

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**IMPLEMENTACIÓN DE DOS SISTEMAS DE CONTROL DE
TEMPERATURA MEDIANTE LA ACCIÓN DE CONTROL ON/OFF
PARA EL LABORATORIO DE TECNOLOGÍA INDUSTRIAL DE LA
ESFOT**

**DESARROLLO DEL ALGORITMO DE CONTROL, HMI Y
REGISTRO DE DATOS**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN ELECTROMECAÁNICA**

MARÍA CRISTINA JIBAJA TRÁVEZ

DIRECTOR: ALAN DANIEL CUENCA SÁNCHEZ

DMQ, julio 2024

CERTIFICACIONES

Yo, María Cristina Jibaja Trávez declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

MARÍA CRISTINA JIBAJA TRÁVEZ

maria.jibaja@epn.edu.ec

cjibaja13@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por María Cristina Jibaja Trávez, bajo mi supervisión.

Alan Daniel Cuenca Sánchez

alan.cuenca@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmo que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

María Cristina Jibaja Trávez

DEDICATORIA

Dedico este trabajo a mis padres, cuya sabiduría y amor incondicional han sido la brújula que ha guiado mi camino. Gracias por sus sacrificios y apoyo inquebrantable. Ustedes son mi mayor inspiración y la razón por la cual persigo mis sueños con tanta pasión.

A mi hermana, mi cómplice y mejor amiga, gracias por tu constante ánimo, tu apoyo en mis noches de estudio y por ser una fuente de alegría en mi vida.

Con todo mi amor y gratitud, les dedico este logro, que no hubiera sido posible sin ustedes.

AGRADECIMIENTO

Quiero agradecer a todas las personas que me han acompañado durante este periodo de mi vida y que forman parte de este logro.

A mi mami, por su amor incondicional, su sacrificio, apoyo y palabras de aliento que me impulsaron a alcanzar este sueño.

A mi papi, por enseñarme a amar el camino de la ciencia, por su amor incondicional, su sacrificio y apoyo que me sirvió como guía todo este tiempo.

A mi ñaña, por darme ánimos y acompañarme durante las largas noches de estudio y proyectos. Gracias por siempre estar a mi lado.

A mi familia, en especial mis abuelitos Papito Julio, Papito Nolito, Mami Rosi, Mamita Gloria, mis tías Nachitas, Conita, Lupi, Lulú y Jova, mis tíos Rober, Andrés y mis primos por su apoyo constante. Sus palabras de aliento y su confianza en mis capacidades me han motivado a seguir adelante.

A mis profesores de carrera, cuyo conocimiento y dedicación han sido pilares en mi formación académica. En especial, agradezco a mi tutor de tesis, Alan Cuenca, por su guía y apoyo durante todo este proceso.

A mi tutor académico, Abraham Loja, quien siempre me motivo y aconsejo para que pueda cumplir mis metas académicas y desarrollarme como profesional.

A mi amiga, Belén Cárdenas, por estar siempre a mi lado durante estos años de carrera. Tu compañía y amistad han hecho de este viaje una experiencia mucho más linda y especial.

A mis amigos politécnicos y mis compañeros, quienes han hecho de mi estancia en la universidad un periodo lleno de momentos memorables y aprendizajes compartidos.

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VI
ABSTRACT	VII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco teórico	2
Control ON/OFF de Temperatura.....	2
PLC S7-1200.....	3
Interfaz Humano Máquina HMI.....	5
2 METODOLOGÍA	6
2.1 Algoritmo de control	6
2.2 Interfaz Humano Máquina.....	17
2.3 Registro de datos.....	18
3 RESULTADOS.....	21
3.1 Pruebas de funcionamiento	21
3.2 Manual de usuario y mantenimiento	24
4 CONCLUSIONES	26
5 Recomendaciones	28
6 REFERENCIAS BIBLIOGRÁFICAS.....	29
7 ANEXOS.....	30
ANEXO I.....	31

RESUMEN

El presente proyecto expone el desarrollo de un algoritmo para el control, monitoreo y supervisión de temperatura y registro de datos, tanto de forma local como remota, de dos módulos didácticos. Para el control local, se empleará un Arduino y se programará en un lenguaje de bajo nivel. Para el control de forma remota, se utilizará un PLC (Controlador Lógico Programable) y se desarrollará su respectiva HMI (Interfaz Humano Máquina). Los módulos deben medir constantemente los datos de temperatura y permitir el ingreso de la referencia (*set point*). Para llevar a cabo, el registro se utilizará un microcontrolador (Arduino) para visualizar los datos y generar tablas en Excel. Los módulos didácticos servirán para fortalecer conocimientos relacionados al acondicionamiento de sensores, control de temperatura y registro de datos.

Este proyecto se organiza en siete capítulos, los cuales se describen a continuación:

El primer capítulo explica la introducción del proyecto, en el cual se describe el componente, los objetivos generales y específicos, se delimita el alcance, y se proporciona una breve explicación de los fundamentos teóricos necesarios para el proyecto.

En el segundo capítulo se explica el procedimiento que se sigue para realizar el proyecto. Se desarrolla un código para el control ON/OFF de temperatura de forma local y remota tomando en cuenta la utilización de un Arduino, PLC y HMI. Por último, se elabora el registro de datos a través de Excel.

En el tercer capítulo se presentan los resultados obtenidos del desarrollo del código para el Arduino, el algoritmo de control para el PLC y HMI y el registro de datos de los módulos, así como los manuales de usuario y mantenimiento necesarios para el correcto uso y funcionamiento del mismo.

El cuarto y quinto capítulo se contienen las conclusiones y recomendaciones obtenidas del proyecto.

Por último, en el sexto y séptimo capítulo se encuentran las referencias bibliográficas y anexos del documento.

PALABRAS CLAVE: Algoritmo de control, PLC, HMI, Registro de datos, Temperatura.

ABSTRACT

This project presents the development of an algorithm for the control, monitoring, and supervision of temperature and data logging, both locally and remotely, for two didactic modules. For local control, an Arduino will be used and programmed in a low-level language. For remote control, a PLC (Programmable Logic Controller) will be used, and its respective HMI (Human-Machine Interface) will be developed. The modules must constantly measure temperature data and allow the input of the reference (*set point*). To carry out the logging, a microcontroller (Arduino) will be used to visualize the data and generate tables in Excel. The didactic modules will help strengthen knowledge related to sensor conditioning, temperature control, and data logging.

This project is organized into seven chapters, which are described below:

The first chapter explains the introduction of the project, in which the component is described, the general and specific objectives are outlined, the scope is defined, and a brief explanation of the theoretical fundamentals necessary for the project is provided.

The second chapter explains the procedure followed to carry out the project. A code is developed for the ON/OFF temperature control both locally and remotely, taking into account the use of an Arduino, PLC, and HMI. Finally, the data logging through Excel is elaborated.

The third chapter presents the results obtained from the development of the code for the Arduino, the control algorithm for the PLC and HMI, and the data logging of the module, as well as the user and maintenance manuals necessary for its correct use and operation.

The fourth and fifth chapters contain the conclusions and recommendations obtained from the project.

Finally, the sixth and seventh chapters include the bibliographic references and annexes of the document.

KEYWORDS: Control algorithm, PLC, HMI, Data logging, Temperature.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

En la industria, el avance tecnológico en el control automático y la instrumentación se han convertido en pieza clave para lograr una mayor eficiencia en los procesos. Los sistemas de control están compuestos por señales de entrada que llegan a un controlador, que puede ser un Arduino, LOGO, PLC, entre otros, los cuales procesan la señal y la envían a un actuador, dependiendo del caso. El proceso puede terminar en la señal de salida, o mediante un sensor se realiza una retroalimentación a la señal de entrada, comenzando así nuevamente el sistema. En el proyecto, se aplicará el control automático de una planta donde, mediante el encendido y apagado de un ventilador y una resistencia de níquelina, se pueda controlar la temperatura de una estación y además a través de un termistor NTC de 10K se realizará la retroalimentación de la señal medida a un Arduino o PLC para el control.

En el Laboratorio de Tecnología Industrial (LTI) de la ESFOT se llevan a cabo diversas prácticas para la medición de magnitudes físicas, y se dispone de varios módulos didácticos donde se aplica el uso de sensores para el control del sistema. Las plantas representadas en los módulos didácticos proporcionan al estudiante una visión del trabajo que puede llevarse a cabo en la industria, utilizando los conocimientos adquiridos de instrumentación, electrónica digital y analógica, control industrial y mecánica. Por esta razón, este proyecto busca brindar conocimientos sobre el uso de controladores para acondicionar sensores, controlar procesos y registrar datos del sistema.

Para conseguir el objetivo general de este proyecto, se desarrollará el programa en Arduino para acondicionar mediante *software* el sensor NTC de 10K y mostrar el valor de lectura en un LCD. Posteriormente, al activar un selector en modo local, se habilitará un teclado matricial, lo que permitirá ingresar un *set point* al LCD. Luego, el Arduino activará la resistencia (níquelina) y el ventilador para mantener la estación a la temperatura deseada.

Si el selector se activa en modo remoto, se controlará el sistema mediante un PLC y se conectará a una HMI. Para lograr esto, se desarrollará un algoritmo en el Portal TIA V15. Para la medición de temperatura, el PLC recibirá una señal acondicionada provista por el Arduino. Luego el algoritmo permitirá ingresar una referencia, y accionará la resistencia (níquelina) y el ventilador, para el control del sistema.

Finalmente, se desarrollará un programa en Arduino para que registre los datos en tiempo en real de la temperatura de la estación, exportando dichos datos a un archivo .CSV en Excel. Además, se elaborarán dos videos correspondientes al manual de usuario y mantenimiento respectivamente.

1.1 Objetivo general

Implementar dos sistemas de control de temperatura mediante la acción de control ON/OFF para el Laboratorio de Tecnología Industrial de la ESFOT.

1.2 Objetivos específicos

- Desarrollar el algoritmo de control, interfaz humano máquina y registro de datos.
- Realizar pruebas de funcionamiento.
- Realizar dos videos del manual de usuario y mantenimiento respectivamente

1.3 Alcance

En el presente proyecto se utilizarán microcontroladores, PLC y pantalla HMI, que permita al operador realizar de forma local y remota el control de temperatura mediante la acción de control ON/OFF de los módulos didácticos. Además, se exportará en tiempo real los datos de temperatura mediante la función *Data Streamer* a un archivo de Excel para su posterior análisis.

Se realizarán las pruebas necesarias para asegurar su correcto funcionamiento, tomando en cuenta la acción de control TODO/NADA, control físico y remoto, lectura de señales, acondicionamiento de sensores y registro de datos. Al finalizar, se elaborará un manual de usuario y mantenimiento, el mismo que contará con las instrucciones respectivas para su uso y recomendaciones para evitar inconvenientes operativos.

1.4 Marco teórico

Control ON/OFF de Temperatura

El control ON/OFF de Temperatura es una de las acciones de control más simples, ya que la señal de salida funciona en solo dos estados encendido y apagado, cuando atraviesa el punto de ajuste. El controlador no permite tener un valor exacto de la variable controlada. Por lo tanto, este sistema utiliza un valor de histéresis, donde el encendido y apagado tiene un margen de tolerancia. [1]

Para este sistema se tiene usualmente: sensores que miden la magnitud física, un controlador que envía señales digitales de 0 y 1 hacia unos reles, que encienden o apagan actuadores como resistencias, ventiladores, bombas, entre otros. [2] En la Figura 1.1 se observa la respuesta de acción de control todo/nada.

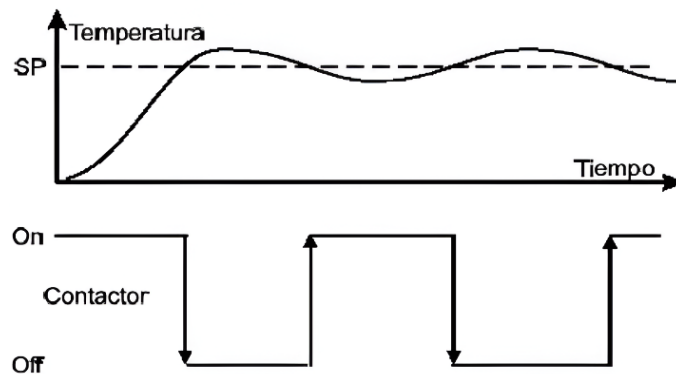


Figura 1.1. Diagrama de control ON/OFF de temperatura [3]

PLC S7-1200

Un Controlador Lógico Programable (PLC) es un computador industrial dedicado para la automatización de maquinaria y procesos. El PLC S7-1200 fue introducido al mercado en 2009 por la fábrica de SIEMENS AG. Desde entonces, se ha convertido en una herramienta para automatizar pequeños y medianos sistemas, como maquinaria, células robotizadas, líneas de producción pequeña y grandes, entre otros.

A continuación, se describen las características principales del PLC S7-1200.

- **Modelos y módulos de expansión**

Existen varios modelos del PLC S7-1200, que se diferencian por el tipo de alimentación, la memoria, tipos de interfaces y el número de módulos de expansión que se pueden añadir. En la Tabla 1.1 se muestran algunos de los modelos con sus respectivas características. Si para el sistema no es suficiente las entradas o salidas del PLC a este se puede añadir módulos de expansión (ver Figura 1.2) como pueden ser entradas o salidas analógicas o digitales, módulos de comunicación, entre otros. [4]

Tabla 1.1. Modelos del PLC S7-1200 [4]

Modelo	Memoria	Entradas/Salidas	Interfaces de Comunicación	Módulos adicionales
S7-1211C	25 KB	6 DI/4 DO	1 X Ethernet	Hasta 2
S7-1212C	50 KB	8 DI/6 DO	1 X Ethernet	Hasta 3
S7-1214C	100 KB	14 DI/6 DO	1 X Ethernet, 1x RS-485	Hasta 8
S7-1215C	125 KB	14 DI/10 DO	2 X Ethernet, 1x RS-485	Hasta 8
S7-1217C	125 KB	14 DI/10 DO	2 X Ethernet, 1x RS-485	Hasta 8
S7-1212FC	75 KB	8 DI/6 DO	1 X Ethernet	Hasta 3
S7-1214FC	100 KB	14 DI/10 DO	2 X Ethernet, 1x RS-485	Hasta 8
S7-1215FC	125 KB	14 DI/10 DO	2 X Ethernet, 1x RS-485	Hasta 8
S7-1217C	125 KB	14 DI/10 DO	2 X Ethernet, 1x RS-485	Hasta 8

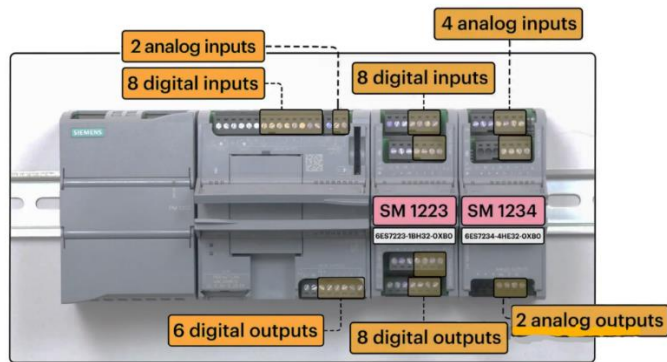


Figura 1.2. Módulos de expansión. [5]

- **Componentes del PLC**

Los PLC están conformados por interfaces de entrada y salida que pueden ser analógicos o digitales, una Unidad de Central de Procesamiento (CPU), y una fuente de alimentación que dependiendo del modelo puede trabajar con 24 (VDC) o 250 (VAC). [6]

A continuación, en la Figura 1.3 se muestran las partes principales del PLC.

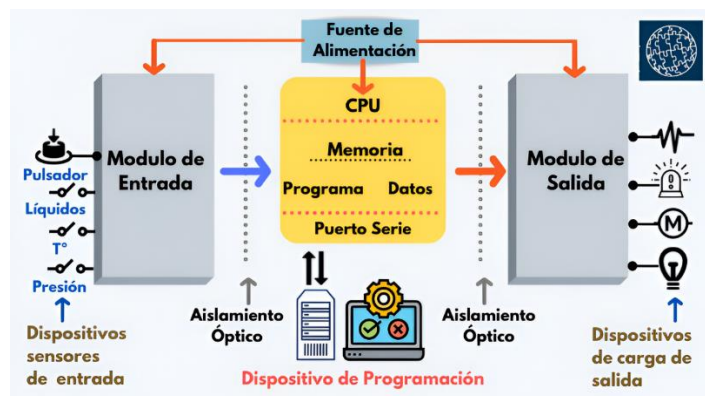


Figura 1.3. Componentes del PLC S7-1200 [6]

- **Lenguaje de programación**

Este PLC trabaja con el software TIA Portal, el cual soporta los siguientes lenguajes:

- LADDER: los diagramas eléctricos se desarrollan de izquierda a derecha.
- FBD: se realiza las operaciones del algoritmo mediante bloques.
- SFC: este lenguaje describe en orden cronológico las acciones del programa.
- IL y ST: es un lenguaje en forma de texto donde se enlista las instrucciones del algoritmo. [7]

- **Protocolos de comunicación**

Para la comunicación del PLC S7-1200 se utilizan un protocolo de tipo PROFINET, pero se puede añadir otros tipos de comunicación, como: Profibus, IO-Link, AS-Interface, entre otros. [4] [8]

Interfaz Humano Máquina HMI

La Interfaz Humano Máquina (HMI) es un medio de interacción que permite al usuario comunicarse con una máquina, equipos o sistema y visualizar datos, gráficas, y conectividad de ellos, para proporcionar información al operador en tiempo real. Las HMI utilizan por una pantalla táctil, botones, indicadores LEDS, y dispositivos de entrada y salida, que permiten al usuario tener una interacción amigable con el proceso. Las HMI utilizan una variedad de protocolos de comunicación como Profinet, Profibus, Modbus, entre otros, que facilita la comunicación confiable y segura en la industria. Este elemento es primordial para integrar un sistema de control y lograr construir un SCADA (Supervisión, Control y Adquisición de Datos) en la industria. [9] [10] En la Figura 1.4 se visualiza una HMI en una pantalla táctil.



Figura 1.4 Dispositivo para HMI [9]

2 METODOLOGÍA

2.1 Algoritmo de control

Para la acción de control *ON/OFF* de temperatura de una planta se optó por desarrollar dos algoritmos, uno para el control local y otro para el control remoto. Se desarrollará un programa utilizando Arduino IDE para el control de manera local. El cuál utilizará un microcontrolador que recibirá la señal de un sensor de temperatura y activará los respectivos actuadores dependiendo del *set point* ingresado. Para el control remoto se elaborará un algoritmo en el Portal TIA V15 ya que se usa un PLC modelo S7-1200 con módulo de expansión de entradas analógicas e Interfaz Humano Máquina.

Algoritmo de programación para control local

- Librerías

Para la programación se utilizaron tres librerías. Como se muestra en la Figura 2.1, se tiene la librería *LiquidCrystal_I2C.h* que sirve para activar el LCD (*Liquid Crystal Display*) con I2C, la *Keypad.h* para configurar el teclado matricial y la *math.h* para realizar las operaciones matemáticas.

```
#include <LiquidCrystal_I2C.h> // Incluimos la librería para el LCD mediante I2C
#include <Keypad.h> // Incluimos la librería para el teclado matricial
#include <math.h> // Incluimos la librería para operaciones matemáticas
```

Figura 2.1 Librerías utilizadas.

- Definición de variables del Teclado Matricial

En el *Hardware* del sistema se va a utilizar un teclado matricial 4x4 para Arduino. Por lo tanto, como se muestra en la Figura 2.2 en el *Software* se determinarán cuantas filas y columnas se tiene en el teclado y luego se define los caracteres.

```
const byte ROWS = 4;
const byte COLS = 4;
// Definimos las variables para el teclado matricial
char keys[ROWS][COLS] = {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};
```

Figura 2.2 Activación de las filas y columnas del teclado matricial.

Para la conexión del teclado con el Arduino, como se muestra en la Figura 2.3 se debe definir los pines del Arduino que van a ser utilizados para las filas y los que van a ser utilizados para las columnas.

```
byte rowPins[ROWS] = {5, 4, 3, 2} ; // Pines de las filas conectados a estos pines de Arduino
byte colPins[COLS] = {9, 8, 7, 6}; // Pines de las filas conectados a estos pines de Arduino

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
```

Figura 2.3 Definición de los pines del Arduino.

- Definición de las variables para el LCD

En el *Hardware* se utiliza un LCD de dos filas y dieciséis columnas. Por lo tanto, para introducir el texto que se desea y ubicar la letra en cada segmento, se declara como se muestra en la Figura 2.4 al LCD y el número de los segmentos que se tiene en las filas y columnas.

```
LiquidCrystal_I2C lcd(0x27, 16, 2); // Dirección del LCD I2C, número de columnas y número de filas
```

Figura 2.4 Definición del LCD.

- Definición de las variables para los actuadores

Se va a utilizar dos relés para el control local de la niquelina y ventilador. Para que el Arduino envíe la señal y active el interruptor, se va a declarar los pines del Arduino que van a controlar estos actuadores como se muestra en la Figura 2.5.

```
// Definimos el pin al que está conectada la salida para la niquelina
const int niquelinaPin = 10;

// Definimos el pin al que está conectada la salida para el ventilador
const int ventiladorPin = 12;

// Definimos el pin PWM
const int pwmPin = 11;
```

Figura 2.5 Definición de los actuadores.

- Definición de variables remotas

Para el control de la planta se realizan ciertas operaciones en el programa. A continuación, en la Figura 2.6 se definen las variables remotas utilizadas, como la temperatura máxima y mínima a la que debe trabajar la planta, el valor de histéresis, la variable de *set point*, y las variables de la fórmula para transformar el valor de voltaje del sensor a temperatura en grados Kelvin y luego transformar los grados Kelvin a grados Centígrados.

```

// Definimos el rango de temperatura y la histeresis
const int temperaturaMinima = 15;
const int temperaturaMaxima = 90;
const int histeresis = 1;

// Definimos el pin que debe estar en alto para empezar el funcionamiento
const int startPin = 13;
int setpoint = 0; // El setpoint que vamos a ingresar

unsigned long previousMillis = 0;
const long interval = 1500;

// Variables usadas en el cálculo
float vm = 0.0;
float rntc = 0.0;
float temperaturaK = 0.0;
float temperatura = 0.0;

float rAux = 10000.0;
float vcc = 5.0;
float beta = 3977.0;
float temp0 = 298.0;
float r0 = 10000.0;

unsigned long lastKeyPressTime = 100;

```

Figura 2.6 Definición de variables remotas

- Configuración del LCD y los pines de salida

Dentro del *void set up* del algoritmo se configuran los pines de los actuadores de control de la niquelina y el ventilador en *pull down* declarando a la variable como *OUTPUT*. Luego se determina el texto y el tiempo que se desea que aparezca en el LCD al encender la planta. Esta configuración se muestra en la Figura 2.7.


```

void setup() {
  Serial.begin(9600);
  // Configuramos los pines de salida
  pinMode(niquelinaPin, OUTPUT);
  pinMode(ventiladorPin, OUTPUT);
  pinMode(startPin, INPUT);

  // Inicializamos la pantalla LCD
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("  Control de ");
  lcd.setCursor(0, 1);
  lcd.print("  Temperatura");
  delay(1000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Jimenez");
  lcd.setCursor(0,1);
  lcd.print("Jibaja");
  delay(1000);
  lcd.clear();
}

```

Figura 2.7 Configuración *void set up*

- Lectura del sensor

Se utilizó un sensor NTC 10K el cual emite una señal de respuesta en ohmios. Por lo tanto, se usó un divisor para transformar esta señal en voltaje y acondicionarla mediante *software*.

Para que el Arduino interprete correctamente la señal analógica del sensor, se debe primero convertir la lectura analógica en un voltaje real, para esto se utilizan las siguientes variables:

vm: Es el voltaje real medido en el pin A0 del Arduino.

vcc: Es el voltaje de referencia que va a variar de 0[V] a 5[V].

analogRead: Valor analógico leído en el pin 0 del Arduino.

1024: Es el número total de posibles valores que tiene el *analogRead*.

En la parte del algoritmo de la Figura 2.8 se muestra la primera fórmula para calcular el voltaje real de entrada que es igual al voltaje de referencia dividido para 1024.

```

if (currentMillis - previousMillis >= interval) {
  previousMillis = currentMillis;
  temperatura = analogRead(0);
  // Bloque de cálculo
  vm=(vcc / 1024)*( analogRead(0) );           // Calcular tensión en la entrada
}

```

Figura 2.8 Lectura de voltaje de entrada

Para el acondicionamiento del NTC se utiliza la Ecuación 2.1 de Steinhart-Hart simplificada, la cual contiene las siguientes variables:

β : Es la constante del termistor NTC, que describe el cambio de la resistencia con la temperatura.

r_0 : Es la resistencia del termistor a la temperatura de referencia.

$temp_0$: Es la temperatura de referencia en [K].

$temperatura$: Es la temperatura de la estación en [°C].

$$temperaturaK = \frac{\beta}{\log\left(\frac{r_{ntc}}{r_0}\right) + \frac{\beta}{temp_0}} [K]$$

Ecuación 2.1 Temperatura en grados Kelvin

En la Figura 2.9 se encuentran las líneas de código para el acondicionamiento del NTC.

```

temperaturaK = beta/(log(rntc/r0)+(beta/temp0)); // Calcular la temperatura en Kelvin
temperatura = (temperaturaK - 273);

```

Figura 2.9 Acondicionamiento del NTC

Por último, en la Figura 2.10 se muestran las líneas de código para mostrar el valor de la temperatura en el LCD.

```

// Restar 273 para pasar a grados Celsius
lcd.setCursor(0, 0);
lcd.print("Temp(PV): ");
lcd.print(temperatura);
lcd.print("C   ");

```

Figura 2.10 Escritura de la temperatura en el LCD

- Condiciones del *Set point*

Mediante el teclado matricial se ingresa el *set point*, como se muestra en la Figura 2.11, en las líneas de código primero se define que al presionar cualquier número se interprete en el código como tal y luego que tal número se imprima en el LCD, después se establecen dos condiciones que si el *set point* ingresado es menor a quince grados centígrados o mayor a noventa no permita ingresar tal número y se imprima en el LCD “Fuera de rango”.

```

if (key != NO_KEY) {
  // Si se presiona una tecla, interpretamos y actualizamos el setpoint
  if (key >= '0' && key <= '9') {
    setpoint = setpoint * 10 + (key - '0');
    // Asegurémonos de que el setpoint esté dentro del rango 0-99
    setpoint = constrain(setpoint, 0, 99);

    // Mostrar el setpoint actualizado en la segunda línea del LCD
    lcd.setCursor(7, 1); // Colocarse en el espacio después de "Setpoint: "
    lcd.print(" "); // Limpiar el valor anterior
    lcd.setCursor(7, 1);
    lcd.print(setpoint);

    if (key != NO_KEY) {

      if ( setpoint < 15 || setpoint > 90) {
        lcd.setCursor(0, 0);
        lcd.print("Fuera de Rango ");
      }

      if (setpoint > 15 || setpoint < 90) {
        lcd.setCursor(0, 0);
        lcd.print("Temp(PV): ");
        lcd.print(temperatura);
        lcd.print("C ");
      }
    }
  }
}

```

Figura 2.11 Ingreso del *set point*

Para reingresar un nuevo *set point* se presiona la letra A y se borrará el antiguo del LCD, y se ingresará un nuevo valor. Para esta acción se escribieron las líneas de código expuestas en la Figura 2.12

```
    } else if (key == 'A') {
    | // Si se presiona 'A', borramos el setpoint
// Si se presiona 'A', borramos el setpoint
    | setpoint = 00;
    | // Mostrar el setpoint borrado en la segunda línea del LCD
    | lcd.setCursor(7, 1); // Colocarse en el espacio después de "Setpoint: "
    | lcd.print(" "); // Limpiar el valor anterior
    | lcd.setCursor(7, 1);
    | lcd.print(setpoint);
    | digitalWrite(ventiladorPin, LOW);
    | digitalWrite(niquelinaPin, LOW);
    | }
}
```

Figura 2.12 Ingreso de un nuevo *set point*

- Control de los actuadores

Para el control de los actuadores (niquelina y ventilador) se usó la lógica descrita en las líneas de código de la Figura 2.13, la cual consiste en que si se encuentra el *set point* fuera de rango la niquelina y ventilador se encontrarán apagados, luego si la temperatura de la estación es mayor al *set point* más la histéresis se debe encender el ventilador y si la temperatura de la estación es menor al *set point* más la histéresis se debe encender la niquelina.

```
    if ( setpoint < 15 || setpoint > 90) {
    |     digitalWrite(ventiladorPin, LOW);
    |     digitalWrite(niquelinaPin, LOW);
    |     }
else if ( setpoint = constrain(setpoint, 0, 99) ){

if ( temperatura >= setpoint + histeresis) {
|     digitalWrite(niquelinaPin, LOW);
|     digitalWrite(ventiladorPin, HIGH);
|     }

if (temperatura <= setpoint - histeresis) {
|     digitalWrite(niquelinaPin, HIGH);
|     digitalWrite(ventiladorPin, LOW);
|     }
}
}
```

Figura 2.13 Control de actuadores

En el Anexo III, lámina 1 se muestra el diagrama de flujo para el algoritmo de control local.

Algoritmo de control remoto

Para el control remoto se realizó el algoritmo en lenguaje FBD en el software Portal TIA V15, este consiste en controlar la niquelina y el ventilador de la estación mediante la lectura de un PWM y el ingreso de un *set point*. En la Figura 2.14 se encuentran todas las variables físicas de entrada como son las perturbaciones de la estación, la activación del control remoto y la señal PWM emitida por el sensor NTC, variables de salida para el control de la niquelina y el ventilador, y por último las variables remotas como marcas, y direcciones para el HMI.

	Nombre	Tipo de datos	Dirección	Rema...	Acces...	Escrib...	Visibl...
1	Temp_in	Word	%IW64	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	Temp_norm	Real	%MD100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	Temp_Sca	Real	%MD150	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	sp	Real	%MD200	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	ventilador	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	Niquelina	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	Out_Range	Bool	%M1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	Histeresis_Mayor	Real	%MD250	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	Histeresis_Menor	Real	%MD300	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	Insade_Range	Bool	%Q10.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	Remote_mode	Bool	%I8.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	Perturbacion_Resistencia	Bool	%I8.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	Perturbacion_Ventilador	Bool	%I8.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	Pertrubacion_R	Bool	%Q11.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
15	Perturbacion_V	Bool	%Q12.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 2.14 Tabla de variables del algoritmo de control

Para el control remoto de la planta en el LTI (Laboratorio de Tecnología Industrial) se necesita un PLC con módulos de expansión de entradas digitales y analógicas. En la Figura 2.15 se tiene el diagrama de conexión del PLC con la planta.

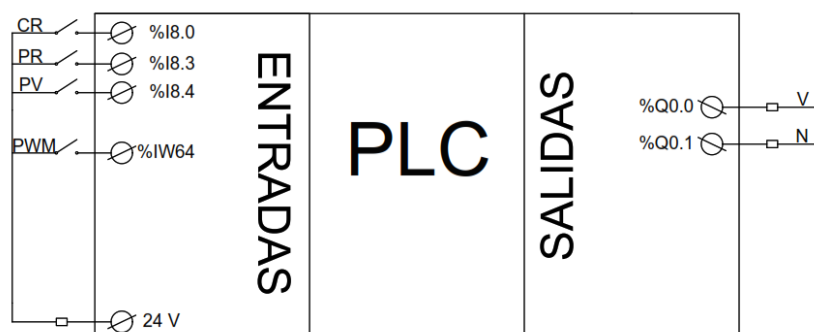


Figura 2.15 Diagrama de entradas y salidas del PLC

- Lectura de sensor

Para la lectura del sensor en el PLC se utilizó la señal acondicionada por *software* para transformarla en una señal PWM que será leída por la entrada analógica %IW64 del PLC. En la Figura 2.16 se encuentran las líneas de código para convertir la señal en una PWM.

```
// Convertir la temperatura a un valor PWM entre 0 y 255
int pwmValue = map(temperatura, 15, 90, 51, 255);

// Convertir el valor PWM de 0-255 a un valor de 0-1023
// (el rango utilizado por la función analogWrite en Arduino Mega)
int pwmValueMapped = map(pwmValue, 51, 255, , 1023);

// Escribir el valor PWM en el pin PWM
analogWrite(pwmPin, pwmValueMapped / 4);
```

Figura 2.16 Código de la temperatura en señal PWM

Luego esta señal PWM será enviada hacia el PLC, para desarrollar el algoritmo de lectura se utilizan tres tipos de variables de temperatura de entrada, normalizada y escalada. Como se muestra en la Figura 2.17 el primer bloque FBD es para normalizar la señal. Se establece el rango de unidades de procesamiento a la cual trabaja la señal PWM, la cual corresponde a 1 [V] a 5 [V] que representa los 2765 a 13824 unidades de procesamiento respectivamente. En el segundo bloque se define la variable normalizada y el rango de temperatura a la que trabaja la planta este va a ir desde 15°C a 90°C y al final se obtiene la temperatura escalada con la cual se desarrollará el algoritmo de control.

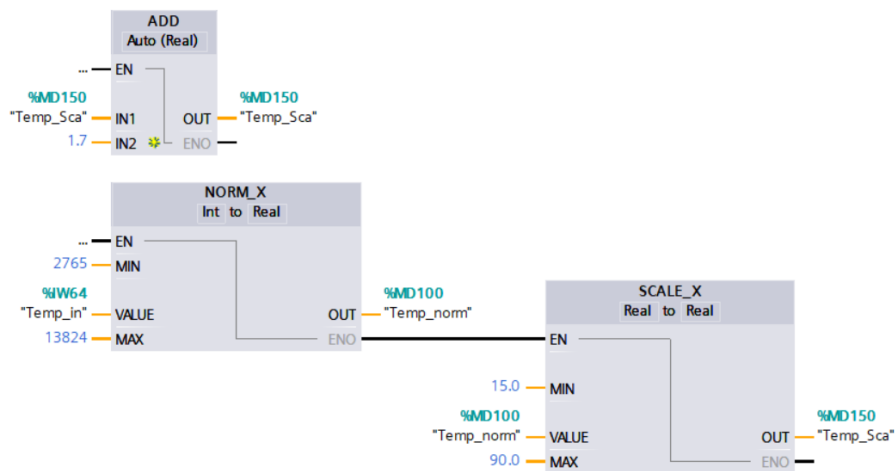


Figura 2.17 Algoritmo de lectura del sensor

- Activación de la niquelina y ventilador

Para el control del ventilador se utilizan bloques de contacto donde si el *set point* es menor a la temperatura escalada este activa el ventilador, y si el *set point* es mayor a la temperatura escalada se enciende la niquelina, a esta lógica se añade una condición de que si la temperatura escalada se encuentra dentro de la histéresis se desactivan los dos actuadores. En la Figura 2.18 se encuentre el algoritmo de control.

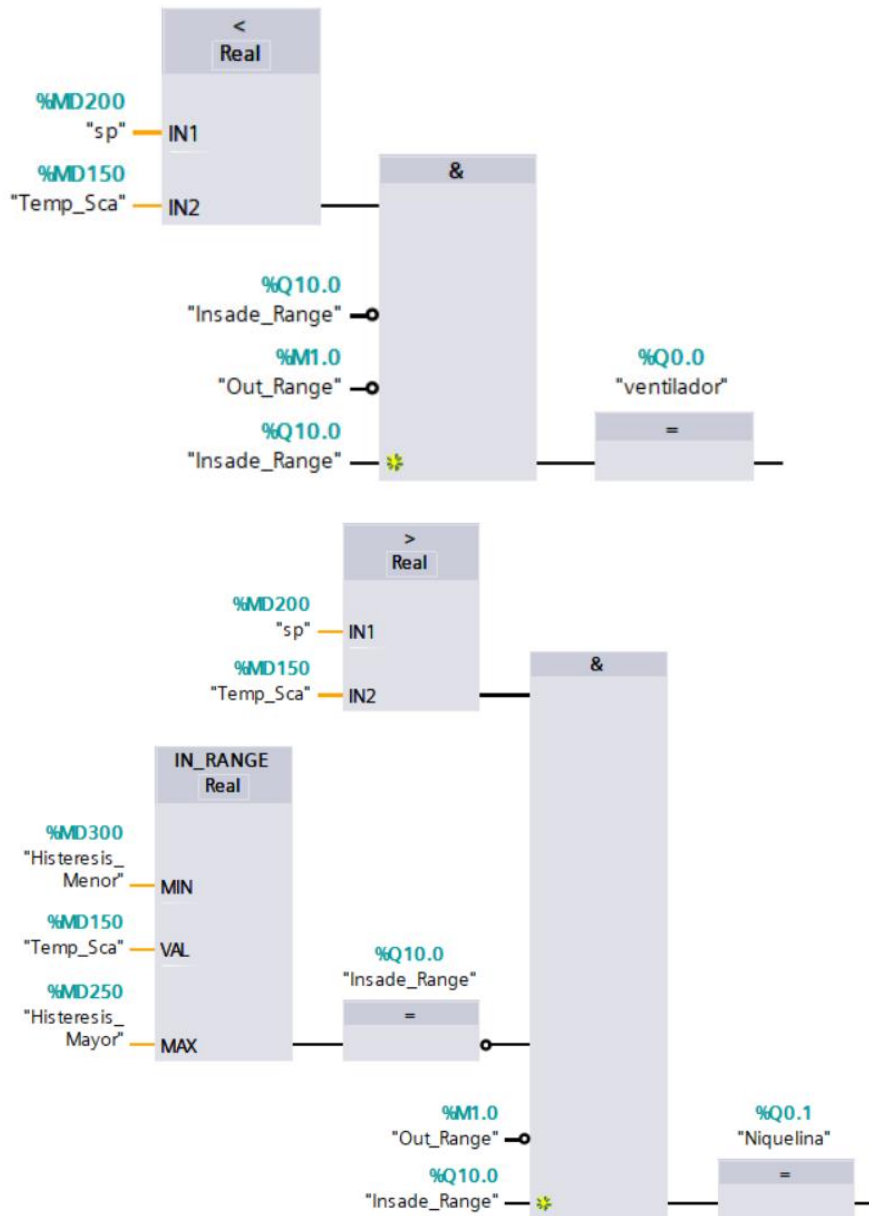


Figura 2.18 Algoritmo de Control de los actuadores

- Fuera de rango

El sistema trabaja de 15 [°C] a 90 [°C], por lo tanto se desarrolló un algoritmo para encender un aviso de “Fuera de Rango” cuando se ingrese un *set point* mayor o menor a este rango. En la Figura 2.19 se tiene los bloques de contacto que definen este rango con el cual trabaja el sistema.

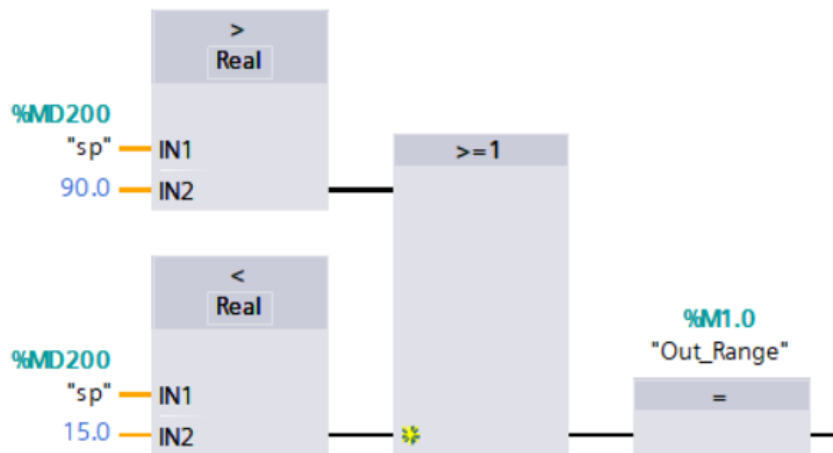


Figura 2.19 Algoritmo para definir el rango de trabajo

- Luces de perturbación

La planta va a trabajar con dos perturbaciones una niquelina y ventilador, por lo tanto, en la Figura 2.20 se encuentran los bloques de contacto que reciben la señal para activar las luces del HMI.

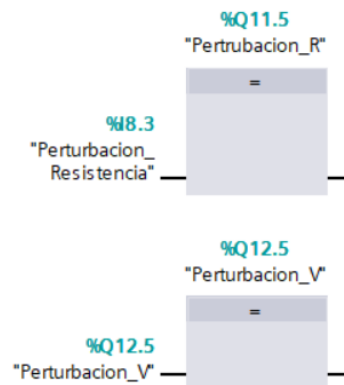


Figura 2.20 Algoritmo de las luces de perturbación

En el Anexo III, lámina 2 se muestra el diagrama de flujo para el algoritmo de control remoto.

2.2 Interfaz Humano Máquina

Se diseñó una Interfaz Humano Máquina (HMI) con tres ventanas que permiten al operador tener un control remoto de los actuadores, monitorear la variable del proceso y supervisar el control *on/off* de temperatura de la planta. Para su diseño se utilizó el *software* Portal TIA V15 y el WinCC Runtime y para la presentación de los gráficos se cumplió con las normas IEC 1082-1 y ANSI/ISA-S5.5-1985.

Se desarrolló una HMI con tres ventanas. En la principal se encuentra el menú de inicio con tres botones que permiten navegar al usuario a las ventanas del proceso de la planta, la presentación del *trend* de curvas, y las conexiones que tiene el PLC. Esta ventana se muestra en la Figura 2.21.



Figura 2.21 Ventana principal de la HMI.

En la Figura 2.22 se observa la ventana del proceso. En esta se ingresa el *set point* deseado, se monitorea la variable del proceso, se supervisa el encendido y apagado de los actuadores de la planta y tiene tres botones que permiten navegar hacia las otras ventanas.



Figura 2.22 Ventana del proceso.

En la ventana de la Figura 2.23 se muestra el gráfico del *trend* de curvas. Aquí se observa el comportamiento en tiempo real de la variable del proceso de la planta con respecto a la banda de histéresis de 2 [°C] y el *set point* deseado.



Figura 2.23 Ventana del *trend* de curvas.

2.3 Registro de datos

La planta didáctica trabaja con dos diferentes controladores (Arduino y PLC) según el modo de control seleccionado local o remoto, por lo tanto, para el registro de datos se utilizan técnicas diferentes para cada caso.

Registro de datos para control local

El registro de datos se realiza mediante la integración del Arduino con Excel *Data Streamer*, esta herramienta permite la transmisión de datos en tiempo real desde el microcontrolador a una hoja de cálculo de Excel.

Para enviar los datos al puerto serial del Arduino, se estableció en las líneas de código la instrucción de enviar el valor de la variable de temperatura y del *set point*, como se muestra en la Figura 2.24.

```
Serial.print(temperatura);  
Serial.print(",");  
Serial.print(setpoint);  
Serial.println();
```

Figura 2.24 Línea de código para enviar datos al puerto serial.

Mediante una hoja de cálculo en *Microsoft Data Streamer for Excel* y la conexión de la PC con el Arduino, se visualizarán los datos de temperatura y *set point* cada cierto tiempo y se tendrá un resumen en un gráfico de dicho comportamiento (Ver Figura 2.25). En la Figura 2.26 se muestra la gráfica generada con los datos registrados.



REGISTRO DE DATOS DE LA PLANTA DE CONTROL ON/OFF DE TEMPERATURA
MODO REMOTO



Entrada de datos (desde Arduino Uno (COM4))

Los datos procedentes del origen de datos actual aparecerán abajo según se reciban.

Datos actuales

TIME	TEMPERATURA [°C]	SETPOINT [°C]
CH8	CH9	CH10
45484.5029	27.38	0

Información histórica

TIME	TEMPERATURA [°C]	SETPOINT [°C]
45484.5029	27.38	0
45484.5029	27.38	0
45484.50289	27.38	0

Figura 2.25 Hoja de cálculo del registro de datos.

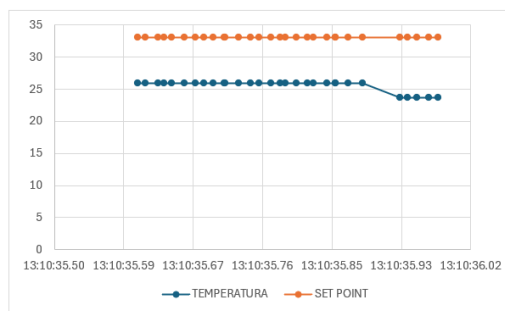


Figura 2.26 Gráfica generada con los datos registrados

Registro de datos para control remoto

El registro de datos en el control remoto no se puede realizar en tiempo real ya que se necesitan extensiones adicionales de la licencia del software Portal TIA, por lo tanto, mediante el *trend* de curvas de la HMI se exportarán los datos tomados de la variable de proceso y el *set point*.

Para exportar los datos se habilitan en propiedades de la curva, la herramienta de “Exportar datos”, este paso se muestra en la Figura 2.27.

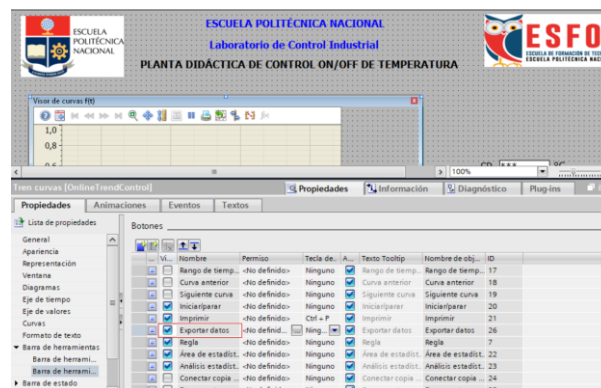


Figura 2.27 Herramienta Exportar datos.

Luego se inicia la simulación en el software WinCC Runtime, donde al haber obtenido los datos deseados, se hace click en el icono con flecha verde que se muestra en la Figura 2.28.

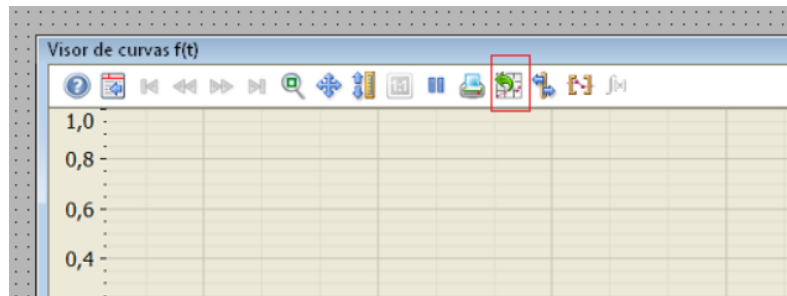


Figura 2.28 Ícono para exportación de datos.

Por último, los datos se exportarán en un archivo de tipo .csv y se guardará en una dirección seleccionada del computador. En la Figura 2.29 se muestra la hoja de Excel con los datos exportados del visor de curvas.

	A	B	C	D	E	F	G	H	I	J	K
1	Curva_1 Time;"Curva_1 ValueY";"Curva_2 Time";"Curva_2 ValueY";"Curva_3 Time";"Curva_3 ValueY";"Curva_4 Time";"Curva_4 ValueY"										
2	12:02:58,873;26,5;12:02:58,873;23,5;12:02:58,873;25;12:02:58,873;25,8576736450195										
3	12:02:59,875;26,5;12:02:59,875;23,5;12:02:59,875;25;12:02:59,875;25,8576736450195										
4	12:03:00,879;26,5;12:03:00,879;23,5;12:03:00,879;25;12:03:00,879;25,8644542694092										
5	12:03:01,879;26,5;12:03:01,879;23,5;12:03:01,879;25;12:03:01,879;25,8712348937988										
6	12:03:02,881;26,5;12:03:02,881;23,5;12:03:02,881;25;12:03:02,881;25,7559452056885										
7	12:03:03,883;26,5;12:03:03,883;23,5;12:03:03,883;25;12:03:03,883;24,9760360717773										
8	12:03:04,885;26,5;12:03:04,885;23,5;12:03:04,885;25;12:03:04,885;25,7152538299561										
9	12:03:05,886;26,5;12:03:05,886;23,5;12:03:05,886;25;12:03:05,886;25,7084732055664										
10	12:03:06,889;26,5;12:03:06,889;23,5;12:03:06,889;25;12:03:06,889;25,8169822692871										
11	12:03:07,891;26,5;12:03:07,891;23,5;12:03:07,891;25;12:03:07,891;27,2547245025635										
12	12:03:08,893;26,5;12:03:08,893;23,5;12:03:08,893;25;12:03:08,893;25,8712348937988										
13	12:03:09,895;26,5;12:03:09,895;23,5;12:03:09,895;25;12:03:09,895;27,2615051269531										
14	12:03:10,897;26,5;12:03:10,897;23,5;12:03:10,897;25;12:03:10,897;27,2615051269531										
15											

Figura 2.29 Hoja de cálculo con los datos exportados.

3 RESULTADOS

3.1 Pruebas de funcionamiento

En las pruebas de funcionamiento de las plantas de control *ON/OFF* de temperatura, se tuvo en cuenta varios parámetros para asegurar la precisión y fiabilidad de los sistemas. A continuación, se detallarán cada uno de ellos.

Calibración de los sensores de temperatura

Para la calibración del sensor, se realizó una comparación entre el sensor NTC 10K de la planta y una termocupla, con esto se asegura de que la lectura del sensor en el Arduino y su acondicionamiento sea preciso y fiable. En la Tabla 3.1 y Tabla 3.2 se muestran los valores tomados con los sensores y el termopar respectivamente para las dos plantas y el cálculo de errores, se observa que el máximo error de la planta 1 es de 3% y el de la planta 2 es de 2.7 %.

Tabla 3.1 Cálculo de errores para lectura del sensor de la planta 1.

Temperatura [°C]	Temperatura termopar [°C]	Error absoluto [°C]	Error relativo [%]
25.7	25.0	0.7	2.8
28.36	28.4	0.04	0.14
35.67	35.9	0.23	0.64
40.5	39.7	0.8	2.01
58.18	60	1.82	3

Tabla 3.2 Cálculo de errores para lectura del sensor de la planta 2.

Temperatura [°C]	Temperatura termopar [°C]	Error absoluto [°C]	Error relativo [%]
26.67	26.1	0.57	2.1
28.3	28.27	0.03	0.1
36.46	35.5	0.96	2.7
41.25	41.3	0.05	0.12
60.37	61.9	1.53	2.5

Pruebas de control *ON/OFF* en modo local

Se probó el programa desarrollado en Arduino para el control local de los actuadores y de la temperatura de la estación. Para esta prueba se consideró, que se pueda ingresar un *set point* con el teclado matricial, que haya una lectura correcta del sensor NTC 10K, y que haya una respuesta lógica al encendido y apagado de los actuadores con respecto a la variación de temperatura y al grado de histéresis usado. En la Tabla 3.3 se muestra el estado de las pruebas de funcionamiento realizadas para las dos plantas.

Tabla 3.3 Pruebas de control en plantas de temperatura (modo local)

Parámetro	Estado	
	Planta 1	Planta 2
Ingreso de <i>set point</i>	Ok	Ok
Lectura de temperatura (sensor NTC 10K)	Ok	Ok
Activación de niquelina de control	Ok	Ok
Activación de ventilador de control	Ok	Ok
Activación de actuadores de perturbación	Ok	Ok
Activación de luces de indicación	Ok	Ok
Control de la variable de proceso	Ok	Ok
Indicación de pantalla LCD	Ok	Ok

Pruebas de control *ON/OFF* en modo remoto

Se probó el algoritmo desarrollado en lenguaje FBD en el PLC y el control y supervisión de las plantas a través de la HMI. Se comprobó que la variable de proceso leída por la entrada del PLC sea igual a la variable del proceso mostrada en el LCD de la planta, luego se procedió a ingresar un *set point* para comprobar el control de los actuadores y verificar su encendido y apagado en la HMI y en la planta. También se evaluó el cambio de temperatura, la histéresis y el *set point* en el visor de curvas. En la Tabla 3.4 se muestra el estado de las pruebas de funcionamiento realizadas para las dos plantas.

Tabla 3.4 Pruebas de control en plantas de temperatura (modo remoto)

Parámetro	Estado	
	Planta 1	Planta 2
Ingreso de <i>set point</i>	Ok	Ok
Lectura de temperatura (sensor NTC 10K)	Ok	Ok
Activación de niquelina de control	Ok	Ok
Activación de ventilador de control	Ok	Ok
Lectura de estado de actuadores de perturbación	Ok	Ok
Activación de luces de indicación	Ok	Ok
Control de la variable de proceso	Ok	Ok
Registro de curvas	Ok	Ok

Pruebas de registros de datos

Se comprobó la configuración en Microsoft Excel con *Data Streamer* para registrar y visualizar en tiempo real la temperatura de la estación y el *set point*. Para esto se ingresaron varios valores en las dos plantas de temperatura para comprobar su registro. En la Tabla 3.5 se muestra el estado de las pruebas de registro de datos realizadas para las dos plantas.

Tabla 3.5 Pruebas de registro de datos (modo local)

Parámetro	Estado	
	Planta 1	Planta 2
Comunicación Arduino y PC	Ok	Ok
Datos leídos	Ok	Ok
Gráfica de datos registrados	Ok	Ok
Archivo de Excel generado	Ok	Ok

Además, se realizaron pruebas del registro de datos en la HMI mediante el *trend* de curvas para los dos sistemas, obteniendo los respectivos archivos .csv con los datos almacenados. En la Tabla 3.6 se muestra el estado de las pruebas de registro de datos realizadas para las dos plantas.

Tabla 3.6 Pruebas de registro de datos (modo remoto)

Parámetro	Estado	
	Planta 1	Planta 2
Comunicación sistema y PLC	Ok	Ok
Datos leídos	Ok	Ok
Gráfica de datos registrados	Ok	Ok
Archivo de Excel generado	Ok	Ok

Finalmente, en la Figura 3.1 se presenta un video de las pruebas realizadas a las plantas de control *ON/OFF* de temperatura, al cual se accede mediante el código QR correspondiente.

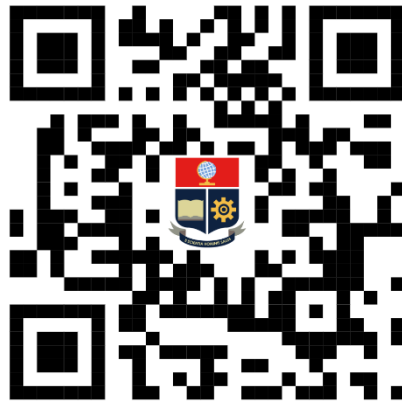


Figura 3.1 Código QR del video de pruebas de funcionamiento

(Enlace: <https://youtu.be/ZXdMYQ-Ymb0>)

3.2 Manual de usuario y mantenimiento

Para el correcto uso de los módulos didácticos, se mostrará en un video de manual de usuario donde se expondrá el encendido del módulo, la selección del control local o remoto, el ingreso del *set point* en el teclado matricial o en la HMI y las conexiones del módulo con el PLC. En la Figura 3.2 se encuentra el código QR para acceder al video del manual de usuario.

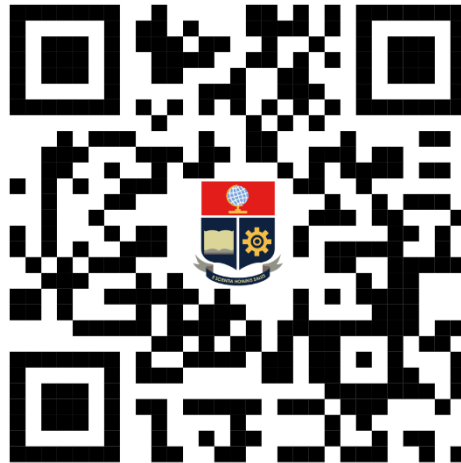


Figura 3.2 Código QR del video del manual de usuario

(Enlace: <https://youtu.be/kmqOPGx5leQ>)

En la Figura 3.3 se encuentra el código QR para acceder al video de manual de mantenimiento, donde se explica las actividades que se realizan a los módulos para mantener la vida útil de los elementos, conocer sobre sus conexiones y las posibles soluciones a problemas que puede presentar.

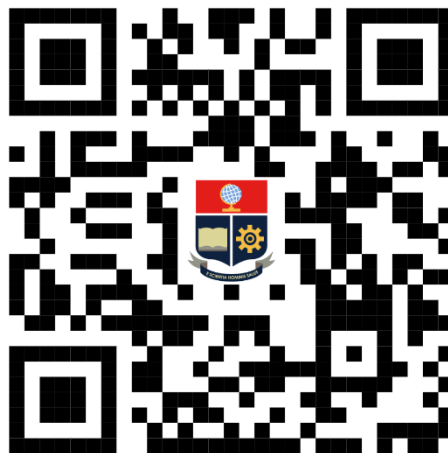


Figura 3.3 Código QR del video del manual de mantenimiento

(Enlace: <https://youtu.be/0Lo26yLfD2E>)

4 CONCLUSIONES

- El acondicionamiento del sensor utilizando un microcontrolador (acondicionamiento por *software*) dio excelentes resultados para implementar la acción de control *ON/OFF* al obtener un error relativo de 2% con la señal tomada como referencia de un termopar. El algoritmo desarrollado en Arduino incluye fórmulas para transformar la variable de voltaje a temperatura y poder visualizarla a través de un LCD y en la ventana HMI.
- El algoritmo desarrollado en Portal TIA V15 permitió controlar la temperatura de las plantas implementadas, ya que a través de la acción de control *ON/OFF* se comandan los actuadores, los cuáles mantienen la variable del proceso dentro de la banda de histéresis establecida.
- Los sistemas construidos disponen de dos algoritmos de control, uno desarrollado a través de la plataforma Arduino IDE y el otro a través del Portal TIA V15 que corresponden a los modos de operación local y remoto respectivamente, por lo que las plantas pueden ser utilizadas de forma didáctica a través de un teclado matricial y una pantalla LCD (modo local) y a través de un PLC y una HMI (modo remoto).
- La lectura de señales PWM en forma de voltaje provista por el sensor y receptada por el PLC fue fundamental para el control remoto de la planta. Tomando en cuenta que el PLC S7 1200 solo trabaja con entradas analógicas de voltaje, fue muy útil la generación de señales PWM para la normalización y escalamiento de la variable del proceso.
- El registro de datos y sus representaciones en gráficas y tablas fue importante para el análisis e interpretación de los resultados del sistema. Esta información permite al operador visualizar e identificar fallos en él módulo, y comprobar el correcto funcionamiento de la planta.
- En las pruebas realizadas, se observaron las ventajas que tienen los dispositivos lógicos programables ante los microcontroladores para el control de variables físicas. Teniendo como resultado, que al usar en modo remoto el controlador PLC muestra una respuesta más rápida en el encendido y apagado de los actuadores, una lectura más precisa de la variable de proceso y un control más eficiente para que la temperatura de la estación se mantenga dentro de la banda de histéresis establecida de ± 1 [°C].

- El manual de usuario y mantenimiento permiten al operador usar de forma correcta los módulos didácticos, arreglar posibles problemas que pueda presentar la planta y mantener la vida útil de los elementos de los módulos.

5 RECOMENDACIONES

- Con las pruebas realizadas a los módulos se recomienda que el rango de trabajo de las plantas sea entre 20 [°C] y 60 [°C] por seguridad y protección de los elementos. Este rango garantiza un rendimiento fiable del sistema y protege al operador.
- Si las plantas han alcanzado una temperatura mayor a 40 [°C] y se desea desconectar, se recomienda enfriar la planta antes de proceder a su desconexión. Con este paso se evita daños por calor residual en la estación y se reduce las tensiones térmicas en los materiales. El enfriamiento previo ayuda a proteger los componentes sensibles del sistema y aumentar la vida útil de los elementos.
- Evitar el desgaste de los actuadores por ciclos frecuentes de encendido y apagado, por lo que se sugiere utilizar un correcto rango de histéresis comprendido entre ± 1 [°C] y ± 2 [°C] o la posible utilización de acciones de control más complejas como PID (Proporcional Integral Derivativa).
- Para mejorar el sistema de registro de datos se recomienda realizar un almacenamiento en la nube. Para esto se podría utilizar un módulo de comunicación inalámbrica, un servidor en la Nube y un protocolo de comunicación.
- Para reducir el consumo energético de los módulos se puede desarrollar un modo de apagado automático. El cuál consista que después de un cierto tiempo de no haber ingresado ningún *set point* y la planta este en modo de inicio, se apague el sistema automáticamente.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] OMEGA a Driver Omega brand, «Controladores de temperatura,» 2023. [En línea]. Available: <https://n9.cl/0726cl>. [Último acceso: 6 Junio 2024].
- [2] Universidad Don Bosco, «Tema: El controlador ON-OFF,» 25 Agosto 2019. [En línea]. Available: <https://n9.cl/1m6hci> [Último acceso: 27 junio 2024].
- [3] Monografías, «Sistemas de Regulación y Control,» 22 Julio 2016. [En línea]. Available: <https://www.monografias.com/trabajos109/sistemas-regulacion-y-control/sistemas-regulacion-y-control2>. [Último acceso: 13 6 2024].
- [4] Iñigo Gútiérrez, «PLC S7-1200 de Siemens: características generales,» 17 abril 2023. [En línea]. Available: <https://programacionsiemens.com/s7-1200/>. [Último acceso: 13 Junio 2024].
- [5] Suiler Altamirano, «SELECCIÓN DE PLC S7-1200: GUÍA DE APLICACIONES INDUSTRIALES,» 21 Julio 2023. [En línea]. Available: <https://n9.cl/kxraf>. [Último acceso: 13 Junio 2024].
- [6] SICMA21, «¿Qué es un PLC y cuándo se usa?,» 14 Octubre 2021. [En línea]. Available: <https://www.sicma21.com/que-es-un-plc/>. [Último acceso: 13 Junio 2024].
- [7] Opiron, «Codesys: IL, SFC, LD, FBD o ST,» 12 Marzo 2017. [En línea]. Available: <https://www.opiron.com/codesys-mejor-lenguaje-programacion/>. [Último acceso: 13 Junio 2024].
- [8] Universidad Don Bosco, «Tema: Red Profinet con S7-1200,» 19 Febrero 2015. [En línea]. Available: <https://n9.cl/dcldk>. [Último acceso: 13 Junio 2024].
- [9] Autycom , «¿Qué es mejor un HMI Siemens o una aplicación de PC?,» 7 Junio 2019. [En línea]. Available: <https://www.autycom.com/hmi-siemens-o-aplicacion-pc/>. [Último acceso: 13 Junio 2024].
- [10] COPADATA, «¿Qué es una HMI?,» 7 Junio 2019. [En línea]. Available: <https://n9.cl/l6zb> . [Último acceso: 13 Junio 2024].

7 ANEXOS

ANEXO I. Resultado obtenido del Turnitin.

ANEXO II. Certificado de funcionamiento del Trabajo de Integración Curricular

ANEXO III. Diagrama de flujo.

ANEXO I. Resultado obtenido del Turnitin.

Reporte de similitud obtenido de Turnitin

Yo, Alan Daniel Cuenca Sánchez, como director del presente Trabajo de Integración Curricular, certifico que el siguiente es el resultado de la evaluación de similitud realizado por la plataforma Turnitin:

Autor de la entrega:	MARIA CRISTINA JIBAJA TRAVEZ
Título del ejercicio:	TIC 2024A
Título de la entrega:	TIC JIBAJA
Nombre del archivo:	TIC_JIBAJA.pdf
Tamaño del archivo:	643.63K
Total páginas:	32
Total de palabras:	6,331
Total de caracteres:	30,832
Fecha de entrega:	19-jul.-2024 01:19p. m. (UTC-0500)
Identificador de la entre...	2419270389

TIC JIBAJA

INFORME DE ORIGINALIDAD

8%

INDICE DE SIMILITUD

8%

FUENTES DE INTERNET

1%

PUBLICACIONES

3%

TRABAJOS DEL
ESTUDIANTE

Excluir citas

Activo

Excluir coincidencias

Apagado

Excluir bibliografía

Activo

ANEXO II. Certificado de funcionamiento del Trabajo de Integración Curricular.

DMQ, 18 de Julio de 2024

Yo, Alan Daniel Cuenca Sánchez, docente a tiempo completo de la Escuela Politécnica Nacional y como director de este trabajo de integración curricular, certifico que he constatado el correcto funcionamiento de los Módulos de control de temperatura mediante la acción de control *on/off*, los cuales fueron implementados por la estudiante María Cristina Jibaja Trávez.

El proyecto cumple con los requerimientos necesarios para que cualquier usuario pueda usar los sistemas.

Ing. Alan Daniel Cuenca Sánchez. MSc.

DIRECTOR

ANEXO III. Diagrama de Flujo.

