

Diseño de invernadero con sistema de control de temperatura y humedad automatizado mediante PID en Trujillo - Perú**Greenhouse design with automated temperature and humidity control system using PID in Trujillo – Peru****Projeto de estufa com sistema automatizado de controle de temperatura e umidade usando PID em Trujillo - Peru**

Castillo Cabrera Rafael Luis¹, León León Ryan Abraham², Rodríguez Vílchez Jonathan³, Rodríguez Vílchez Jonathan⁴

Resumen

Un sistema invernadero crea un ambiente adecuado para el cultivo de cualquier fruto u hortalizas en un espacio sin importar el clima y la humedad del terreno, es un gran paso para la agricultura controlada, que nos permite obtener los valores de temperatura y humedad de manera óptima para un adecuado crecimiento de los cultivos. En este trabajo, se presenta una alternativa de automatización con base en plataformas abiertas de hardware y software, con los objetivos de disminuir costos y aumentar los rendimientos del cultivo frente a sistemas comerciales, mediante la re-configuración por software Labview para el monitoreo constante y la interfaz de Arduino la cual permite activar el sistema solo cuando sea necesario. Para el desarrollo del hardware se utilizó al Arduino ya que al ser de código abierto permite añadir las funciones que se requiere. El sistema propuesto pretende demostrar el control y uso adecuado del PID para su uso automatizado siendo como eje principal la temperatura en el ambiente. Sin embargo, es necesario una evaluación minuciosa de las cosechas para corroborar las ventajas competitivas del invernadero y sus variables de (Kc, Ti y de Td) del sistema PID creado en Labview para este artículo.

Palabras clave: *Invernadero, Labview, PID, Arduino, Automatización.*

Abstract:

A greenhouse system creates a suitable environment for the cultivation of any fruit or vegetables in a space regardless of the climate and humidity of the land, it is a great step for controlled agriculture, which allows us to obtain the values of temperature and humidity in an optimal way for proper crop growth. In this work, an automation alternative based on open hardware and software platforms is presented, with the objectives of reducing costs and increasing crop yields compared to commercial systems, through the re-configuration by Labview software for constant monitoring, and the Arduino interface which allows activating the system only when necessary. For the hardware development the Arduino was used since being open source allows adding the required functions. The proposed system aims to demonstrate the control and adequate use of the PID for its automated use, the main axis being the temperature in the environment. However, a careful evaluation of the crops is necessary to corroborate the competitive advantages of the greenhouse and its (Kc, Ti and Td) variables of the PID system created in Labview for this article.

Keywords: *Greenhouse, Labview, PID, Arduino, Automation.*

Resumo:

Um sistema de estufa cria um ambiente adequado para o cultivo de qualquer fruta ou hortícola num espaço independentemente do clima e humidade do terreno, é um grande passo para uma agricultura controlada, que nos permite obter os valores de temperatura e humidade ideal para o crescimento adequado das culturas. Neste trabalho, é apresentada uma alternativa de automação baseada em plataformas abertas de hardware e software, com os objetivos de reduzir custos e aumentar a produtividade das culturas em relação aos sistemas comerciais,

¹Escuela de Ingeniería Industrial. Facultad de Ingeniería. Magister. Universidad Privada del Norte. Trujillo, La Libertad, Perú. <https://orcid.org/0000-0001-6804-5852>

²Escuela de Ingeniería Industrial. Facultad de Ingeniería. Magister. Universidad Privada del Norte. Trujillo, La Libertad, Perú. ryan.leon@upn.edu.pe, <https://orcid.org/0000-0002-0599-0141>.

³Escuela de Ingeniería Industrial. Facultad de Ingeniería. Estudiante. Universidad Privada del Norte. Trujillo, La Libertad, Perú. n000136269@upn.pe, <https://orcid.org/0000-0002-5631-1962>

⁴Escuela de Ingeniería Industrial. Facultad de Ingeniería. Estudiante. Universidad Privada del Norte. Trujillo, La Libertad, Perú. n00083745@upn.pe, <https://orcid.org/0000-0001-6692-8440>

através da reconfiguração pelo software Labview para monitoramento constante. ser ativado somente quando necessário. Para o desenvolvimento do hardware, foi utilizado o Arduino, pois, por ser open source, permite adicionar as funções necessárias. O sistema proposto visa demonstrar o controle e uso adequado do PID para seu uso automatizado, tendo como eixo principal a temperatura no ambiente. No entanto, é necessária uma avaliação criteriosa das culturas para corroborar as vantagens competitivas da casa de vegetação e suas variáveis (Kc, Ti e Td) do sistema PID criado em Labview para este artigo.

Palavras-chave: Estufa, Labview, PID, Arduino, Automação.

Introducción

El riego en la industria agrícola es un proceso de suma importancia para el desarrollo del cultivo en el cual está enfocado en tener un ambiente estable el cual no sea muy excesivo o muy reactivo con el consumo del agua, siendo este uno de los recursos que poco a poco ha ido escaseando con el paso del tiempo, por ello se busca el desarrollo de un sistema controlado el cual ayude a la mejora de este problema. Según (Chulde, 2017) el riego por goteo automatizado ayuda con una reducción de hasta el 40% del recurso que es el agua haciendo evidencia el problema actual del riego manual.

La idea de poder crear un ambiente adecuado para el cultivo de cualquier fruto u hortalizas en un espacio sin importar el clima y la humedad del terreno es un gran paso para la agricultura controlada el cual mediante los sistemas actuales de tecnología se pueden aprovechar para un adecuado crecimiento de los cultivos, Según (Perea, 2016), un sistema de manera controlada nos permite obtener los valores de temperatura y humedad de manera óptima, concluyendo que es una herramienta funcional y versátil para estar pendiente al obtener los datos dentro del invernadero.

La implementación del sistema de automatización para invernaderos con control de cultivos de la industria agro comprende cuatro principales factores a tener en cuenta: Temperatura, humedad y riego por goteo; esta implementación se realiza utilizando la plataforma de creación electrónica de código abierto Arduino UNO el cual según (Arduino, 2021), es ideal para el control de distintos controladores y sensores de hasta 16 puertos.

Según (Wheather Spark, 2021) la ciudad de Trujillo una de las ciudades las cuales los veranos son cortos y los inviernos son más largos durante todo el año. La temperatura generalmente varía de 17 °C a 26 °C. El siguiente trabajo busca desarrollar cultivos no aptos para la costa peruana y que se desean desarrollar mediante un invernadero casero, estos pueden variar ya se desee cultivar desde piñas que necesita un clima tropical óptimo para su desarrollo, la temperatura debe estar entre 25 ° C y 32 ° C durante el día y entre 15 ° C y 20 ° C durante la noche, así mismo requiere un porcentaje de humedad del 85-90%, esto según (Neri, y otros, 2021), hasta cultivo de espinaca la cual es una especie de hortaliza que no soporta los climas cálidos por ende su cultivo óptimos términos para el desarrollo de esta especie se encuentra entre los 15 y 18 °C y una humedad relativa de entre 95 y 98%., esto según (Jayo, 2018).

El objetivo de este trabajo es diseñar un sistema de control de invernaderos de fácil uso y sea automatizado el cual permita el seguimiento y modificación de todas las variables allí presentes en tiempo real.

Con este prototipo, busca a dar un uso de la aplicación adecuada del sistema PID para una adecuada automatización de un invernadero haciendo que este sistema funcione con conocimientos de programación muy básicos y hasta nulos.

Materiales y Métodos

Los componentes del prototipo se muestran a continuación; tales como hardware y software del mismo, tipos de sensores y actuadores utilizados, parte del software, hardware, programación, interfaz gráfica, diagrama de conexión, así como sus resultados.

Estructura

Para cumplir con los requisitos del proyecto, se desarrolló una estructura de invernadero en forma de

semi cilindro adaptando el sistema de control de temperatura y humedad dentro del mismo apoyo al sistema estructural de PVC y con cubierta de polietileno de baja densidad. Se dispondrá de una base de madera, se diseñó un modelo escala de 1 a 10 por medio de Tinkercad

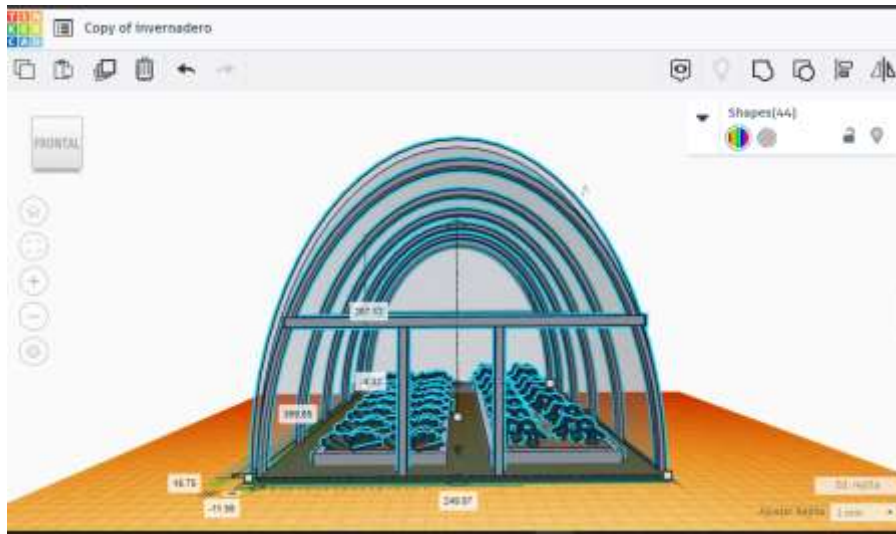


Figura 1. Vista Frontal del Invernadero.
Fuente: Elaboración Propia mediante Tinkercad.



Figura 2. Vista lateral del Invernadero.
Fuente: Elaboración Propia mediante Tinkercad

Materiales Electrónicos

Para llevar a cabo este proyecto se utilizó una placa Arduino UNO, como el que podemos ver en la (Figura 2), es una plataforma de creación electrónica gratuita, flexible y fácil de usar de hardware y software para creadores y desarrolladores. Esta plataforma permite la creación de diferentes tipos de tarjetas únicas a las que la comunidad de creadores puede dar diferentes usos.

El microcontrolador Arduino tiene lo que se llama una interfaz que es una conexión en la que podemos conectar tipos de dispositivos en la placa. La información de estos dispositivos que conectes será transmitida al microcontrolador, y el microcontrolador se encargará de procesar los datos que pasan por este mismo. También tiene una interfaz de salida, que es para transmitir información que ha sido procesada a otros dispositivos.



Figura 3 Placa Arduino uno.
Fuente: Iberobotics

Módulo sensor de temperatura y humedad DHT11 KY015, envía una señal digital a su salida. Tiene un tamaño muy compacto, bajo consumo de energía y una distancia de transmisión de señal de hasta 20 m, lo que lo hace ampliamente utilizado en una gran cantidad de aplicaciones básicas e incluso aplicaciones más difíciles.

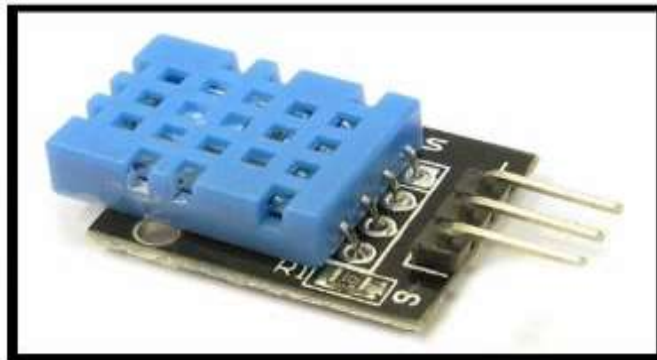


Figura 4 Sensor de temperatura y humedad DHT11
Fuente: Maker Electrónico

En la (Figura 5) podemos visualizar el sensor de humedad de suelo FC-28 permite medir de forma sencilla la humedad del suelo por medio de 2 electrodos resistivos. Compatible con Arduino. El sensor es ideal para monitorear el nivel de humedad en sus cultivos, recordando así cuándo necesita regar o incluso para crear un sistema de riego completamente automático agregando una válvula o bomba de agua. Si el sistema está conectado a Internet, podemos controlar / monitorear nuestro jardín desde cualquier parte del mundo.

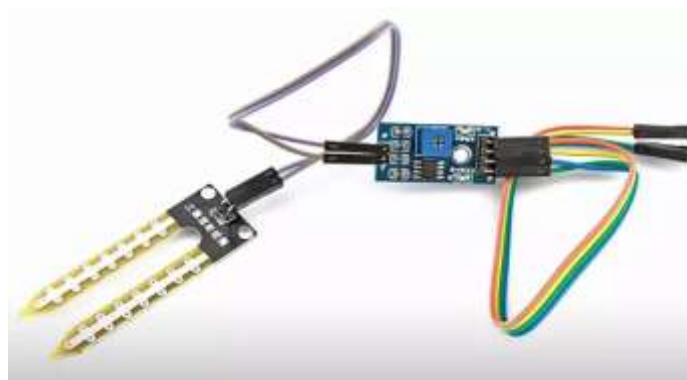


Figura 5 Sensor de Humedad FC-28
Fuente: Murky Robot

Un módulo de 4 relés para Arduino que funcionan a 5 Voltios, capaces de manejar cargas de hasta 10 Amperes en 250 Voltios, convenientemente aislados mediante optoacopladores de las entradas,

aquellos con LED individuales actúan como indicadores de estado. Como se puede ver en la (Figura 6), la tarjeta contiene un conector de entrada (IN1 a IN4), una fuente de alimentación (GND tierra o negativo y Vcc positivo), cuatro LED que indican el estado de las entradas, un selector de puente a los relés de alimentación, cuatro optoacopladores del tipo FL817C, cuatro diodos de protección, cuatro relés marca SONGLE con bobinas de 5V y contactos capaces de controlar hasta 10 Amperes en una tensión de 250V y cuatro borneras, con tres contactos cada una (Común, Normal abierto y Normal cerrado), para las salidas de los relés.

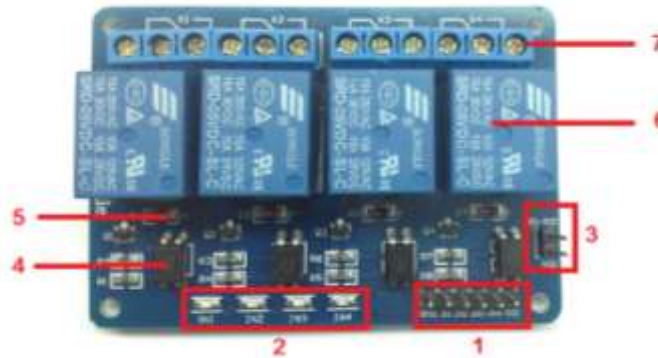


Figura 6 Módulo de 4 relés para Arduino.
Fuente: Murky Robot

Y por último se implementará el uso de un módulo I2C, el cual es una interfaz de comunicación serial que utiliza el protocolo I²C para controlar una pantalla LCD de una forma más fácil, al solo utilizar dos pines para comunicarse con nuestro Arduino, lo cual nos deja más pines del Arduino libres para usarlos en con otros sensores. Este módulo tiene un chip I²C PCF8574 el cual convierte los códigos de I²C en datos paralelos para que puedan ser leídos por la pantalla LCD, funciona tanto en pantallas LCD 16x02 como en LCD 20x04.



Figura 7 Módulo I2C y LCD 16x02.
Fuente: Acortes Software

Potencia disipada por el circuito en corriente directa (DC)

El consumo de corriente continua depende del funcionamiento de cada componente electrónico incluido en el circuito, y el consumo de corriente continua (DC) de cada componente se muestra a continuación. Según la (Tabla 1).

Tabla 1
Cálculo de la corriente total consumida en la etapa electrónica

Componente	Voltage	Intensidad (mA)	Cant.	Subtotal (mA)
Arduino UNO	+5 V	150	x 1	150
Modulo I2C	+5 V	30	x 1	30
Pantalla de Cristal Líquido, LCD 16x02	+5 V	150	x 1	150
Módulo Sensor DHT11	+5 V	2.5	x 1	2.5
Módulo Sensor FC-28	+5 V	35	x 1	35
Modulo relay de 4 canales	+5 V	10	x 1	10
Total corriente consumida				377.5

Nota: Esta tabla fue elaborada mediante la obtención de los datos de los componentes anteriormente mencionados

Fuente: Elaboración Propia

Cálculo de Etapa Electrónica

Por lo tanto, se requiere una fuente de alimentación eléctrica de +5V a 377,5 mA (0,377 5 A).

La potencia consumida por esta etapa es la siguiente:

$$PDC = V I$$

$$PDC = 5 \text{ V } (0,377 \text{ 5 A})$$

$$PDC = 1,887 \text{ 5 W}$$

Potencia disipada por el circuito en corriente alterna (AC)

El consumo de energía eléctrica se basa en la referencia de un foco de resistencia incandescente de 100W, por lo tanto, la mayoría de artefactos eléctricos están referenciados a una determinada cantidad de focos. Según la (Tabla 2).

Tabla 2
Cálculo de la corriente total consumida en la etapa eléctrica

Componente	Consumo (W)	# Focos	Cant.	Subtotal
Fuente de alimentación	100	1	x 1	100
Electroválvula de Agua	55	0.5	x 1	55
Ventilador	25	0.25	x 1	25
Humidificador	300	3	x 1	300
Calefactor	1500	15	x 1	1500
Total, de potencia consumida				1980

Nota: Esta tabla fue elaborada mediante la obtención de los datos de los componentes anteriormente mencionados

Fuente: Elaboración Propia

Cálculo de Etapa Eléctrica: $P_{AC} = 1980 \text{ W}$

Conexión de Elementos

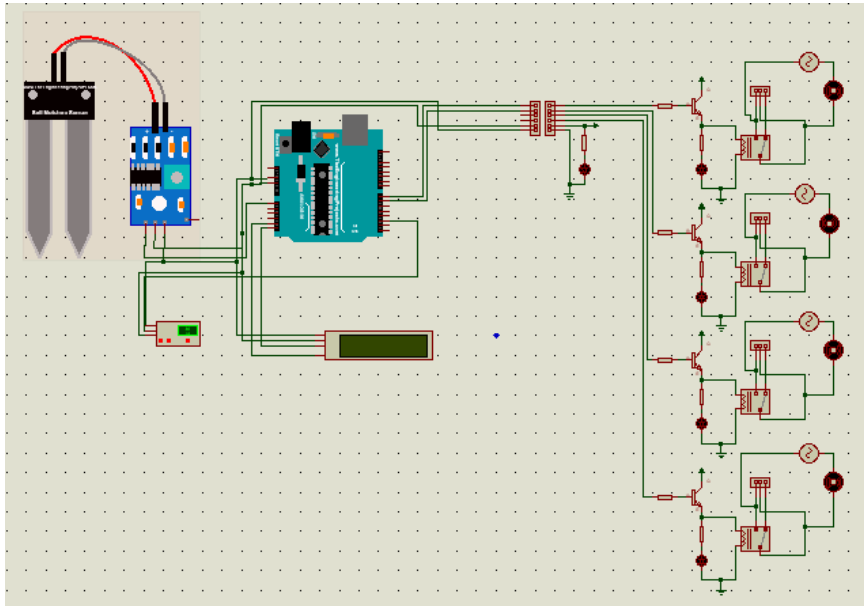


Figura 8 Conexión de Arduino mediante Proteus
Fuente: Elaboración Propia

En la (Figura 8) se observa la visión general de la totalidad del circuito en la automatización del invernadero, en el cual se aprecia la configuración del Arduino, así como el uso de relé, también el uso de sensor de temperatura y humedad DHT11 y sensores de humedad de suelo FC-28 para el control del huerto. Este circuito se puede dar su medición por la pantalla LCD el cual nos indica si la temperatura es baja o alta así mismo con la temperatura de suelo el cual mediante parámetros establecidos se visualiza su completa activación hacia la bomba mediante el puerto de relé número 2.



Figura 9 Conexión de controlador
Elaboración Propia

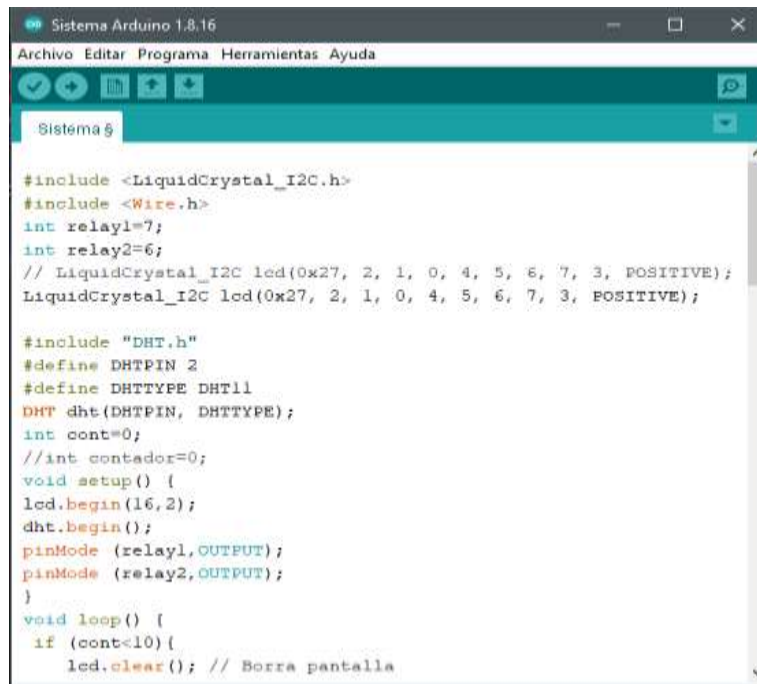
Para la conexión mediante el controlador físico se implementó adicionalmente la ayuda de un protoboard para poder realizar el correspondiente circuito del plano en proteus, así mismo se puede apreciar en la (Figura 9) el adecuado funcionamiento del mismo

Código Arduino utilizado

Para el siguiente código se utilizó el uso de librerías el cual permitieron programar la placa Arduino uno, estas librerías fueron las siguiente:

- Adafruit_Unified_Sensor
- DHT_sensor_library
- LiquidCrystal_i2c

Y para la programación del código fue la siguiente:



```
Sistema Arduino 1.8.16
Archivo Editar Programa Herramientas Ayuda

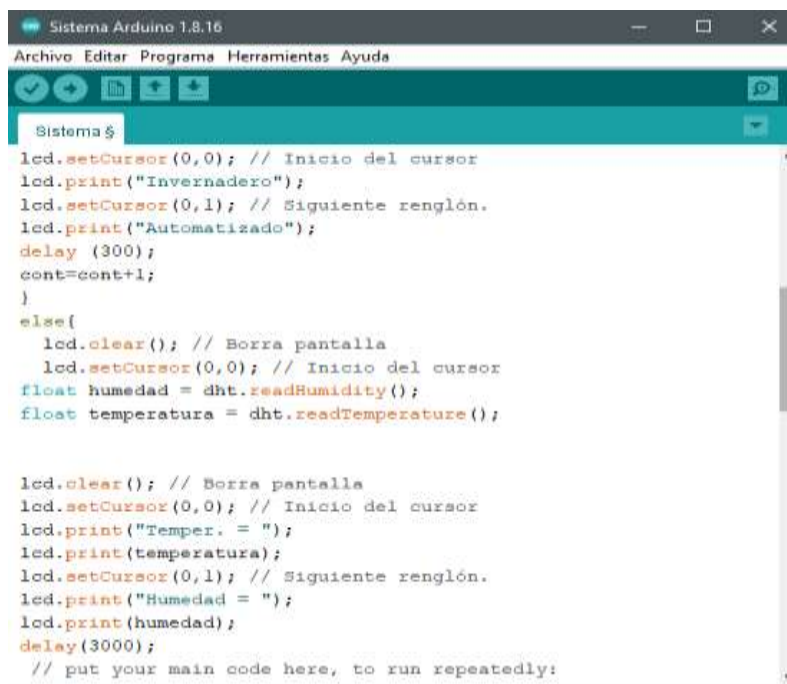
Sistema $

#include <LiquidCrystal_I2C.h>
#include <Wire.h>
int relay1=7;
int relay2=6;
// LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
int cont=0;
//int contador=0;
void setup() {
  lcd.begin(16,2);
  dht.begin();
  pinMode (relay1,OUTPUT);
  pinMode (relay2,OUTPUT);
}
void loop() {
  if (cont<10){
    lcd.clear(); // Borra pantalla
```

Figura 10 Código Arduino parte 1

Fuente: Elaboración Propia



```
Sistema Arduino 1.8.16
Archivo Editar Programa Herramientas Ayuda

Sistema $

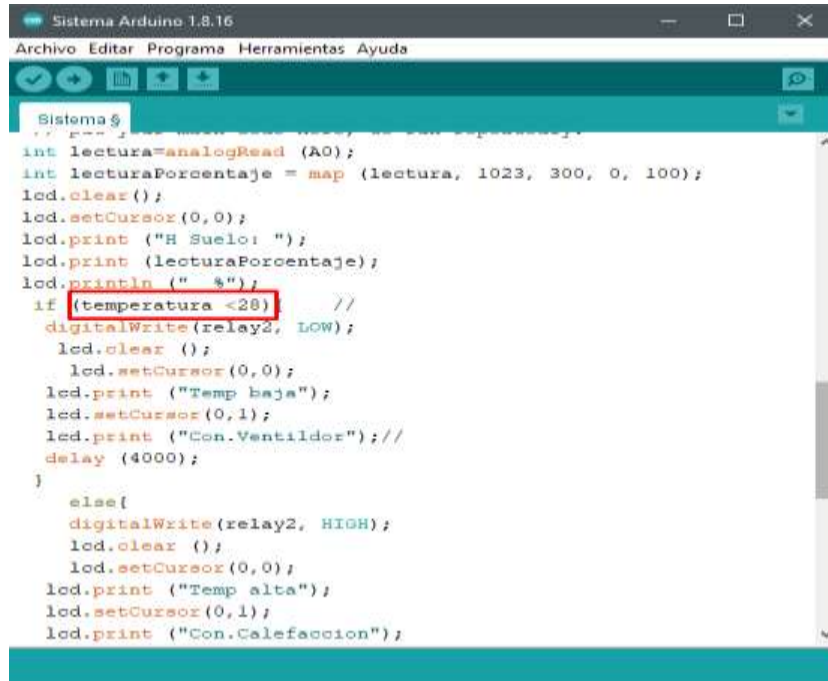
lcd.setCursor(0,0); // Inicio del cursor
lcd.print("Invernadero");
lcd.setCursor(0,1); // Siguiete renglón.
lcd.print("Automatizado");
delay (300);
cont=cont+1;
}
else{
  lcd.clear(); // Borra pantalla
  lcd.setCursor(0,0); // Inicio del cursor
  float humedad = dht.readHumidity();
  float temperatura = dht.readTemperature();

  lcd.clear(); // Borra pantalla
  lcd.setCursor(0,0); // Inicio del cursor
  lcd.print("Temper. = ");
  lcd.print(temperatura);
  lcd.setCursor(0,1); // Siguiete renglón.
  lcd.print("Humedad = ");
  lcd.print(humedad);
  delay(3000);
  // put your main code here, to run repeatedly:
```

Figura 11 Código Arduino parte 2

Fuente: Elaboración Propia

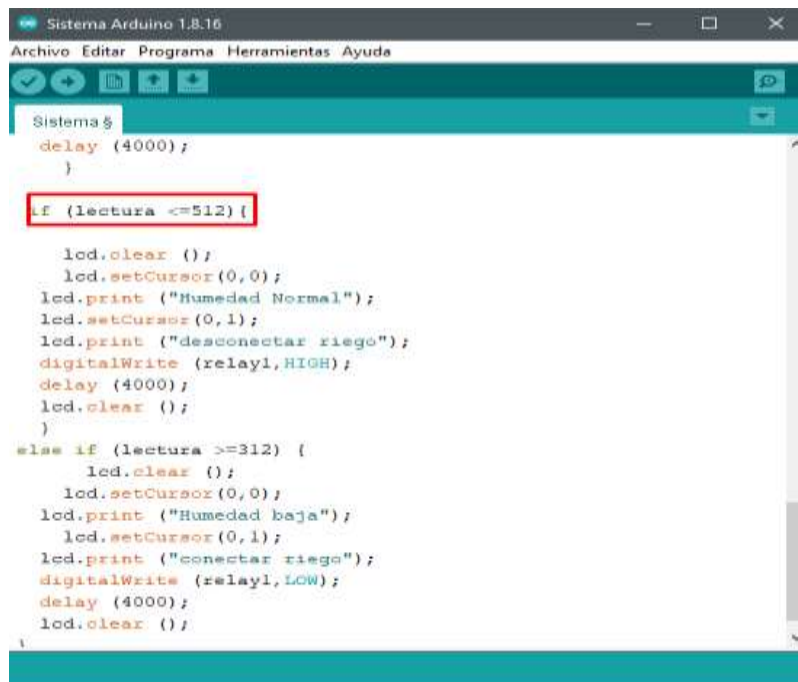
Para la parte inicial del código en las (Figuras 10 y 11) se hace la descripción de las librerías así mismo se realiza la programación para que el panel LCD se figuren los datos obtenidos por el sensor DHT11 el cual nos mostrara la temperatura y humedad.



```
Sistema Arduino 1.8.16
Archivo Editar Programa Herramientas Ayuda
Sistema $
int lectura=analogRead (AO);
int lecturaPorcentaje = map (lectura, 1023, 300, 0, 100);
lcd.clear();
lcd.setCursor(0,0);
lcd.print ("H Suelo: ");
lcd.print (lecturaPorcentaje);
lcd.println (" %");
if (temperatura <28) //
digitalWrite(relay2, LOW);
lcd.clear ();
lcd.setCursor(0,0);
lcd.print ("Temp baja");
lcd.setCursor(0,1);
lcd.print ("Con.Ventildor");//
delay (4000);
}
else{
digitalWrite(relay2, HIGH);
lcd.clear ();
lcd.setCursor(0,0);
lcd.print ("Temp alta");
lcd.setCursor(0,1);
lcd.print ("Con.Calefaccion");
```

Figura 12 Código Arduino parte 3
Fuente: Elaboración Propia

Para la (Figura 12) se crea el comando indicador de la temperatura adecuada para el cultivo que estemos realizando y así mismo si esta temperatura es alta encenderá el ventilador hasta que la temperatura del invernadero sea la adecuada o similar a la misma



```
Sistema Arduino 1.8.16
Archivo Editar Programa Herramientas Ayuda
Sistema $
delay (4000);
}
if (lectura <=512){
lcd.clear ();
lcd.setCursor(0,0);
lcd.print ("Humedad Normal.");
lcd.setCursor(0,1);
lcd.print ("desconectar riego");
digitalWrite (relay1,HIGH);
delay (4000);
lcd.clear ();
}
else if (lectura >=312) {
lcd.clear ();
lcd.setCursor(0,0);
lcd.print ("Humedad baja");
lcd.setCursor(0,1);
lcd.print ("conectar riego");
digitalWrite (relay1,LOW);
delay (4000);
lcd.clear ();
```

Figura 13 Código Arduino parte 4
Fuente: Elaboración Propia

De esta misma forma mediante el módulo y sensor FC-28 Se obtendrá los rangos necesarios para que se pueda activar y desactivar el sistema de riego adecuado tendiendo un rango adecuado para que el suelo este correctamente húmedo.

Funcionamiento del sistema PID tradicional

A continuación, se describe la fórmula PID tradicional que permite ver y comprender cómo funcionan los tres gráficos:

$$C(s) = k_c \left(1 + \frac{1}{\tau_i s} + \tau_d s \right)$$

Figura 14 Formula de PID Tradicional

Fuente: <https://controlautomaticoeducacion.com/control-realimentado/control-pid-por-asignacion-de-polos/>

donde K_c es la ganancia proporcional, T_i es el tiempo integral del controlador y T_d es el tiempo derivativo del controlador.

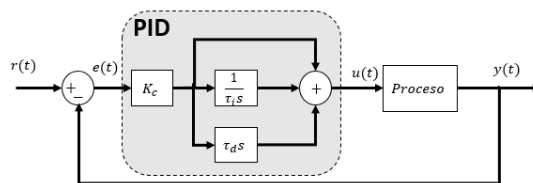


Figura 15 Esquema PID tradicional

Fuente: <https://controlautomaticoeducacion.com/control-realimentado/control-pid-por-asignacion-de-polos/>

DESARROLLO

Simulación de un sistema PID en acción

Para realizar la explicación del fenómeno obtenido mediante el software LabVIEW, se requirió el diseño de un sistema de control de temperatura y humedad relativa tal que pueda simular la pérdida y ganancia en tiempo real y el fenómeno ocurrido entre estos, asimismo cuenta con un botón de encendido/apagado, indicadores led que permiten reconocer el momento en que se requiera activar los artefactos, además el sistema fue diseñado para la obtención de datos en tiempo real mediante Excel.

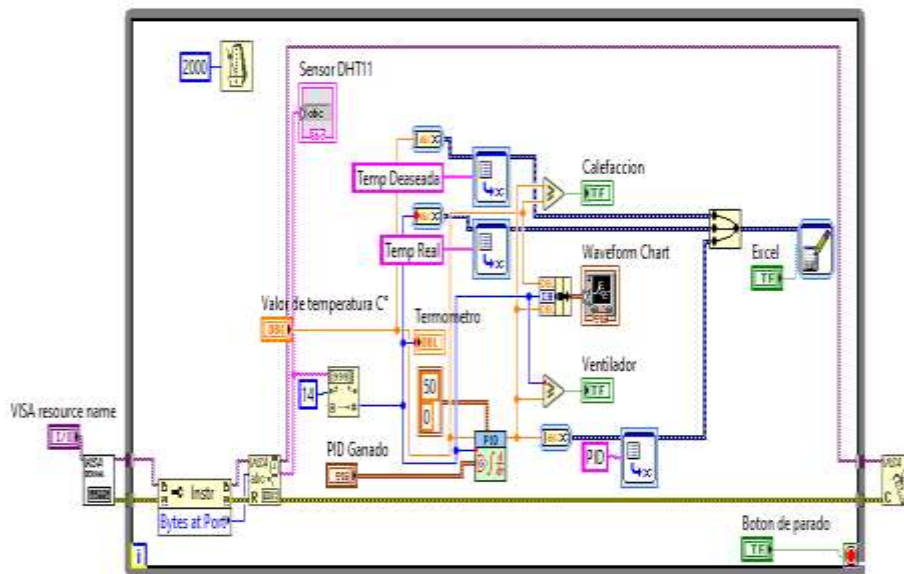


Figura 16 Diagrama de controlador PID

Fuente: Elaboración Propia

Interfaz Gráfico

A continuación, se muestra el panel frontal (Figura 14) diseñado para la simulación y toma de datos donde se pueden controlar y monitorear los factores más importantes en un antes mencionados en el presente documento. Así mismo se pueden establecer a criterio de los usuarios los valores con los cuales simula una situación real del sistema de control de la placa Arduino de una manera autónoma, facilitando y mejorando su visualización.

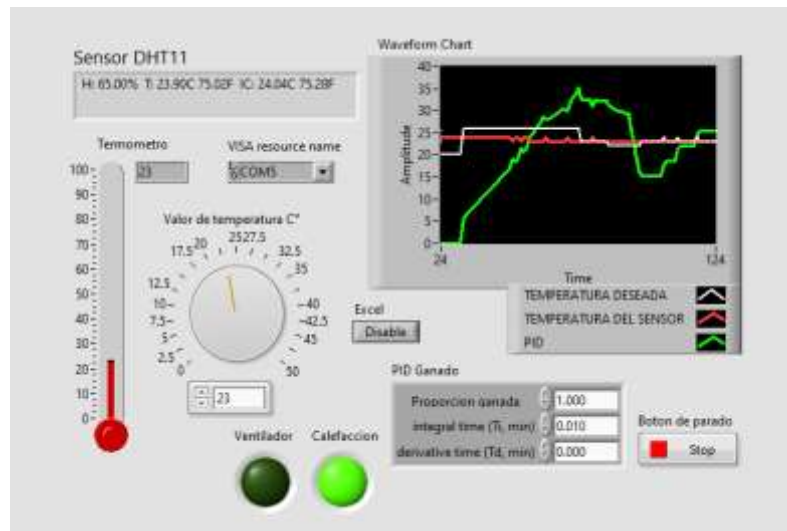


Figura 17 Parte frontal de controlador PID
Elaboración Propia

Resultados

Para la toma de resultados se hará una comparativa de 4 muestras diferentes mediante diferentes variables de T_i y T_d :

Para la primera muestra se puede apreciar que nuestra integral de tiempo sea muy baja y también nuestra derivativa de tiempo sea nula hace que las oscilaciones de la misma sean muy abruptas y en una situación real este mismo hubiera ocasionado un cortocircuito y hasta un accidente por la misma variación.

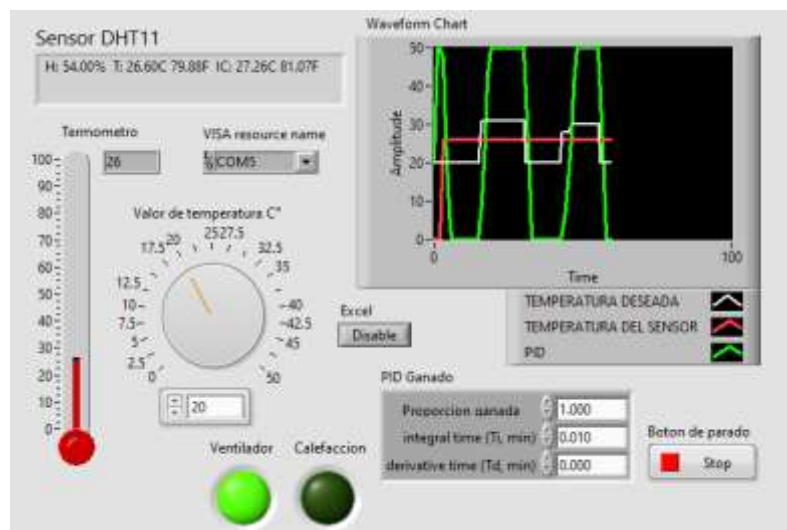


Figura 18 Muestra 1 de variación de PID
Fuente: Elaboración Propia

En nuestra segunda muestra que pusimos la integral del tiempo a 0,5 se obtuvo que la oscilación es más recta sin embargo al no contar con una derivativa de tiempo el retorno es muy brusco, por ende, tampoco puede ser considerado adecuado para el uso correcto del PID

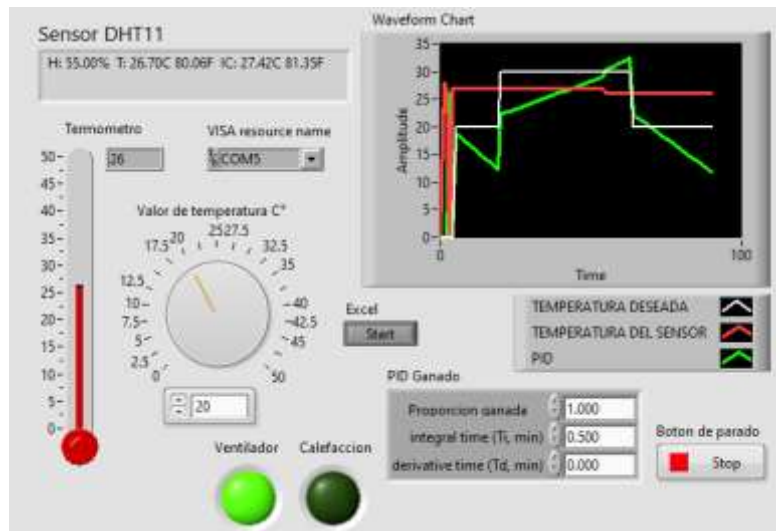


Figura 19 Muestra 2 de variación de PID
Fuente: Elaboración Propia

Para la tercera muestra se puede apreciar que nuestra integral de tiempo sea de 0,5 y también nuestra integral derivativa sea de un 0,1 que las oscilaciones de la misma sean muy lentas y la respuesta del mismo demore en alcanzar una respuesta adecuada a la temperatura deseada y en una situación real este mismo tipo de control sea la adecuada para un funcionamiento esperado.

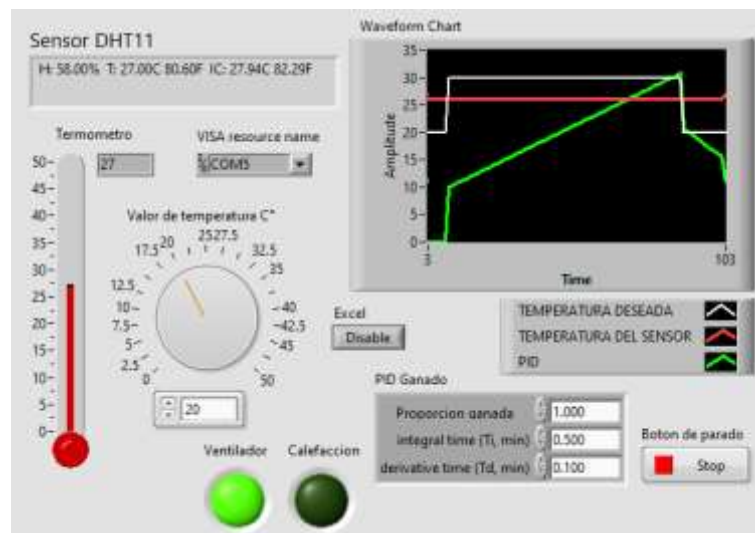


Figura 20 Muestra 3 de variación de PID
Fuente: Elaboración Propia

Para la cuarta muestra se puede apreciar que nuestra integral de tiempo sea de 0,5 y también nuestra integral derivativa sea de un 0,5 que las oscilaciones de la misma sean muy lentas, sin embargo, hace también que nuestra toma de la respuesta del mismo demore en alcanzar una respuesta adecuada a la temperatura deseada y en una situación real este mismo tipo de control sea la adecuada para un funcionamiento esperado, pero en comparación con los anteriores resultados se puede decir que hay señales de que suceda un cierto error como fallos esporádicos que se muestran en la (Figura 19).

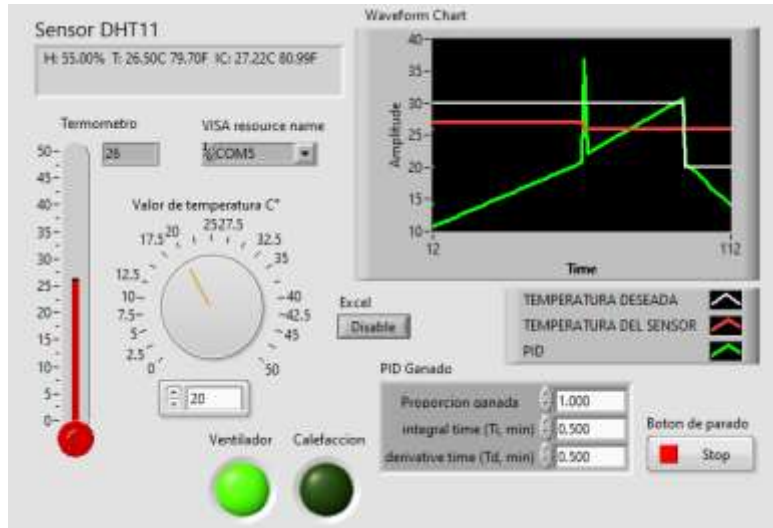


Figura 21 Muestra 4 de variación de PID
Fuente: Elaboración Propia

Análisis al control de temperatura PID

Según la (Figura 15) la podemos apreciar que el controlador PID es inversamente proporcional a la temperatura obtenida mediante el sensor DHT11 así mismo se puede apreciar que el controlador PID se toma su tiempo para poder aumentar y depreciarse en la curva, esto se debe debido a que se altera su T_i y su T_d el cual al ser mayor inclinan que la curva se eleve de manera muy rápida y que los ventiladores y calefactores que se utilicen no sean activados constantemente y se pueda producir algún tipo de corte circuito por parte de los mismo sistema.

Discusión

A diferencia del sistema implementado por (Perea, 2016) en su trabajo de Diseño de un sistema de monitoreo, registro y control de temperatura y humedad para un cultivo de invernadero, quien implementa un sistema de monitoreo mediante Arduino Mega y un sensor DHT22 se simplifico el uso de los recursos del mismo sin desaprovechar la potencia de estos controladores cambiándolos por un Arduino Uno así también como el uso de un módulo DHT11. También podemos afirmar que el sistema actual es más eficiente al incluir el sistema implementa el uso el módulo FC-28 haciendo el uso adecuado de la programación para Arduino y regularizando el control de humedad de la tierra.

Conclusiones

Como conclusión logramos cumplir con nuestros objetivos, principalmente el de implementar un sistema de temperatura y riego automatizado por PID, este sistema no necesita supervisión continua ya que estará activándose solo cuando sea necesario, esto significa que solo cuando registre que la temperatura y la humedad sean muy alta o muy baja, entonces el PID activará la calefacción y el sistema de riego respectivamente en caso sea necesario.

Se pudo corroborar que la lectura de temperatura del software corresponde a la misma que se obtiene en tiempo real, concluyendo la efectividad del sistema para ser implementada en un vivero automatizado y ya no se tenga que regar de forma manual para que los trabajadores puedan centrarse en otras tareas en vez del riego en ese sector.

Referencias

- Arduino. (2021). *Arduin CC*. Obtenido de Arcuino Uno, Revisado: <https://store-usa.arduino.cc/products/arduino-uno-rev3>.
- Chulde, K. (2017). Implementación de un sistema automatizado de riego por goteo parcial, a campo abierto en el Centro Experimental del Riego de la Facultad de Recursos Naturales. *Escuela Superior Politécnica de Chimborazo*.
- Gracia, M. (2021). *Deloitte*. Obtenido de IoT - Internet Of Things, Revisado: <https://www2.deloitte.com/es/es/pages/technology/articles/IoT-internet-of-things.html>.
- Jayo, J. (2018). *Niveles de guano de isla y dosis de microorganismos eficaces*. Ayacucho - Perú: UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL.
- Neri, J., Meleández, J., Vilca, N., Huaman, E., Silva, R., & Oliva, M. (2021). Efecto de la densidad de plantación sobre el rendimiento agronómico y la calidad del fruto de tres cultivares de piña (*Ananas comosus* L. Merr.). *Hindawi Publishing Corporation*.
- Perea, J. (2016). Diseño de un sistema de monitoreo, registro y control de temperatura y humedad para un cultivo de invernadero. *Universidad Tecnológica de Pereira*, 42-43.
- Wheather Spark. (3 de octubre de 2021). *Wheather Spark*. Obtenido de El clima y el tiempo promedio en todo el año en Trujillo: <https://es.weatherspark.com/y/19239/Clima-promedio-en-Trujillo-Per%C3%BA-durante-todo-el-a%C3%B1o>.