoint equivalentes a 28.30% y 63.20% en un tiempo de muestreo de 200 (ms) mili-segundo, registrado estos datos se promedia, obtenido los valores equivalentes en RPM; como se muestra en la figura 2.8. Estos valores se reemplazan en la siguiente formula:

	A	B	C	D	E	F	G	H	I	J	K	L
1		28,3 (%)				63,20 (%)						
2	Timepo (MS)	Setpoint (int)	Endocer (RPM)	Timepo (MS)	Setpoint (int)	Endocer (RPM)				Setpoint (int)	Sensor (°C)	
3	0	72,42	1347,58	0	161,16	2998,84			Promedio 1	72,42	1347,58	
4	200	72,42	1347,58	200	161,16	2998,84			Promedio 2	161,16	2998,84	
5	400	72,42	1347,58	400	161,16	2998,84			Ganancia	18,60784314		
6	600	72,42	1347,58	600	161,16	2998,84			$Ganancia (K) = \frac{RPM\ 2 - RPM\ 1}{Setpoint\ 2 - Setpoint\ 1} \left(\frac{RPM}{int} \right)$			
7	800	72,42	1347,58	800	161,16	2998,84						
8	1000	72,42	1347,58	1000	161,16	2998,84						
9	1200	72,42	1347,58	1200	161,16	2998,84						
10	1400	72,42	1347,58	1400	161,16	2998,84						
11	1600	72,42	1347,58	1600	161,16	2998,84						
12	1800	72,42	1347,58	1800	161,16	2998,84						
13	2000	72,42	1347,58	2000	161,16	2998,84						
14	2200	72,42	1347,58	2200	161,16	2998,84						
15	2400	72,42	1347,58	2400	161,16	2998,84						
16	2600	72,42	1347,58	2600	161,16	2998,84						
17	2800	72,42	1347,58	2800	161,16	2998,84						
18	3000	72,42	1347,58	3000	161,16	2998,84						
19	3200	72,42	1347,58	3200	161,16	2998,84						
20	3400	72,42	1347,58	3400	161,16	2998,84						
21	3600	72,42	1347,58	3600	161,16	2998,84						

Figura 2.8 Cálculo de la ganancia del sistema

En la curva de reacción se traza un tangente en el punto de inflexión de la curva y obtiene el tiempo de retador del sistema (L) y la constante del tiempo (T) como se muestra en la figura 2.9.

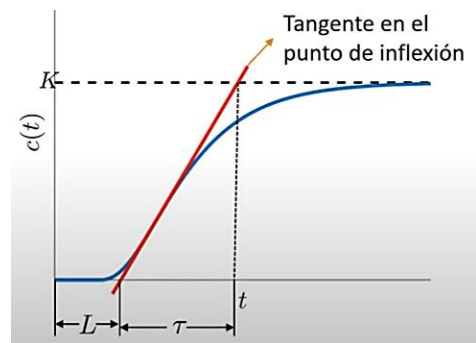


Figura 2.9. (L) y (T) en la curva de Reacción del sistema

Fuente [1]

Calculado la Ganancia del sistema, el tiempo de retador del sistema y la constante del tiempo se establece la función de transferencia equivalente del sistema de control.

$$G(S) = \frac{K * e^{-Ls}}{Ts + 1}$$

Tabla 2.1 Método Ziegler y Nichols

Control	Kp	Ti	Td
P	$T/(K*L)$	n/a	n/a
PI	$0.9*T/(K*L)$	$L/ (0.3)$	n/a
PID	$1.2*T/(K*L)$	$2L$	$0.5L$

Fuente: [3]

Los valores obtenidos del método de lazo abierto de Ziegler y Nichols están expresados en el modelo estándar de control PID, MyOpenLab emplea el modelo paralelo, para esto se emplean las siguientes fórmulas, para transformar estos valores:

$$Kp = Kp$$

$$Ki = Kp / Ti$$

$$Kd = Kp * Td$$

2.6.4 Sintonización utilizando PID Tuner, herramienta de Matlab

Para lograr la sintonización del control con esta herramienta se debe seguir los siguientes pasos:

7. Se deben cargar los datos de Encoder (RPM) y del Setpoint (int) en la ventana "Workspace". Para cargar los datos, se hace clic derecho sobre la ventana, se desplegará una ventana donde se selecciona "NEW", a continuación, se define un nombre "Setpoint" y otra con el nombre "Encoder", creadas estas dos variables se hace doble clic en cualquiera de ellas, se desplegará

una hoja de cálculo (padecida a Excel), en se debe pegar los datos obtenidos de MyOpenLab del Setpoint e Encoder.

- Una vez cargado los datos, se emplea el comando (`>>ident`) en la ventana “Command Window” se presiona la tecla “Entre”, se desplegará una ventana con el nombre “System identification” en donde se importarán los datos del sistema de control en función del tiempo, como se puede ver en la figura 2.10.

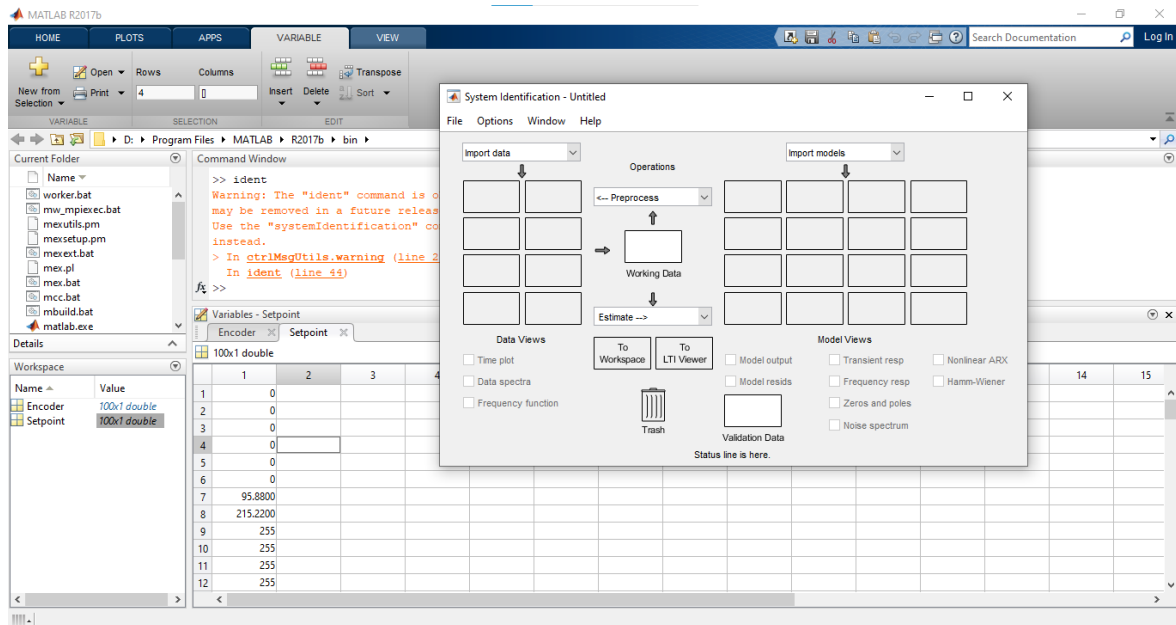


Figura 2.10 Vista de datos de Setpoint e Intensidad en el Workspace

- Dentro de la ventana “System Identification” se da clic en “Import data” y se selecciona “Time Dimain Data”, se desplegará una ventana con el nombre “Import Data”, se debe llenar con los siguientes datos:

Workspace Variable

- Input:** “Setpoint”
- Output:** “Encoder”

Se debe ingresar el nombre de las variables ingresadas en la ventana “Workpace”.

Nota: Revisar las mayúsculas y minúsculas

Data Information

- Starting time: “0” (es en tiempo desde donde inicia el registro de datos, MyOpenLab registra los datos desde $t = 0s$)
- Sample time: “0.2” (Tiempo entre Muestras 200(ms))

En la figura 2.11 se puede visualizar la ventana “Import Data” a continuación, se da clic en “Import”, obtenido así la función de transferencia de lazo abierto del sistema.

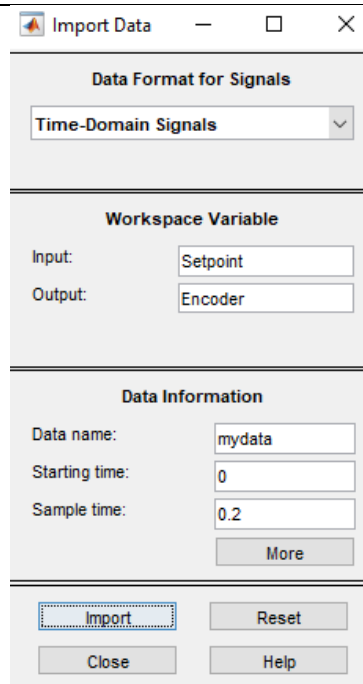


Figura 2.11 Ventana “Import Data”

Nota: Para visualizar las funciones de transferencia del Setpoint y Encoder, se debe dar clic “Time plot” en la ventana “System Identification”.

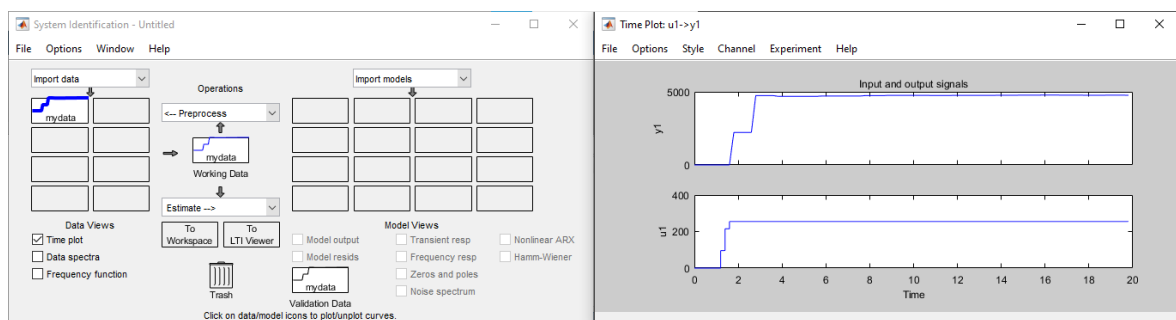


Figura 2.12 Ventana System Identification y gráfica de datos resultante

10. En “Estimate -->” se selecciona “Process Model”, se desplegará la ventana “Process Model” al final de la venta se debe dar clic en estimar, para obtener las constantes K, Ti y Td, de este paso se obtiene el “Model output” final, como se ve en la figura 2.13

Nota: para visualizar el modelo de salida se debe dar clic en “Model output”

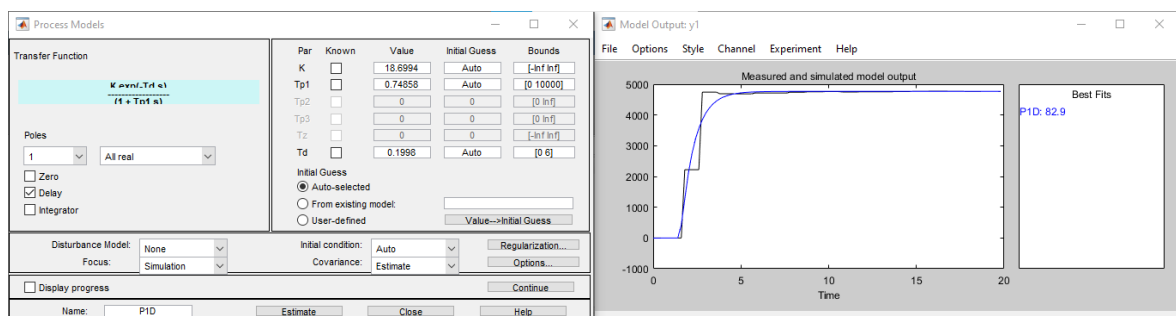


Figura 2.13 Ventana Process Model y gráfica del modelo PID de salida

Se selecciona la ventana del “PID” y se arrastra hasta la opción “To workspace” en la ventana “System Identification”.

- Transferido el (PID) a “To workspace” se cierra la ventana “System Identification” (no es necesario guardar nada). En la ventana principal de Matlab en “Command Window” se escribe el siguiente comando “ $Control = tf(PID)$ ” y se obtiene la función de transferencia del control como se ve en la figura 2.14.

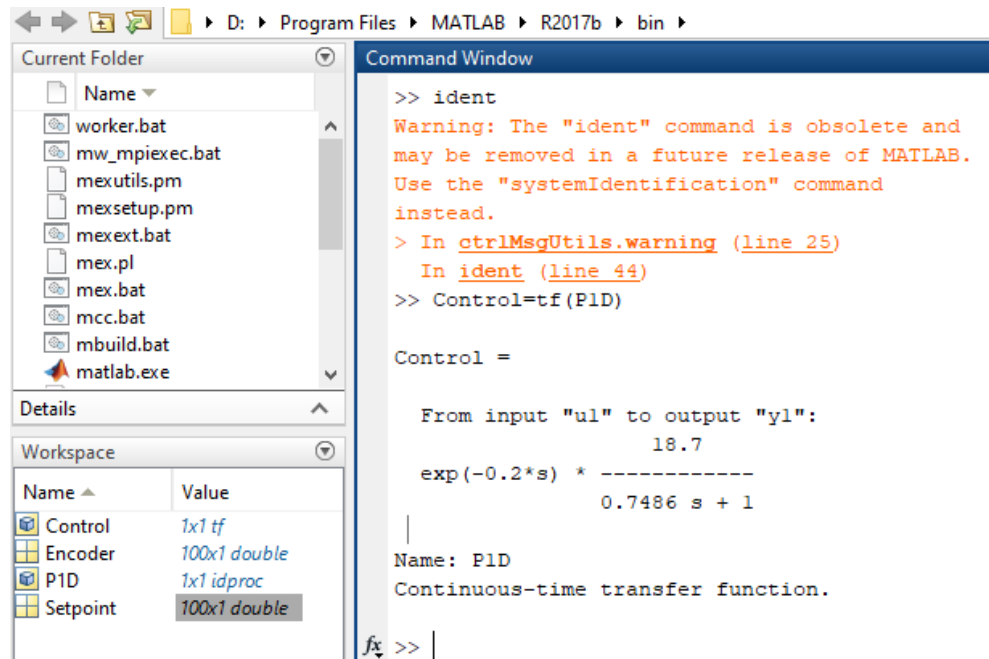


Figura 2.14 función de transferencia obtenida del Matlab

- En la parte superior de Matlab se selecciona “APPS”, y dentro de este se elige la herramienta “PID Tuner”, a continuación, se desplegará una ventana “PID Tune”, en esta ventana en la parte superior izquierda en “Plant” se puede importar el modelo del control, como se ve en la Figura 2.15.

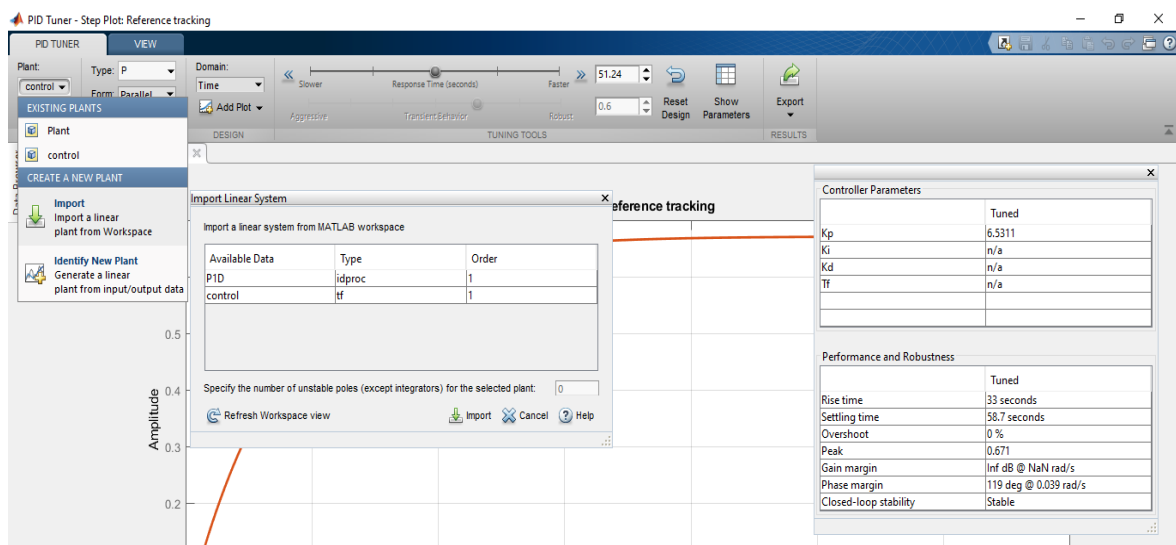


Figura 2.15 Ventana PID Tuner

En PID Tuner tenemos las opciones de elegir el tipo de proceso (P, PI, PD o PID) y elegir entre el modelo estándar (K, T_i, T_d) y el modelo paralelo ($K_p, K_i, y K_d$).

3.6.1. Sintonización con método alternativo a PID Tuner, herramienta de Matlab

Una alternativa de uso gratuito y en líneas es Tune your PID (<https://pidtuner.com/>), los pasos son similares a Matlab, se deber cargar los datos del Setpoint y Sensor en función del tiempo y se obtienen las gráficas de Relación de la curva.

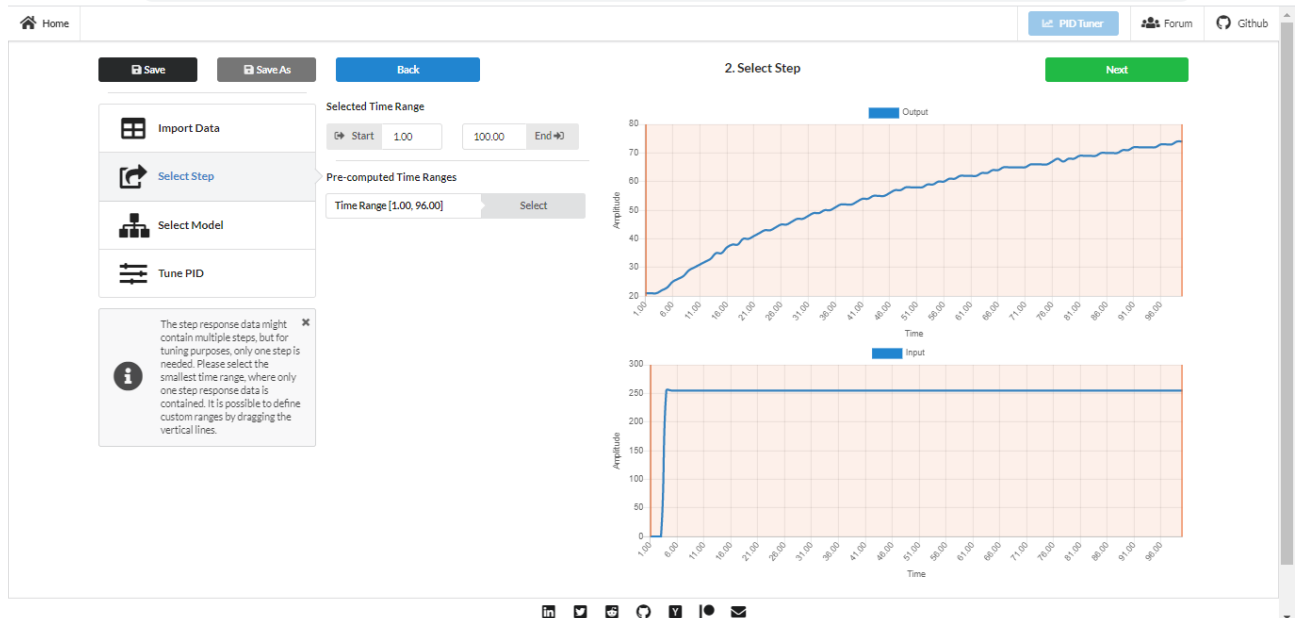


Figura 2.16. Ventana Tune your PID

Esta página también nos entrega la función de transferencia de proceso o control, como se ve en la figura 2.12

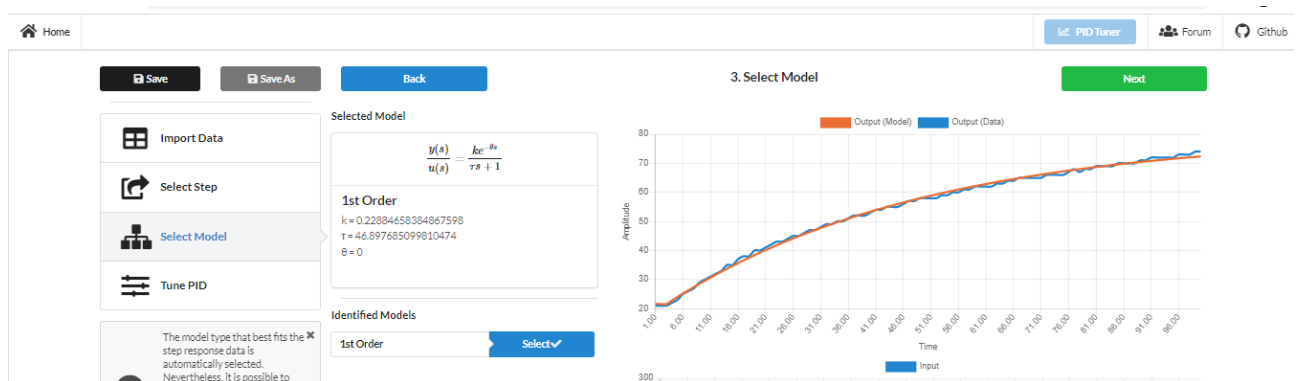


Figura 2.17. Valores de la función de transferencia en Tune your PID

Las constantes que entre Tune your PID, se encuentra en función del modelo estándar que son: (K, T_i y T_d). MyOpenLab emplea el modelo paralelo, para esto se emplea las siguientes fórmulas, para transformar esto valores:

$$K_p = K_p$$

$$K_i = K_p / T_i$$

$$K_d = K_p * T_d$$

INFORME DE PRÁCTICA

3 RESULTADOS Y DISCUSIÓN

Para realizar el análisis del comportamiento del sistema de control, se debe establecer un Setpoint diferente a la RPM Actual y se empieza a registrar los datos, el “Tiempo de Muestreo” es el tiempo entre muestra tomada. En la figura 3.1 se visualiza la gráfica de los datos tomado, con el siguiente comportamiento:

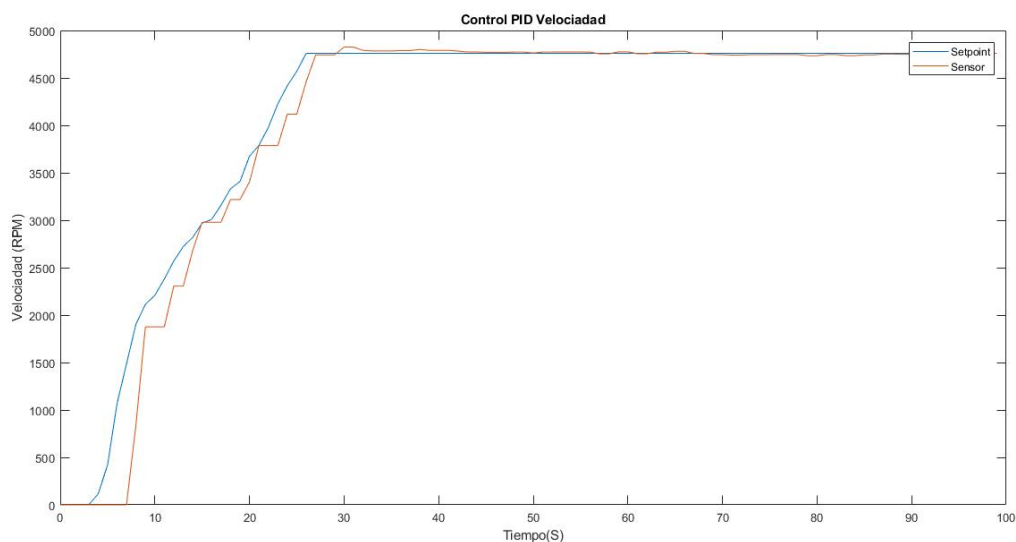


Figura 3.1 Comportamiento del PID de motor dc con un Setpoint de 4700 RPM

3.5 ANÁLISIS DE RESULTADOS:

-

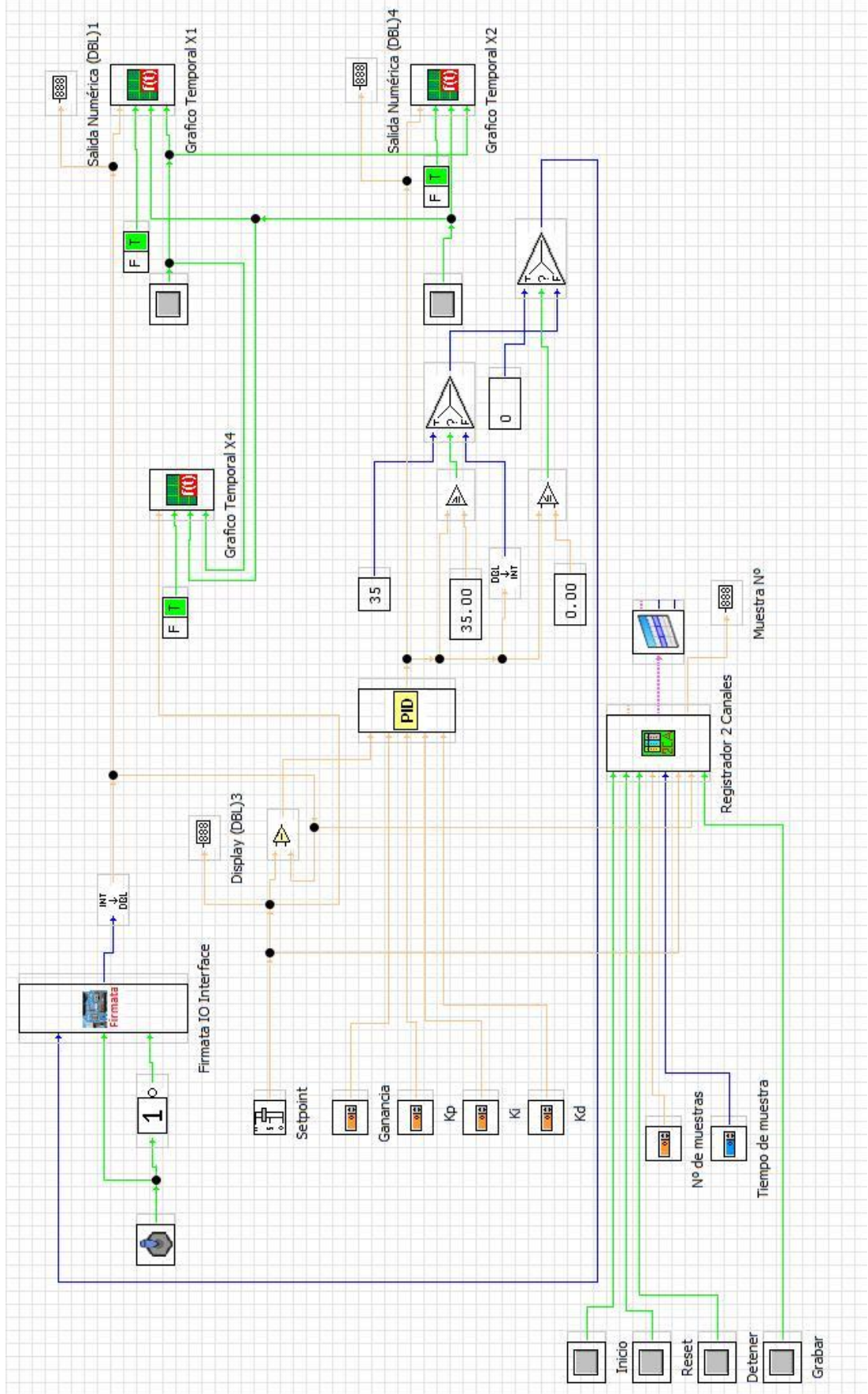
5 CONCLUSIONES

-

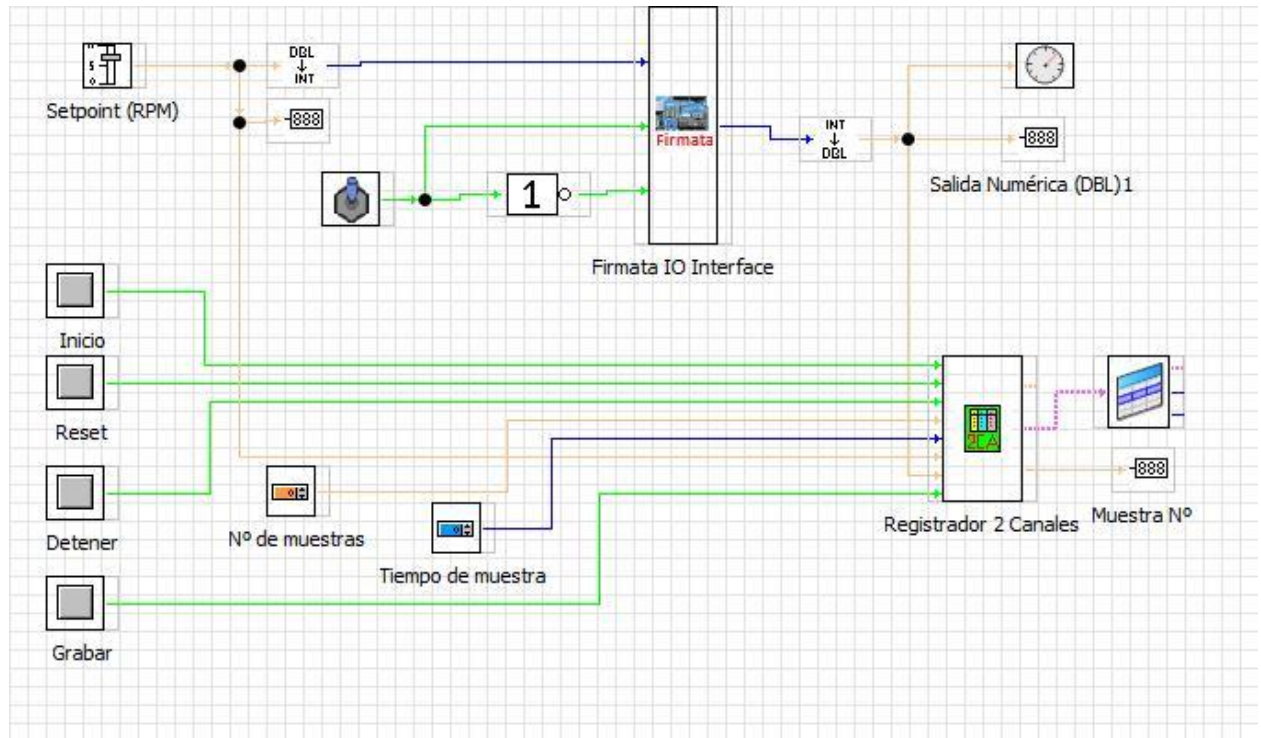
6 RECOMENDACIONES

- Para graficar la señal del Sensor y del Setpoint en la misma gráfica como se muestra en la figura 2.3 del panel frontal, se emplea dos elementos “Gráfico Temporal” X1 y X2 en el panel frontal, los dos gráficos deben tener las mismas unidades y dimensiones y no tener fondo (activar Background Transparent en las propiedades de la gráfica), para interpolar una gráfica sobre la otra.
-

7 REFERENCIAS BIBLIOGRÁFICAS (Utilizar las normas IEEE)



Anexo 1 Control PID de control de velocidad, panel de circuito



Anexo 2 Diagrama lazo abierto del Control PID de velocidad

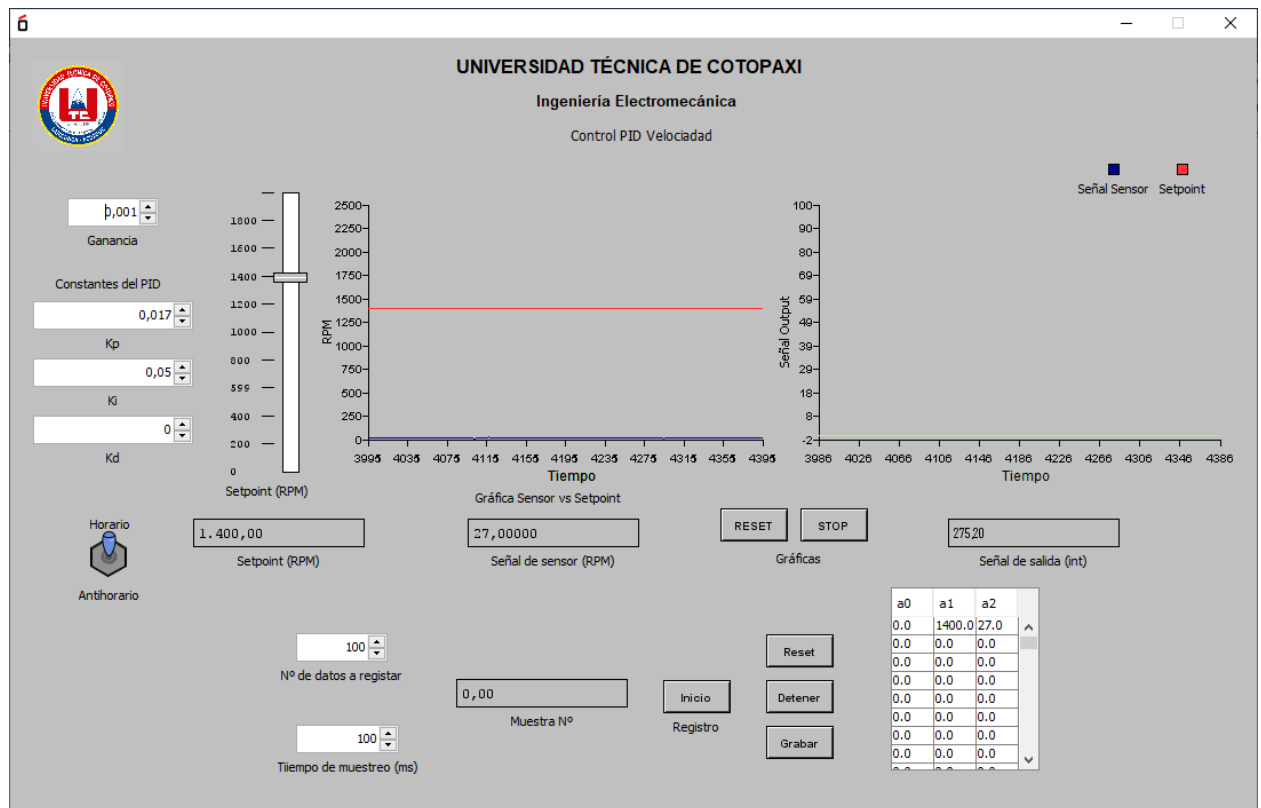
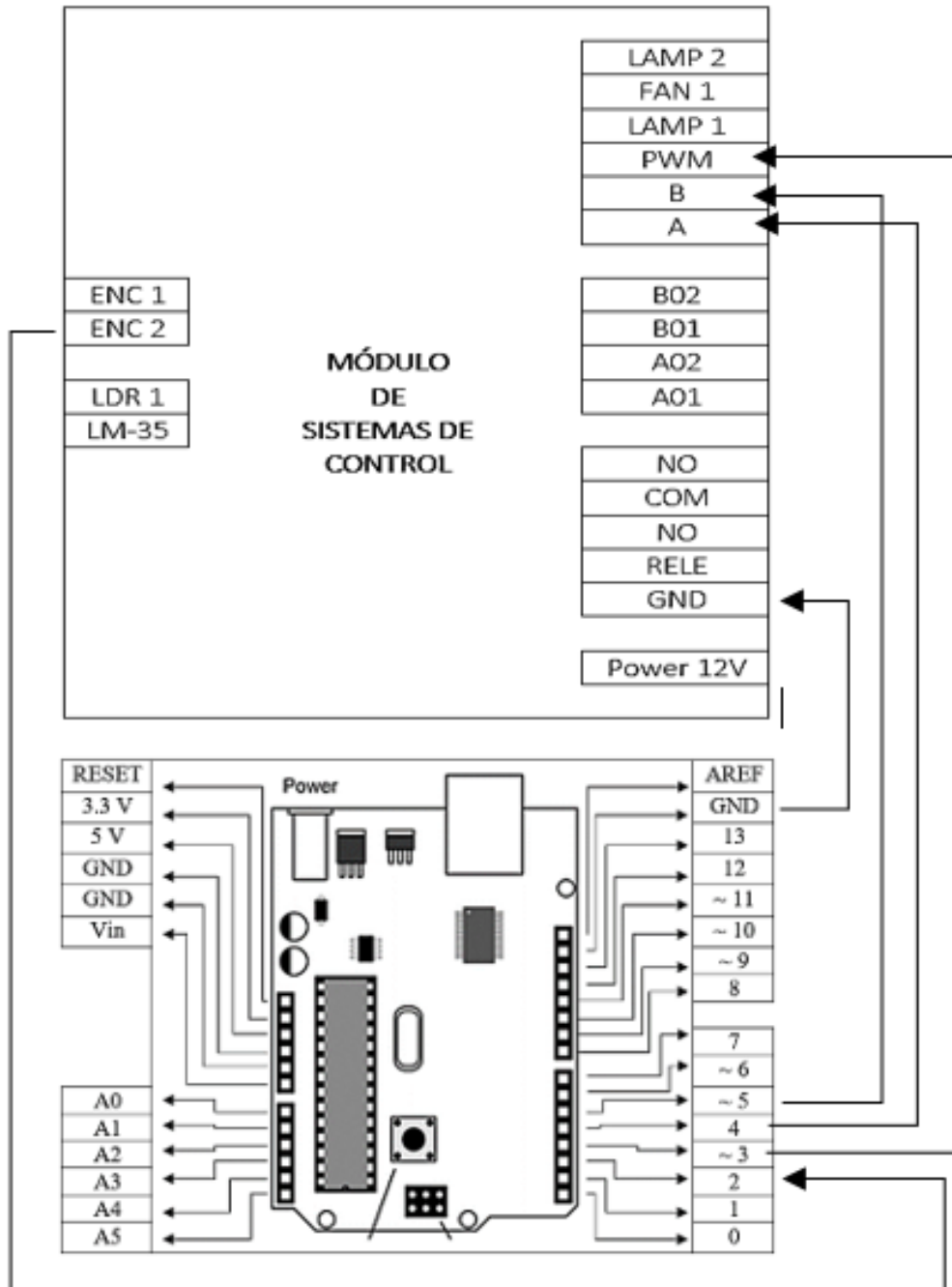


Figura 2.4. Control PID de control de motor DC, panel Frontal.



Anexo 3 Diagrama de conexión Módulo/Arduino, Control PID de velocidad

Anexo 39	UNIVERSIDAD TÉCNICA DE COTOPAXI	
----------	--	--

PRÁCTICA DE LABORATORIO

CARRERA	CÓDIGO DE LA ASIGNATURA	NOMBRE DE LA ASIGNATURA
INGENIERÍA ELECTROMECAÁNICA	ILM702M2	SISTEMAS DE CONTROL

PRÁCTICA N°	LABORATORIO:		DURACIÓN (HORAS)
04	NOMBRE DE LA PRÁCTICA:	Control RPM motor DC	3

1	OBJETIVOS
	<ul style="list-style-type: none"> • Cargar el código del protocolo “StandardFirmata” a la tarjeta Arduino • Conectar el módulo a las salida y entradas; análogas y digitales de la tarjeta de adquisición (Arduino) • Recopilar datos del comportamiento del Sistema de Control y realizar la gráfica. • Analizar la forma de onda resultante del control.

2	FUNDAMENTACIÓN TEÓRICA
	<p>2.1. INTRODUCCIÓN:</p> <p>El Laboratorio de Sistemas de control cuenta con equipos para realizar de prácticas, a través de las cuales se pretende, complementar el estudio teórico con el práctico y crear en el estudiante habilidades para el diseño e implementación de Sistemas de control, generalmente utilizados en plantas industriales, procesos de producción, en la automatización de procesos, etc. El término control hace referencia a los métodos y maneras de gobernar el comportamiento de un aparato, máquina o proceso. En un sistema de control, hay varios</p>

parámetros de entrada que actúan sobre otros parámetros de salida, organizados en un orden lógico de control.

La adquisición de datos y señales consiste en la toma de muestras de sistemas análogos (mundo real) con el fin de generar datos que puede ser manipulados, modificados e interpretados por un ordenador o sistema digital. También poder definirse como: tomar un conjunto de señales físicas y convertirlas en tensión y corriente eléctrica y digitalizarlas para ser procesados por un computador. Para todo ello nos enfocaremos en control de cambio de giro de un motor DC para su posterior funcionamiento y eficacia.

2.2 EQUIPO, INSTRUMENTOS Y MATERIALES NECESARIOS

- Módulo de Sistemas de Control
- Fuente de 12 voltios.
- Destornillador plano tipo bornera
- Cables Jumper
- Computadora

2.3 SOFTWARE REQUERIDO

- MyOpenLab
- Arduino
(revisar manual de uso, para su instalación)

2.4 MEDIDAS DE SEGURIDAD

- Revisar el manual de uso del módulo de sistemas de control
- Realizar la conexión de la tarjeta Arduino con el módulo des-energizado y cuando los leds indicadores no estén encendidos

2.5 ACTIVIDADES A DESARROLLAR:

2.5.1 Cargar el código del protocolo “StandardFirmata” a la tarjeta Arduino, siguiendo el siguiente enlace: [Abrir Arduino/Archivo/Ejemplos/Firmata/Estándar Firmata](#), como se muestra en la figura 2.1

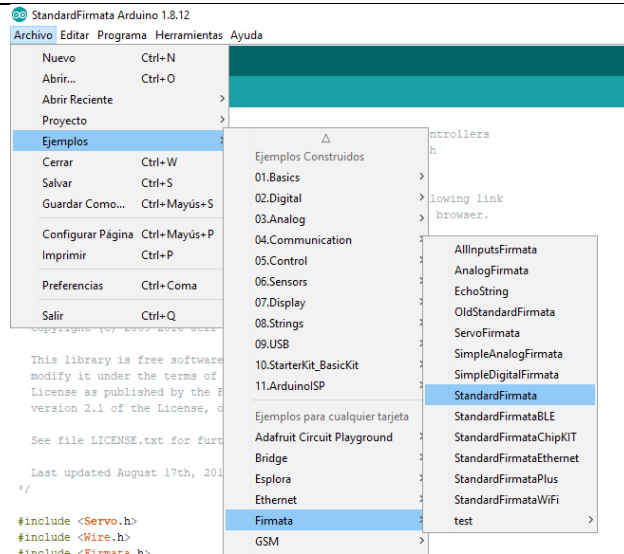


Figura 2.1. Guía para cargar el protocolo “StandardFirmata”

2.5.1.1 Se desplegará el código del protocolo StandardFirmata, sobre la pantalla, la tarjeta debe estar conectada previamente, se da clic en verificar, para revisar que no exista errores en el código y a continuación se da clic en Subir, si la ventana de comandos inferior cambia a naranja (error) revisar el puerto de comunicación de la tarjeta Arduino.

2.5.2 Abrir el programa MyOpenLab y crear un nuevo proyecto.

En la parte superior de la ventana de MyOpenLab, se debe seguir el siguiente enlace para crear un nuevo proyecto: Fichero/Nuevo Proyecto, dentro de la ventana “Nuevo Proyecto” se puede cambiar el nombre del proyecto y seleccionar la ubicación; como se muestra en la siguiente figura.

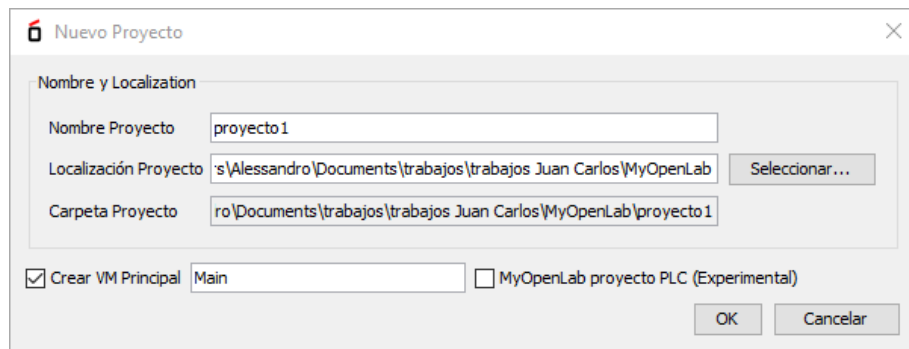


Figura 2.2 Ventana “Nuevo Proyecto”

2.5.3 Repicar el código de programación que se muestra en el **Anexo 1**, los pines de entrada y salida del Firmata IO Interface son:

- **1 Analog_Input:** ENC 2 (pin 14, es necesario habilitar este por la programación, ya que la señal del encoder 2 (señal 0,1) entra por el pin 2, el StandardFirmata_Editado acondiciona la señal y la entrega en el Pin 14 como señal (int) dentro de MyOpenLab).
- **2 Digital_Output:** A, B (cambio de giro) pines 4 y5
- **1 PWM_Ouput:** PWM, pin 3

Para configurar los pines se debe dar clic sobre el “Firmata IO Interface”, aparecerá una ventana en la esquina inferior “Componentes”, al final de la ventana se debe dar clic sobre los 3 puntos (...) en “Config Pins” a continuación, se desplegará una ventana donde se poder habilitar cada una de las entradas y salidas deseadas como se muestra en la siguiente imagen.

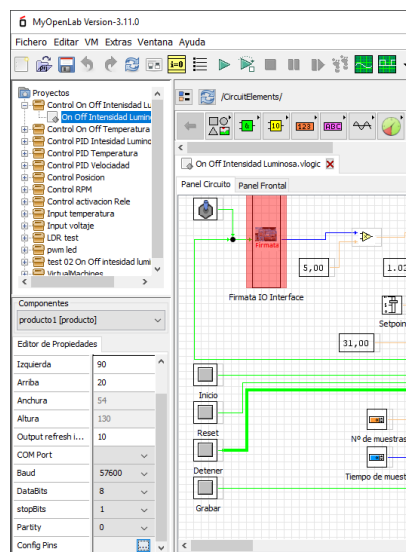


Figura 2.3 configuración de los pines de Firmata IO Interface

Antes de programar, la tarjeta Arduino debe estar conectada, para que MyOpenLab la reconozca automáticamente. Se debe editar el código del StandardFirmata, para transforma la señal del encoder 2 (dato booleano) a dato entero (int), esta son las modificaciones del código:

La modificación del código estándar Firmata se encuentre en la guía de laboratorio para PID de Velocidad

Si no se logra modificar el código del protocolo StandardFirmata, se puede copiar el código del siguiente link:

https://create.arduino.cc/editor/juan_carlos_capa/331e9f20-6d20-4686-ba00-83134384e82a/preview

Comunicación CdN. Logic

Para enviar la función de cambio de giro en el motor DC trabajamos con un CdN es decir una programación interna dentro de la interfaz de MyOpenLab, la cual permite utilizar constantes booleanas en falso o verdadero como podemos ver en la **Anexo 2** Esto a su vez enviando la señal de control PWM para la velocidad en medio del cambio de giro.

En el **Anexo 3** se visualiza la apariencia del control de RPM

2.5.7 Se debe conectar con el módulo des-energizado, al lado izquierdo de módulo se encuentran los sensores que deben conectarse a las entradas análogas de la tarjeta Arduino, y al lado derecho se encuentran dispuestos los actuadores; en el **Anexo 4** se visualiza un ejemplo de conexión del control RPM del motor DC

- ENC 2, se debe conectar en el pin 2 (por la programación)
- PWM, se conecta en el pin 3
- A y B, se conecta en el pin 4 y 5

INFORME DE PRÁCTICA

3 RESULTADOS Y DISCUSIÓN

Para realizar el análisis del comportamiento del sistema de control, se debe establecer un Setpoint variante para poder observar el control del motor DC y su funcionamiento, se puede observar la manera en la cual se dan los cambios de giro con un pequeño retardo en el cambio.

Todo esto como lo vemos en la Figura 3.1

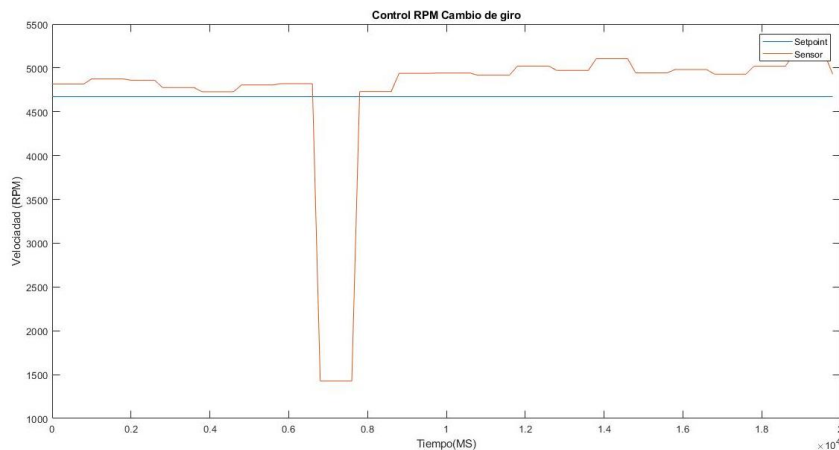


Figura 3.1. Inversión de giro al tiempo de 50

3.6 ANÁLISIS DE RESULTADOS:

-

5 CONCLUSIONES

-

6 RECOMENDACIONES

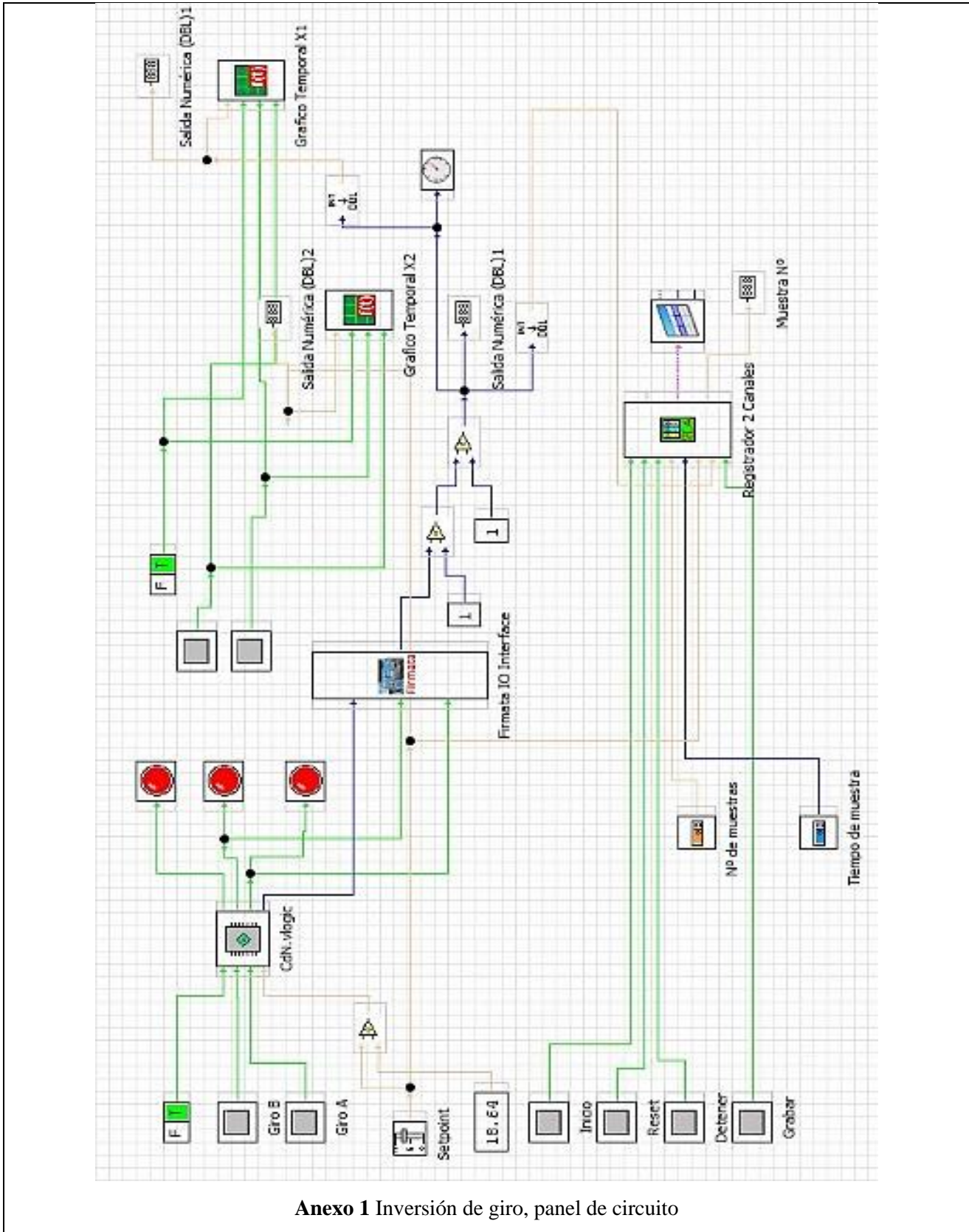
- Para graficar la señal del Sensor y del Setpoint en la misma gráfica como se muestra en la figura 2.3 del panel frontal, se emplea dos elementos “Gráfico Temporal” X1 y X2 en el panel frontal, los dos gráficos deben tener las mismas unidades y dimensiones y no tener fondo (activar Background Transparent en las propiedades de la gráfica), para interpolar una gráfica sobre la otra.
-

7 REFERENCIAS BIBLIOGRÁFICAS (Utilizar las normas IEEE)

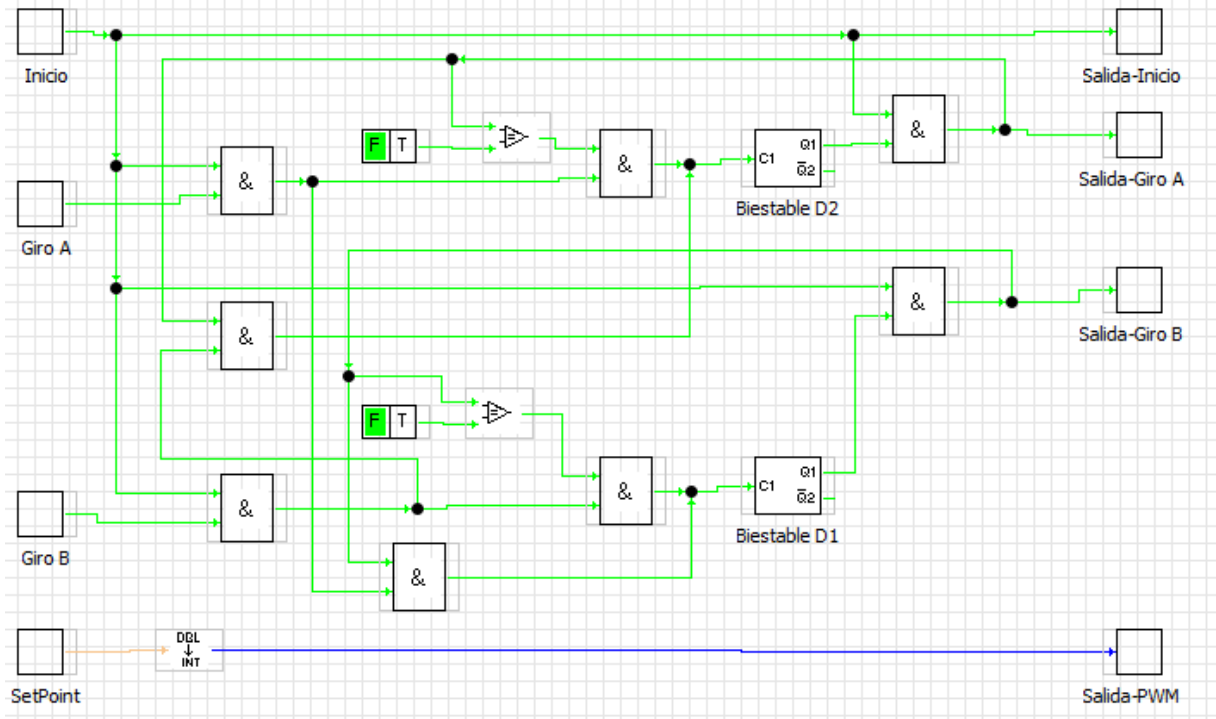
METODOLOGÍA

- [1] Marcelo Diez, “Tipos de control,” *Controlador de comunicación Cdn y compuertas*, Jul. 02, 2010. <http://educativa.catedu.es/44700165/aula/archivos/repositorio/4750/4926/html/13> (accessed Aug. 23, 2020).

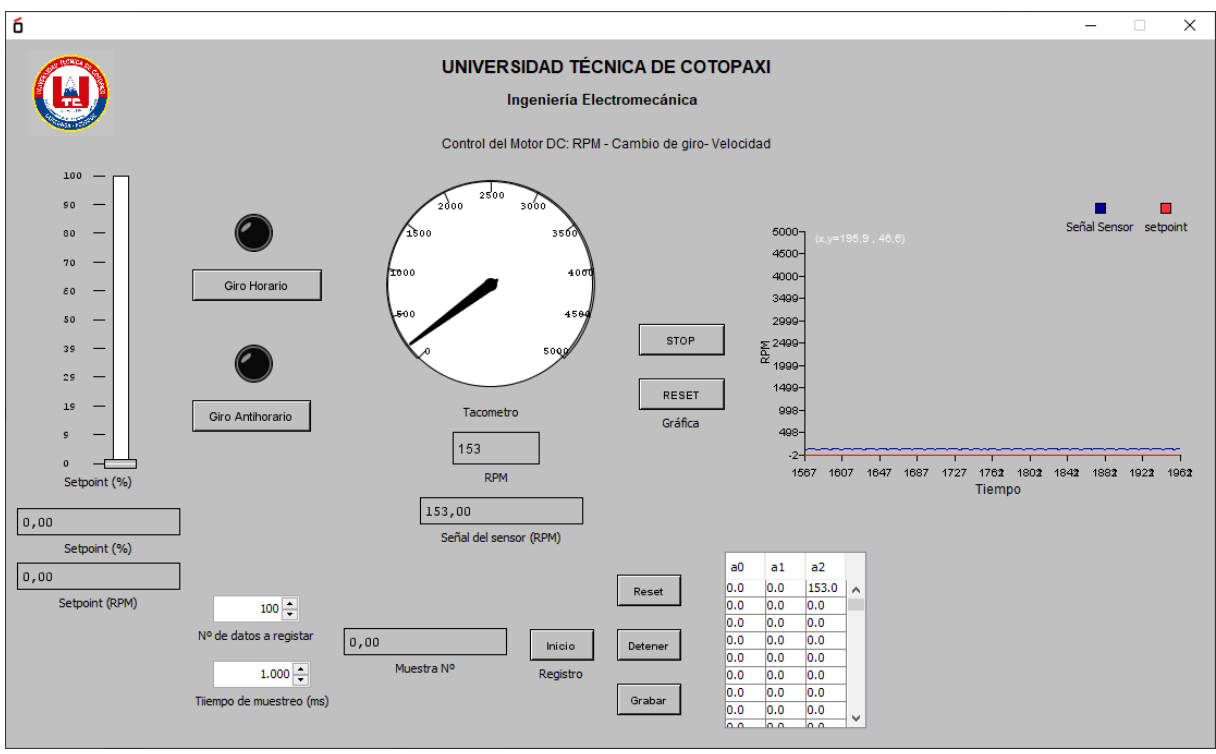
8 ANEXOS



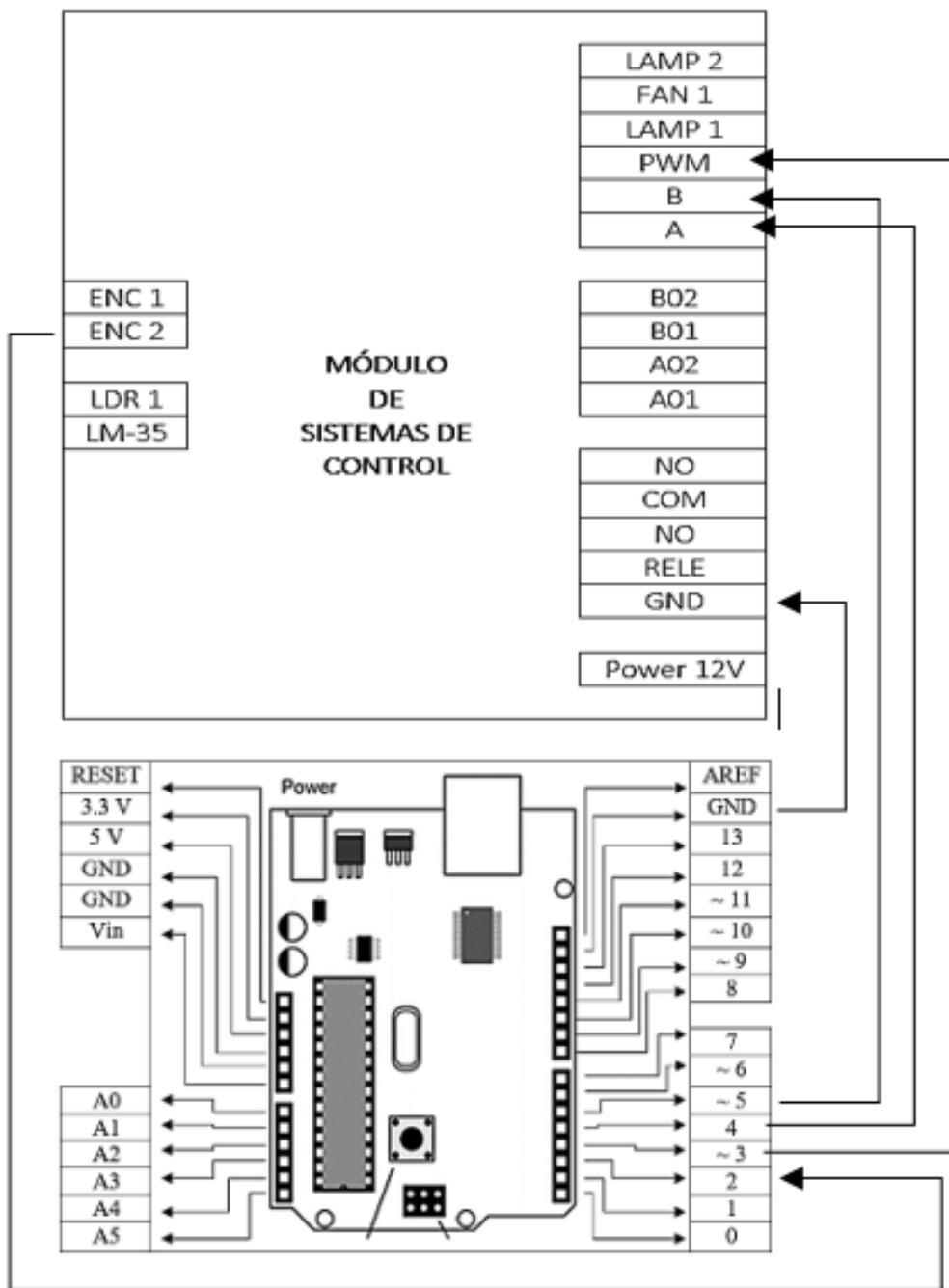
Anexo 1 Inversión de giro, panel de circuito



Anexo 2 Programación CdN. Logic



Anexo 3 Control PID de control de motor DC, panel Frontal.



Anexo 4 Diagrama de conexión Módulo/Arduino, Control de cambio de giro

Anexo 40	UNIVERSIDAD TÉCNICA DE COTOPAXI	
PRÁCTICA DE LABORATORIO		

CARRERA	CÓDIGO DE LA ASIGNATURA	NOMBRE DE LA ASIGNATURA
INGENIERÍA ELECTROMECAÁNICA	ILM702M2	SISTEMAS DE CONTROL

PRÁCTICA N°	LABORATORIO:	Sistemas de Control	DURACIÓN (HORAS)
04	NOMBRE DE LA PRÁCTICA:	Control de Posición	3

1	OBJETIVOS
	<ul style="list-style-type: none"> • Cargar el código del protocolo “StandardFirmata” a la tarjeta Arduino • Conectar el módulo a las salida y entradas; análogas y digitales de la tarjeta de adquisición (Arduino) • Recopilar datos del comportamiento del Sistema de Control y realizar la gráfica. • Analizar la forma de onda resultante del control.

2	FUNDAMENTACIÓN TEÓRICA
	<p>2.1. INTRODUCCIÓN:</p> <p>El Laboratorio de Sistemas de control cuenta con equipos para realizar de prácticas, a través de las cuales se pretende, complementar el estudio teórico con el práctico y crear en el estudiante habilidades para el diseño e implementación de Sistemas de control, generalmente utilizados en plantas industriales, procesos de producción, en la automatización de procesos, etc. El término control hace referencia a los métodos y maneras de gobernar el</p>

comportamiento de un aparato, máquina o proceso. En un sistema de control, hay varios parámetros de entrada que actúan sobre otros parámetros de salida, organizados en un orden lógico de control.

La manera en la cual vamos a determinar cada uno de los pasos que va a tener el motor step serán mediante tablas y matrices configuradas desde MyOpenLab y la respectiva comunicación con Arduino.

2.2 EQUIPO, INSTRUMENTOS Y MATERIALES NECESARIOS

- Módulo de Sistemas de Control
- Fuente de 12 voltios.
- Destornillador plano tipo bornera
- Cables Jumper
- Computadora

2.3 SOFTWARE REQUERIDO

- MyOpenLab
- Arduino
(revisar manual de uso, para su instalación)

2.4 MEDIDAS DE SEGURIDAD

- Revisar el manual de uso del módulo de sistemas de control
- Realizar la conexión de la tarjeta Arduino con el módulo des-energizado y cuando los leds indicadores no estén encendidos

2.5 ACTIVIDADES A DESARROLLAR:

2.5.1 Cargar el código del protocolo “StandardFirmata” a la tarjeta Arduino, siguiendo el siguiente enlace: [Abrir Arduino/Archivo/Ejemplos/Firmata/Estándar Firmata](#), como se muestra en la figura 2.1

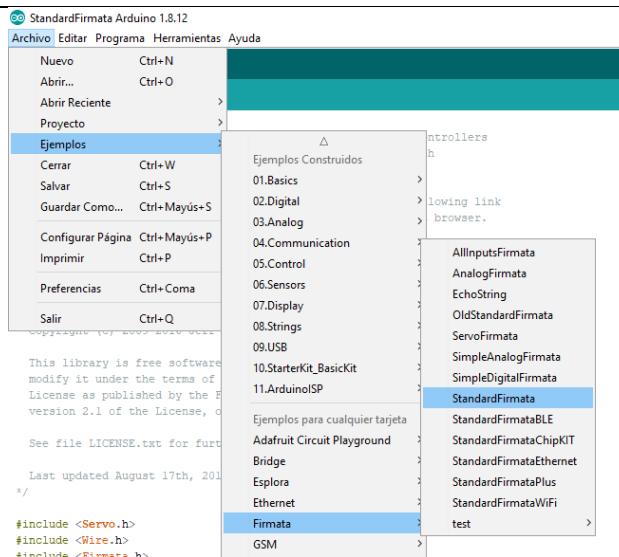


Figura 2.1. Guía para cargar el protocolo “StandardFirmata”

2.5.1.1 Se desplegará el código del protocolo StandardFirmata, sobre la pantalla, la tarjeta debe estar conectada previamente, clic en verificar para revisar que no exista errores en el código y a continuación se da clic en Subir, si la ventana de comandos inferior cambia a naranja (error) revisar el puerto de comunicación de la tarjeta Arduino.

2.5.2 Abrir el programa MyOpenLab y crear un nuevo proyecto.

2.5.3 Replicar el código de programación que se muestra en el anexo 1, los pines de entrada y salida del Firmata IO Interface son:

- **INPUT: Pin 2 señal de encoder**
- **OUTPUT: 5 – 6 – 9 – 10 (dos bobinas 4 hilos bipolar)**
- **GND del Arduino y GND del Módulo.**

Antes de programar, la tarjeta Arduino debe estar conectada, para que MyOpenLab la reconozca automáticamente. Tener muy en cuenta este punto ya que si no se encuentra conectada la tarjeta Arduino o a su vez reconociendo el puerto de comunicación tendremos errores en el arranque del software.

2.5.4 Interfaz de uso de MyOpenLab

Esta fue diseñada con una tabla en la cual mostramos los ángulos a ser establecidos a través de un setpoint y la comparación de los ángulos dados y los pasos que da el motor, tomamos como referencia 0° y 360° a los 48 pasos que tiene el motor step.

Aquí podemos tener el control de los ángulos a disposición de un setpoint de muestra, anexo a esto tenemos una tabla en la interfaz de MyOpenLab en donde nos muestra los grados y números de pasos que nuestro motor da.

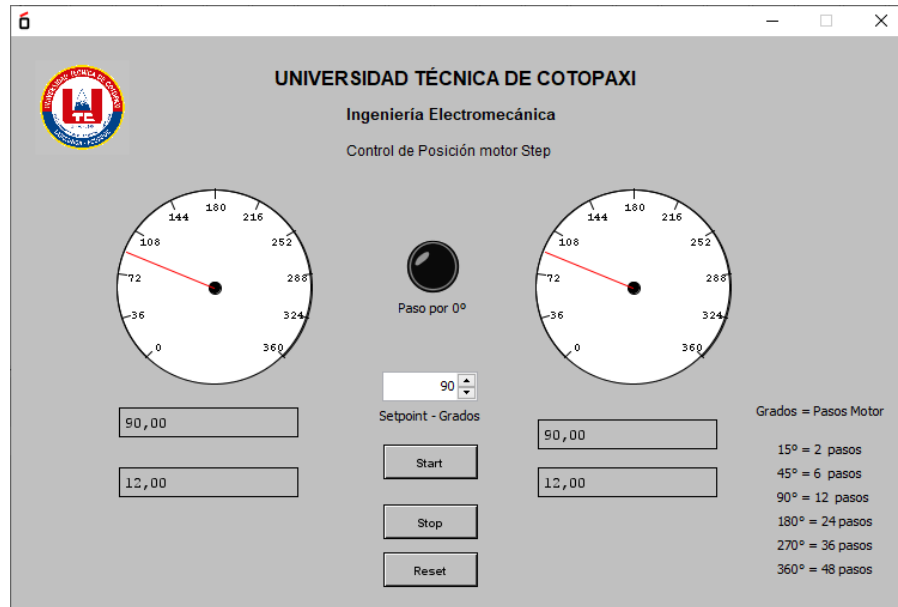


Figura 2.2. Interfaz de control de motor a pasos

2.6. METODOLOGÍA Y TÉCNICA EXPERIMENTAL

Para poder tener la visualización y recorrido de pasos con este motor se procedió con el uso de matrices conociendo los pasos del motor y la configuración que este tenga, en este caso tenemos un motor step bipolar de 48 pasos. En la figura 2.3 y 2.4 podemos observar la matriz a utilizarse y que se entiende por esto.

2.7. MOTOR BIPOLAR.

Para hacer girar un motor paso a paso bipolar, se aplican impulsos en secuencia los devanados, la secuencia de estos impulsos, se aplican externamente con un controlador electrónico. Dichos controladores, se diseñan de manera que el motor se pueda mantener en una posición fija y también para que se le pueda hacer girar en ambos sentidos. Los motores bipolares, se pueden hacer avanzar a frecuencias generales de audio, lo que les permite girar muy velozmente en cada uno de sus comandos. Este es el motivo por que se suele decir que un motor “acelera”, debido a la frecuencia a la que se produce la conmutación. Con un controlador apropiado, se les puede hacer arrancar y detenerse en cualquier instante y en una posición determinada.

Estos motores, tienen varios bobinados que, para producir el avance de un paso, deben ser alimentados en una secuencia adecuada. Al invertir el orden de esta secuencia, se

logra que el motor gire en sentido opuesto. El torque de detención, hace que un motor paso a paso (esto vale para los bipolares y unipolares) se mantenga firmemente en su posición, estando alimentado aun cuando no esté girando.

Hay tres características que, son comunes a los motores paso a paso:

- Voltaje. Los motores paso a paso tienen una tensión eléctrica de trabajo. Este valor viene impreso en su carcasa o por lo menos se especifica en su hoja de datos.
- Resistencia eléctrica. La resistencia de los bobinados, esta resistencia determinará la corriente que consumirá el motor y su valor afecta la curva de torque del motor y su velocidad máxima de operación.
- Grados por paso. Generalmente, este es el factor más importante al elegir un motor paso a paso para un uso determinado. Este factor define la cantidad de grados que rotará el eje para cada paso completo.

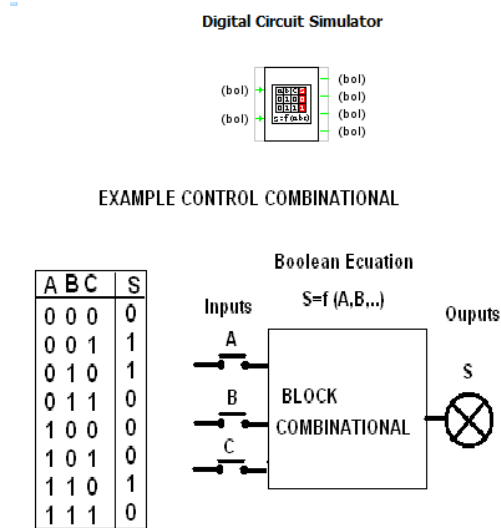


Figura 2.3 Simulador de circuito, este nos da la configuración de activación de las bobinas para los pasos del motor

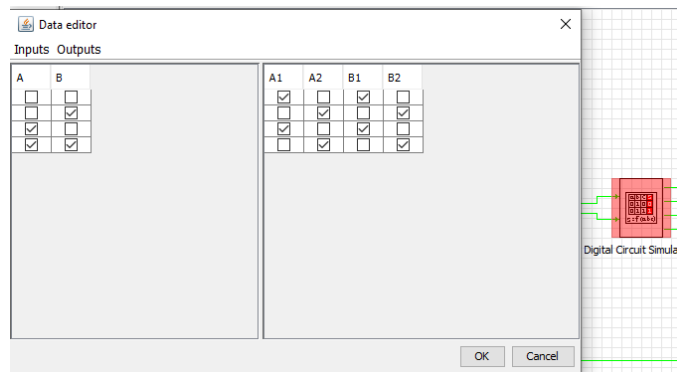
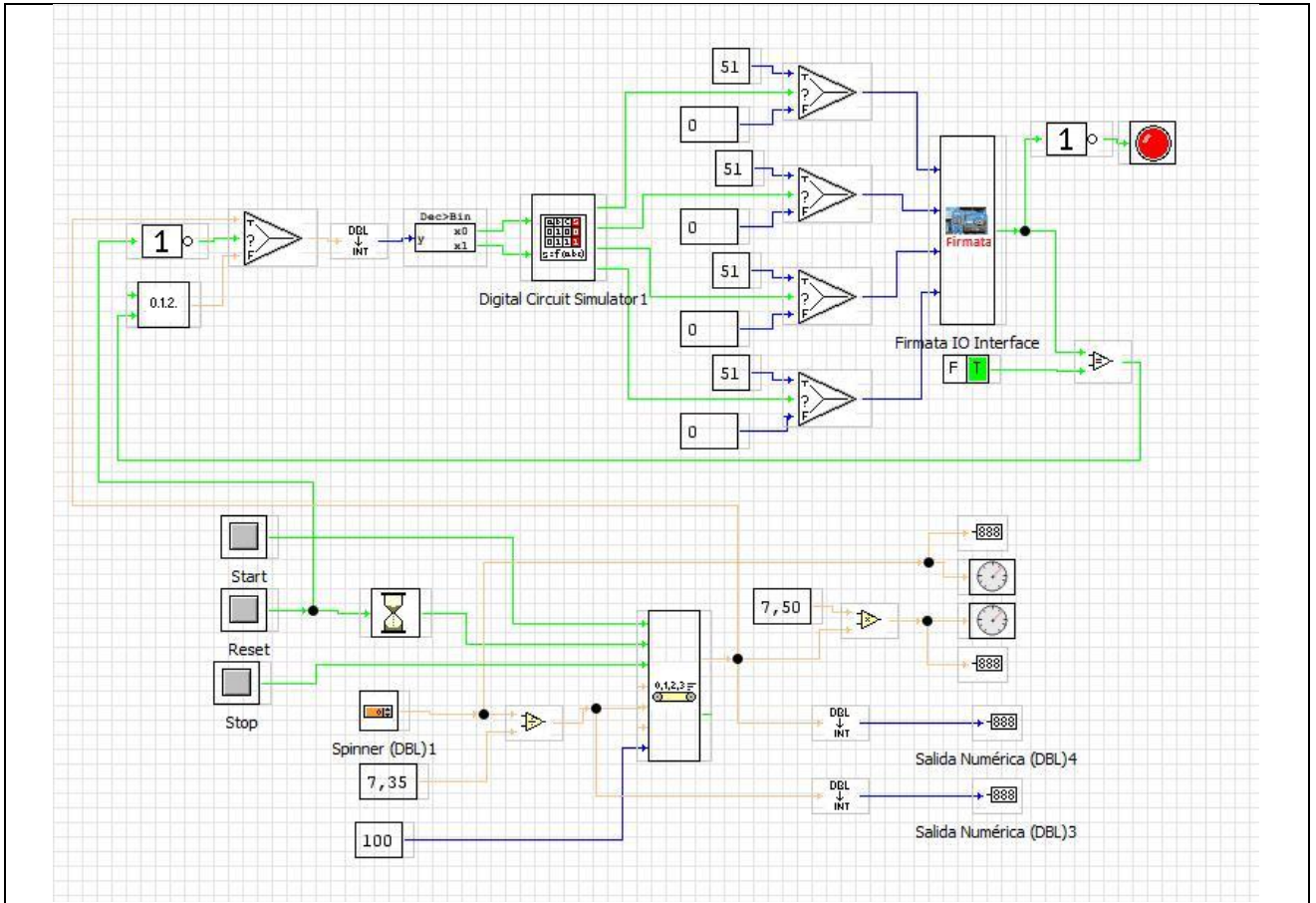


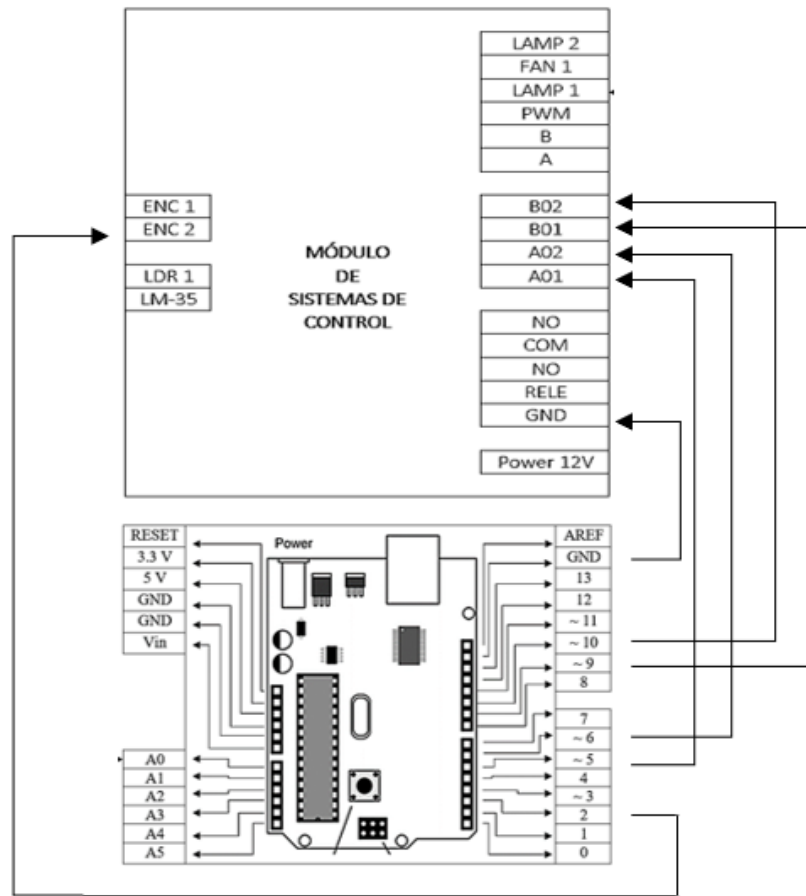
Figura 2.4 Matriz de configuración de activación de bobinas

INFORME DE PRÁCTICA

3	RESULTADOS Y DISCUSIÓN
<p>Aquí se observa el giro tanto para 45°, 90°, 180°, 270°, 360°, se puede notar que el Angulo de fallo respecto a la precisión es mínimo.</p> <p>3.7 ANÁLISIS DE RESULTADOS:</p> <ul style="list-style-type: none">•	
5	CONCLUSIONES
<ul style="list-style-type: none">•	
6	RECOMENDACIONES
<ul style="list-style-type: none">• Se debe tener en cuenta que hay que disponer de los datos precisos en la matriz para no tener fallos al momento de la activacion de cada bobina y el giro de posicion que se le vaya a dar.	
7	REFERENCIAS BIBLIOGRÁFICAS (Utilizar las normas IEEE)
METODOLOGÍA	
8	ANEXOS

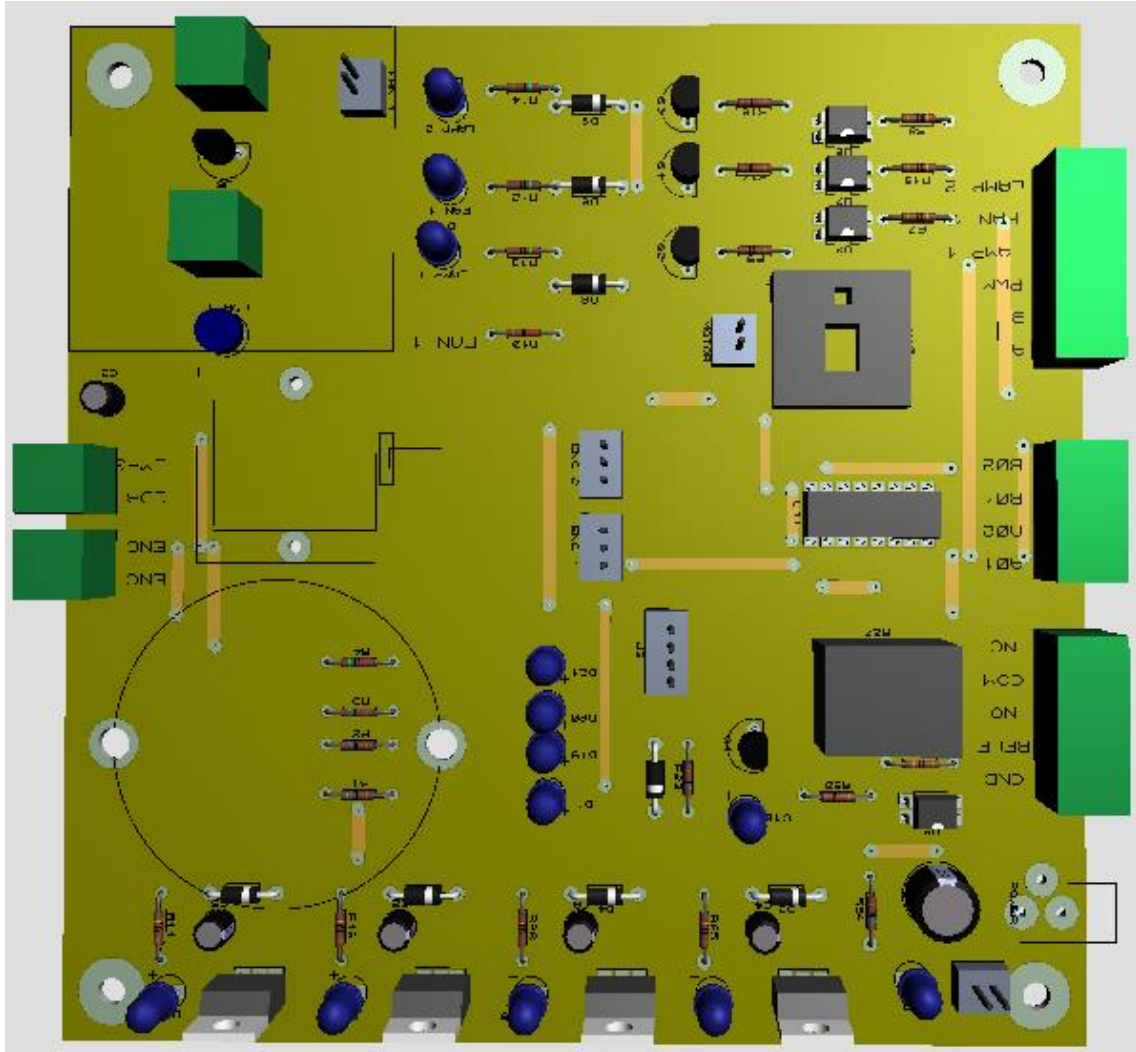


Anexo 1 Control de motor a pasos, panel de circuito



Anexo 2 Disposición de pines

Manual de mantenimiento del módulo



Antes de manipular el módulo se debe leer el presente manual de mantenimiento

1. Descripción

El presente manual tipo como objetivo el proporcionare información necesaria en caso de avería y daño de algún componente con el fin de cambiar el componente por otro y así solucionar el problema.

Este manual este diseño para apoyar en el mantenimiento y reparación, para conserva la vida útil del equipo, este manual dese ser apoyado con los data-sheet de cada uno de los componentes que serán mencionados a continuación. El mantenimiento dese ser llevado por una persona autorizada y capacitado con el condominito técnico de electrónico

2. Listado de componentes

Componentes de Módulo de Sistemas de Control			
Título del diseño		Diagrama módulo de Sistemas de Control	
Número de documento		001	
Creado del diseño		domingo, 14 de junio de 2020	
Última modificación del diseño		lunes, 24 de agosto de 2020	
Componentes en el diseño		81	
Categoría	Cantidad	Nomenclatura	Valor
Condensadores	1	C1	1000uF
	4	C2, C4-C6	100uF
	1	C3	6 uF
Resistencias	9	R1-R4, R12-R14, R23-R24	560
	8	R7-R8, R11, R15-R16, R22, R25-R26	300
	3	R9, R17-R18	1k
	1	R10	220
	1	R20	1K
	1	R27	1-1721150-3
Integrados	4	U1-U2, U4-U5	7805

	4	U3, U7-U9	PC817
	1	U6	LM35
	1	U10	ROB-14450
	1	U11	L293D
Transistores	4	Q1-Q4	2N2222
Diodos	1	D1	O1
	8	D2, D4-D6, D8-D10, D15	1N4007
	1	D3	VDD3 5V
	1	D7	VDD1 5V
	1	D11	VDD2 5V
	1	D12	FAN 1
	1	D13	LAMP 1
	1	D14	LAMP 2
	1	D16	RELE
	1	D17	FUENTE
	1	D18	VDD4 5V
	1	D19	O2
	1	D20	O3
	1	D21	O4
Miscelánea	1	ENC 1	SIL-100-03
	1	ENC 2	ENCODER 2
	1	F1	fusible
	4	FAN 1, LAMP 1-LAMP 2, POWER	12V
	1	J1	TBLOCK-I5
	1	J2	TBLOCK-I4
	1	J3	MOTOR STEP
	2	J4-J5	TBLOCK-I3
	2	J7-J8	TBLOCK-I2

	1	LDR 1	TORCH_LDR
	1	MOTOR	MOTOR DC

3. Mascarilla de componentes

En la siguientes grafica se puede visualizar la posición de los componentes sobre la placa sobre la placa de circuito integrado y el diseño de la pista y ruteo de cada componente

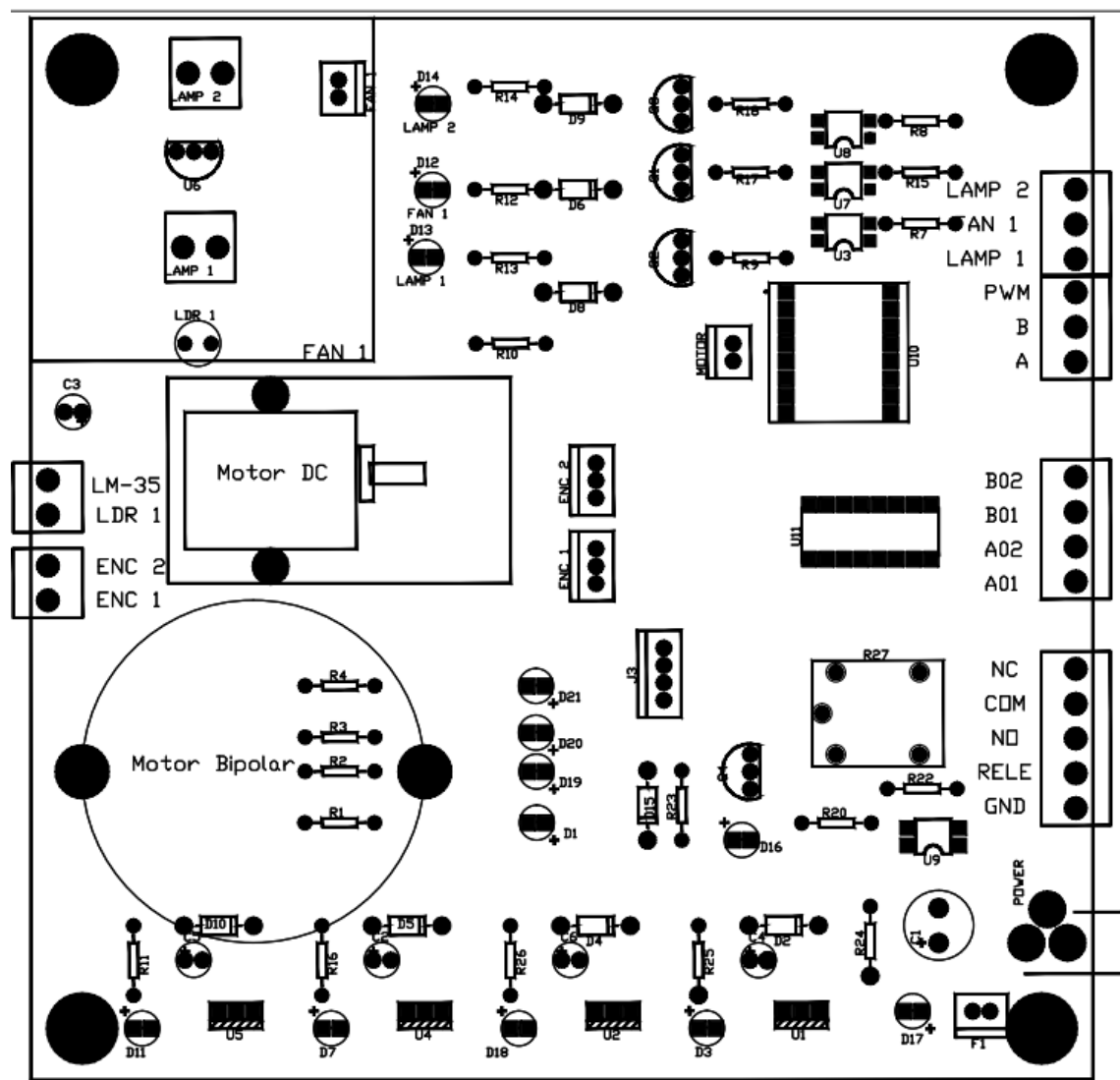


Figura 1. Mascarilla de componentes

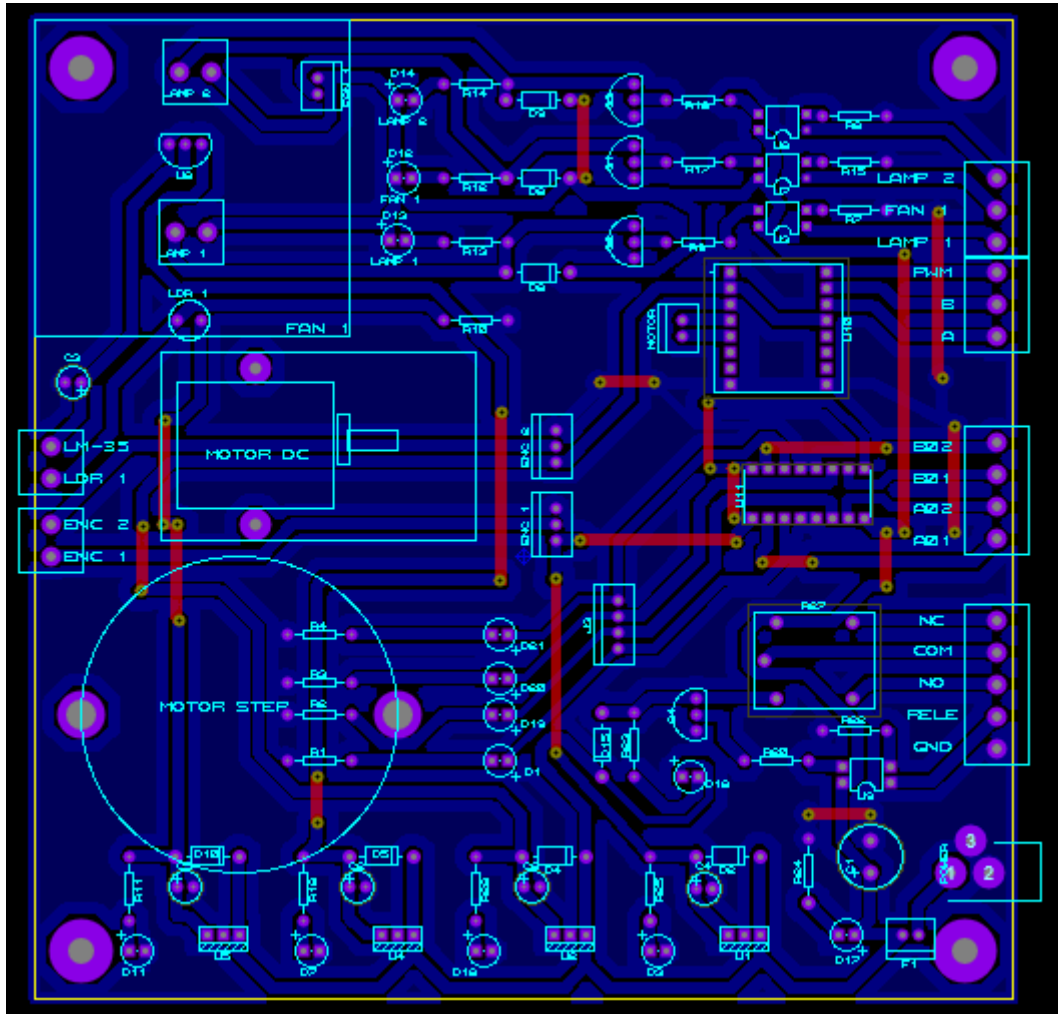


Figura 2. Diseño de las pistas sobre la placa de circuito integrado

4. Descripción del esquema eléctrico

Diseño de la alimentación y filtrado de tensión para el módulo

Se trabajará con dos niveles de tensión, una a 5 voltios para la transmisión de la señal de los sensores de temperatura, LDR, encoders, hacia la tarjeta de adquisición de datos, así como los circuitos integrados y driver de control de los actuadores; y otro voltaje de 12

voltios para potencia y la activación de los actuadores como lámpara, ventilador, motor DC, motor paso a paso, etc.

En la figura 4.1 se observa el diagrama electrónico de los componentes utilizados para regular el voltaje tanto de 5V. y 12V.; a continuación, se describen la característica de los componentes:

- LM-7805 es un regulador de voltaje a 5 V. y 1 A. a la salida. Se utiliza un regulador por control para evitar ruido eléctrico en la señal generada por los sensores.
- Condensadores de 1000 uf. En la línea de 12 V. y 100 uf. En las líneas de 5 V.
- Diodo rectificador 1N4007 polarizado inversamente en los pines 1 y 3 de LM-7805.
- Resistencia de 220 Ω conectado a un diodo led indicador de encendido y apagado.

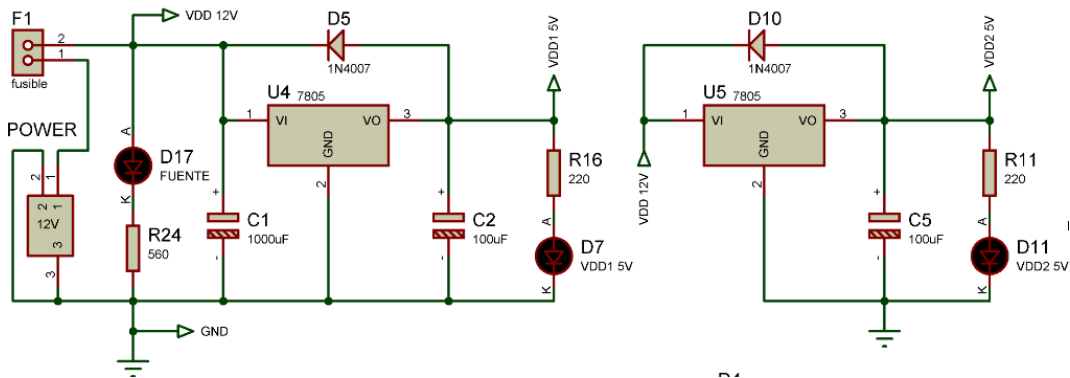


Figura 3. Diagrama eléctrico de la alimentación de 5 V. y 12 V.

Diseño del sensor y control de temperatura

El LM-35 es un sensor de temperatura. Con rango de medición desde -55°C hasta 150°C esta calibrado para trabajar en grados Celsius, la tensión de salida es proporcional a la temperatura, con una precisión de 0.5°C a 25°C con una corriente baja de alimentación (60 uA.) y una alimentación de 4 V. hasta 20 V [35]. El LM-35 trabaja con dos configuraciones como sensor de rango completo de temperatura (-55°C hasta 150°C) y Sensor básico de temperatura (2°C hasta 150°C).

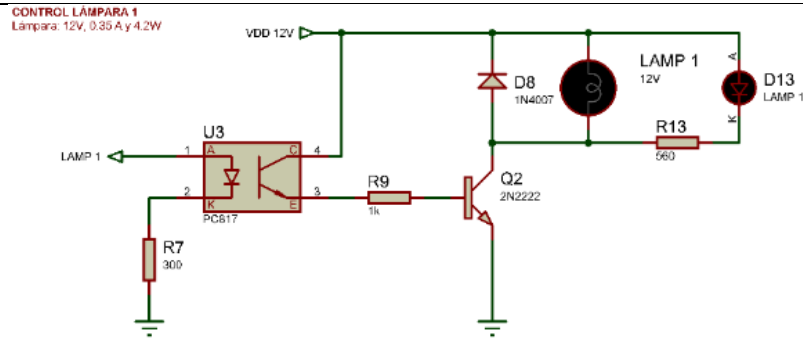


Figura 5. Diagrama control de lámparas

El diagrama electrónico es similar para las dos lámparas ya que poseen las mismas características:

- Lámpara incandescente de 12 V.
- Corriente medida de 0.35 A.
- Y un consumo de 4.2 W.

Diseño del control del ventilador

Es necesario utilizar un actuador que acelere la caída de temperatura generada por las lámparas incandescentes.

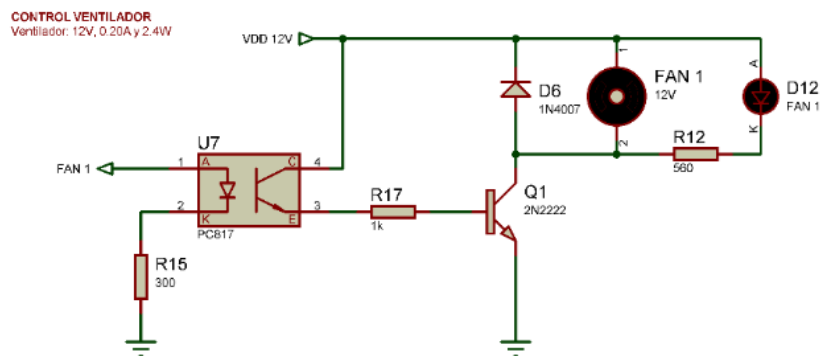


Figura 6 Diagrama control del ventilador

En el diagrama se puede apreciar un ventilador de 30 x 30 mm. Con un voltaje de funcionamiento de 12 V. y un consumo de 0.2 A.

Diseño del sensor y control de intensidad luminosa

El Light Dependent Resistor o LDR es un resistor que varía su valor de resistencia eléctrica dependiendo de la intensidad de luz que intensidad sobre su fotorresistencia, la

resistencia del LDR es baja cuando la intensidad de luz es alta y en ausencia de luz la resistencia varía en los mega-ohmios $M\Omega$.

- EL LDR tiene una tensión máxima a 25 °C igual a 150 VDC.
- Una potencia disipable de 90 mW.
- Resistencia de 50 Ω a 100 ΩK .
- Temperatura ambiente desde -25 a 75 °C [38].

Para que el LDR funcione como sensor de intensidad luminosa se debe emplear un divisor de voltaje que es un circuito de resistencias en serie el voltaje de salida es una fracción del voltaje de entrada.

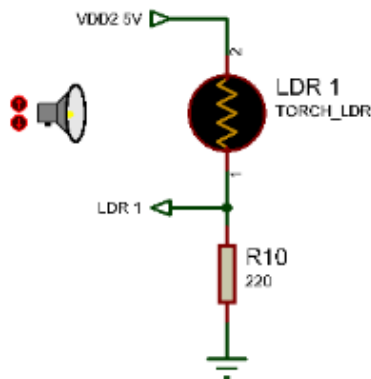


Figura 7 Diagrama sensor de intensidad luminosa

Para determinar el voltaje de salida en el pin LDR 1 se emplea la siguiente ecuación:

El control de intensidad luminosa se utilizará el mismo circuito de las lámparas incandescentes, ya que nos brinda las mismas características necesarias para ambos controles.

Diseño del control del motor DC

El TB6612FNG de motores que nos permite manejar dos motores de corriente continua desde Arduino, variando tanto la velocidad como el sentido de giro. En este caso lo usaremos para controlar un solo motor y direccionar las aplicaciones antes mencionadas a este. Este controlador se encarga principalmente del funcionamiento correcto del motor dc. El TB6612FNG también permite controlar intensidades de corriente altas, siendo capaz de suministrar 1.2A por canal de forma continua, y 3.2A de pico. En cada canal podemos controlar el sentido de giro y la velocidad, para lo cual admite una señal PWM

de frecuencia máxima de 100 kHz (muy por debajo del rango normal de PWM en Arduino). Con un voltaje de 5v y a una corriente de arranque de 1300 mA.

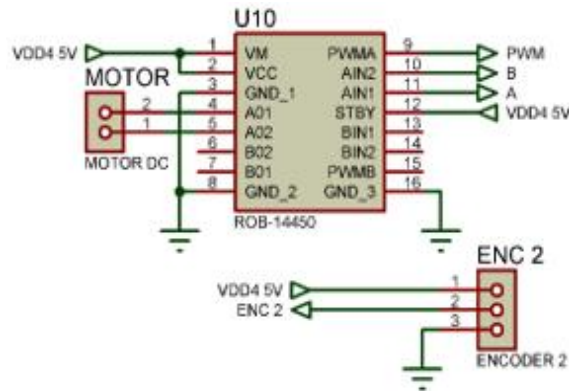


Figura 8 Diagrama de control del motor DC

Diseño del control de motor Step o motor paso a paso

El L293D es un driver de 4 canales capaz de proporcionar una corriente de salida de hasta 600mA por canal y puede soportar picos de hasta 1.2 A. Cada canal es controlado por señales TTL y cada pareja de canales dispone de una señal de habilitación para conectar o desconecta las salidas de los mismos. En esencia utilizamos este integrado para el control del motor a pasos siendo factible con sus características de brindadas en voltaje e intensidad.

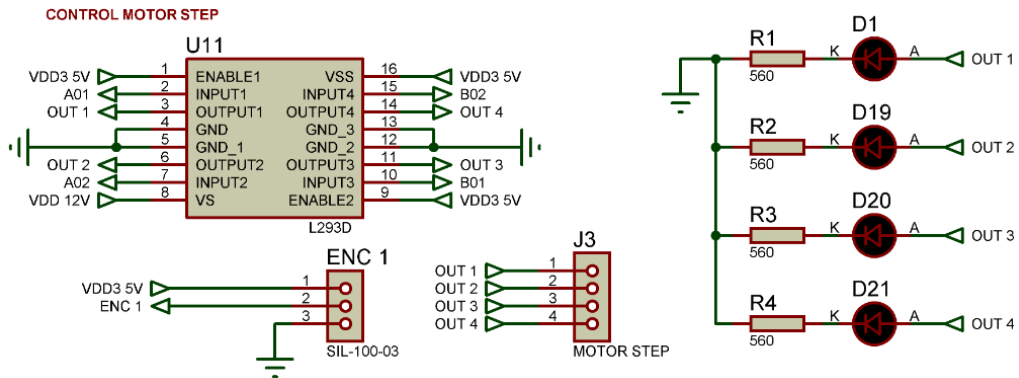


Figura 9 Diagrama de control del motor a pasos bipolar

Diseño de circuito del encordar óptico para los motores

En este caso usamos el módulo sensor de velocidad IR con el comparador LM393 podemos calcular la velocidad de rotación de cada uno de los motores tanto a pasos como el motor dc. Si colocamos una corona dentada que gira unida a nuestra rueda. También se

podría usar como un interruptor óptico, así es como detectamos las RPM en cada uno de los motores. El funcionamiento básico de este sensor es el siguiente; Si se hace pasar cualquier cosa entre la ranura del sensor, este crea un pulso digital en el pin D0. Este pulso va de 0V a 5V y es una señal digital TTL.

Diseño del circuito para el control del relé

La finalidad de este control es la de realizar el control de otros procesos que necesiten un sistema de control, el relé permitirá la activación y la desactivación de cualquier actuador, al igual que los controles anteriores se emplea el opto-acoplador PC-817 para el control de alta potencia. las especificaciones del relé.

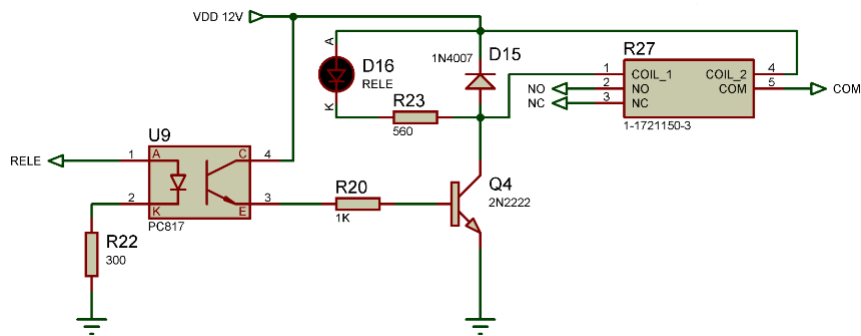


Figura 10 Diagrama del control del relé

5. Posibles fallas y averías

Actuadores		
Componente	Falla o avería	Solución
Lámpara incandescente	No enciende	Revisar y ajusta los terminales de conexión Cambia de lámpara por fundición del tungsteno Revisar el transistor
	Se mantiene encendido	El transistor sufrió un daño que lo dejó en circuito cerrado
	Lámpara titila	Los contactos no están ajustados correctamente
Ventilador	No enciende	Revisar y ajusta los terminales de conexión Revisar el transistor
	Se mantiene encendido	El transistor sufrió un daño, que lo dejó en circuito cerrado
Motor DC	No enciende	Revisar y ajusta los terminales de conexión

		El driver ROB-14450 no está correctamente comentado sobre el socket
	Se mantiene encendido	revisar el driver, talvez se encuentra encorto y deber ser cambiado por uno nuevo
Motor Step	No enciende	Revisar y ajusta los terminales de conexión El driver L293d no está correctamente conectado sobre el socket
	Leds enciende y no el motor	Revisar y ajusta los terminales de conexión Talvez el motor sufrió una avería
Sensores		
Componente	Falla o avería	Solución
LM-35	No da una lectura adecuada	Revisar el led de alimentación, talvez el regulador de tensión sufrió un daño y no está alimentando al sensor
		Si el led de alimentación este encendido, talvez el LM-35 sufrió un dalo, debe ser cambiado
LDR	No da una lectura adecuada	Revisar el led de alimentación, talvez el regulador de tensión sufrió un daño y no está alimentando al sensor
		Si el led de alimentación este encendido, talvez el LDR sufrió un dalo, debe ser cambiado
Encoder 1 y Encoder 2	No da una lectura adecuada	Revisar el led de alimentación, talvez el regulador de tensión sufrió un daño y no está alimentando al sensor Si el led indicador de alineaciones esta encendió y no censa, resisar el conector mólex del encoder