

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas  
Carrera De Ingeniería En Mantenimiento Eléctrico

**IMPLEMENTACIÓN DE UN HORNO AUTOMÁTICO DE CONVECCIÓN PARA EL  
SOLDADO DE COMPONENTES ELECTRÓNICOS DE MONTAJE SUPERFICIAL.**

Trabajo de grado presentado ante la Universidad Técnica del Norte previo a la  
obtención del título de grado de Ingeniero en Mantenimiento Eléctrico.

Autor:

Rotman Wladimir Quimbita Mosquera

Director:

Ing. Francisco Naranjo Cobo MSc.

Ibarra – Ecuador

Octubre 2020



# UNIVERSIDAD TÉCNICA DEL NORTE

## BIBLIOTECA UNIVERSITARIA

### AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

#### 1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que se publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
<b>CÉDULA DE IDENTIDAD:</b>	050398974-1		
<b>APELLIDOS Y NOMBRES:</b>	Quimbita Mosquera Rotman Wladimir		
<b>DIRECCIÓN:</b>	Latacunga, Parroquia Aláquez, calles Vicente León y García Moreno		
<b>EMAIL:</b>	rwquimbitam@utn.edu.ec		
<b>TELÉFONO FIJO:</b>	032262452	<b>TELÉFONO MÓVIL:</b>	0987838966

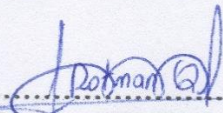
DATOS DE LA OBRA	
<b>TÍTULO:</b>	IMPLEMENTACIÓN DE UN HORNO AUTOMÁTICO DE CONVECCIÓN PARA EL SOLDADO DE COMPONENTES ELECTRÓNICOS DE MONTAJE SUPERFICIAL.
<b>AUTOR (ES):</b>	Quimbita Mosquera Rotman Wladimir
<b>FECHA:</b> <b>DD/MM/AAAA</b>	08/10/2020
<b>SOLO PARA TRABAJOS DE GRADO</b>	
<b>PROGRAMA:</b>	<input checked="" type="checkbox"/> <b>PREGRADO</b> <input type="checkbox"/> <b>POSTGRADO</b>
<b>TITULO POR EL QUE OPTA:</b>	Ingeniería en Mantenimiento Eléctrico
<b>ASESOR/DIRECTOR</b>	Ing. Francisco Naranjo Cobo MSc.

## 2. CONSTANCIA

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 08 días del mes de octubre de 2020

**EL AUTOR:**

  
.....  
Rotman Wladimir Quimbita Mosquera  
050398974-1



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO**

Yo, Francisco Naranjo en calidad de tutor del señor estudiante, Quimbita Mosquera Rotman Wladimir certifico que ha cumplido con las normas establecidas en la elaboración del trabajo de investigación titulado: **“IMPLEMENTACIÓN DE UN HORNO AUTOMÁTICO DE CONVECCIÓN PARA EL SOLDADO DE COMPONENTES ELECTRÓNICOS DE MONTAJE SUPERFICIAL”**. Para la obtención del título de Ingeniero en Mantenimiento Eléctrico; aprobando la defensa, impresión y empastado.



.....  
Ing. Francisco Naranjo MSc.

**DIRECTOR DE TRABAJO DE GRADO**

## DEDICATORIA

El presente trabajo de grado se lo dedico a mis padres, quienes desde niño me inculcaron que la mejor herencia que me pueden dar es la educación, por su incondicional apoyo tanto emocional como económico y sobre todo por su inmenso amor. A mi tío Víctor Hugo (+) quién fue mi guía y fomentó mi deseo de superación con su gran ejemplo. A mi esposa Diana, por sus palabras de aliento en momentos difíciles a lo largo de la carrera, por su compañía, por su amor y sin lugar a dudas por siempre creer en mí y en mis capacidades. A mis hijos Alejandro y Adrián quienes son mi mayor motivación para salir adelante, por sus ocurrencias que alegran mis días, sus muestras de cariño que hacían mis días más llevaderos en momentos de mayor estrés. A mis amigos y compañeros de clase, trabajo en equipo, conocimiento compartido, alegrías, tristezas, experiencias vividas en el transcurso de estos 5 años de carrera universitaria.

## **AGRADECIMIENTOS**

Agradezco a Jehová por bendecirme y brindarme salud y vida, sabiduría para poder finalizar con éxito el presente trabajo de grado y con ello la finalización de la carrera de pregrado. Agradezco a mi casona universitaria, mi querida Universidad Técnica del Norte, la cual me permitió acceder a una educación de calidad gracias a sus docentes de excelencia. A mi familia, principalmente a mis padres y a mi esposa quienes me han ayudado a levantarme de mis caídas, siempre creyeron en mí y han hecho de mí un hombre de bien, útil a la sociedad. A mi Tutor, Francisco Naranjo quien demostró interés y me sacó de muchos atascos en el desarrollo de este proyecto. A mis abuelitos Cristóbal, Hilda y Dolores quienes se preocupan por mí, se alegran de mis triunfos y me alientan en mis tropiezos.

## TABLA DE CONTENIDO

IDENTIFICACIÓN DE LA OBRA .....	ii
CONSTANCIA .....	iii
CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO.....	iv
DEDICATORIA .....	v
AGRADECIMIENTOS.....	vi
RESUMEN .....	xv
INTRODUCCIÓN .....	i
Planteamiento Del Problema.....	ii
Objetivos .....	iii
Objetivo General .....	iii
Objetivos Específicos.....	iii
Justificación .....	iii
Alcance .....	iv

### CAPÍTULO 1

Análisis Bibliográfico .....	1
1.1. Componentes electrónicos para circuitos impresos .....	1
1.1.1. Componentes electrónicos THD .....	1
1.1.2. Componentes electrónicos SMD .....	2
1.1.2.1. Ventajas de los componentes SMD .....	3
1.2. Aleaciones para soldadura de componentes de montaje superficial .....	3
1.3. Soldadura por Reflujo .....	4
1.3.1. Perfil de soldadura (curva de temperatura) .....	6
1.3.1.1. Pre calentamiento .....	7
1.3.1.2. Remojo .....	7
1.3.1.3. Reflujo .....	7
1.3.1.4. Enfriamiento.....	7
1.3.2. Refusión por Convección Forzada de Aire .....	8

1.3.2.1.	Transferencia de calor por convección forzada.....	9
1.3.2.2.	Ventiladores .....	11
1.3.2.3.	Ventajas y Desventajas de los Sistemas de Convección Forzada .....	12
1.3.3.	Hornos de Reflujo para Soldadura SMD .....	13
1.4.	Automatización del sistema térmico.....	14
1.4.1.	Sensores de temperatura .....	15
1.4.1.1.	Termopares (termocupla) .....	16
1.4.2.	Elemento final de control (Triacs) .....	17
1.4.3.	Sistemas de control .....	18
1.4.3.1.	Clasificación de los sistemas de control .....	18
1.4.3.1.	Componentes en un sistema de control.....	19
1.4.3.2.	Control digital.....	20
1.4.3.3.	Control de temperatura PID digital .....	21
1.4.4.	Procesamiento y adquisición de datos .....	24
1.4.4.1.	Tarjeta de desarrollo Arduino Mega 2560.....	24
1.4.5.	Interfaz Gráfica de Usuario.....	25
1.4.5.1.	Raspberry Pi 3 modelo B+ .....	25
1.4.5.2.	Instrumentación Virtual .....	26

## CAPÍTULO 2

Diseño Electrónico y Automatización .....	27
2.1. Introducción.....	27
2.1.1. Diseño de Hardware.....	27
2.1.1.1. Planta de temperatura .....	27
2.1.1.2. Módulo de control.....	28
2.1.2. Diagrama de bloques del sistema automatizado .....	29
2.1.1.1. Controlador .....	30
2.1.1.2. Sensor de temperatura .....	30
2.1.1.3. Lámparas Halógenas.....	32



2.1.1.4.	Ventilador convección .....	33
2.1.1.5.	Módulo de disparo .....	34
2.1.1.6.	Ventilador enfriamiento .....	38
2.1.1.7.	Transistor .....	39
2.1.1.8.	HMI (Interfaz Hombre Máquina) .....	41
2.1.1.9.	Fuente de alimentación DC .....	41
2.1.1.10.	Dispositivos de protección .....	42
2.1.2.	Control de fase directo .....	42
2.1.2.1.	Control de fase con Arduino .....	43
2.1.3.	Circuito detector de cruce por cero .....	45
2.1.4.	Diagrama de flujo del microcontrolador Arduino .....	47
2.1.5.	Diseño de diagrama eléctrico general .....	49
2.1.6.	Esquemático general distribución de pines Raspberry - Arduino .....	50
2.1.7.	Diseño de placas de circuito impreso (PCB) .....	52
2.1.7.1.	Circuito detector de cruce por cero y Módulo de disparo 3 .....	52
2.1.7.2.	Módulo de disparo 2 y 4 .....	52
2.1.7.3.	Módulo de disparo 1 y 5 .....	53
2.1.7.4.	Módulo sensores de temperatura .....	54
2.1.7.5.	Módulo ventiladores .....	55
2.2.	Diseño del control de temperatura .....	56
2.2.1.	Diseño del control PID .....	56
2.2.1.1.	Introducción .....	56
2.2.1.2.	Identificación de la dinámica del sistema .....	56
2.2.1.3.	Sintonía del controlador PID .....	59
2.3.	Diseño de la Interfaz Gráfica de Usuario (GUI) .....	60
2.4.	Pruebas de funcionamiento .....	61
<b>CAPÍTULO 3</b>		
	Implementación del Equipo de Soldadura .....	64

3.1.	Modificación del Horno.....	64
3.1.1.	Extracción de elementos innecesarios .....	64
3.1.2.	Incorporación de elementos necesarios .....	65
3.1.2.1.	Elementos calefactores .....	65
3.1.2.2.	Incorporación de ventiladores: convección y enfriamiento .....	65
3.1.2.3.	Incorporación de sensores de temperatura .....	66
3.1.2.4.	Aislamiento térmico .....	67
3.2.	Implementación del módulo de control.....	69
3.2.1.	Construcción de placas de control .....	69
3.2.2.	Instalación del sistema eléctrico y circuitos de control .....	70
3.3.	Implementación final del equipo .....	70
3.4.	Implementación del Control PID digital .....	71
3.5.	Pruebas y ajustes necesarios del control PID de temperatura .....	72
3.5.1.	PID sintonizado con el criterio de Ziegler and Nichols (sin sintonía fina) .....	72
3.5.2.	PID sintonizado con el criterio de Ziegler and Nichols (con sintonía fina) .....	73
3.5.3.	Filtrado de ruido en sensores aplicado al control PID .....	75
3.6.	Implementación de la Interfaz Gráfica de Usuario.....	75
3.6.1.	Configuración de la Raspberry PI.....	76
3.6.2.	Adecuación de la Interfaz Gráfica.....	77
3.7.	Prueba real de soldado .....	78
3.8.	Presupuesto para la construcción del equipo de soldadura.....	81
3.9.	Comparación de costos de equipos de soldadura para componentes SMD.....	82
3.10.	Manual de operación y mantenimiento del equipo automatizado .....	83
	CONCLUSIONES .....	86
	RECOMENDACIONES.....	87
	REFERENCIAS .....	88
	ANEXOS .....	94

# ÍNDICE DE FIGURAS

Fig. 1.1. Circuito integrado con encapsulado THT del tipo DIP.....	1
Fig. 1.2. Perfil de soldadura THT donde se aprecian los conos de soldadura.....	2
Fig. 1.3. Componentes THT radial y axial. ....	2
Fig. 1.4. Tecnología THT vs tecnología SMD.....	2
Fig. 1.5. Encapsulado SMD con y sin terminales. ....	3
Fig. 1.6. Opciones de soldadura por reflujo.....	5
Fig. 1.7. Perfil sugerido por el fabricante para soldadura con plomo Sn63Pb37.....	6
Fig. 1.8. Proceso de soldadura con horno.....	6
Fig. 1.9. Vista frontal de un horno de convección forzada.....	8
Fig. 1.10. Ejemplo de transferencia de calor por convección.....	9
Fig. 1.11. Hélice Axial de: a) Perfil delgado. b) Perfil sustentador.....	11
Fig. 1.12. Rodete centrífugo: a) Álabes radiales. b) Álabes hacia adelante. c) Álabes hacia atrás.....	12
Fig. 1.13. Elementos de un sistema de control de temperatura. ....	14
Fig. 1.14. Unión caliente y fría de un termopar.....	16
Fig. 1.15. Curvas características de voltaje vs temperatura en termopares.....	17
Fig. 1.16. Símbolo y nombres de terminales de un Triac.....	18
Fig. 1.17. Diagrama básico de un sistema de control digital. ....	21
Fig. 1.18. Aproximación rectangular para integral discreta. ....	22
Fig. 1.19. Diagrama de bloques del sistema de control PID digital.....	23
Fig. 2.1. Estructura del equipo de soldadura.....	27
Fig. 2.2. Diagrama de bloques del sistema automatizado.....	29
Fig. 2.3. Lámparas halógenas de cuarzo.....	32
Fig. 2.4. Medidas para rodete de ventilador.....	33
Fig. 2.5. Ventilador para horno de convección: a) Elaborado, b) Comercial. ....	33
Fig. 2.6. Módulo de disparo con TRIAC para carga resistiva.....	35
Fig. 2.7. Módulo de disparo con TRIAC para carga Inductiva.....	37
Fig. 2.8. Ventilador Brushless 4 Pines Con Control PWM.....	38
Fig. 2.9. Ventilador Brushless 2 Pines.....	39
Fig. 2.10. Circuito de control para ventiladores de enfriamiento.....	40
Fig. 2.11. Circuito reductor de voltaje.....	42
Fig. 2.12. Control de fase directo con Triac.....	43
Fig. 2.13. Pulsos de detección de cruce por cero.....	45

Fig. 2.14. Circuito detector de cruce por cero.....	45
Fig. 2.15. Señal de cruce por cero medida en osciloscopio. ....	47
Fig. 2.16. Diagrama de flujo del sistema automatizado. ....	48
Fig. 2.17. Diagrama eléctrico.....	50
Fig. 2.18. Esquemático general distribución de pines RPI3 - Arduino. ....	51
Fig. 2.19. Módulo de disparo 3 y circuito detector de cruce por cero - esquemático general. ....	52
Fig. 2.20. Módulo de disparo 3 y circuito detector de cruce por cero - diseño PCB y visualización 3D. .....	52
Fig. 2.21. Módulo de disparo 2 y 4 - esquemático general.....	53
Fig. 2.22. Módulo de disparo 2 y 4 - diseño PCB y visualización 3D. ....	53
Fig. 2.23. Módulo de disparo 1 y 5 - esquemático general.....	54
Fig. 2.24. Módulo de disparo 1 y 5 - diseño PCB y visualización 3D. ....	54
Fig. 2.25. Módulo sensores de temperatura - esquemático general.....	55
Fig. 2.26. Módulo sensores de temperatura - diseño PCB y visualización 3D. ....	55
Fig. 2.27. Módulo para ventiladores - esquemático general.....	55
Fig. 2.28. Módulo para ventiladores - diseño PCB y visualización 3D.....	56
Fig. 2.29. Respuesta en lazo abierto. ....	57
Fig. 2.30. Ventana de Identificación de sistemas. ....	58
Fig. 2.31. Ventana modelo de proceso.....	58
Fig. 2.32. Modelo matemático estimado. ....	59
Fig. 2.33. Respuesta del sistema con el método de Ziegler and Nichols. ....	60
Fig. 2.34. Interfaz gráfica de usuario HMI o GUI. ....	61
Fig. 2.35. Ventana para verificar salida de la interfaz gráfica de usuario. ....	61
Fig. 2.36. Conexión para la prueba de funcionamiento. ....	62
Fig. 2.37. Funcionamiento de la interfaz gráfica de usuario.....	62
Fig. 3.1. Horno Homotech TO-162H-1.....	64
Fig. 3.2. Extracción de elementos y modificación del Horno.....	64
Fig. 3.3. Elementos calefactores: a) Resistencias. b) Lámparas halógenas de cuarzo. ....	65
Fig. 3.4. Instalación de ventilador de convección. ....	66
Fig. 3.5. Instalación de ventilador de enfriamiento.....	66
Fig. 3.6. Instalación de sensores de temperatura. ....	67
Fig. 3.7. Barrera que retarda la transferencia de calor. ....	67
Fig. 3.8. Materiales termoaislantes: a) Fibra de vidrio. b) Silicato de calcio. c) Fibra cerámica. d) Lana de roca. e) Poliuretano. ....	68

Fig. 3.9. Aislamiento térmico del Horno.....	68
Fig. 3.10. Planta térmica terminada.....	69
Fig. 3.11. Placas de control.....	69
Fig. 3.12. Caja de acrílico para alojar toda la circuitería.....	70
Fig. 3.13. Instalación del sistema eléctrico y circuitos de control.....	70
Fig. 3.14. Equipo de soldadura SMD finalizado.....	71
Fig. 3.15. Diagrama de bloques del control de temperatura digital.....	71
Fig. 3.16. Implementación del control PID digital.....	72
Fig. 3.17. Implementación de las constantes de sintonía en Arduino.....	72
Fig. 3.18. Respuesta del control PID sin sintonía fina.....	73
Fig. 3.19. Respuesta del sistema con un aumento de ganancia proporcional.....	74
Fig. 3.20. Respuesta del control PID con sintonía fina.....	74
Fig. 3.21. Señal filtrada del sensor de temperatura.....	75
Fig. 3.22. Conexión remota de la Raspberry Pi.....	76
Fig. 3.23. Pantalla táctil configurada.....	77
Fig. 3.24. Programación para la transmisión de datos Raspberry Pi - Arduino.....	78
Fig. 3.25. Soldadura Sn63Pb37 y perfil de temperatura.....	78
Fig. 3.26. PCB diseñada para componentes SMD.....	79
Fig. 3.27. Colocación de soldadura en pasta (Sn63Pb37) en pads.....	79
Fig. 3.28. Ubicación de componentes SMD en pads.....	79
Fig. 3.29. Colocación de PCB en equipo de soldadura.....	80
Fig. 3.30. Medición de temperatura.....	80
Fig. 3.31. Prueba de soldadura.....	81
Fig. 3.32. Comparación de costos de producción de diferentes modelos de Hornos.....	83
Fig. 3.33. Operación del equipo de soldadura.....	84
Fig. 3.34. Hollín generado en lámpara halógena.....	84
Fig. 3.35. Bornes de conexión en base cerámica.....	85
Fig. 3.36. Revisión de partes móviles del motor universal.....	85

## ÍNDICE DE TABLAS

TABLA 1.1 RANGOS DE FUSIÓN Y LAS TEMPERATURAS MÁXIMAS DE REFLUJO DE ALGUNAS ALEACIONES DE SOLDADURA.....	4
TABLA 1.2 PROBLEMAS MÁS FRECUENTES CON SUS POSIBLES CAUSAS .....	8
TABLA 1.3 VALORES TÍPICOS DEL COEFICIENTE DE TRANSFERENCIA DE CALOR POR CONVECCIÓN...	10
TABLA 1.4 PRECIO DE HORNOS DE REFLUJO COMERCIALES.....	13
TABLA 1.5 TIPO DE SENSORES Y CARACTERÍSTICAS.....	15
TABLA 1.6 TIPO DE TERMOPARES Y CARACTERÍSTICAS .....	16
TABLA 2.1 ESPECIFICACIONES TÉCNICAS DEL TRANSDUCTOR DE TEMPERATURA .....	31
TABLA 2.2 PINES DE CONEXIÓN ENTRE MODULO MAX6675 Y ARDUINO MEGA.....	31
TABLA 2.3 DISTRIBUCIÓN PINES VENTILADOR BRUSHLESS .....	38
TABLA 2.4 ESPECIFICACIONES TÉCNICAS DE LA PANTALLA TÁCTIL TFT-LCD .....	41
TABLA 2.5 DESCRIPCIÓN DE PINES A UTILIZAR EN ARDUINO .....	51
TABLA 2.6 REGLA DE SINTONÍA DE ZIEGLER AND NICHOLS.....	59
TABLA 3.1 PROPIEDADES DE MATERIALES TERMOAISLANTES .....	68
TABLA 3.2 PRESUPUESTO DEL EQUIPO DE SOLDADURA .....	81
TABLA 3.3 COSTOS DE PRODUCCIÓN DE DIFERENTES MODELOS DE HORNOS.....	82

## RESUMEN

Por años se ha utilizado el método de soldadura tradicional con la utilización de pistolas de calor o cautines, esto debido a que los componentes electrónicos en su mayoría se encontraban en encapsulados grandes de agujeros pasantes. En la actualidad los componentes electrónicos de última generación son fabricados en encapsulados de montaje superficial, pero al ser sus pines pequeños se dificulta el proceso de soldado por métodos manuales “tradicionales” especialmente cuando la placa de circuito impreso tiene bastantes componentes a soldar. En el mercado existen equipos de soldadura automatizados con la utilización de un Horno de Reflujo, básicamente lo que hacen estos equipos es elevar la temperatura hasta un punto máximo que requiere el tipo de soldadura para fundirse y después se enfrían hasta llegar a temperatura ambiente, lastimosamente estos equipos se encuentran a un precio elevado. Por esta razón este estudio va dirigido al diseño e implementación de un Horno de Reflujo cuyas características se asemejen lo más posible a un equipo industrial. Para llevar a cabo tal propósito se realiza un control de temperatura basado en un control PID digital ya que es un controlador sencillo de implementar, pero con la robustez necesaria para presentar buenos resultados al ser sintonizado adecuadamente. En este sentido se ha utilizado un sistema embebido como lo es Arduino para implementar el controlador principal de temperatura y un miniordenador Raspberry Pi como contralor maestro que permite interactuar con el horno a través de una pantalla táctil. En el miniordenador se aloja la interfaz gráfica de usuario que permite enviar las ordenes al Arduino para iniciar o detener el proceso de soldadura, además de que permite mostrar los cambios de temperatura tanto numérica como gráfica en tiempo real. Los resultados obtenidos con el equipo finalizado se reflejan en la obtención de una curva de temperatura muy similar al perfil de temperatura proporcionado por el fabricante del tipo de suelda gracias a la automatización del equipo y al correcto desempeño del control de temperatura PID, por lo tanto, se concluye en la obtención de una soldadura confiable y dentro de los parámetros esperados.

Palabras clave: Horno Reflujo, Arduino, Raspberry Pi, Control Temperatura, Control PID, Interfaz Gráfica.

## Abstract

For years, the traditional welding method has been used with the use of heat guns or soldering irons, because the electronic components were mostly in large packages with through holes. At present, the latest generation electronic components are manufactured in surface-mount packages, but since their pins are small, the soldering process is difficult by “traditional” manual methods, especially when the printed circuit board has many components to solder. In the market there are automated welding equipment with the use of a Reflow Oven, basically what these equipment do is raise the temperature to a maximum point that requires the type of welding to melt and then cool down to room temperature, unfortunately, these equipment have a high price. For this reason, this study is aimed at the design and implementation of a Reflow Oven whose characteristics are as close as possible to industrial equipment. To carry out this purpose, a temperature control based on a digital PID control is carried out since it is a simple controller to implement, but with the necessary robustness to present good results when properly tuned. In this sense, an embedded system such as Arduino has been used to implement the main temperature controller and a Raspberry Pi minicomputer as a master controller that allows interaction with the oven through a touch screen. The minicomputer houses the graphical user interface that allows sending the orders to the Arduino to start or stop the welding process, in addition to showing the numerical and graphical temperature changes in real-time. The results obtained with the finished equipment are reflected in obtaining a temperature curve very similar to the temperature profile provided by the manufacturer of the type of weld thanks to the automation of the equipment and the correct performance of the PID temperature control, so it is concluded in obtaining a reliable weld and within the expected parameters.

Keywords: Reflow Oven, Arduino, Raspberry Pi, Temperature Control, PID Control, Graphic Interface.



## INTRODUCCIÓN

Imaginar la tecnología en el mundo actual sin soldadura con estaño o soldadura blanda es prácticamente imposible, siendo su campo de utilización verdaderamente extenso. Así pues, no importa si hablamos de tecnologías de la energía, de su conducción, telecomunicaciones, automatización o desarrollo de prototipos eléctricos y electrónicos. En todos estos campos las uniones de soldadura suponen una parte decisiva para que cada dispositivo trabaje tal y como ha sido planeado y desarrollado por el creador e inventor del producto (Manual\_Ersa, 2015).

Para el desarrollo de dispositivos tecnológicos se requiere que los diseños electrónicos sean plasmados en una placa de circuito impreso o PCB (Printed Circuit Board) en la cual los diferentes componentes electrónicos serán montados y soldados. De modo que, la función básica de la unión de soldadura es proporcionar una fuerte unión eléctrica y mecánica entre los componentes electrónicos y la placa de circuito impreso, proporcionando así una interconexión eléctrica mecánicamente robusta, segura y de baja impedancia entre varios componentes que comprenden el circuito (Gurmukh, 2014).

Actualmente, para la soldadura de componentes electrónicos se utilizan dos tecnologías de montajes: la tecnología de sujeción mediante orificios o agujeros pasantes Through Hole Technology (THT) y la Tecnología de Montaje Superficial (SMT). La sujeción mediante agujeros es la tecnología tradicional en donde los componentes son montados en agujeros que atraviesan la tarjeta, mientras la SMT usa dispositivos que se montan, como su nombre lo indica, sobre las superficies de la tarjeta (Placeres, 2002).

En la Tecnología Through Hole uno de los métodos para realizar el proceso de soldadura se emplean pistolas de calor o cautines, el uso de los mismos conlleva soldar punto a punto cada patilla del componente TH en el circuito impreso debiendo hacer esto en el menor tiempo posible ya que, algunas componentes electrónicas pierden sus propiedades semiconductoras o químicas si se someten a calor excesivo (Acuña, Chica, & Marquez, 2011).

El proceso de soldadura con pistolas de calor o cautines presenta algunas desventajas como son: el uso de lentes y herramientas especiales, requerimientos de iluminación, inhalación de humo e incluso peligro de quemaduras para el soldador, al finalizar el proceso se tiene acabados poco estéticos (Masabanda & Yauripoma, 2009).

Una de las variables que influyen en la calidad de una soldadura realizada por caufín o pistola de calor es la temperatura, especialmente difícil en condiciones de trabajo permanente

e intenso, debe ser muy estable y permanecer dentro de cierto rango; bajo el mínimo, se produce soldadura fría, generándose microgrietas que en condiciones de vibración conducen a fracturas de la unión y pérdida del contacto eléctrico; sobre el máximo, se recalienta el componente, disminuyendo su vida útil y se produce errores en el funcionamiento del circuito impreso (Lund, 2004).

La SMT es el sistema o conjunto de procesos utilizados para soldar componentes de montaje superficial sobre una placa de circuito impreso (Verrone, 2006). Tanto los equipos así construidos como los componentes de montaje superficial suelen ser llamados dispositivos de montaje en superficie o SMD (Surface Mount Device).

La Tecnología de Montaje Superficial fue desarrollada por los años '60 volviéndose ampliamente utilizada a fines de los '80, debido a que la industria del circuito se había hecho muy sofisticada y los circuitos integrados muy complejos (Carrasco, 2018).

En las últimas dos décadas la industria de la SMT está creciendo a pasos agigantados, los componentes SMD se usan en casi todos los equipos electrónicos de última generación debido a sus reducidos tamaños (Merchan, 2018).

Existen dos procesos de soldadura, el primero por ola de calor y el segundo por refusión o inducción de calor; el proceso por ola de calor es usado predominantemente para la Tecnología Through Hole. Debido a la baja eficiencia del proceso de ola de calor se originó el proceso de refusión, que consiste en depositar un tipo de soldadura especial, después mediante la aplicación de calor con equipos especializados la soldadura fundirá dando como resultado la unión física del componente a la placa (Cervantes & García, 2013).

## **Planteamiento Del Problema**

En el mercado se puede encontrar Hornos para la soldadura de componentes SMD o de montaje superficial, desde hornos industriales para la producción en masa, hasta los más pequeños de uso personal o para microempresas, desafortunadamente sus precios son elevados.

En nuestro país no existe producción nacional de hornos destinado a la soldadura de componentes de montaje superficial, debido a esto se sigue utilizando el modo tradicional de soldadura manual que es mediante el uso de cautines o pistolas de calor para la unión de componentes SMD en el circuito impreso, teniendo que soldar patilla por patilla cada componente lo cual dificulta el proceso de suelda volviéndose demoroso y tedioso.

Debido al tamaño reducido de los componentes electrónicos del tipo SMD la soldadura manual se vuelve dificultoso, tiempos excesivos y mayor precisión del soldador son requeridos; además el riesgo de quemaduras, inhalación de gases tóxicos, pérdida de propiedades semiconductoras de los componentes electrónicos por exceso de calor o microgrietas o fracturas al tener una soldadura fría, además de acabados poco estéticos son algunos problemas que presentan los estudiantes.

Al momento la Carrera de Electricidad no cuenta con un equipo especializado para soldar componentes SMD, limitando la aplicación práctica de sus estudiantes, lo que ocasiona gastos innecesarios al mandar a elaborar las placas en centros especializados que cuenten con el equipamiento necesario o por daños del circuito impreso al realizar el proceso de soldadura de manera manual.

La Carrera de Electricidad al no contar con un equipo automatizado para la soldadura de componentes SMD hace dificultoso y demoroso el proceso de soldadura, peor aún si se tiene varias placas por soldar el tiempo que se requeriría sería un problema, en consecuencia, no aporta al desarrollo tecnológico en sus estudiantes.

## **Objetivos**

### **Objetivo General**

Implementar un Horno Automático de Convección para la soldadura de componentes electrónicos de montaje superficial mediante dispositivos eléctricos, electrónicos e instrumentación virtual.

### **Objetivos Específicos**

- Analizar las tecnologías de soldadura para circuitos impresos.
- Realizar el diseño electrónico y automatización aplicado a un Horno de Convección para soldadura SMD.
- Implementar un Horno Automático de Convección para soldadura SMD.

## **Justificación**

En la malla curricular de la Carrera de Ingeniería Eléctrica y de la Carrera de Ingeniería en Mantenimiento Eléctrico cuenta con materias orientadas al prototipado de circuitos impresos. En cada una de ellas se realizan prácticas de laboratorio, proyectos de curso, e incluso trabajos de grado que involucran el soldado de componentes de montaje superficial, que al

contar con un equipo automatizado para soldar facilitará su implementación ahorrando tiempo y dinero.

En la actualidad existen hornos diseñados específicamente para este fin, que realizan el proceso de soldadura de componentes SMD, los precios de estos hornos son muy elevados y considerando que no existe producción nacional los precios tienden a subir por el pago de impuestos, es por ello que de existir un equipo construido en el país podría incluso ser comercializado lo que mejoraría la economía local.

Para poder brindar a los estudiantes de la Carrera de Electricidad de la Universidad Técnica del Norte un equipo útil para la soldadura de componentes electrónicos de montaje superficial, se propone la implementación del horno automático de convección capaz de aportar lo necesario para tener soldaduras de calidad, como lo es precisión, eficacia, rapidez y funcionalidad además del ahorro energético al permitir soldar placas pequeñas y medianas en el mismo tiempo.

El proyecto beneficiará principalmente a los estudiantes de la Carrera de Electricidad, lo que permitirá impulsar el desarrollo tecnológico en los estudiantes optimizando tiempo y recursos, así este equipo será parte de CIELE. Al implementar el Horno automático de convección se dispondrá de factores muy importantes como sus características tecnológicas, funcionamiento, diseño, adquisición de datos, medición de temperatura, sistema de control de temperatura, supervisión en tiempo real e instrumentación, lo que pondrá en manifiesto lo aprendido en el transcurso de la carrera.

## **Alcance**

La implementación de un horno automático de convección para el soldado de componentes electrónicos SMD se presenta con el fin de brindar un proceso de soldadura de calidad.

Para esta aplicación como controlador principal se considera una buena opción el uso de una placa Raspberry Pi ya que al ser una mini-PC brinda buenas prestaciones para realizar la interfaz software/hardware e integrar diferentes elementos como: termopares, elementos calefactores y de enfriamiento, indispensables en el proceso. Su precio económico lo hace más accesible.

Para facilidad de comunicación y si en el desarrollo del equipo se ve en la necesidad del uso de un ARDUINO como complemento de la automatización del Horno de Convección se considerará su implementación.

Se considera el uso de una pantalla HMI ya que su uso esta optimizado para satisfacer necesidades de interfaz hombre-máquina específicos utilizando interfaces abiertas y estandarizadas en hardware y software, que permiten la integración eficiente en los sistemas de automatización.

Para el funcionamiento de los elementos de Potencia se utilizará circuitos impresos (Electrónica de Potencia) para controlar o transformar voltajes y corrientes requeridas.

Para la instrumentación Virtual se utilizará software específico que ofrezca un enfoque de programación gráfica, configuración de hardware, datos de medidas y depuración requeridos en el proceso de soldadura.

El Horno será capaz de proporcionar temperaturas que van desde los 25°C hasta los 250°C dependiendo el tipo de suelda a utilizar que lógicamente tendrá que tener curvas de fusión menores o iguales a la máxima producida por el horno.

La estructura mecánica será adquirida a partir de un horno eléctrico domestico convencional, cuyas dimensiones serán menores a 60cm en cualquiera de sus ejes pudiendo ser de 33x52x44 cm lo que definirá el tamaño de las placas a soldar; en dicha estructura mecánica se realizará las modificaciones pertinentes para la incorporación de los elementos eléctricos, electrónicos y de visualización.

El presente trabajo incluirá desde el diseño electrónico, construcción de los circuitos de control y potencia hasta la Automatización del equipo, lo que en conjunto formará un Horno Automático de Convección cuyas características permitirá soldar componentes SMD, el mismo que será implementado en el laboratorio de la Carrera de Electricidad.

# CAPÍTULO 1

## Análisis Bibliográfico

En el siguiente capítulo se presenta la fundamentación teórica acerca de las tecnologías de soldadura para circuitos impresos, haciendo un análisis de los dispositivos automatizados para tal fin y los conceptos fundamentales que implica la realización del presente proyecto tanto en la parte eléctrica, electrónica y de control.

### 1.1. Componentes electrónicos para circuitos impresos

Según sus terminales existen dos tipos de componentes electrónicos ya sean pasivos o activos. El autor Bellido (2016), clasifica los componentes electrónicos según sus terminales en:

- Componentes electrónicos THD “Through Hole Devices”
- Componentes electrónicos SMD “Surface Mount Devices”

Cada uno de estos tipos de componentes cuenta con características propias que los diferencian significativamente una de otras como su tamaño, velocidad de funcionamiento, tipo de terminal, precio y su resistencia al calor principalmente.

#### 1.1.1. Componentes electrónicos THD

Los componentes electrónicos del tipo THD son componentes en su mayoría de tamaño mediano que poseen patas metálicas que se introducen a través de orificios realizados en la tarjeta de circuito impreso PCB, se sueldan generalmente al lado opuesto al que se encuentra el componente y se recorta la patilla en exceso. (Dpto. Ing. Electrónica, 2011). En la Figura 1.1, muestra un circuito integrado con encapsulado THT, en la misma se puede apreciar el volumen del componente.

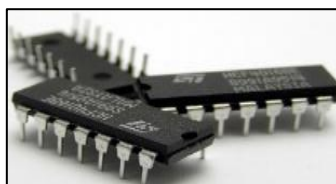


Fig. 0.1. Circuito integrado con encapsulado THT del tipo DIP.

Fuente: (Polosaari, 2006)

Suelen ser de costo económico y fácilmente manipulables por las personas, aunque su producción en masa complican los procesos de automatización y presentan problemas de volumen ocupado y de resistencia parásita. Su perfil de soldado lo muestra la Figura 1.2.



Fig. 0.2. Perfil de soldadura THT donde se aprecian los conos de soldadura.

Fuente: (Humanoc, 2010)

Los componentes de agujeros pasantes vienen en uno de dos tipos de cables, radial y axial. En los componentes de agujeros pasantes axial, los cables se colocan en ambos lados extremos del componente. Los componentes radiales tienen los cables en un solo lado del componente (Tamari, 2017). La Figura 1.3, muestra estos dos tipos de componentes THT.

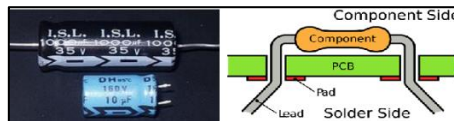


Fig. 0.3. Componentes THT radial y axial.

Fuente: (Tamari, 2017)

### 1.1.2. Componentes electrónicos SMD

Surface Mounted Device (SMD) es una forma de encapsulado de componentes electrónicos con o sin terminales que se caracterizan por su pequeño tamaño que los dispositivos de agujeros pasantes (through hole), no atraviesan la placa de circuito impreso ya que van montados sobre la placa y las conexiones se realizan mediante contactos planos, matrices de bolas en el interior del encapsulado o terminaciones metálicas en los bordes del componente (De Blas, 2018). En la Figura 1.4, se puede observar los componentes THT y SMD respectivamente.

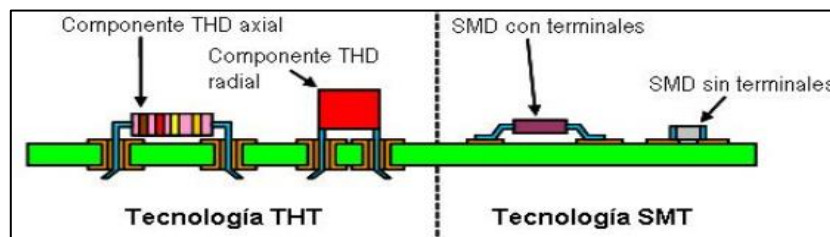


Fig. 0.4. Tecnología THT vs tecnología SMD

Fuente: (Merchan, 2018)

La Figura 1.5, muestra una placa de circuito impreso con componentes SMD con y sin terminales.



Fig. 0.5. Encapsulado SMD con y sin terminales.

Fuente: (Gouvea, 2014)

El sistema o conjunto de procesos destinado a soldar este tipo de componentes SMD es conocido como SMT (Surface Mount Technology) o Tecnología de Montaje Superficial que como su nombre lo indica suelda los componentes sobre la placa de circuito impreso (PCB). (Cervantes & García, 2013).

#### 1.1.2.1. Ventajas de los componentes SMD

Al emplear componentes SMD proporciona mayor igualdad y facilita la automatización al proceso de montaje, lo que aumenta la calidad y la fiabilidad. De acuerdo con los autores Martín & De Soto Gorroño (2005) se tiene entre otras las siguientes ventajas:

- Reducción de tamaño y ausencia de hilos por lo tanto disminución de volumen y de peso. Especial para equipos portátiles.
- Mecanizado más simple por no tener terminales (no es necesario cortarlos ni doblarlos).
- Buenas características mecánicas: Resistencia a choques y ausencia de vibraciones y de resonancias mecánicas.
- Pistas más cortas entre componentes y menor número de capas.
- Disminución de capacidades e inductancias parásitas en encapsulados sin terminales.
- Mejor comportamiento a altas velocidades de trabajo.
- Mejor disipación térmica por estar en contacto directo con el sustrato.
- Están mejor preparados para las últimas tecnologías en cuanto a la eliminación de residuos resultantes de la soldadura.

#### 1.2. Aleaciones para soldadura de componentes de montaje superficial

Las soldaduras para electrónica están compuestas principalmente por estaño (Sn), pero existen aleaciones. El Plomo (Pb) es el elemento más común para realizar estas aleaciones; la existencia de aleaciones se debe a que el estaño sin plomo tiene un punto de fusión más alto, por lo que para soldar el componente hay que aplicar más calor (JMN Electronics, 2018).



Del estaño con plomo hay dos aleaciones principales: 63/37 (63% estaño y 37% plomo) y 60/40 (60% estaño y 40% plomo). Existe aleaciones especiales donde la temperatura de fusión de los diferentes metales se vuelve uniforme y presentan únicamente dos estados de la materia: sólido y líquido, esta aleación obtenida recibe el nombre de eutéctica y presenta un punto de fusión más bajo que los puntos de fusión de los diferentes metales que la forman (JMN Electronics, 2018).

Por ejemplo, para la fusión del plomo se requiere una temperatura de 327 °C y del estaño de 232 °C. Una aleación (Sn63Pb37) de 63% de estaño con 37% de plomo forma una unión eutéctica cuyo punto de fusión es de 183 °C (Méndez, 2018).

Dependiendo de la aplicación que se requiera, además del plomo, existen otros metales con los que se combina, estos son: plata (Ag) y Cobre (Cu) en diferentes porcentajes (FCTA-México, 2018). En la Tabla 1.1 se muestra los intervalos de fusión, así como la máxima temperatura de reflujo de las aleaciones de soldadura más comerciales en el mercado.

TABLA 0.1  
RANGOS DE FUSIÓN Y LAS TEMPERATURAS MÁXIMAS DE REFLUJO DE ALGUNAS ALEACIONES DE SOLDADURA

Aleación de soldadura	Intervalos de fusión °C	Típica temperatura de reflujo máxima °C
Sn63Pb37	183	210 - 230
Sn60Pb40	183 - 188	210 - 230
Sn96.5Ag3Cu0.5	217 - 221	240 - 250
Sn96.5Ag3.5	217 - 221	240 - 250
Sn99.0Ag0.3Cu0.7	217 - 221	240 - 250
Sn95.6Ag3.5Cu0.9	217	235 - 250
SnAg3.5	221	235 - 250

Fuente: (FCTA-México, 2018)

Para montar prototipos algo muy habitual es el uso de soldaduras formadas a base de aleaciones estaño-plomo, y para producción masiva es estaño-plata-cobre.

Existen diferentes métodos y equipos empleados para realizar el proceso de soldadura de componentes electrónicos de manera automática, pero el sistema que utiliza un horno como medio principal para realizar el proceso de soldadura es la soldadura por refusión o reflujo (Reflow Soldering), debido a esto se analiza a continuación en que consiste.

### 1.3. Soldadura por Reflujo

La soldadura por reflujo es aquel método que se utiliza para soldar componentes electrónicos SMD a través de un horno de reflujo, para calentar los paneles ensamblados a una temperatura adecuada durante el período de tiempo especificado a la velocidad de calentamiento adecuada acorde al tipo de suelda (Tsai, 2012).

El proceso de reflujo calienta todo el conjunto a una temperatura superior al punto de fusión de la aleación de soldadura que se esté utilizando, lo que permite que las partículas de soldadura individuales dentro de la pasta se unan formando un conjunto líquido para luego formar una unión sólida entre el pad del PCB y los pines del componente (Whalley, 2004).

Actualmente la soldadura por reflujo es el proceso principal utilizado para la unión de componentes electrónicos a placas de circuito impreso (PCB). Antes del proceso de reflujo, en el PCB se deposita pasta de soldadura sobre los pads, generalmente usando un proceso de serigrafía, y los componentes se colocan en los depósitos de pasta de soldadura.

El equipo de refusión ha variado con mayor frecuencia que cualquier otro equipo de montaje superficial. Durante las últimas décadas han emergido cronológicamente cuatro conceptos diferentes de diseño: fase vapor, lámparas de infrarrojos, paneles infrarrojos y convección forzada de aire (Strauss, 1998). En la Figura 1.6, se ilustra de mejor manera estas tres topologías.

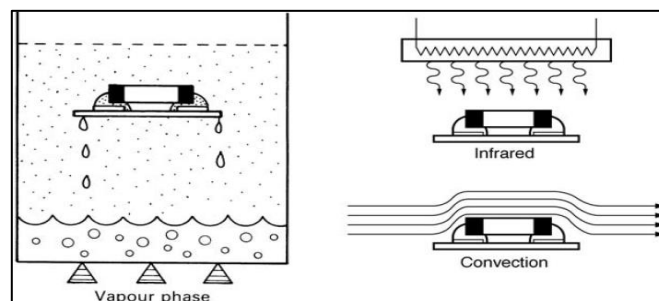


Fig. 0.6. Opciones de soldadura por reflujo

Fuente: (Strauss, 1998)

La tecnología de refusión por fase vapor fue la primera en desarrollarse y la única por varios años pese a que estos equipos estuvieron plagados de problemas de mantenimiento y de inconvenientes por los costos de operación; luego, y aunque no libres de problemas los sistemas infrarrojos una vez que maduraron se convirtieron en el enfoque preferido y aunque hoy en día son el tipo de equipo de uso más común, los sistemas de convección forzada está ganando mayor aceptación en el medio (Verrone, 2006).

Marcoux (1992) refiere que el éxito del proceso de reflujo depende de varias variables:

- La tasa de transferencia de calor de una fuente de calor.
- La tasa de absorción de calor y la uniformidad del conjunto.
- La capacidad de soldadura de los metales a unir.
- El punto de fusión de la soldadura.
- La tolerancia al calor de los materiales en el ensamblaje.

- La química del flujo y la limpieza de los metales a unir.

Para ajustar los parámetros de calentamiento del circuito impreso en periodos específicos de tiempo, y para procesos de control es generada una gráfica a lo cual se le llama perfil de temperatura (Tsai, 2012).

### 1.3.1. Perfil de soldadura (curva de temperatura)

Fabricantes de diferentes tipos de soldadura sugieren un perfil de soldadura también conocido como perfil térmico el cual es proporcionado en una hoja de datos o datasheet, este perfil se lo representa a manera de gráfica o curva de soldadura, en la cual se muestra la variación de la temperatura con respecto al tiempo como se ilustra en la Figura 1.7.

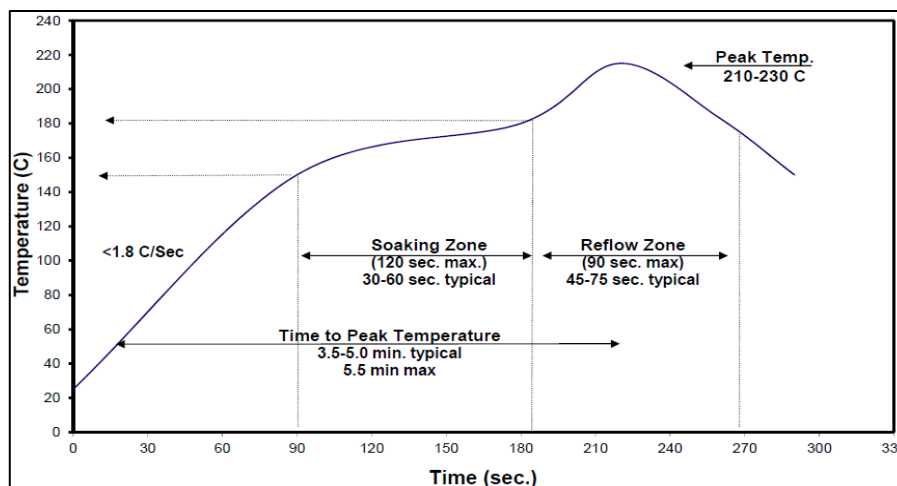


Fig. 0.7. Perfil sugerido por el fabricante para soldadura con plomo Sn63Pb37

Fuente: (Kester, 2009)

Todos los perfiles de soldadura constan de cuatro etapas distintas, las cuales son: precalentamiento, remojo, reflujo, enfriamiento (preheat, soak, reflow, cooling). Es fundamental respetar cada etapa ya que cada una tiene un objetivo y un propósito dentro del proceso de soldadura, además de controlar la velocidad de cambio de temperatura, la temperatura máxima y la duración del tiempo dentro de cada etapa para crear uniones de soldadura confiables, de no ser así, se presentarán diversos problemas como se muestra en la Tabla 1.2 (Cervantes & García, 2013).



Fig. 0.8. Proceso de soldadura con horno

Fuente: El Autor.

Las características de cada etapa son:

#### **1.3.1.1. Precalentamiento**

El precalentamiento es importante porque amortigua el choque térmico, que golpearía la placa y los componentes en ella, si tuviera que enfrentar la onda de soldadura directamente del frío (Strauss, 1998). Normalmente la tasa de cambio ( $dT/dt$ ) en ésta zona debe ser de 2 a 4 °C/s con temperaturas máximas comprendidas entre 100 a 150 °C durante 90 segundos aproximadamente (Corporación-Peregrine, 2016). El propósito del precalentamiento es permitir que los solventes se evaporen, y sirve también para eliminar tensiones internas y gases residuales de la PCB (Dpto. Electrónica, 2015).

#### **1.3.1.2. Remojo**

La zona de remojo está diseñada para permitir que la placa y los componentes alcancen una temperatura uniforme, minimizando los gradientes térmicos. Durante este período, se eliminan los volátiles de la pasta de soldadura y se activa el fundente (flux) y solventes, lo que permite eliminar capas de óxido en los componentes en su preparación para la soldadura (RFICs, 2016). La velocidad de rampa en esta zona es muy baja generalmente entre 0.5 a 1 °C/s y la temperatura se eleva cerca del punto de fusión de la soldadura (+183 °C para la soldadura estándar Sn63PB37 y +217 °C para la soldadura Sn96.5Ag3.5); las temperaturas típicas en ésta zona está alrededor del rango de 150 - 200 °C durante 60 a 90 segundos aproximadamente (Corporación-Peregrine, 2016).

#### **1.3.1.3. Reflujo**

Es la zona donde la soldadura cambia de sólido a líquido es decir donde ocurre la soldadura, y es en esta etapa en donde el control de temperatura se vuelve crítico. Aquí se alcanza el valor más alto de temperatura en el proceso de soldadura, la temperatura debe ser excedida por aproximadamente 25°C para una actividad del fundente adecuado y para obtener una buena humectación logrando así asegurar en todos los puntos una mejor unión (Lee, 2002). Para soldaduras eutécticas Sn63Pb37, un rango de temperatura pico de 210-230 °C se considera aceptable; la temperatura aumenta a una velocidad de aproximadamente 2 °C/s durante 30 a 60 segundos según la temperatura pico (Prasad, 1998).

#### **1.3.1.4. Enfriamiento**

Esta es la etapa final, y se refiere al período después del reflujo donde la temperatura está disminuyendo y el horno se enfría durante la cual la soldadura se solidifica. Se debe tener cuidado de realizar esta etapa en forma gradual; cuanto más rápida sea la velocidad de

enfriamiento, menor será el tamaño de grano de la junta de soldadura y, por lo tanto, mayor será la resistencia a la fatiga de la junta de soldadura, sin embargo, un enfriamiento muy rápido provocará un shock térmico en los componentes y la placa (Prasad, 1998). En ésta zona se recomienda tener un decremento en un rango de -1 a -3 °C/s y no exceder de -6°C (Dpto. Electrónica, 2015).

TABLA 0.2  
PROBLEMAS MÁS FRECUENTES CON SUS POSIBLES CAUSAS

PROBLEMA	POSIBLE CAUSA
Rotura de capacitores cerámicos tipo chip. Explosión de la pasta de soldadura, formación de bolas de soldadura.	Un incremento excesivo de la temperatura en la etapa de precalentamiento.
Bolas de soldadura o microgrietas.	Etapa de remojo incompleta. Temperatura excesiva en la zona de remojo.
Soldadura fría.	Tiempo insuficiente en la etapa de reflujo.
La soldadura no se adhiere a la terminal. Oxidación.	Tiempo de remojo excesivo. Temperatura excesiva en reflujo.
La soldadura no se adhiere al pad.	La terminal se calienta más rápido que la placa.
Placa o componente quemado.	Exceder la temperatura de reflujo.

Fuente: (Cervantes & García, 2013)

### 1.3.2. Refusión por Convección Forzada de Aire

La convección forzada de aire es el último desarrollo de la tecnología de refusión, la mayoría de los hornos de reflujo de hoy en día usan la convección forzada como el modo primario de transferencia de calor. El calor es trasferido a la placa mediante aire caliente que circula a baja velocidad, logrando el calentamiento por conducción, en lugar de radiación (Verrone, 2006).

Los equipos de soldadura por convección forzada calientan el aire transformando la energía eléctrica en dos tipos de energías: energía térmica (mediante el efecto Joule de elementos resistivos) como fuente de calor y energía mecánica (ventilador) para la circulación de aire, además de contar con uno o varios instrumentos de medida (termopares o termoresistencias) (Verrone, 2006). En la Figura 1.9, se puede observar la conformación de un horno de convección forzada incluyendo la placa de circuito impreso (PCB).

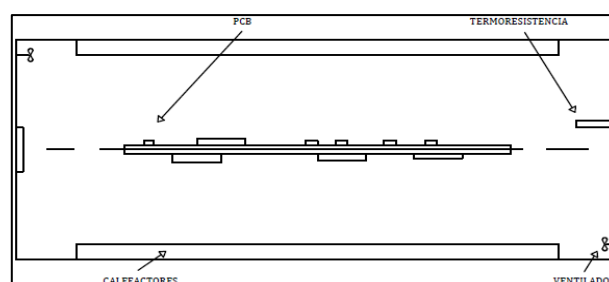


Fig. 0.9. Vista frontal de un horno de convección forzada.

Fuente: (Verrone, 2006)

### 1.3.2.1. Transferencia de calor por convección forzada

La transferencia de calor es la ciencia que pretende predecir la transferencia de energía que puede tener lugar entre cuerpos como resultado de una diferencia de temperatura, la termodinámica enseña que esta transferencia de energía se define como calor (Holman, 2010). La ciencia de la transferencia de calor busca no solo explicar cómo se puede transferir la energía térmica, sino también predecir la velocidad a la que se realizará el intercambio bajo ciertas condiciones específicas.

La transferencia de calor por convección ocurre cuando un fluido (como aire, nitrógeno o agua) a una temperatura dada entra en contacto con una masa sólida (como un conjunto SMT) a una temperatura diferente de modo que, si el fluido está más caliente que la masa, la masa se calentará, por el contrario, si el fluido está más frío que la masa, la masa se enfriará tal como lo muestra la Figura 1.10 (Norman, 2001).

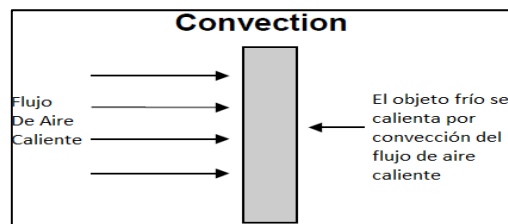


Fig. 0.10. Ejemplo de transferencia de calor por convección.

Fuente: (Norman, 2001)

La convección es el modo de transferencia de energía entre una superficie sólida y el líquido o gas adyacente que está en movimiento, e involucra los efectos combinados de la conducción y el movimiento del fluido, de tal manera que cuanto más rápido sea el movimiento del fluido, mayor será la transferencia de calor por convección y en el caso de ausencia de cualquier movimiento de fluido, la transferencia de calor es por conducción pura (Yunus & Afshin, 2015).

Para expresar el efecto general de la convección, se utiliza la ley de enfriamiento de Newton (ecuación 1) (Yunus & Afshin, 2015):

$$\dot{Q}_{conv} = hA_s(T_s - T_\infty) \quad (W) \quad (1)$$

Dónde:

- $\dot{Q}_{conv}$ : es la tasa de calor convectiva (W).
- $h$ : es el coeficiente de transferencia de calor por convección  $\left(\frac{W}{m^2 \cdot K}\right)$ .
- $A_s$ : es el área de la superficie de transferencia de calor ( $m^2$ ).

- $T_s$ : es la temperatura de la superficie ( $^{\circ}\text{C}$ ).
- $T_{\infty}$ : es la temperatura de los alrededores ( $^{\circ}\text{C}$ ).

Por sus unidades, el coeficiente de transferencia de calor por convección  $h$  se definirse como la tasa de transferencia de calor entre una superficie sólida y un fluido por unidad de superficie por unidad de diferencia de temperatura (Yunus & Afshin, 2015). La Tabla 1.3 lista algunos valores típicos de coeficiente de convección  $h$ .

TABLA 0.3  
VALORES TÍPICOS DEL COEFICIENTE DE TRANSFERENCIA DE CALOR POR CONVECCIÓN

TIPO DE CONVECCIÓN	$h, \text{W}/\text{m}^2 \cdot \text{K}$
Convección libre de gases	2-25
Convección libre de líquidos	10-1000
Convección forzada de gases	25-250
Convección forzada de líquidos	50-20,000
Ebullición y condensación	2500-100,000

Fuente: (Yunus & Afshin, 2015)

En aplicaciones de reflujo SMT, aumentar la velocidad del flujo aumentará el coeficiente de transferencia de calor  $h$  y por tanto mejora la cantidad de transferencia de calor por convección, sin embargo, la velocidad debe ser controlada bajo límites prácticos, ya que una velocidad muy elevada puede hacer que los componentes se muevan.

La presencia de superficies sólidas afecta el flujo de fluido conocido como la condición de no deslizamiento, esto ocurre porque un fluido en contacto directo con un sólido se “adhiera” a la superficie a causa de los efectos viscosos y carece de deslizamiento, ésta situación conlleva a que la transferencia de calor desde el sólido a la capa de fluido inmóvil adyacente a la superficie, se lleve a cabo mediante conducción (ecuación 2) (Yunus & Afshin, 2015).

$$\dot{q}_{conv} = \dot{q}_{cond} = -k_{fluido} \left. \frac{\partial T}{\partial y} \right|_{y=0} = 0 \quad \left( \frac{\text{W}}{\text{m}^2} \right) \quad (2)$$

Donde:

- $k_{fluido}$ : es la conductividad térmica del fluido  $\left( \frac{\text{W}}{\text{m} \cdot \text{K}} \right)$ .
- $T$ : representa la distribución de temperatura en el fluido.
- $(\partial T / \partial y)_{y=0}$ : es el gradiente de temperatura en el sistema.

La convección forzada requiere una fuerza externa (ventilador) que empuja o tira el fluido sobre el objeto (PCB). A continuación, se analizan algunas topologías de ventiladores.

### 1.3.2.2. Ventiladores

Un ventilador es una máquina rotativa capaz de poner el aire, o un gas, en movimiento. Se lo define también como una turbomáquina que transmite energía mecánica para generar la presión necesaria con la cual mantiene un flujo continuo de aire (Salvador Escoda S.A., 2014).

Un ventilador consta esencialmente de un motor de accionamiento, generalmente eléctrico y un propulsor giratorio en contacto con el aire, al que le transmite energía. Éste propulsor adquiere la forma de rodete con álabes en el caso de ventiladores del tipo centrífugo, o de una hélice con palas de silueta y en número diverso, si son del tipo axiales (Salvador Escoda S.A., 2014).

Según esta condición se los puede clasificar en ventiladores centrífugos y ventiladores axiales.

- Ventiladores axiales. - El movimiento del flujo de aire a través de la hélice es en dirección paralela al eje del ventilador (dirección axial) (Soler & Palau, 2012). El aire es impulsado por una hélice dotada de un número determinado de álabes o palas que están ancladas en un núcleo o cubo que es el elemento que transmite la energía del eje a los álabes. Se usan para mover grandes cantidades de aire con una presión relativamente baja y se los conoce con el nombre de extractores o inyectores de aire (Soler & Palau, 2012).

Las hélices de los ventiladores axiales pueden ser de dos tipos: de hélice axial de perfil delgado y de hélice axial de perfil sustentador tal como indica la Figura 1.11.

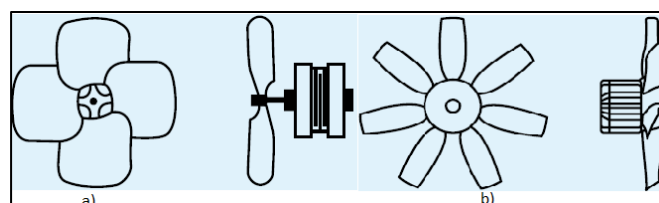


Fig. 0.11. Hélice Axial de: a) Perfil delgado. b) Perfil sustentador.

Fuente: (Soler & Palau, 2012)

- Ventiladores centrífugos. - Son llamados centrífugos debido a que una parte considerable de la energía transferida al aire en un ventilador de flujo radial se debe a la acción centrífuga. En éstos ventiladores el movimiento del aire lo hace a través de un rodete con un numero variable de álabes, la trayectoria del aire sigue una dirección axial a la entrada y sale en dirección perpendicular, es decir que la entrada y salida están en ángulo recto (Soler & Palau, 2012).



Los rodetes de los ventiladores centrífugos pueden ser de tres tipos: álabes radiales, álabes hacia adelante, álabes hacia atrás (Soler & Palau, 2012).

La Figura 1.12 ilustra de mejor manera los tres tipos de rodetes.

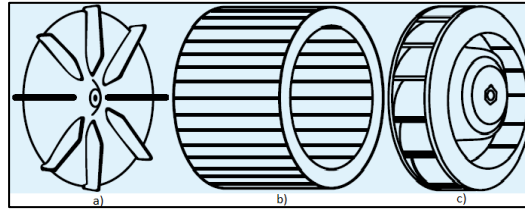


Fig. 0.12. Rodete centrífugo: a) Álabes radiales. b) Álabes hacia adelante. c) Álabes hacia atrás.

Fuente: (Soler & Palau, 2012)

- a) Álabes Radiales. Presentan un menor número de álabes que los de álabes hacia adelante o hacia atrás; son utilizados para impulsar aire o gases a elevada temperatura, gracias a la facilidad con que son eliminados los depósitos sólidos por la fuerza centrífuga (Fernández, 2009).
- b) Álabes Hacia Adelante. Presentan un gran número de álabes y son empleados en ventiladores debido al bajo ruido que presentan, pero su rendimiento es bajo (Fernández, 2009).
- c) Álabes Hacia Atrás. Son empleados en bombas y ventiladores centrífugos y presentan un mejor rendimiento que los anteriores, de igual manera el nivel de ruido es bajo (Fernández, 2009).

### 1.3.2.3. Ventajas y Desventajas de los Sistemas de Convección Forzada

Como se ha mencionado anteriormente, los sistemas de convección forzada están ganando acogida muy rápidamente frente a sistemas como fase vapor e infrarrojos y son los de mayor utilidad a nivel industrial por las mejoras que se obtienen en el proceso de soldadura. Verrone (2006) detalla algunas ventajas y desventajas de los sistemas de convección forzada.

Ventajas:

- Producen un calentamiento próximo al equilibrio.
- Calientan uniformemente.
- Son fáciles de mantener.
- En los sistemas de convección es muy fácil definir un perfil.
- No son sensibles al calor (emisividad y absorbancia de los componentes).

- Mayor confiabilidad y precisión.

Desventajas:

- El aire tiene que ser controlado con presión.
- Los humos de la pasta de soldar y de la placa se reciclan para hacer más eficiente el sistema.
- Los equipos de producción continua podrían perder calor por los bordes.

### 1.3.3. Hornos de Reflujo para Soldadura SMD

En la industria electrónica existen diferentes tipos de hornos de refusión, algunos son por fase vapor (VP), otros por radiación infrarroja (IR) y, por último, están los de convección forzada de aire caliente (FC).

Hornos VP utilizan la temperatura latente de condensación de un vapor para producir el calentamiento de la superficie que debe ser soldada; en los Hornos IR el calor es suministrado por emisores de rayos infrarrojos y se transfiere a la placa y a las uniones tanto por radiación como por convección del gas contenido en el recinto, por último, los Hornos FC calientan mediante la circulación de un fluido o gas caliente el cual es transferido a la placa a soldar (Martín & De Soto Gorroño, 2005).

En la Tabla 1.4 se detalla algunos de los modelos de hornos de reflujo y sus precios en los países de origen.

TABLA 0.4  
PRECIO DE HORNOS DE REFLUJO COMERCIALES

MODELO	TIPO DE TECNOLOGÍA	POTENCIA (W)	PAÍS DE ORIGEN	PRECIO (\$)
T962A	IR	1500	China	700
T937M	IR + Convección	3300	China	700
SUNY-SMT100	Convección	1800	China	1000
ZB630RF	Convección	7500	China	1600
T200A	IR + Convección	4800	China	2000
TYR108	IR + Convección	4200	USA	2750
QHL320A	IR + Convección	3300	USA	2800
SMD-2007	IR + Convección	3500	USA	2900
NEODEN IN6	Convección	1700	USA	3000

Fuente: (Alibaba, 2020)

El calor que genera el Horno para soldadura SMD puede ser proveniente de materiales resistivos como niquelinas, lámparas infrarrojas, lámparas halógenas, etc., entre los elementos más comunes cuyas características térmicas se diferencian entre cada tipo.

#### 1.4. Automatización del sistema térmico

Automatizar una instalación o equipo consiste en desarrollar una serie de tareas aprovechando la capacidad de las máquinas sin que sea necesaria la intervención humana en su proceso (Entrena, 2016). Mediante dispositivos tecnológicos (autómatas y microprocesadores) se provee de cierta inteligencia al sistema a automatizar los cuales son sometidos a acciones de control para su buen comportamiento con mayor rapidez y mejor de lo que podría hacer un ser humano.

Izaguirre Castellanos (2012), afirma “Un sistema automatizado es el conjunto de elementos (equipamiento, sistema de información, y procedimientos) interrelacionados funcionalmente entre sí que conforman una estructura jerárquicamente expandida cuya función es garantizar el desempeño independiente del proceso a través de operaciones de control y supervisión total del sistema, bajo las técnicas más modernas y cumpliendo los requisitos establecidos de acuerdo al tipo de planta.” (p.10).

Bajo estos conceptos, se focaliza en los sistemas de control de temperatura, el sistema debe cumplir con los parámetros anteriormente mencionados con el propósito de llevar a cabo su proceso de manera autónoma y cumpliendo los requerimientos de manera confiable.

Un sistema de control de temperatura, adquiere la temperatura del ambiente a medir mediante un sensor, y esta señal es tratada, ya sea digital o análogamente para posteriormente pasar a un sistema de control el cual activa, desactiva, aumenta o disminuye el sistema que estará encargado de mantener la temperatura deseada a través de un elemento actuador (Carrillo, 2011).

En la Figura 1.13, se puede observar los elementos fundamentales dentro de un sistema de control de temperatura. A continuación, se detalla cada uno de ellos.

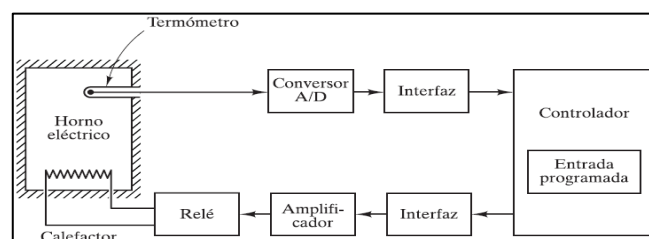


Fig. 0.13. Elementos de un sistema de control de temperatura.

Fuente: (Ogata, 2010)

### 1.4.1. Sensores de temperatura

Es de vital importancia medir la temperatura en varios procesos industriales, debido a que muchos materiales requieren llegar a cierto rango de temperatura, para esto la temperatura debe ser medida y controlada por medio de un sensor más un transductor. Un sensor se puede definir como un dispositivo primario que, a partir de la energía del medio donde se mide, genera una señal de salida transducible que es función de la variable medida (Pallás Areny, 2003).

Un elemento sensor con un dispositivo acondicionamiento de señal se denomina transductor. Un transductor es un dispositivo que transforma un tipo de magnitud física como la temperatura en una señal eléctrica correspondiente (Pallás Areny, 2003).

Según el aporte de energía, los sensores se pueden dividir en moduladores y generadores. En los sensores moduladores o activos como el caso de los termistores, la señal eléctrica de salida se deriva de una fuente de energía auxiliar de voltaje o corriente; en cambio en los sensores generadores o pasivos como el termopar, la energía de salida es suministrada por la entrada, es decir no requieren de una fuente externa (Pallás Areny, 2003).

En la Tabla 1.5 se listan los sensores más comunes empleados para medir temperatura con instrumentación electrónica.

TABLA 0.5  
TIPO DE SENSORES Y CARACTERÍSTICAS

	Rango de medición	Salida	Características principales	Desventaja
Termopar	-200 °C a 2495 °C	No lineal	Bajo voltaje de salida. Alta precisión y repetitividad. Rápida respuesta. Fiabilidad elevada. Costo bajo.	Baja sensibilidad. Requieren unión de referencia.
RTD	-200 °C a 700 °C	Lineal	Alta precisión. Mayor exactitud.	Requiere excitación. Costo elevado.
Termistor	0 °C a 100 °C	No lineal	Alta sensibilidad.	Requiere excitación.
Circuito Integrado	-55 °C a 150 °C	Lineal	Alta sensibilidad. Salida lineal a 10mV por grado.	Respuesta lenta. Requiere excitación.

Fuente: (Pallás Areny, 2003)

La soldadura por reflujo llega a temperaturas que bordean los 250°C por lo que de entrada se descarta el uso de termistores y circuitos integrados como sensores de temperatura por lo que se analiza los termopares ya que el uso de RTD implicaría en costos elevados.

### 1.4.1.1. Termopares (termocupla)

Los termopares son sensores activos formados por dos alambres metálicos distintos, cuyas uniones (unión de medida o caliente y unión de referencia o fría) se mantienen a distinta temperatura (Alzate, Montes, & Silva, 2016). Un voltaje en milivolts es generado en los terminales de los metales que no se encuentran en contacto, el cual varía en función de la diferencia de temperatura en sus uniones (Gutiérrez & Iturralde, 2017). La unión caliente es la que se expone a la temperatura a medir y la unión fría o de referencia es la que va al circuito acondicionador o transductor (Figura 1.14). El voltaje producido en la unión es no lineal con respecto a la temperatura y de baja amplitud, por lo que necesita ser amplificada, filtrada y compensada (Cervantes & García, 2013).

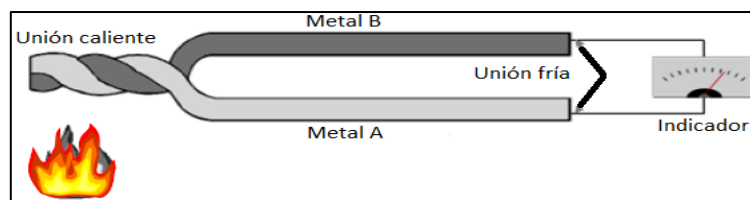


Fig. 0.14. Unión caliente y fría de un termopar.

Fuente: (Rodríguez, 2015)

En el mercado existe varios tipos de termopares, su criterio de selección radica en el intervalo de temperatura a medir y su precisión. En la Tabla 1.6 se indican los tipos de termopares más comunes y sus principales características, mientras que en la Figura 1.15 se muestra la respuesta de salida en voltaje de un termopar con respecto a la temperatura.

TABLA 0.6  
TIPO DE TERMOPARES Y CARACTERÍSTICAS

Tipo de termopar	Rango de temperatura (°C)	Error de medición
J-(Hierro-Constantan)	-200 a 1200	±1.0 °C
K-(Cromel-Alumel)	-40 a 1100	±2.2 °C
T-(Cobre-Constantan)	-200 a 260	±1.0 °C
E-(Cromel-Constantan)	-200 a 900	±1.7 °C
R-(Platino-13% Rodio/Platino)	0 a 1500	±1.5 °C
S-(Platino-10%Rodio/Platino)	0 a 1500	±1.5 °C
B-(Platino-30% Rodio/Platino-6% Rodio)	0 a 1800	±0.5 °C

Fuente: (Cervantes & García, 2013)

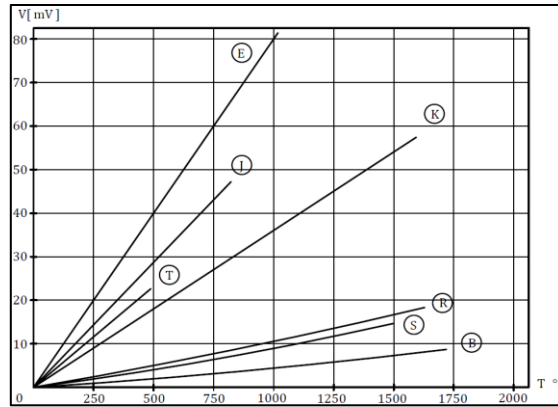


Fig. 0.15. Curvas características de voltaje vs temperatura en termopares.

Fuente: (Creus, 2011)

### 1.4.2. Elemento final de control (Triacs)

El Triac (triodo de corriente alterna), es un dispositivo semiconductor de tres terminales perteneciente a la familia de los tiristores que, a diferencia del SCR, éste puede conducir en cualquier dirección cuando se encuentra en conducción, y son empleados para el control de corriente alterna (Maloney, 2006).

El Triac se puede considerar como dos SCR conectados en antiparalelo con una conexión de compuerta común y es usado para controlar la corriente promedio que fluye a una carga y por consiguiente la potencia que consume siempre y cuando haya una señal de disparo en la compuerta (Rashid, 2004). El símbolo esquemático del Triac se muestra en la Figura 1,16 con los nombres y abreviaturas de sus terminales.

Una vez que el Triac se encuentra encendido, se crea una ruta de flujo de corriente de baja resistencia de una terminal hacia la otra cuya dirección del flujo depende de la polaridad del voltaje aplicado externamente (Maloney, 2006). Al actuar el Triac como un interruptor se conecta en serie a la carga, pudiendo variar la corriente promedio suministrada a la carga mediante el cambio de la cantidad de tiempo por ciclo que el Triac se mantiene en estado de conducción.

Con un apropiado circuito de disparo el Triac puede conducir durante 360° completos por ciclo a diferencia del SCR que puede conducir 180°, en consecuencia, el Triac proporciona un control de potencia de onda completa el cual puede ser por ángulo de fase o cruce por cero (Maloney, 2006).

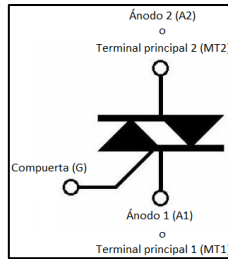


Fig. 0.16. Símbolo y nombres de terminales de un Triac

Fuente: (Maloney, 2006)

El ánodo 2 normalmente es conectado al neutro y el ánodo 1 va entre el cable neutro y la carga a controlar, bastará que exista una corriente en el gate superior a la umbral para poder ser activado.

### 1.4.3. Sistemas de control

Carrillo (2011), define al sistema de control como “El conjunto de elementos que funcionan de manera concatenada para proporcionar una salida o respuesta deseada.” (p.20).

Por otro lado, Hernández (2010), menciona que un sistema de control automático es un sistema interconectado de elementos que proporcionará una respuesta deseada cuyo arreglo resultante es capaz de controlarse por sí mismo.

#### 1.4.3.1. Clasificación de los sistemas de control

De acuerdo con Hernández (2010), existen dos tipos de sistemas de control que se pueden considerar: sistemas de bucle abierto (o no automáticos) y sistemas de bucle cerrado (retroalimentados o automáticos).

##### a) Sistema de control en bucle abierto

Los sistemas de control en bucle abierto son aquellos sistemas en los cuales la salida no tiene efecto sobre la acción de control (Ñeco, Reinoso, García, & Aracil, 2013). Es decir, en este tipo de sistema no se tiene medición de la salida para poder compararla con la entrada es por ello que se conocen como sistemas no retroalimentados.

Los sistemas de control en bucle abierto operan en base a tiempos tal es el caso de los hornos microondas y molinos eléctricos, éste esquema de control es utilizado cuando se conoce la relación entre la entrada y la salida y cuando no existe perturbaciones, su precisión depende de la calibración (Ogata, 2010). Los elementos que componen este tipo de sistema son: señal de entrada o referencia, controlador, sistema a controlar y señal de salida (Ñeco et al., 2013).

#### b) Sistema de control en bucle cerrado

En los sistemas de control de bucle cerrado se compara permanentemente la señal de salida o variable a controlar con la señal de entrada y en función de esta diferencia (señal de error) el controlador cambia la acción de control sobre los actuadores de la planta o sistema (Ñeco et al., 2013). Es decir, en un sistema de bucle cerrado la acción de control depende directamente de la salida.

Una de las ventajas de los sistemas de bucle cerrado es la poca afectación ante perturbaciones gracias a que su implementación requiere siempre el uso de una acción de control realimentado para reducir el error y llevar la salida del sistema a un valor deseado, generalmente la retroalimentación se la lleva a cabo mediante un conjunto de sensores (Ogata, 2010). Los elementos que componen este tipo de sistema son: señal de entrada, controlador, sistema a controlar, elementos de medición (sensores) y señal de salida (Ñeco et al., 2013).

#### 1.4.3.1. Componentes en un sistema de control

En cualquier sistema de control existen una serie de elementos que es necesario definirlos. Ñeco et al. (2013) presenta una clasificación de la siguiente forma:

- a) Variable controlada. - La variable controlada es la cantidad o condición que se mide y se controla, normalmente representa la salida del sistema (Ogata, 2010).
- b) Planta o sistema. – La planta es la instalación de un sistema destinada a realizar un proceso específico (Carrillo, 2011).
- c) Sensor. – Es el elemento que permite adquirir el valor de la variable controlada en instantes de tiempo determinados (Ñeco et al., 2013).
- d) Señal de referencia. – Es la señal consigna, que se aplica a voluntad del usuario con la finalidad de encontrar una respuesta deseada a la salida (Carrillo, 2011).
- e) Controlador. – El controlador o regulador es el encargado de proporcionar la energía necesaria al actuador modificando la señal de control para así afectar el valor de la variable controlada en función del objetivo de control (Ñeco et al., 2013).
- f) Actuador. – Es también conocido como el elemento final de control y es aquel dispositivo de potencia que produce la entrada para la planta de acuerdo con la señal de control modificando de esta forma la señal de salida (Ogata, 2010).

Un sistema de control de temperatura requiere un control retroalimentado debido a las ventajas que este presenta es por ello que a continuación se analizan técnicas avanzadas de control en bucle cerrado.



### 1.4.3.2. Control digital

Con el avance tecnológico en el desarrollo de microprocesadores en cuanto a velocidad de procesamiento, capacidades de cálculo, reducción de tamaño y consumo de potencia, la forma de vida ha sufrido cambios drásticos con la aplicación de computadoras, específicamente en el control de procesos llamado también control por computadora o control digital (Del Busto, 2013).

Al referirse a control digital se habla de control en tiempo discreto, es decir que necesitaremos discretizar las señales continuas para que pueda interpretar el microprocesador.

Los sistemas de tiempo discreto, son sistemas dinámicos en los cuales una o más variables pueden variar únicamente en ciertos instantes de tiempo, llamado tiempo de muestreo que se indican por  $kT$  ( $k=0,1, 2,\dots$ ) pueden especificar el instante en el cual se realiza una medición física o el tiempo en el cual se lee la memoria del computador (García, 2009).

En la Figura 1,17, se muestra un diagrama básico de un control digital con realimentación, en el cual  $r(kT)$  es la señal de referencia o punto de ajuste establecida de manera digital por un computador; la señal de control  $u(t)$  es convertida por un conversor digital análogo (CDA), a una forma analógica y es mantenida constante entre los instantes de muestreo; por medio de un conversor análogo digital (CAD) la señal de salida de planta  $y(t)$  es convertida a una forma digital a instantes de muestreo  $kT$  definidos por los ciclos del reloj (García, 2009).

Al ser un sistema en lazo cerrado las señales del sensor también son discretizadas obteniendo una versión muestreada de la señal de salida de la planta ( $m(kT)$ ), una señal de perturbación  $w(t)$  puede ser añadida a planta la cual intentara ser corregida por medio del controlador; el microprocesador interpreta la diferencia entre la referencia  $r(kT)$  y la respuesta de la planta discretizada  $m(kT)$  como una secuencia de números los cuales serán procesados por medio de un algoritmo de control para generar la señal de control  $u(kT)$  (Del Busto, 2013).

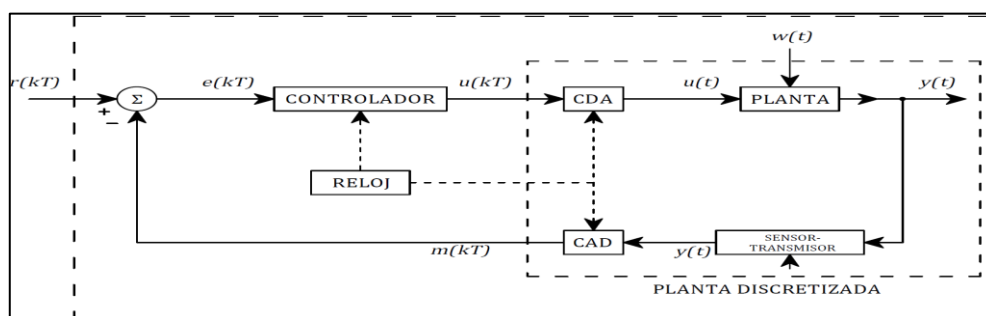


Fig. 0.17. Diagrama básico de un sistema de control digital.

Fuente: (Del Busto, 2013)

### 1.4.3.3. Control de temperatura PID digital

Más de la mitad de los controladores industriales que se utilizan en la actualidad en procesos térmicos emplean esquemas de control PID o PID modificado. A pesar de los avances tecnológicos en temas de control no ha podido ser reemplazado, debido a que con un ajuste adecuado en sus ganancias  $K_p$ ,  $K_i$  y  $K_d$  muestran muy buenos resultados, además de que por su sencillez pueden ser fácilmente implementados (Reyes Cortés & Cid Monjaraz, 2015).

Para entender el control PID digital, se parte por su versión continua. En tiempo continuo la salida del controlador del tipo PID es representado idealmente por la siguiente ecuación (Del Busto, 2013):

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (3)$$

Donde:

$u(t)$ : salida de control PID

$e(t)$ : función del error

$K_p$ : ganancia proporcional

$T_i$ : tiempo integral<sup>1</sup>

$T_d$ : tiempo derivativo<sup>2</sup>

Al aplicar la transformada de Laplace a la ecuación (3) se puede obtener la función de transferencia del regulador PID, el mismo que relaciona la transformada de Laplace de la señal de error con la transformada de Laplace de la acción de control (Salt, Cuenca, Casanova, & Correcher, 2015). Se representa con la siguiente ecuación:

$$G_{PID}(s) = \frac{U(s)}{E(s)} = K_p \left[ 1 + \frac{1}{T_i s} + T_d s \right] \quad (4)$$

Para poder obtener la versión digital del regulador PID se procede a aproximar (discretizar) la acción de control integral mediante el método de aproximación rectangular y la acción derivativa mediante el método de Euler (Reyes Cortés & Cid Monjaraz, 2015). De esta manera

---

<sup>1</sup> Tiempo requerido para que la acción integral iguale el valor de la acción proporcional.

<sup>2</sup> Tiempo requerido para que la acción derivativa iguale el valor de la acción proporcional.

se obtiene a la salida del controlador la ecuación discreta en  $u(kT)$ , donde  $kT$  es el tiempo discreto, el cual se define como múltiplos del periodo de muestreo  $T_s$ .

La Figura 1.18, ilustra de mejor manera la aproximación rectangular para el proceso de integración discreta.

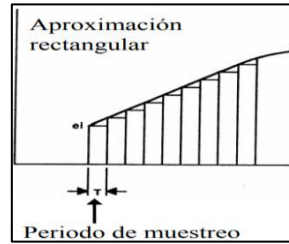


Fig. 0.18. Aproximación rectangular para integral discreta.

Fuente: (Ramos Lara, 2007)

$$u(kT) = Kp \left[ e(kT) + \frac{T_s}{T_i} \sum_{i=0}^k e(i) + \frac{T_d}{T_s} (e(kT) - (e_{(kT-1)})) \right] \quad (5)$$

O de la forma:

$$u(kT) = Kp \left[ e(kT) + \frac{1}{T_i} (Int(kT)) + T_d (euler(kT)) \right] \quad (6)$$

Siendo:

$$Int(kT) = Int(kT - 1) + T_s(e(kT)) \rightarrow Integral\ discreta$$

$$euler(kT) = \frac{(e(kT) - e(kT - 1))}{T_s} = \frac{\Delta e(kT)}{T_s} \rightarrow Derivada\ discreta$$

Donde:

$u(kT)$ : salida de control PID discreto

$e(kT)$ : error en el instante actual

$e(kT - 1)$ : error en el instante anterior

$T_s$ : periodo de muestreo

Al aplicar la transformada Z la cuál es análoga a la transformada de Laplace en tiempo continuo se obtiene la función de transferencia en tiempo discreto (Salt et al., 2015):

$$G_{PID}(z) = \frac{U(z)}{E(z)} = Kp \left[ 1 + \frac{T_s}{T_i} \frac{1}{(1 - z^{-1})} + \frac{T_d}{T_s} (1 - z^{-1}) \right] \quad (7)$$

Si bien se tiene una idea matemática del algoritmo de control PID es necesario conocer que función realizan cada una de las acciones básicas de control (proporcional, integral y derivativa).

- **Acción Proporcional ( $K_p * e(t)$ ).** - Se llama así porque es proporcional al valor de la señal de error en cada instante de tiempo conocido también como control del presente, se caracteriza por un parámetro conocido como la ganancia proporcional ( $K_p$ ). Si la ganancia proporcional es pequeña las acciones sobre la planta serán suaves y la respuesta será lenta y sin oscilaciones, por el contrario, si la ganancia es demasiado grande las acciones serán fuertes y la respuesta será rápida, pero con tendencia a inestabilizarse (Salt et al., 2015).
- **Acción Integral ( $K_i * \int_0^t e(t) dt$ ).** - Básicamente es un acumulador de errores en el tiempo que dejará de crecer cuando se consigue anular el error, por esta razón es conocido como un control del pasado y ayuda a reducir el error en estado estacionario que mantiene el control proporcional (Salt et al., 2015). La acción integral mejora las características del régimen permanente, pero empeora las características del transitorio donde se desea que modifique la respuesta lo menor posible, motivo por el cual la ganancia integral ( $K_i$ ) debe ser pequeña (Reyes Cortés & Cid Monjaraz, 2015).
- **Acción Derivativa ( $K_p * \frac{de(t)}{dt}$ ).** - Ésta acción utiliza la derivada para realizar una predicción de la evolución de la señal de error, es por ello que es conocido como control del futuro y permite actuar sobre la planta y corregir los errores antes de que se produzcan (Salt et al., 2015). La acción de control derivativa funciona como un freno o amortiguador que elimina sobreimpulsos y oscilaciones en la fase transitoria (Reyes Cortés & Cid Monjaraz, 2015).

En la Figura 1.19, se puede observar el diagrama de bloques de un sistema de control PID digital, en ella se muestra la retroalimentación del sistema y como se integran las tres acciones de control.

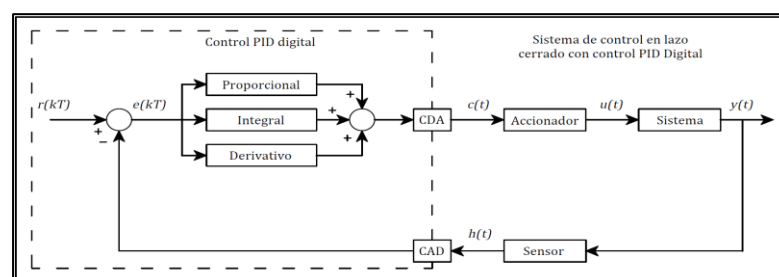


Fig. 0.19. Diagrama de bloques del sistema de control PID digital.

Fuente: (Del Busto, 2013)

Una vez definido el control a utilizar es necesario conocer en donde implementarlo, para ello la manera más fácil y menos costosa es mediante la utilización del microcontrolador Arduino, dispositivo muy amigable al usuario, y ampliamente utilizado a lo largo de la carrera.

#### **1.4.4. Procesamiento y adquisición de datos**

La tarjeta de desarrollo Arduino permite realizar tareas de procesamiento y adquisición de datos, su capacidad de procesamiento posibilita la implementación de algoritmos de control. Arduino puede interactuar con el mundo exterior al tomar información y datos a través de sus pines de entrada por medio de la variedad de sensores que existen en el mercado, por esta razón puede ser usado para controlar y actuar sobre todo aquello que le rodea (Tojeiro, 2014).

##### **1.4.4.1. Tarjeta de desarrollo Arduino Mega 2560**

Tojeiro (2014), define al Arduino como una plataforma electrónica y de programación para la creación de prototipos basada en software y hardware libre.

El lenguaje de programación de Arduino se basa en C/C++ y un entorno de desarrollo (IDE), además debido a que cuenta con un puerto USB se puede establecer una comunicación serial RS232 que expande sus prestaciones y potencialidades al ser capaz de comunicarse con otros paquetes como MATLAB, LabView, Python, entre otros (Reyes Cortés & Cid Monjaraz, 2015).

Arduino Mega 2560 es una placa basada en el microcontrolador ATmega2560 cuyas características más relevantes son (Torrente, 2016):

- 54 pines entradas/salidas digitales (de los cuales 14 pueden ser utilizados como salidas analógicas PWM).
- 16 pines de entradas analógicas y sus 4 receptores/transmisores serie TTL-UART.
- Memoria Flash de 256 kilobytes.
- Memoria SRAM de 8KB y una EEPROM de 4KB.
- Oscilador de cristal de 16MHz.
- Voltaje de trabajo de (5V) y límite máximo de corriente admitida por pin es de (40mA).
- 6 interrupciones disponibles: (INT0 en pin 2), (INT1 en pin 3), (INT2 en pin 21), (INT3 en pin 20), (INT4 en 19) y (INT5 en pin 18).
- 6 temporizadores distribuidos: (Timer0 en pines 4 y 13), (Timer1 en pines 11 y 12), (Timer2 en pines 9 y 10), (Timer3 en pines 2,3 y 5), (Timer4 en pines 6,7 y 8) y (Timer5 en pines 44,45 y 46)

### 1.4.5. Interfaz Gráfica de Usuario

Al emplear Arduino, la interfaz con el usuario mediante software es únicamente a través de la consola o terminal de Arduino IDE. Para que los proyectos tengan una apariencia amigable y cómoda con el usuario es preciso implementar una Interfaz gráfica de usuario o también llamada GUI por sus iniciales en inglés (Graphical User Interface), lo que permite dotar de un aspecto profesional mediante la elaboración de entornos gráficos de ventanas (Tojeiro, 2016).

La GUI puede ser implementada dentro de un mini ordenador como la Raspberry Pi 3, programada mediante el lenguaje de programación de Python, utilizando como herramienta gráfica Tkinter, la cual es un binding<sup>3</sup> de la biblioteca gráfica Tcl/Tk para Python (Tojeiro, 2016).

#### 1.4.5.1. Raspberry Pi 3 modelo B+

La Raspberry Pi 3 B+ es una tarjeta de desarrollo embebido basada en un procesador ARM Cortex a 1.4GHz, la cual requiere la instalación de un sistema operativo basado en Linux para su funcionamiento (Tojeiro, 2016). La RPI al no contar con memoria de almacenamiento interno, necesita de una memoria micro-SD externa para poder ejecutar el sistema operativo.

Según el datasheet del fabricante Raspberry Pi Foundation (2018), las especificaciones técnicas del modelo de Raspberry Pi 3 B+ son:

- Procesador: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- RAM: 1GB LPDDR2 SDRAM
- Wi-Fi + Bluetooth: 2.4GHz y 5GHz IEEE 802.11.b/g/n/ac, Bluetooth 4.2, BLE
- Ethernet: Gigabit Ethernet sobre USB 2.0 (300 Mbps)
- GPIO de 40 pines. HDMI. 4 puertos USB 2.0
- Puerto CSI para conectar una cámara
- Puerto DSI para conectar una pantalla táctil
- Salida de audio estéreo y vídeo compuesto
- Micro-SD. Power-over-Ethernet (PoE)
- Alimentación: 5VDC/2.5A mediante conector micro USB

---

<sup>3</sup> Un binding es una adaptación de una biblioteca para ser usada en un lenguaje de programación distinto de aquel en el que ha sido escrita.

#### **1.4.5.2. Instrumentación Virtual**

A diferencia de la instrumentación tradicional en la que se requieren instrumentos físicos para medir múltiples variables o botoneras reales para ejecutar determinadas acciones, un instrumento virtual se ejecuta en una computadora y tiene sus funciones definidas por software (Pineda, 2013). En un instrumento tradicional no se tiene interacción con equipos de cómputo en tiempo real, anulando la posibilidad de comunicación inalámbrica y aplicaciones móviles.

Debido a esto ha llevado al nacimiento de la instrumentación virtual, el cual es un término asociado al uso de instrumentos virtuales dado que se encuentran dentro de la computadora y mediante los cuales se pueden desarrollar aplicaciones computacionales basadas en GUI's, proporcionando mejores formas de innovar y reducir significativamente los costos al reducir hardware mediante software (Pineda, 2013).

Python ha aumentado considerablemente su popularidad debido a que es el lenguaje principal de programación utilizado en mini ordenadores como lo es la Raspberry Pi, en Python mediante la librería Tkinter se puede llevar a cabo aplicaciones como sistemas de instrumentación virtual remota con el fin de elaborar prácticas de control de procesos en una plataforma web (Machado & Coto, 2017).

# CAPÍTULO 2

## Diseño Electrónico y Automatización

### 2.1. Introducción

El sistema de control y automatización del equipo es el responsable de obtener una soldadura adecuada. Consta de varios elementos los cuales interactúan de forma conjunta y aportan al cumplimiento de los requerimientos necesarios del sistema.

En el presente capítulo se abordará el diseño electrónico, control de la temperatura, así como la interfaz de usuario, los cuales en conjunto conforman el sistema automatizado.

#### 2.1.1. Diseño de Hardware

El diseño del hardware consiste en la adecuación de la planta de temperatura y diseño de un módulo de control. A continuación, se detalla cada una:

##### 2.1.1.1. Planta de temperatura

La planta de temperatura debe contar con los elementos necesarios para que puedan llevar a cabo los requerimientos del proceso. El proceso para llevar a cabo la soldadura por reflujo requiere de dos etapas: Etapa de calentamiento y etapa de enfriamiento.

En la Figura 2.1 se detalla la estructura general del equipo de soldadura con el fin de conocer sus componentes.

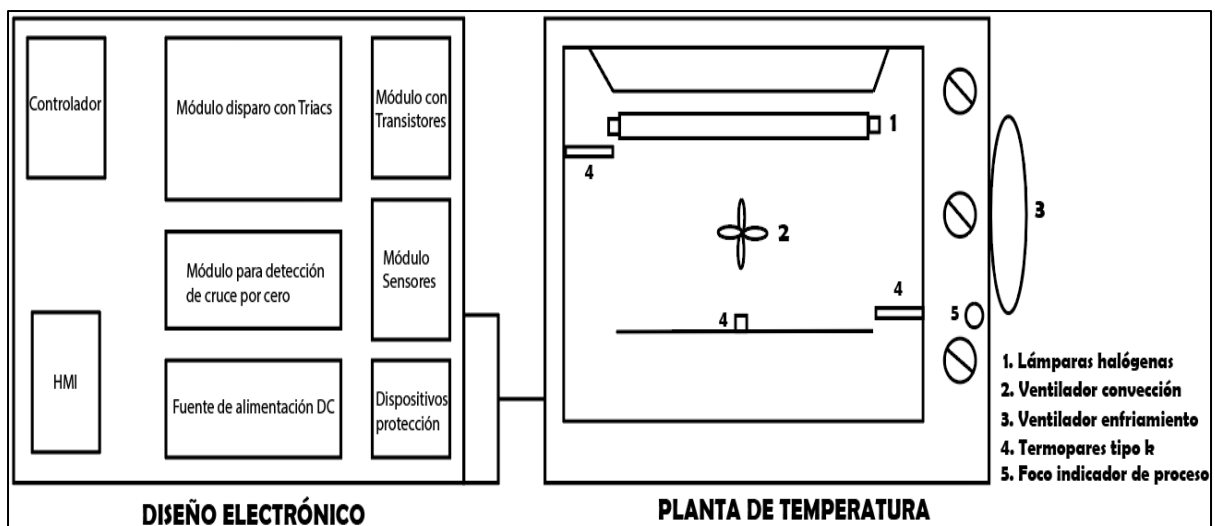


Fig. 0.1. Estructura del equipo de soldadura.

Fuente: (El Autor, 2020)



Para la etapa de calentamiento se utilizará:

- Lámparas halógenas de cuarzo lineal de 1500W como elementos calefactores.
- Ventilador universal para la circulación del aire caliente.

Para la etapa de enfriamiento se utilizará:

- Ventilador DC Brushless.

#### **2.1.1.2. Módulo de control**

El control del Horno se lo realiza a través de un módulo de control con el fin de proteger la circuitería del calor generado por el horno. El proceso requiere el seguimiento de una curva de temperatura que se aproxime a la proporcionada por el fabricante de la suelda a utilizar.

A continuación, se detalla los elementos necesarios para llevar a cabo el control de temperatura.

Interfaz Hombre-Máquina (HMI):

- Raspberry Pi con pantalla táctil 3.5"

Toma de datos y controlador:

- Arduino Mega 2560
- Termopar tipo k
- Módulo Max 6675

Módulos para poder controlar la planta:

- Módulo para la detección del cruce por cero.
- Módulos de disparo con Triacs.
- Módulo para accionamiento de ventiladores a través de transistores.
- Módulo para incorporar todos los sensores a utilizar.

Fuente de alimentación y dispositivos de protección:

- Fuente de voltaje de 12 y 5 VDC.
- Interruptor termomagnético y fusible.

### 2.1.2. Diagrama de bloques del sistema automatizado

Un sistema puede ser tan complejo que resulta difícil su interpretación, por esta razón se emplea diagrama de bloques para la representación del funcionamiento de un sistema de manera gráfica mediante bloques, cuyas interconexiones definen sus entradas y salidas (Hernández, 2010). En los bloques se debe detallar los elementos que interactúan en el sistema para comprender su estructura, comportamiento y conexión.

En la Figura 2.2 se indica el diseño de diagrama de bloques del sistema automatizado para un horno de convección destinado a la soldadura de componentes electrónicos de montaje superficial, donde participan los elementos necesarios para llevar a cabo el control y monitoreo de la temperatura.

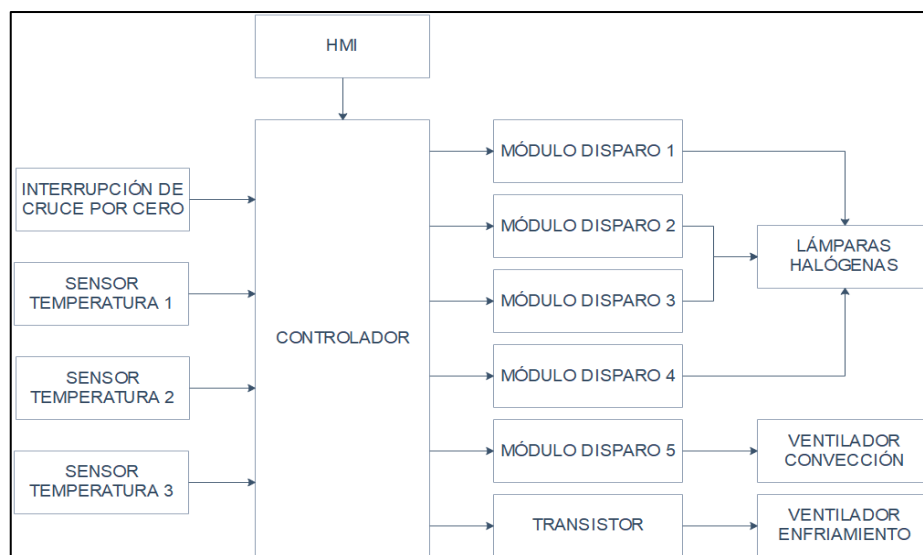


Fig. 0.2. Diagrama de bloques del sistema automatizado.

Fuente: (El Autor, 2020)

Este sistema debe realizar varias funciones a la vez ya que incorpora elementos que permiten proporcionar datos (sensores de temperatura), iniciar o parar el proceso y adquisición de datos en la GUI (comunicación serial), controlar la potencia en corriente alterna de los elementos calefactores (Triacs) y por ende la temperatura del horno, permitir encendido o apagado de ventilador de convección (Triac), regular la velocidad del ventilador de enfriamiento (transistor) y principalmente procesar datos de los sensores en el controlador para tomar decisiones de acuerdo a los valores de referencia.

Para ejecutar todos los requisitos del sistema anteriormente mencionados, se requieren elementos adecuados que se adapten a las necesidades, elementos que puedan soportar altas temperaturas y dispositivos que puedan funcionar de manera continua y que su precisión sea buena.

A continuación, se detalla la elección de los elementos de acuerdo a su funcionalidad, características y conexiones dentro del sistema para poder llevar a cabo el proceso de soldadura por reflujo.

#### **2.1.1.1. Controlador**

Si bien se puede utilizar la Raspberry Pi para realizar todo el control del proceso térmico, existe una limitante en cuanto a la cantidad de entradas/salidas (I/O) utilizables, otro inconveniente es que al utilizar un sistema operativo para su funcionamiento los tiempos de respuesta que se consiguen programando en Python se vuelve un problema.

La solución a estas problemáticas se resuelve con la combinación de la RPI3 con Arduino, siendo este el encargado de llevar a cabo todo el control del hardware y empleando la RPI3 como controlador maestro.

Debido a las características del Arduino Mega 2560 descritas en el capítulo I se opta por utilizarlo como controlador del hardware para esta aplicación.

#### **2.1.1.2. Sensor de temperatura**

Uno de los elementos esenciales para llevar a cabo el control y monitoreo es a través de sensores de temperatura los cuales captan el valor de la temperatura en el interior del horno y permiten brindar la información necesaria al controlador para la toma de decisiones.

En base al análisis realizado en el capítulo anterior y debido a que el termopar es el elemento mayormente utilizado en la industria de fabricación de hornos se considera apropiado su utilización, además que sus características satisfacen en gran medida los requerimientos y sus precios son accesibles.

El sensor a utilizar es la termocupla tipo k de tornillo usado en aplicaciones con Arduino, pero como se mencionó en el capítulo anterior, este sensor necesita de un elemento acondicionador de señal para poder ser utilizado, este proceso se conoce como adecuación del termopar.

En el mercado existen diferentes circuitos integrados que a más de compensar la unión fría amplifican la señal de salida lo que facilita la interpretación y la conexión del termopar a las entradas analógicas y digitales de un microcontrolador como Arduino.

Tal es el caso del circuito integrado MAX6675 el cual es un convertidor analógico/digital especialmente utilizado para termopares tipo K. En el interior de este pequeño circuito se encuentra la electrónica necesaria para amplificar, compensar y convertir a digital el voltaje

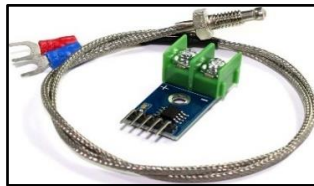
generado por el termopar, lo que hace más sencilla la tarea de conectar un termopar a un microcontrolador mediante una interfaz SPI unidireccional.

En resumen, el transductor de temperatura seleccionado es el módulo MAX6675 el cual incorpora una termocupla tipo K. El MAX6675 no solo amplifica la señal, sino que también linealiza la respuesta del termopar y realiza la compensación de unión fría.

En la Tabla 2.1 se observa las especificaciones técnicas del transductor.

TABLA 0.1  
ESPECIFICACIONES TÉCNICAS DEL TRANSDUCTOR DE TEMPERATURA

CARACTERÍSTICAS DE LA TERMOCUPLA TIPO K	CARACTERÍSTICAS ADC MAX6675
<ul style="list-style-type: none"> <li>• Termocupla tipo K de tornillo</li> <li>• Rosca M6</li> <li>• Rango de Temperatura: 0 – 800°C</li> <li>• Longitud del cable: 1 metro</li> <li>• Longitud del sensor: 25mm</li> </ul>	<ul style="list-style-type: none"> <li>• Interfaz compatible con SPI solo de lectura</li> <li>• Resolución de 12 bits, Precisión: <math>\pm 0.25</math> °C</li> <li>• Medición hasta 1024 °C</li> <li>• Alimentación de 3.3 a 5 volts</li> <li>• Frecuencia de reloj SPI máxima 4.3 MHz</li> <li>• Tiempo de conversión 0.22 segundos (mínimo 0.17 s)</li> <li>• Consumo máximo de 1.5 mA</li> </ul>



Fuente: (Maxim Integrated Products, 2014)

El módulo MAX6675 se comunica con el Arduino a través de una interfaz periférica serial o SPI por sus siglas en inglés (Serial Peripheral Interface). Arduino dispone de ciertos pines para lograr dicha comunicación, aunque por software es posible configurar cualquier otro grupo de pines.

En la Tabla 2.2 se puede observar la distribución de pines para su conexionado. Siendo la descripción de los pines de la siguiente manera:

- GND (Ground). Conexión a tierra.
- VCC (Positive Supply). Conexión a fuente de 5VDC positiva.
- SCK (Serial Clock Input). Entrada de reloj serial.
- CS (Chip Select). Selección de chip.
- SO (Serial Data Output). Salida de datos seriales.

TABLA 0.2  
PINES DE CONEXIÓN ENTRE MODULO MAX6675 Y ARDUINO MEGA

MODULO MAX6675	ARDUINO MEGA	GRÁFICO
SO	50	
CS	53	
SCK	52	
VCC	5V	



Por esta razón, el elemento que se empleará en el proyecto para generar calor será un conjunto de lámparas halógenas gracias a su gran capacidad de generar calor.

#### 2.1.1.4. Ventilador convección

Tras el análisis bibliográfico de los tipos de ventiladores empleados para realizar la convección forzada se concluye que el rodete a utilizar es de tipo álabe radial ya que permite impulsar aire a elevada temperatura, ideal para éste aplicativo.

Debido al costo y dificultad para conseguir un ventilador para un horno de convección, se opta por construir uno. El motor utilizado para la convección es un motor universal de 120VAC con una potencia de 25W, obteniendo una corriente de 0.208 A. En la Figura 2.4 se muestra el álabe radial metálico con sus medidas.

- Diámetro = 110 mm
- Longitud de aspa = 10 mm



Fig. 0.4. Medidas para rodete de ventilador.

Fuente: (El Autor, 2020)

Se obtiene un ventilador para el proceso de convección como el mostrado en la Figura 2.5a), que se asemeja tanto en forma como en funcionamiento a uno comercial Figura 2.5b).



Fig. 0.5. Ventilador para horno de convección: a) Elaborado, b) Comercial.

Fuente: (El Autor, 2020)

### 2.1.1.5. Módulo de disparo

Debido a que se va usar cuatro lámparas halógenas de 1500W a 120V en corriente alterna y un Arduino Mega para controlar los Triacs, la salida del microcontrolador debe estar aislada para la protección del mismo, es decir el circuito de control debe estar aislado eléctricamente del circuito de potencia, por esta razón para activar los Triacs se debe diseñar un módulo de disparo.

El primer paso para diseñar el módulo de disparo es conocer la corriente que va consumir la carga. Para no sobrecargar al Triac y evitar problemas de calentamiento se decide utilizar 4 Triacs, cada Triac va controlar una lámpara.

Al aplicar la ley de Ohm se obtiene una corriente de 12.5 amperios por lámpara funcionando sin ningún tipo de control es decir a su máxima potencia, por esta razón se decide emplear el Triac BTA41 que soporta una corriente máxima de 41 amperios.

En la Figura 2.6 se muestra el módulo de disparo propuesto el cual cuenta con un optoacoplador MOC3021 (circuito integrado que constituye la parte de aislamiento entre el circuito de control y la carga conectada a corriente alterna), Triac BTA41 (dispositivo semiconductor que permite controlar el flujo de potencia entregada a las lámparas halógenas), resistencias R2 y R3 cuyo cálculo es una de las partes principales de este diseño. Para calcular dichas resistencias es necesario conocer las características tanto del Triac como del optoacoplador. A continuación, se muestran las principales características eléctricas:

#### MOC3021 (Anexo H)

- $I_{Fmax} = 60mA$  (*Forward Current*), corriente máxima en polarización directa del Led.
- $I_{FT} = 15mA$ , corriente mínima necesaria en LED para disparar el Triac.
- $V_{Fmax} = 1.5V$  (*Forward Voltage*), voltaje máximo del Led en polarización directa.
- $V_R = 3V$  (*Reverse Voltage*), tensión máxima inversa del Led.

#### TRIAC BTA41 (Anexo I)

- $V_{GT} = 1.3V$ , tensión máxima de disparo de compuerta, valor típico 0.7V.
- $I_{GM} = 8A$ , pico de corriente de compuerta.
- $I_{GT} = 50mA$ , corriente máxima de disparo de compuerta.

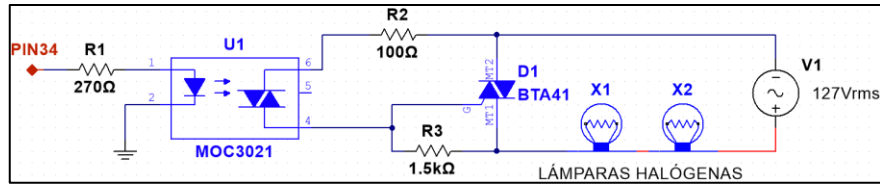


Fig. 0.6. Módulo de disparo con TRIAC para carga resistiva.

Fuente: (El Autor, 2020)

Para calcular la resistencia de disparo  $R_1$  se parte por la ley de Ohm y considerando el voltaje de entrada de 5V (proporcionado por el Arduino) y la corriente que circula por esta que es la corriente necesaria para activar el Triac  $I_{FT}$ .

$$V_{Arduino} = I_{FT} * R_1 + V_{Fmax}$$

$$R_1 = \frac{V_{Arduino} - V_{Fmax}}{I_{FT}} = \frac{(5 - 1.5)V}{15mA} = 233.333\Omega$$

$$R_1 = 270\Omega \text{ Valor comercial}$$

Para que el circuito pueda funcionar correctamente es necesario precisar la corriente que debe pasar por  $R_3$ . El objetivo de  $R_3$  es evitar que el Triac BTA41 se dispare cuando el optocoplador MOC3021 no conduce, ya que puede haber interferencias capaces de producir tensiones de disparo en la puerta del Triac cuando ésta queda suelta. Con  $R_3$  se confirma que la tensión en la puerta  $V_{GT}$  solo será suficiente cuando el optocoplador esté activado. No es necesario que la corriente de  $R_3$  sea muy alta, bastará tomar un valor de 1mA

$$R_3 = \frac{V_{GT}}{I_{R3}} = \frac{1.3V}{1mA} = 1.3k\Omega$$

$$R_3 = 1.5k\Omega \text{ Valor comercial}$$

Se calcula  $R_2$  a partir de las tensiones que hay en los componentes dentro de su malla ( $V_T$ ,  $V_R$  y  $V_{GT}$ ) y de la corriente que pasa por esta, que sería la suma de la corriente de  $R_3$  y la corriente de disparo de compuerta  $I_{GT}$ . El objetivo de  $R_2$  es limitar la corriente a través del Triac.

$$R_2 = \frac{V_T - V_R - V_{GT}}{I_{GT} + I_{R3}} = \frac{(5 - 3 - 0.7)V}{(15 + 1)mA} = 81.25\Omega$$

$$R_2 = 100\Omega \text{ Valor comercial}$$



Para controlar la potencia del ventilador de convección se empleará el mismo circuito de disparo que el anterior, pero al usar un Triac para controlar una carga reactiva como es el ventilador de convección hay que tener precaución para asegurar el apagado del Triac al final de cada medio ciclo de la onda sinusoidal. También hay que considerar la corriente de carga para poder seleccionar el Triac a usar.

Al medir con una pinza amperimétrica se tiene una corriente de 0.21 amperios funcionando a su máxima capacidad, por esta razón se decide emplear el Triac BT136 que soporta una corriente máxima de 4 Amperios.

Las características principales del Triac seleccionado son (Anexo J):

- $V_{GT} = 0.7V$ , tensión de disparo de compuerta.
- $I_{GM} = 2A$ , pico de corriente de compuerta.
- $I_{GT} = 5mA$ , corriente máxima de disparo de compuerta, valor típico 2.5mA.

En la Figura 2.7 se muestra el módulo de disparo propuesto para el ventilador de convección y los cálculos se detalla a continuación:

$$V_{Arduino} = I_{FT} * R_4 + V_{Fmax}$$

$$R_4 = \frac{V_{Arduino} - V_{Fmax}}{I_{FT}} = \frac{(5 - 1.5)V}{15mA} = 233.333\Omega$$

$$R_4 = 270\Omega \text{ Valor comercial}$$

$$R_6 = \frac{V_{GT}}{I_{R6}} = \frac{0.7V}{1mA} = 700\Omega$$

$$R_6 = 680\Omega \text{ Valor comercial}$$

$$R_5 = \frac{V_T - V_R - V_{GT}}{I_{GT} + I_{R6}} = \frac{(5 - 3 - 0.7)V}{(2.5 + 1)mA} = 371.43\Omega$$

$$R_5 = 390\Omega \text{ Valor comercial}$$

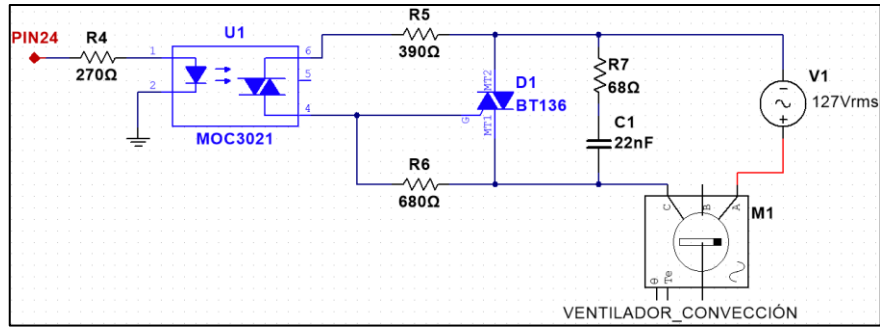


Fig. 0.7. Módulo de disparo con TRIAC para carga Inductiva.

Fuente: (El Autor, 2020)

Un desfase entre voltaje y corriente causado por cargas reactivas puede provocar un salto de voltaje que encienda el Triac por error, estos encendidos accidentales se pueden evitar empleando un circuito Snubber, dicha red consiste en un circuito RC conectado en paralelo al Triac.

La red Snubber se calcula a partir de los datos del fabricante del Triac indicado en el Anexo 9 y aplicando las ecuaciones de Rashid (2004):

Se parte por calcular la corriente de carga y resistencia de carga ( $I_C, R_C$ ), se sabe que el ventilador funciona a 120V con una potencia real de 25W y factor de potencia aproximadamente de 0.95.

$$I_C = \frac{P}{V * \cos\phi} = \frac{25W}{120V * (0.95)} = 0.22A$$

$$R_C = \frac{V}{I_C} = \frac{120V}{0.22A} = 545.45\Omega$$

Si se considera que la corriente de descarga  $I_{TD}$  es 10 veces la corriente nominal, la resistencia en serie del circuito Snubber (R7) se calcula de la siguiente manera:

$$R_S = \frac{V_S}{I_{TD}} = \frac{120V}{10 * 0.22A} = 54.54\Omega = 68\Omega \text{ valor comercial}$$

Para calcular la capacitancia en serie (C1) de la red Snubber se realiza a partir de la ecuación:

$$C_S = \frac{0.632R_C V_S}{\frac{dv}{dt} (R_S + R_C)^2} = \frac{0.632(545.45)(120)}{5(68 + 545.45)^2} = 21.98nF = 22nF \text{ valor comercial}$$

### 2.1.1.6. Ventilador enfriamiento

Para la etapa de enfriamiento se decide utilizar un ventilador de hélice axial ya que son empleados para extraer o inyectar grandes cantidades de aire a presiones bajas, esta característica es requerida para la etapa de enfriamiento del proceso de soldadura.

El ventilador es un motor Brushless de 4 Pines cuyas características eléctricas son las siguientes:

- Voltaje: 12VDC
- Potencia: 15.6 W

En la Tabla 2.3 se detalla la distribución de pines del ventilador a utilizar para la etapa de ventilamiento. Dicha configuración de pines lo hace ideal para poder controlarlo desde una tarjeta arduino sin necesidad de utilizar un driver externo popularmente conocido como puente H, pero, se deberá prestar atención a la frecuencia con la que debe trabajar el PWM para poder controlarlo (Figura 2.8).

El control de la velocidad del ventilador tiene su importancia en el proceso de soldadura debido a que el enfriamiento no debe ser abrupto sino mas bien controlado.



Fig. 0.8. Ventilador Brushless 4 Pines Con Control PWM.

Fuente: (El Autor, 2020)

TABLA 0.3  
DISTRIBUCIÓN PINES VENTILADOR BRUSHLESS

PIN	COLOR	NOMBRE
1	Azul	Control PWM
2	Negro	GND
3	Rojo	+12VDC
4	Blanco	Sensor RPM

Fuente: (El Autor, 2020)

Hay que tener en cuenta que para variar la velocidad del ventilador se lo realiza por medio de PWM (Pulse Width Modulation) o modulación por ancho de pulsos a una frecuencia de 25KHz utilizando el Timer 2 del Arduino, pero existe una relación fija entre los temporizadores y las salidas PWM, en este caso, los pines 9 y 10 son controlados por el Timer 2.

Para ayudar al enfriamiento de las placas de potencia se decide utilizar un motor Brushless a 12 VDC con 1.5W de potencia. Tal como se muestra en la Figura 2.9 el ventilador posee dos cables de conexión: 1 cable rojo (12VDC), 1 cable negro (GND). Para poder controlar el encendido y apagado del ventilador desde Arduino es necesario emplear un transistor.

En este ventilador a diferencia del anterior, no es posible regular la velocidad de giro directamente, pero, como se va utilizar únicamente para ayudar a enfriar los disipadores de calor de los Triacs no se necesita controlar la velocidad sino más bien los encendidos y apagados a través de transistores.



Fig. 0.9. Ventilador Brushless 2 Pines.

Fuente: (El Autor, 2020)

### 2.1.1.7. Transistor

El transistor es el dispositivo semiconductor necesario para conmutar los ventiladores de corriente continua a través de Arduino, el cual además brindará la protección necesaria a los pines del microcontrolador para encender cargas con corrientes superiores a los 40mA que es lo que normalmente soporta.

Debido a que se va emplear dos ventiladores Brushless, se empleará transistores BJT para poder accionarlos. El transistor tiene 3 pines de conexión: emisor, colector y base. La elección del tipo de transistor se la realiza en base a la corriente de colector.

El ventilador de enfriamiento del proceso va consumir una corriente de 1.30A por lo tanto se opta por utilizar el transistor TIP122 NPN, mientras que el ventilador de enfriamiento para las placas de potencia va consumir una corriente de 0.12A por tanto se opta por utilizar un transistor 2N3904 NPN.

En la Figura 2.10 se puede observar el circuito para lograr la conmutación de los ventiladores a través del Arduino. Es necesario asegurar que la corriente de base sea lo suficiente para que el transistor entre en saturación, esto se lo obtiene por medio de una resistencia.

Para calcular la resistencia R1 se obtiene los datos del fabricante del transistor TIP122 en el Anexo K, a partir del cual se obtiene:

$$I_B = \frac{I_C}{h_{FE}} = \frac{1A}{1000} = 1mA$$

$$V_{RB} = V_{PIN} - V_{BE} = (5 - 1.1)V = 3.9V$$

$$R_B = \frac{V_{RB}}{I_B} = \frac{3.9V}{1mA} = 3.9K\Omega$$

Para calcular la resistencia R2 se obtiene los datos del fabricante del transistor 2N3904 en el Anexo L, a partir del cual se obtiene:

$$I_B = \frac{I_C}{h_{FE}} = \frac{0.12A}{100} = 1.2mA$$

$$V_{RB} = V_{PIN} - V_{BE} = (5 - 0.65)V = 4.35V$$

$$R_B = \frac{V_{RB}}{I_B} = \frac{4.35V}{1.2mA} = 3625\Omega = 3.9K\Omega \text{ valor comercial}$$

El microcontrolador permite conmutar los ventiladores por medio del pin 45 y 41 respectivamente, para lograr variar la velocidad del primer ventilador se lo realiza a través del pin 9 conectado en el terminal del ventilador que es destinado para este fin.

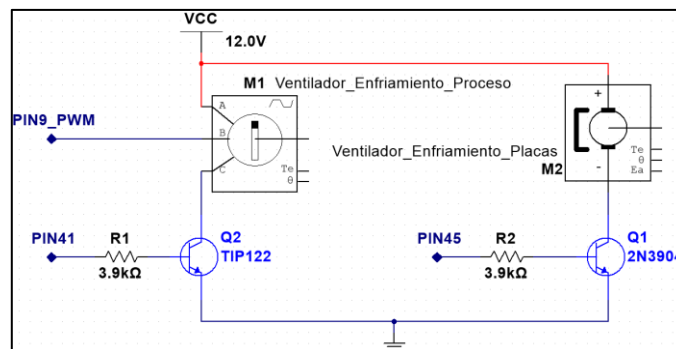


Fig. 0.10. Circuito de control para ventiladores de enfriamiento.

Fuente: (El Autor, 2020)

### 2.1.1.8. HMI (Interfaz Hombre Máquina)

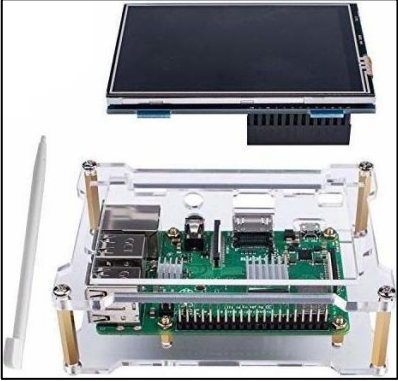
Para la implementación de un HMI se toma en cuenta la facilidad de uso del equipo y la parte económica pues se podría diseñar una interfaz gráfica a través de un software industrial especializado para tal fin como Labview, pero esto implicaría costos adicionales por usos de licencias además de requerir un computador con dicho software incorporado para poder utilizar el equipo de soldadura.


Por tal situación se opta por utilizar un miniordenador Raspberry Pi, el cual es un dispositivo económico que cuenta con periféricos de salida ideales para implementar una interfaz gráfica a pequeña escala.

Para poder visualizar las variables e interactuar con el Horno de soldadura se provee de una pantalla TFT-LCD (Thin Film Transistor-Liquid Crystal Display, Pantalla de cristal líquido de transistores de película fina) de 3.5 pulgadas con función táctil, la cual se puede montar directamente en los pines de propósito general (GPIO) de la Raspberry PI sin requerir alimentación externa.

Las características se muestran en la Tabla 2.3.

TABLA 0.4  
ESPECIFICACIONES TÉCNICAS DE LA PANTALLA TÁCTIL TFT-LCD

CARACTERÍSTICAS PRINCIPALES	MONTAJE
<ul style="list-style-type: none"><li>- Modelo MPI3501</li><li>- Resolución de 320x480</li><li>- Tecnología IPS</li><li>- Control Touch Resistivo</li><li>- Ajuste perfecto a la RPI</li><li>- No necesita drivers para Raspbian/Ubuntu</li></ul>	

Fuente: (AV Electronics, 2020)

### 2.1.1.9. Fuente de alimentación DC

La fuente de alimentación DC es el elemento que se encarga de energizar a los ventiladores de enfriamiento tanto para el proceso como para las placas. El voltaje con el que trabaja los ventiladores es de 12VDC; el ventilador destinado al enfriamiento del proceso tiene un consumo de corriente de 1.30A y el ventilador destinado al enfriamiento de las placas tiene un consumo de 0.12A lo que resulta en una corriente total de 1.42A a 12 VDC.

Utilizando una fuente de alimentación de 12VDC a 5A se puede cubrir el consumo de los ventiladores y se aprovecha también este voltaje rectificado para reducirlo a 5VDC mediante el regulador de voltaje LM7805 para alimentar a los tres sensores de temperatura empleados en este proyecto.

En la Figura 2.11 se muestra la fuente de alimentación en DC y el circuito reductor.

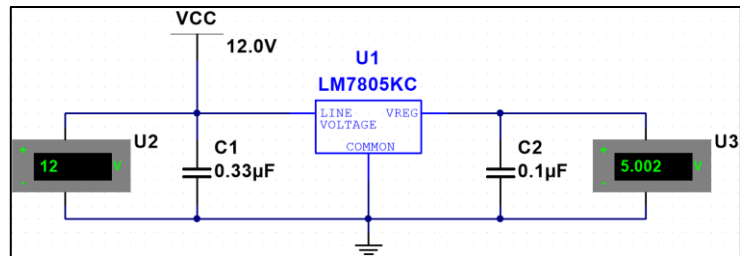


Fig. 0.11. Circuito reductor de voltaje.

Fuente: (El Autor, 2020)

#### 2.1.1.10. Dispositivos de protección

Para la protección del circuito de fuerza es necesario utilizar un interruptor termomagnético y para el circuito de control un fusible.

El interruptor termomagnético se dimensiona tomando en consideración lo siguiente:

Para alcanzar la temperatura máxima del horno (230 °C) es suficiente utilizar la mitad de la potencia de cada lámpara halógena, obteniendo una corriente aproximada de 17 A.

El interruptor termomagnético a utilizar debe ser superior a 17 A, por lo tanto, un valor comercial superior es de 20 A.

Para el dimensionamiento del fusible se considera la corriente necesaria para el funcionamiento del miniordenador Raspberry Pi, la misma es de 2 A. Para el funcionamiento de ventiladores tanto de convección y enfriamiento se tiene una corriente prevista de 1.7 A resultando una corriente de 3.7 A.

El fusible a utilizar se encuentra con un valor comercial de 5 A.

#### 2.1.2. Control de fase directo

El control de fase directo es el método empleado para variar el valor eficaz de una tensión alterna por medio del control por ángulo de fase; la carga recibe un voltaje alterno cuya

amplitud depende del ángulo de disparo, al variar el valor eficaz del voltaje se varía el flujo de potencia al permitir que el Triac conduzca en ciertos intervalos de cada semiciclo de la onda.

La Figura 2.12 muestra el voltaje tanto en la carga como en los terminales del Triac para ángulos de fase de 30° y 120°.

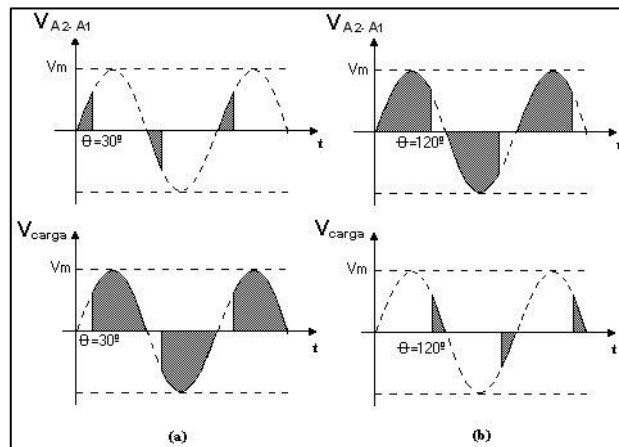


Fig. 0.12. Control de fase directo con Triac.

Fuente: (Paredes, 2020)

### 2.1.2.1. Control de fase con Arduino

Antes de empezar con cualquier proyecto de automatización en el cual involucre el control en CA, es necesario conocer la velocidad a la cual trabaja el dispositivo en el cual se va a implementar el control automático, en este caso se decide utilizar una tarjeta de desarrollo Arduino. El control de fase de CA se realiza utilizando el control interno hardware Timer1 en Arduino.

El reloj del Arduino funciona a 16 MHz, es decir 16 000 000 ciclos por segundo.

$$T = \frac{1}{16\,000\,000\text{ Hz}} = 0.0625\mu\text{s}$$

Se determina que un solo ciclo de reloj toma 0.0625 microsegundos. La red eléctrica del Ecuador tiene una frecuencia de 60 ciclos por segundo, por lo tanto, cada semiciclo tendrá una frecuencia de 120 Hz, entonces:

$$\frac{16\,000\,000\text{ Hz}}{120\text{ Hz}} = 133333.3333$$

Un solo medio ciclo de la señal de CA de 60 Hz contiene 133333 ciclos de reloj. Esto es importante porque se determinará los intervalos de tiempo por conteo de reloj en el código Arduino, no por segundos.



Para controlar con precisión el tiempo de disparo de la compuerta del TRIAC con la señal alterna se emplea interrupciones y los temporizadores del Arduino.

Como se mencionó anteriormente, debido al cristal de 16MHz los temporizadores del Arduino pueden incrementar sus contadores a 16MHz (una velocidad demasiado rápida). El Timer1 es un temporizador de 16 bits, lo que quiere decir que puede almacenar un valor de contador máximo de  $2^{16} = 65536$ . Cuando el contador alcanza su máximo valor, éste volverá a cero, esta acción es conocida como desbordamiento.

Para controlar la velocidad del incremento del temporizador y que no se produzcan interrupciones muy rápidas se utiliza el *prescaler*, el cual establece la velocidad de temporización con la siguiente ecuación:

$$\text{velocidad del temporizador (Hz)} = \frac{\text{velocidad del reloj Arduino}}{\text{prescaler}}$$

El preescalador puede ser igual a 1, 8, 64, 256 y 1024.

De este modo se calcula el registro de comparación para que vaya en sincronía con la señal en alterna con la siguiente ecuación:

$$\text{Registro de comparación} = \frac{\text{velocidad del reloj Arduino}}{\text{preescalador} * \text{frecuencia de interrupción desea}} - 1$$

$$\text{Registro de comparación} = \frac{16\,000\,000\text{ Hz}}{256 * 120\text{ Hz}} - 1$$

$$\text{Registro de comparación} \approx 520$$

El temporizador de hardware funciona a 16MHz, usando un divisor por 256 en el mostrador, cada cuenta es de 16 microsegundos; 1/2 onda de una señal de CA de 60Hz es de aproximadamente 520 recuentos (8.333 milisegundos).

Cómo  $520 < 65536$  es adecuado utilizar el Timer1. A partir de aquí solo basta configurar los registros del Timer1 con las consideraciones anteriormente descritas y en base al datasheet del microcontrolador Atmel ATMEGA 2560.

Debido a que el microcontrolador del Arduino no es perfecto, se va perder un poco de tiempo a la hora de detectar el pulso de disparo y para evitar pasarse a la siguiente onda se decide dejar unos márgenes, resultando valores del registro comparador de salida (OCR1A)

que van de 15 a 465. Éstos valores serán los equivalentes al disparo del Triac de 0 a 360°, y en función de esa variación se tendrá mayor o menor potencia en la carga.

### 2.1.3. Circuito detector de cruce por cero

La detección de cruce por cero de la onda sinusoidal se vuelve necesario para realizar una regulación de potencia en AC debido a que se debe sincronizar la señal de disparo del Triac con la frecuencia de la red, de lo contrario provocará parpadeos en la carga.

El circuito detector de cruce por cero se lo realiza con el optoacoplador H11AA1 el cual es un aislador acoplado ópticamente de entrada bidireccional, dispositivo que emite un pulso cuando la señal alterna de entrada pasa por 0V o cuando pasa del semiperiodo negativo al positivo, tal como se indica en la Figura 2.13.

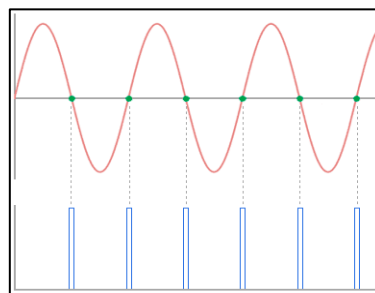


Fig. 0.13. Pulsos de detección de cruce por cero.

Fuente: (Llamas, 2017a)

Esquema de montaje:

El circuito diseñado para la detección de la señal de cruce por cero se indica en la Figura 2.14, en la cual se incluye un arreglo resistivo necesario para reducir los voltajes ya que se necesita alimentar con corriente alterna de 120V, de esta forma se evita quemar el circuito integrado H11AA1.

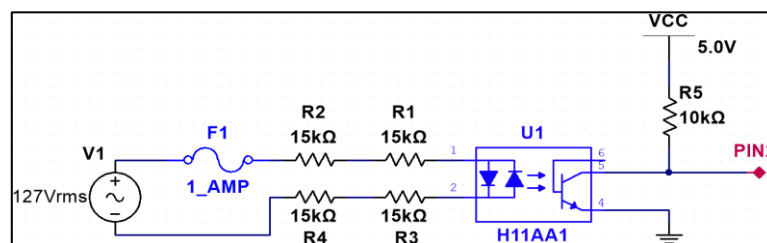


Fig. 0.14. Circuito detector de cruce por cero.

Fuente: (El Autor, 2020)

Por supuesto, se puede conectar este sistema de detección de cruce por cero a un Arduino con una resistencia Pull Up que normalmente es de 10KΩ para evitar que en estado de reposo

se mida un valor erróneo suprimiendo la influencia de agentes externos como el ruido eléctrico sobre la medición.

Arduino tiene incorporado una resistencia Pull Up que se lo puede utilizar mediante software sin necesidad de añadir un circuito extra, simplemente dentro del void setup {} se incluye los siguientes comandos: pinMode (PIN2, INPUT\_PULLUP), de esta manera queda activada la resistencia Pull Up en el Pin 2.

Al tener un voltaje alterno es necesario limitar la corriente a la entrada del optoacoplador H11AA1, se lo realiza mediante las resistencias R1, R2, R3 y R4. En el Anexo M se tiene información del fabricante del optoacoplador, en la cual se tiene  $I_F = 10mA$ ,  $V_F = 1.2V$ , entonces R1 y R3 se calcula mediante la ley de Ohm:

$$V_R = I * R$$

$$-V_{fuente} + V_F + V_R = 0$$

$$V_R = V_{fuente} - V_F$$

$$I * R = V_{fuente} - V_F$$

$$R = \frac{V_F - V_D}{I_F} = \frac{(127\sqrt{2} - 1.2) V}{0.01 A} = 17840.51\Omega$$

$$R = 18k\Omega \text{ valor comercial}$$

donde:

$I_F$ : Corriente continua directa

$V_F$ : Tensión directa

$V_{fuente}$ : Tensión de fuente

Para mayor seguridad se opta emplear una resistencia de 30kΩ con lo cual se tiene una corriente de 6mA, suficiente para polarizar al diodo.

La potencia a disipar se calcula con la siguiente ecuación:

$$P = I^2 * R = (0.006A)^2 * 30k\Omega = 1W$$

Por lo tanto, se precisa emplear dos resistencias en serie de 15kΩ de 1/2W.

Con la ayuda del osciloscopio se puede observar en la Figura 2.15 la señal sinusoidal de entrada y los pulsos en los cruces por cero.

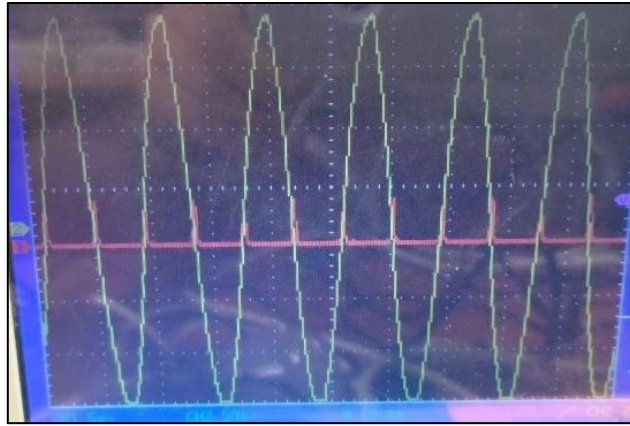


Fig. 0.15. Señal de cruce por cero medida en osciloscopio.

Fuente: (El Autor, 2020)

#### **2.1.4. Diagrama de flujo del microcontrolador Arduino**

Con el propósito de un mejor entendimiento del programa se realiza un diagrama de flujo como se indica en la Figura 2.16, en el mismo se entiende el orden del algoritmo que ejecuta el microcontrolador.

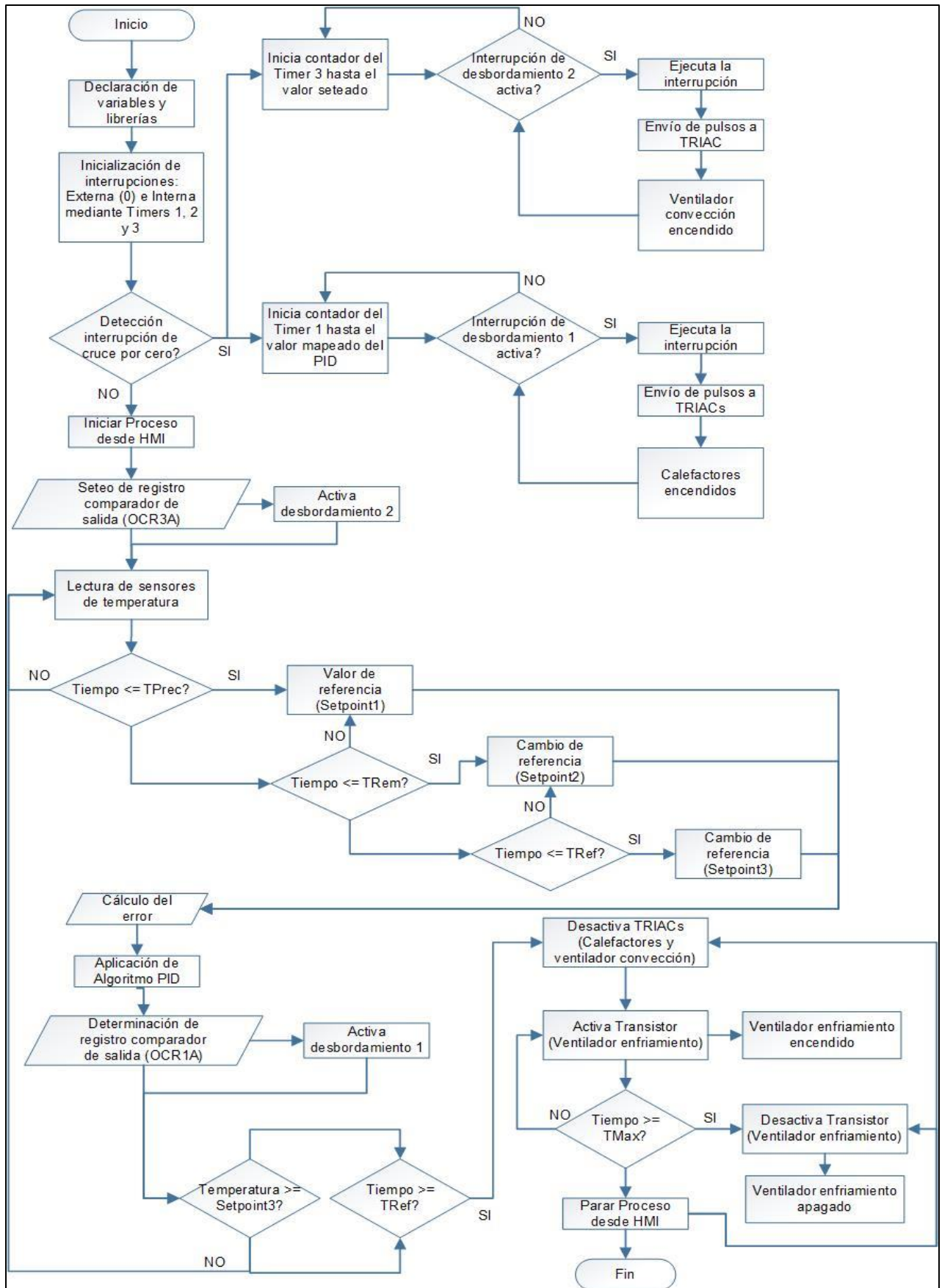


Fig. 0.16. Diagrama de flujo del sistema automatizado.

Fuente: (El Autor,2020)

Al iniciar el programa se comienza ejecutando la declaración de variables y librerías. Con esto el microcontrolador asigna espacios de memoria y los pines a utilizar. Además, las variables son necesarias para almacenar los datos receptados por los sensores y la posterior ejecución adecuada del algoritmo de control PID.

Después de la ejecución de variables el programa entra en un bucle en donde se configura la velocidad de transmisión de datos a través del puerto serial, procedimiento de operación de variables ya sea como entradas o salidas, modo de funcionamiento, asignación de valores iniciales para su posterior utilización, configuración de Timers 1 y 3 y configura la interrupción externa 0. Este bucle se ejecuta una vez al iniciar el programa.

Posteriormente ejecuta una función cuando detecta el cruce por cero. En la misma se inicializa los temporizadores y registros contadores, Se configura los comparadores para que cada vez que el temporizador se dispare el microcontrolador ejecute una función de interrupción que permitirá sincronizar los disparos de los TRIAC únicamente en los cruces por cero de la señal alterna.

Para que la descripción anterior se cumpla es necesario configurar el registro comparador de salida, esta configuración detectará el microcontrolador una vez que reciba la orden de inicio del proceso a través del HMI, al activarse el desbordamiento del Timer ejecuta las interrupciones y en consecuencia da inicio a la activación de los elementos de potencia.

Al mismo tiempo que se da INICIO desde el HMI, empieza la toma de datos de los sensores para su procesamiento dentro del control PID y verifica los tiempos establecidos para los cambios de referencia o setpoint.

Al llegar a la temperatura de Reflujo o máxima del proceso y tras verificar el tiempo establecido para esta etapa, se desactivan las señales que permiten la activación de los Triacs que accionan a los elementos calefactores y se envía una señal de activación al transistor que permite accionar al ventilador de enfriamiento del proceso.

Seguidamente verifica que se cumpla un tiempo máximo establecido y se desactiva todos los ventiladores. En cualquier momento es posible detener el proceso desde el HMI.

### **2.1.5. Diseño de diagrama eléctrico general**

En este apartado se detalla el diagrama eléctrico de potencia en el cual se indica la distribución y conexionado de los elementos de corriente alterna, así como las protecciones necesarias ante posibles cortocircuitos, cabe mencionar que los Triac al ser los encargados de permitir el paso de la corriente alterna entre sus ánodos se comportan como interruptores, pero de tipo electrónicos cuyo funcionamiento se detalló en el módulo de disparo propuesto.

También se incluyen dos convertidores que permite transformar la corriente alterna de 120VAC en 12VDC para el funcionamiento de los ventiladores de enfriamiento y 5VDC para la alimentación del mini ordenador Raspberry Pi 3. Los elementos de potencia permitirán ser accionados a través del interruptor termomagnético mientras q para el encendido y apagado del miniordenador se utiliza un selector de 2 posiciones.

En la Figura 2.17 se muestra el diagrama eléctrico general.

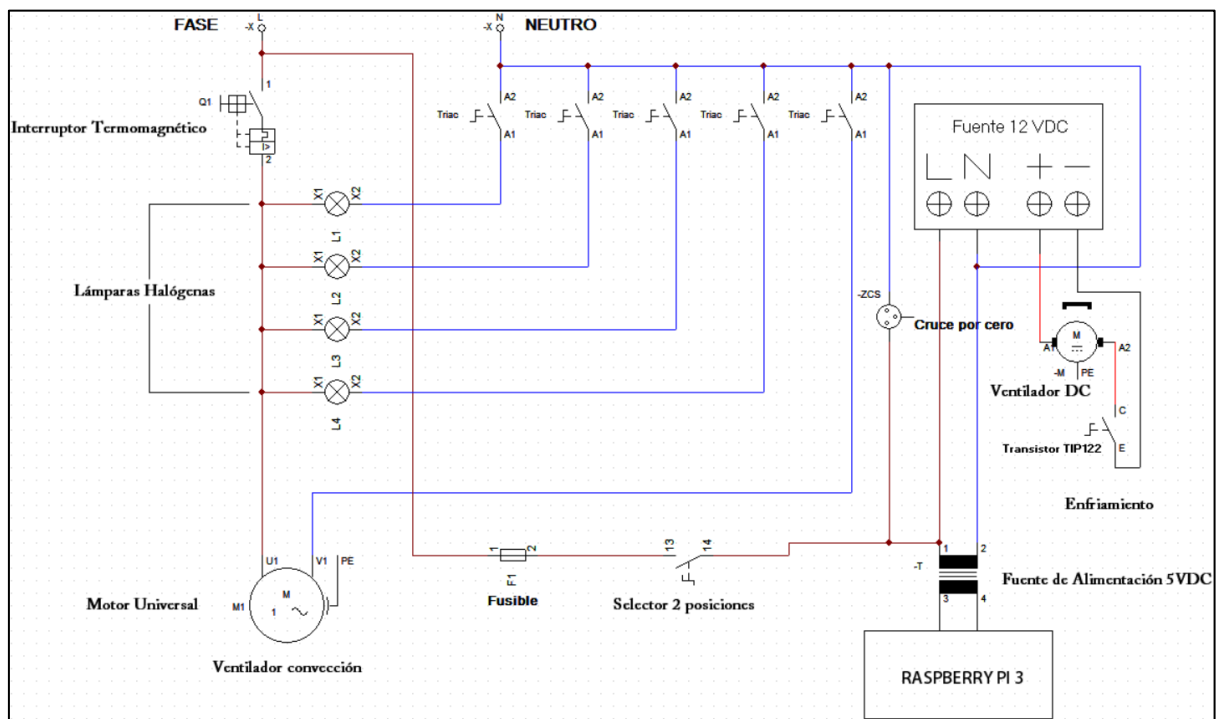


Fig. 0.17. Diagrama eléctrico.

Fuente: (El Autor, 2020)

### 2.1.6. Esquemático general distribución de pines Raspberry - Arduino

En esta sección se detalla el conexionado de los elementos encargados de realizar el control. El RPI3 funciona como controlador maestro desde el cual se puede mandar las ordenes al Arduino y adquirir los datos de los sensores mediante la pantalla táctil, esto es posible gracias a la comunicación serial a través del cable USB. Para que el mini ordenador funcione es necesario alimentarle con una fuente de 5VDC y 3A de corriente.

En la Figura 2.18 se muestra el diagrama de conexionado general para el sistema de control:

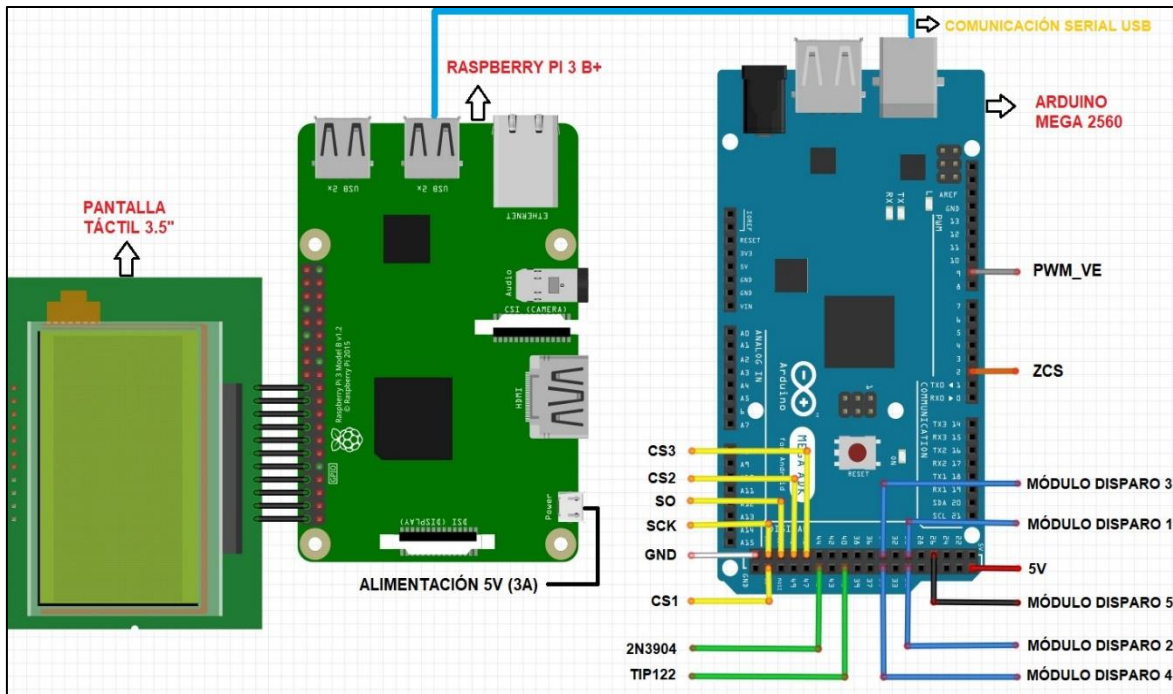


Fig. 0.18. Esquemático general distribución de pines RPI3 - Arduino.

Fuente: (El Autor, 2020)

En la siguiente Tabla se realiza una descripción de los pines utilizados del Arduino para el proceso de automatización.

TABLA 0.5  
DESCRIPCIÓN DE PINES A UTILIZAR EN ARDUINO

PIN CONFIGURADO	DESCRIPCIÓN
PIN 2 → ZCS	Pin configurado como interrupción externa para recibir los pulsos de la señal de cruce por cero.
PIN 9 → PWM_VE	Pin configurado para realizar el control de la velocidad del ventilador de enfriamiento del proceso mediante PWM.
PIN 26 → MÓDULO DE DISPARO 5	Pin configurado para enviar señales de activación al Triac destinado para el control de fase en el ventilador de convección.
PIN 30 → MÓDULO DE DISPARO 1	Pin configurado para enviar señales de activación al Triac destinado para el control de fase de una lámpara halógena.
PIN 31 → MÓDULO DE DISPARO 2	Pin configurado para enviar señales de activación al Triac destinado para el control de fase de una lámpara halógena.
PIN 34 → MÓDULO DE DISPARO 3	Pin configurado para enviar señales de activación al Triac destinado para el control de fase de una lámpara halógena.
PIN 35 → MÓDULO DE DISPARO 4	Pin configurado para enviar señales de activación al Triac destinado para el control de fase de una lámpara halógena.
PIN 41 → TIP122	Pin configurado para enviar señales de activación al transistor TIP122 que permitirá el paso de corriente al ventilador de enfriamiento del proceso.
PIN 45 → 2N3904	Pin configurado para enviar señales de activación al transistor 2n3904 que permitirá el paso de corriente al ventilador de enfriamiento de las placas.
PIN 46 → CS3	Pin configurado para el Chip Select del Módulo Max6675.
PIN 48 → CS2	Pin configurado para el Chip Select del Módulo Max6675.
PIN 50 → SO	Pin configurado para el Serial Data Output del Módulo Max6675.
PIN 52 → SCK	Pin configurado para el Serial Clock Input del Módulo Max6675.
PIN 53 → CS1	Pin configurado para el Chip Select del Módulo Max6675.
PIN 5V	Pin de 5VDC.
PIN GND	Pin de Tierra.

Fuente: (El Autor, 2020)



### 2.1.7. Diseño de placas de circuito impreso (PCB)

Una vez realizado los cálculos necesarios de los circuitos, se procede a realizar los circuitos en PCB utilizando el software KiCad.

#### 2.1.7.1. Circuito detector de cruce por cero y Módulo de disparo 3

En la Figura 2.19 se muestra el esquemático general:

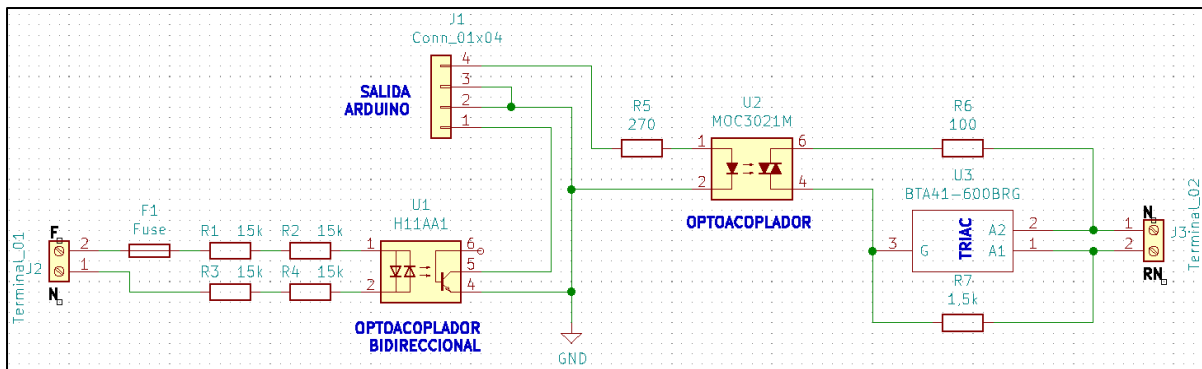


Fig. 0.19. Módulo de disparo 3 y circuito detector de cruce por cero - esquemático general.

Fuente: (El Autor, 2020)

En la Figura 2.20 se muestra el diseño de la PCB como su visualización en 3D.

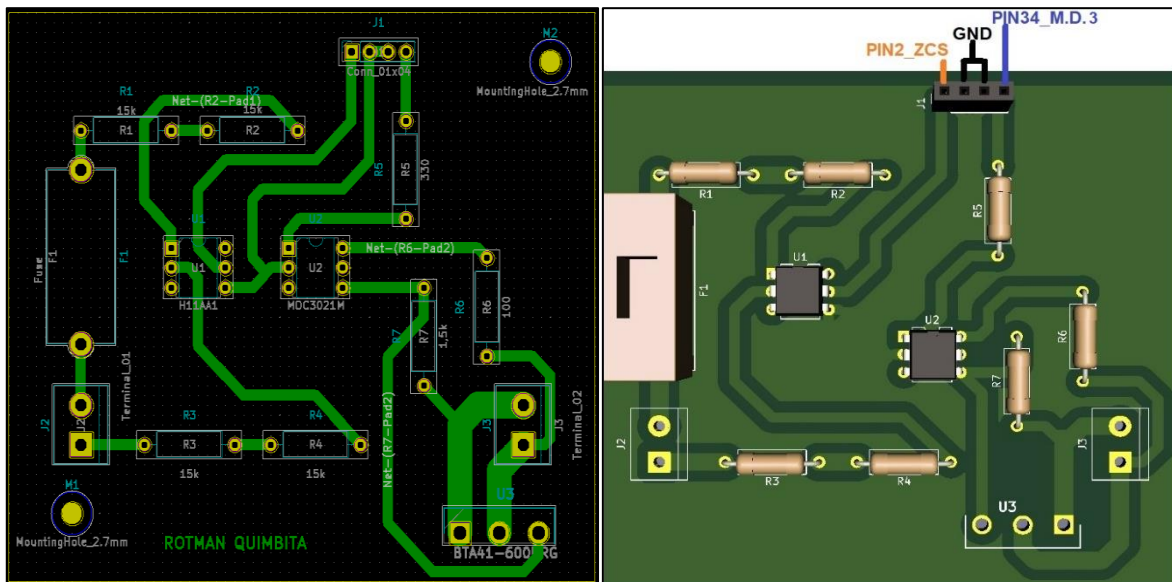


Fig. 0.20. Módulo de disparo 3 y circuito detector de cruce por cero - diseño PCB y visualización 3D.

Fuente: (El Autor, 2020)

#### 2.1.7.2. Módulo de disparo 2 y 4

En la Figura 2.21 se muestra el esquemático general:

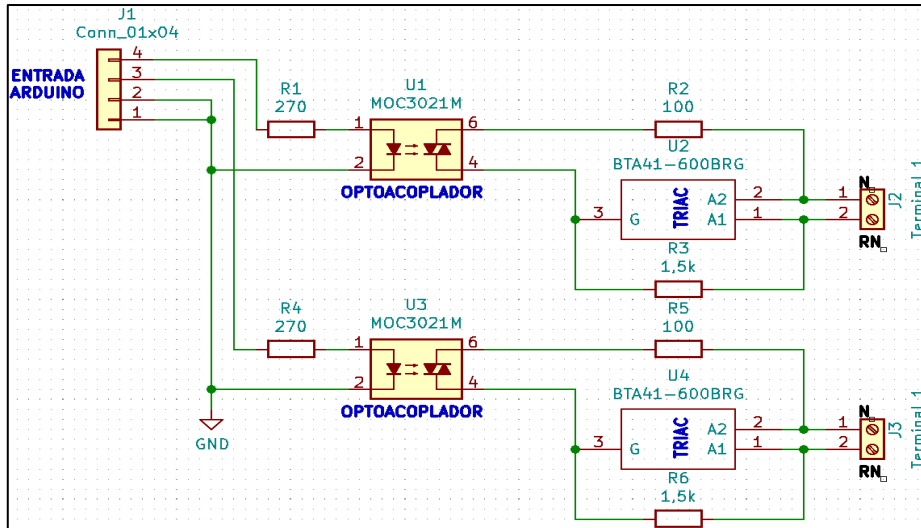


Fig. 0.21. Módulo de disparo 2 y 4 - esquema general.

Fuente: (El Autor, 2020)

En la Figura 2.22 se muestra el diseño de la PCB como su visualización en 3D.

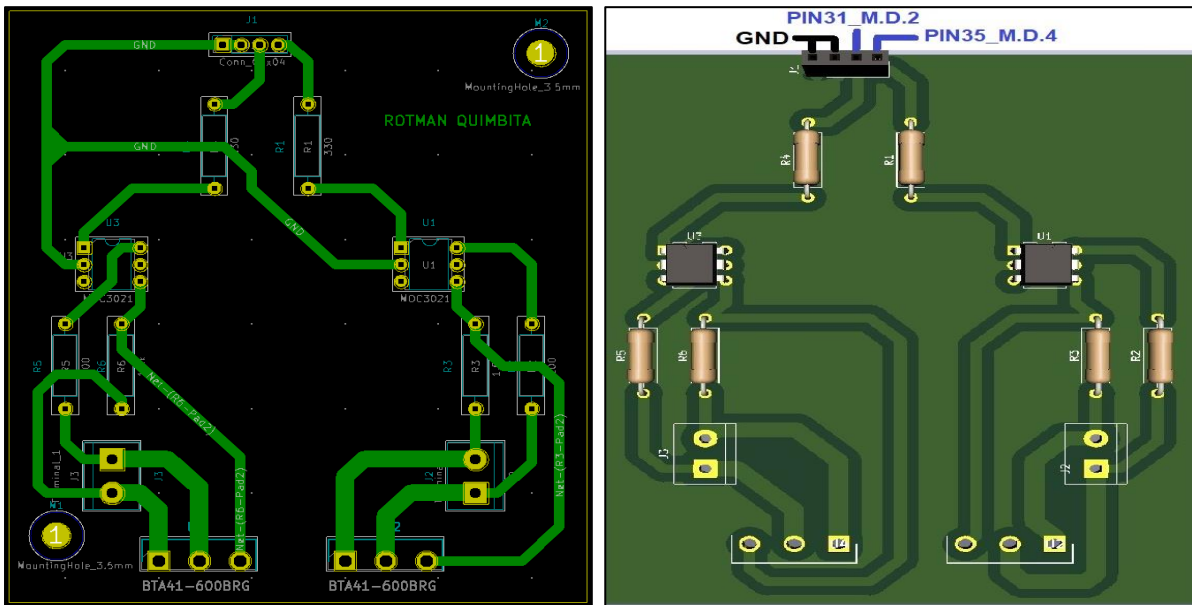


Fig. 0.22. Módulo de disparo 2 y 4 - diseño PCB y visualización 3D.

Fuente: (El Autor, 2020)

### 2.1.7.3. Módulo de disparo 1 y 5

En la Figura 2.23 se muestra el esquema general:

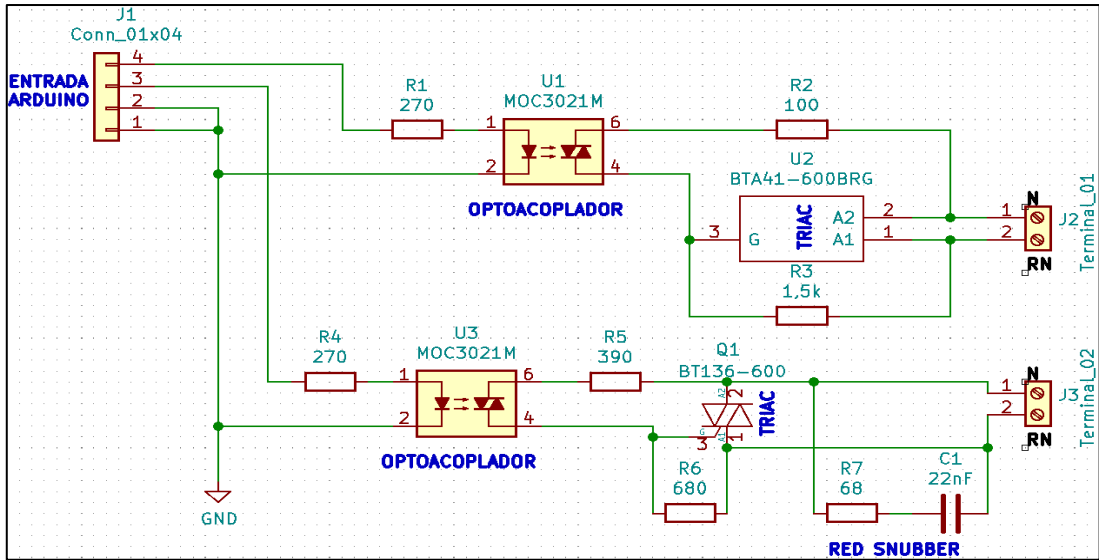


Fig. 0.23. Módulo de disparo 1 y 5 - esquema general.

Fuente: (El Autor, 2020)

En la Figura 2.24 se muestra el diseño de la PCB como su visualización en 3D.

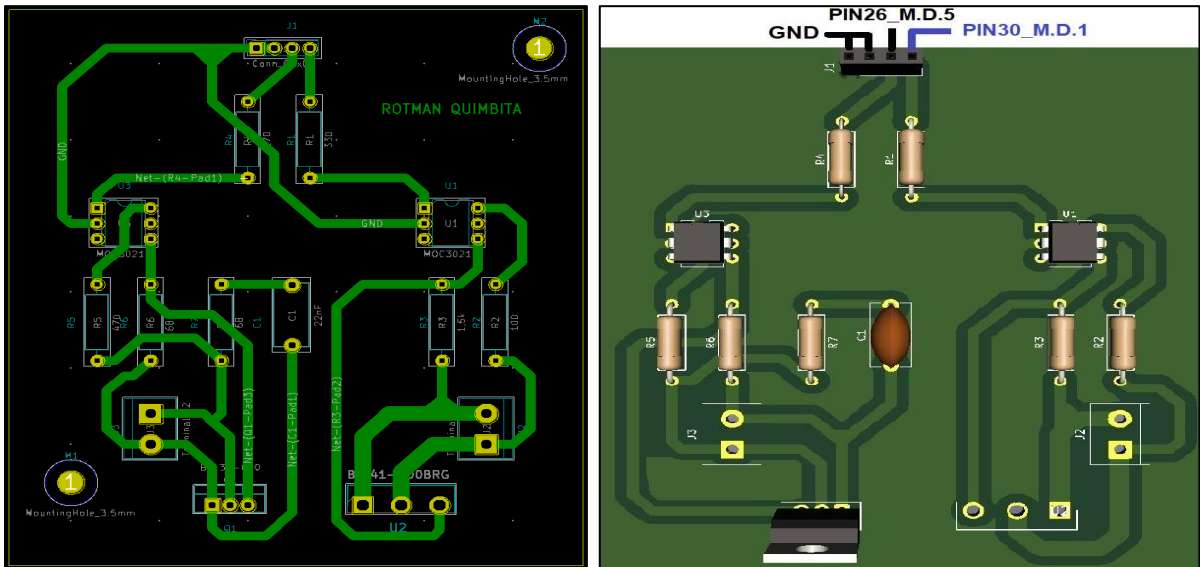


Fig. 0.24. Módulo de disparo 1 y 5 - diseño PCB y visualización 3D.

Fuente: (El Autor, 2020)

#### 2.1.7.4. Módulo sensores de temperatura

En la Figura 2.25 se muestra el esquema general:

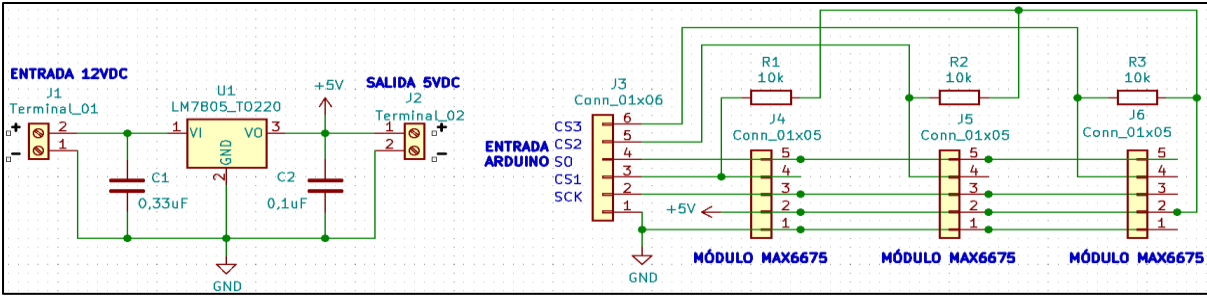


Fig. 0.25. Módulo sensores de temperatura - esquemático general.

Fuente: (El Autor, 2020)

En la Figura 2.26 se muestra el diseño de la PCB como su visualización en 3D.

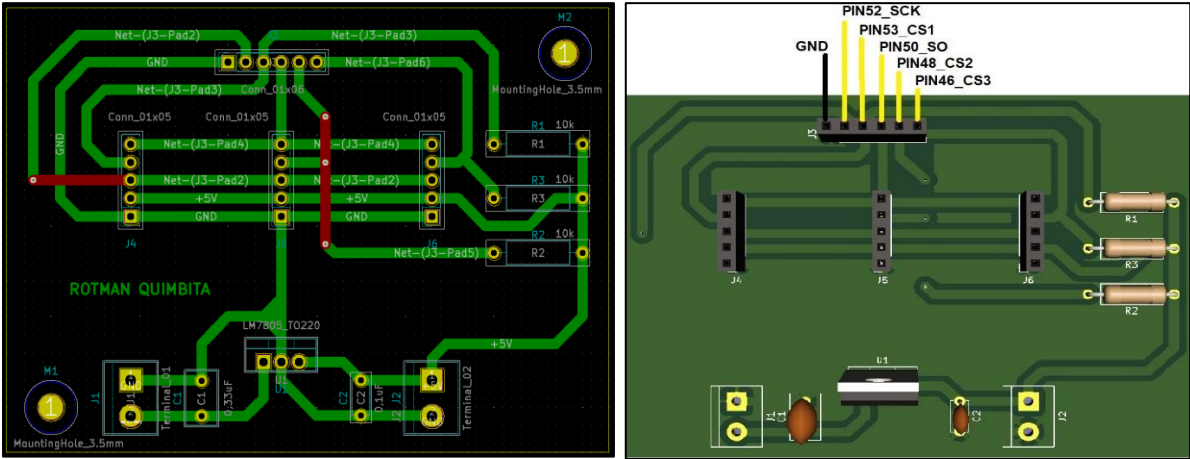


Fig. 0.26. Módulo sensores de temperatura - diseño PCB y visualización 3D.

Fuente: (El Autor, 2020)

**2.1.7.5. Módulo ventiladores**

En la Figura 2.27 se muestra el esquemático general:

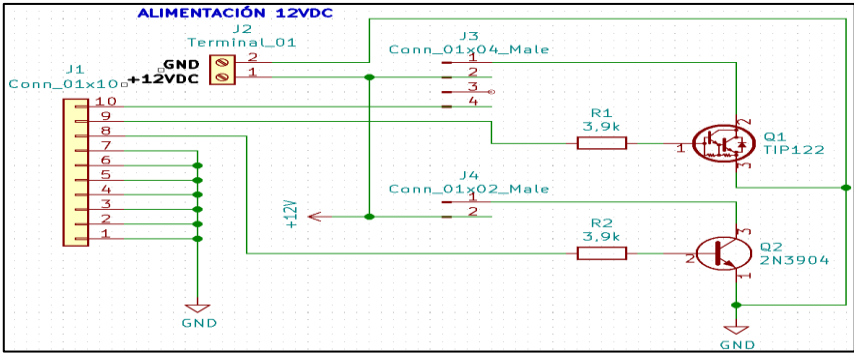


Fig. 0.27. Módulo para ventiladores - esquemático general

Fuente: (El Autor, 2020)

En la Figura 2.28 se muestra el diseño de la PCB como su visualización en 3D.

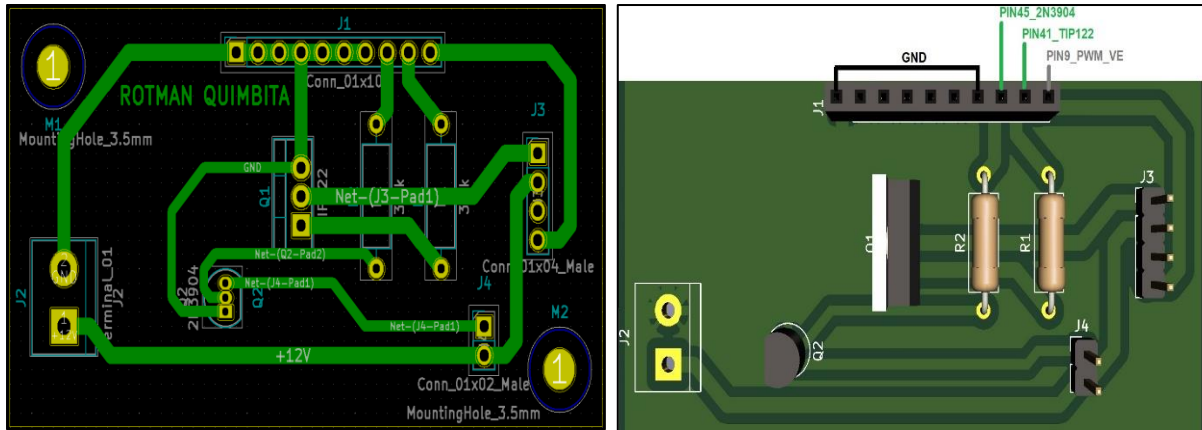


Fig. 0.28. Módulo para ventiladores - diseño PCB y visualización 3D

Fuente: (El Autor, 2020)

## 2.2. Diseño del control de temperatura

Para poder controlar el proceso térmico se realiza un control retroalimentado tipo PID (Proporcional-Integral-Derivativo). A continuación, se indica los pasos necesarios para el diseño del controlador.

### 2.2.1. Diseño del control PID

#### 2.2.1.1. Introducción

El proceso de diseño consiste en ajustar los parámetros del regulador PID para conseguir el cumplimiento de las especificaciones del proceso.

Es importante recordar que la “inteligencia” del controlador viene determinada por la complejidad de las operaciones matemáticas que es capaz de desempeñar, cuanto más elaborado sea el cálculo a partir del cual se generan las acciones de control, más eficiente será su aplicación sobre el proceso controlado.

La parte proporcional es el único elemento de control, la parte integral y derivativa deben ser interpretadas como acciones de control, es decir, no representan esquemas de control, más bien son elementos necesarios que ayudan a corregir o perfeccionar las características analíticas y cualitativas del control proporcional.

#### 2.2.1.2. Identificación de la dinámica del sistema

Para poder diseñar y posteriormente implementar un control PID digital es importante entender el funcionamiento del proceso que deseamos controlar, para ello se necesita la respuesta de la planta de temperatura al proporcionar una entrada escalón, este procedimiento se conoce como la identificación de sistemas por medio de la curva de reacción

y su salida se representa con una función de transferencia o también llamada dinámica del proceso.

Para obtener la función de transferencia en lazo abierto se coloca a la entrada del sistema una señal de tipo escalón que vaya del 25 al 30% del valor máximo, para este caso el valor del escalón está determinado por el registro comparador o ángulo de disparo del Triac, el cual se estableció en 465 como el 0% y 300 el 100%. Con el 25% de la variable de control (CV) se espera que la variable de proceso (PV) tienda a estabilizarse.

La adquisición de datos se la realiza a través de la comunicación serial Arduino-Matlab en el cual ambas variables se van guardando para posteriormente utilizar los datos para el análisis, la medición se lleva a cabo por aproximadamente treinta minutos con un periodo de muestreo de 0.1s ya que, por naturaleza de las lámparas halógenas del horno, el cambio de temperatura en el mismo es un poco lenta; la respuesta resultante de cada señal se muestra en la Figura 2.29.

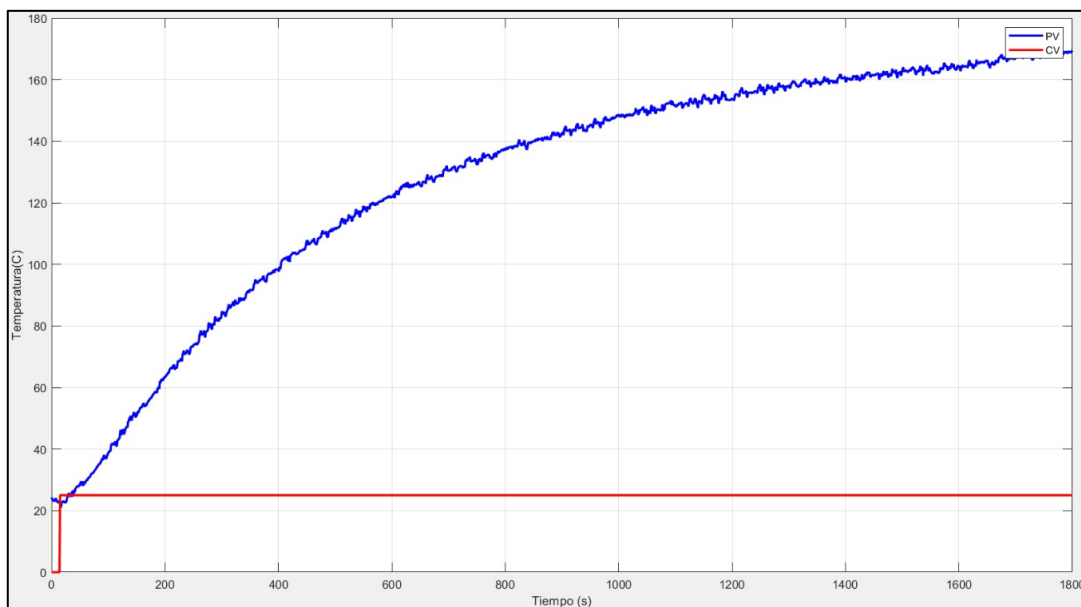


Fig. 0.29. Respuesta en lazo abierto.

Fuente: (El Autor, 2020)

Una vez finalizada la adquisición se tiene a disposición los datos en el Workspace de Matlab 2018, software a utilizar para determinar el modelo matemático que describe el proceso de temperatura; a continuación, se detalla el proceso realizado.

En el Comand Window se ejecuta la función "systemIdentification", desplegándose una pantalla mostrada en la Figura 2.30. Se importa los datos en el dominio del tiempo y se ingresa

los datos de entrada y salida con el mismo periodo de muestreo utilizado en la adquisición de datos.

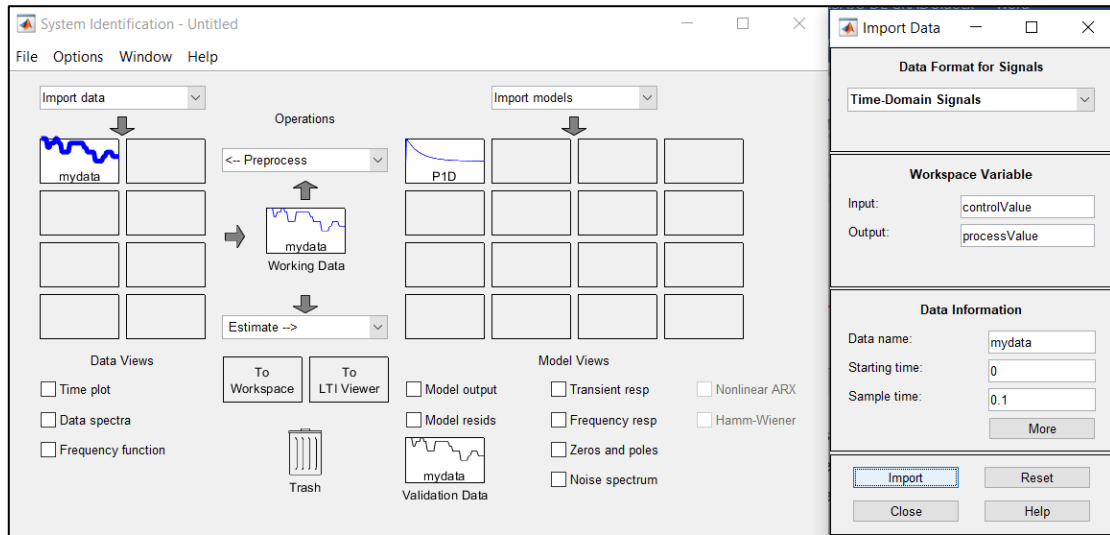


Fig. 0.30. Ventana de Identificación de sistemas.

Fuente: (El Autor, 2020)

Seguidamente, se selecciona Process Models sabiendo que se desea aproximar a un sistema de primer orden más tiempo muerto, y presionar el botón “Estimate” como se indica en la Figura 2.31.

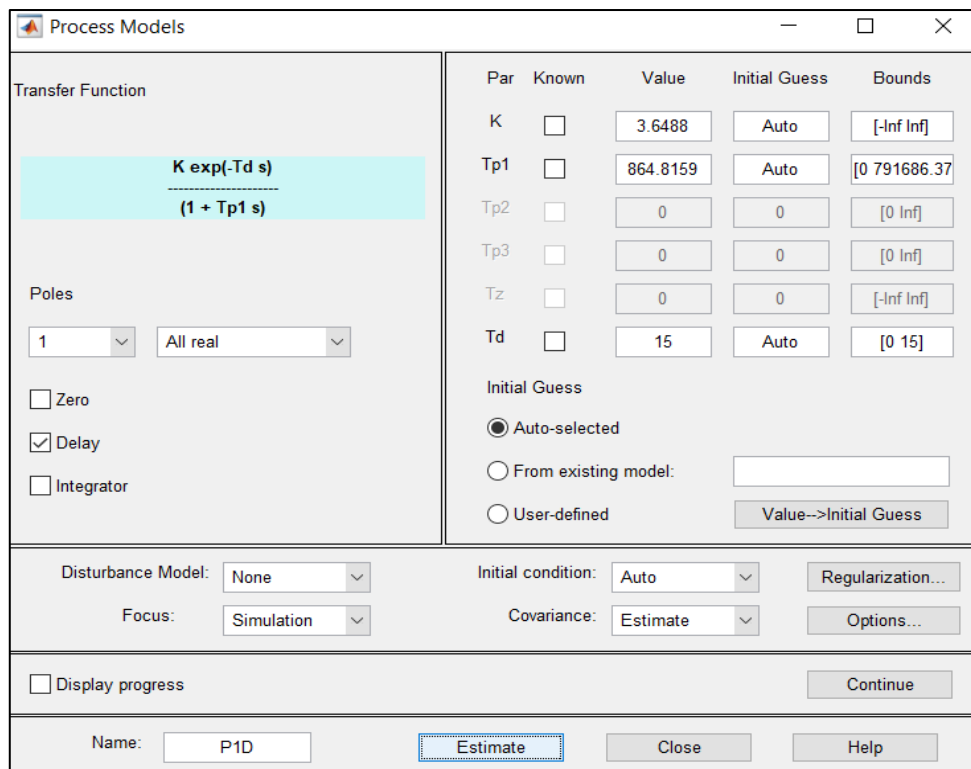


Fig. 0.31. Ventana modelo de proceso.

Fuente: (El Autor, 2020)

Finalmente, en la pantalla principal habilitar la opción de “Model output” para visualizar la respuesta del modelo matemático estimado y el porcentaje de aceptación. La Figura 2.32 muestra el modelo obtenido con un porcentaje del 96.53 de aceptación.

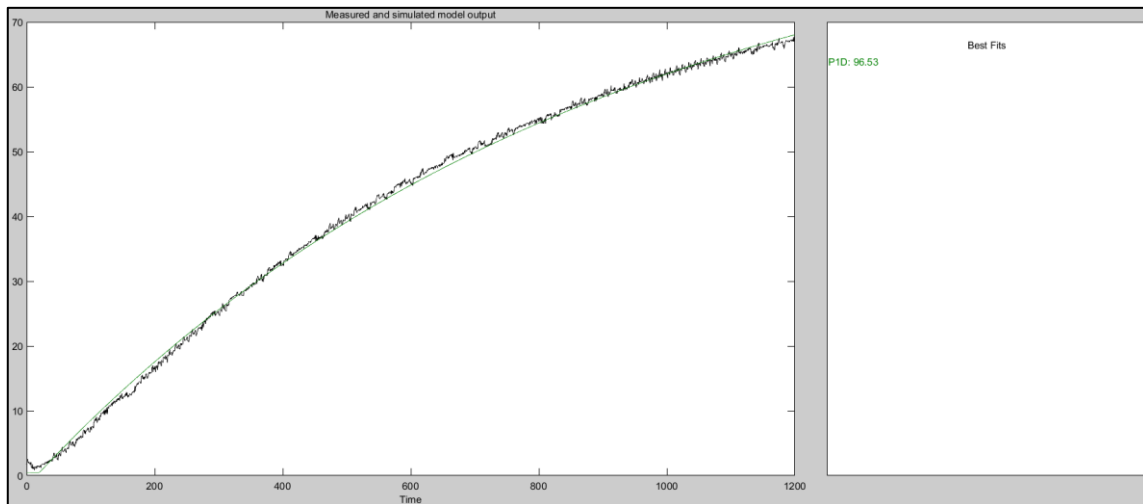


Fig. 0.32. Modelo matemático estimado.

Fuente: (El Autor, 2020)

El modelo matemático estimado que representa la dinámica de la planta de temperatura se muestra en la ecuación 8.

$$G(s) = \frac{3.6488}{1 + 864.8159 s} e^{-15s} \quad (8)$$

Donde  $K_p = 3.6488$ ,  $T = 864.8159$  y  $L = 15$  con estos valores se puede encontrar las constantes de sintonización del algoritmo de PID con los diferentes métodos existentes.

### 2.2.1.3. Sintonía del controlador PID

Para éste aplicativo se decide por el método de Ziegler and Nichols ya que presenta un buen comportamiento realizando un ajuste fino. En la Tabla 2.5 se indica la regla de sintonía de Ziegler and Nichols para diferentes tipos de controladores ya sea P, PI, o PID.

TABLA 0.6  
REGLA DE SINTONÍA DE ZIEGLER AND NICHOLS

TIPO DE CONTROLADOR	$K_p$	$T_i(s)$	$T_d(s)$
P	$\frac{T}{L}$	$\infty$	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{T}{L}$	$2L$	$0.5L$

Fuente: (El Autor, 2020)



Como se va implementar un controlador PID las constantes quedan definidas por las ecuaciones:

$$\text{Ganancia Proporcional} \rightarrow K_P = 1.2 \left( \frac{T}{L} \right) = 1.2 \left( \frac{864.8159}{15} \right) = 69.1853$$

$$\text{Constante de tiempo integral} \rightarrow T_i = 2L = 2(15) = 30$$

$$\text{Constante de tiempo derivativa} \rightarrow T_d = 0.5L = 0.5(15) = 7.5$$

Estas constantes serán ingresadas al controlador PID implementado en Arduino ( $K_p$ ,  $T_i$ ,  $T_d$ ).

En Figura 2.33 se puede observar la respuesta simulada en Simulink del sistema con control PID al aplicar el método de sintonización de Ziegler and Nichols.

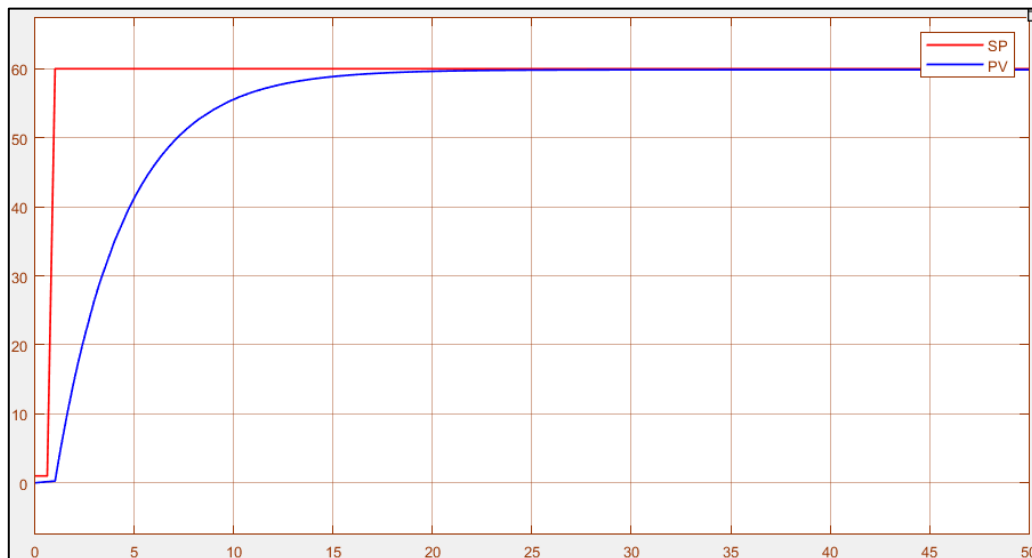


Fig. 0.33. Respuesta del sistema con el método de Ziegler and Nichols.

Fuente: (El Autor, 2020)

### 2.3. Diseño de la Interfaz Gráfica de Usuario (GUI)

Para realizar la interfaz gráfica de usuario destinada a la visualización y control del proceso se la realiza en Python mediante el módulo Tkinter, se procede a crear una figura mediante el módulo de Matplotlib para graficar el valor obtenido del sensor e insertarle a la ventana principal a través de un canvas; creación de un frame<sup>4</sup> para ubicar los labels de medición de temperatura y un label de conteo del tiempo en segundos; un frame para ubicar los botones de inicio y apagado así como un menú desplegable para la selección del puerto conectado al

<sup>4</sup> Un frame es un widget, que sirve de contenedor para los demás widgets dentro de la ventana principal.

Arduino y un botón para establecer la comunicación. La forma final de la interfaz gráfica de usuario se puede apreciar en la Figura 2.34.

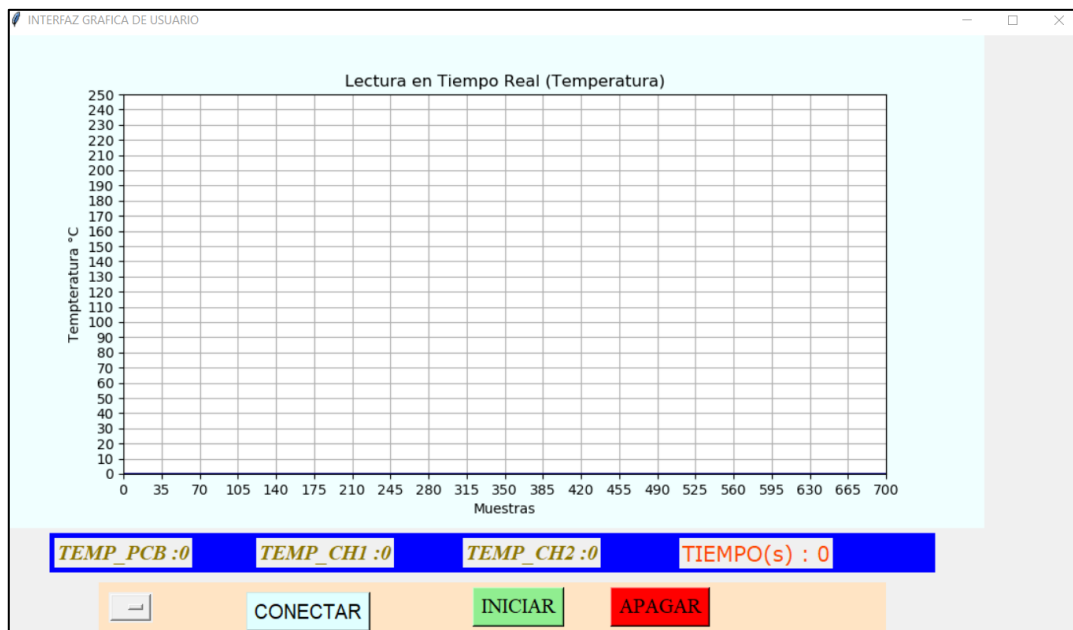


Fig. 0.34. Interfaz gráfica de usuario HMI o GUI.

Fuente: (El Autor, 2020)

Para evitar cerrar la ventana principal por error se programa una sub-ventana que pregunta al usuario si desea salir del programa lo que implicaría finalizar el proceso de soldadura de manera abrupta. La forma de la ventana se indica en la Figura 2.35.

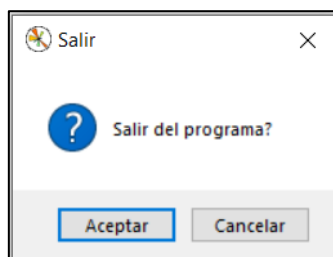


Fig. 0.35. Ventana para verificar salida de la interfaz gráfica de usuario.

Fuente: (El Autor, 2020)

## 2.4. Pruebas de funcionamiento

Para comprobar el correcto funcionamiento del módulo de control como de la comunicación a través de la Raspberry Pi se realiza una prueba. En la Figura 2.36 se muestra los elementos utilizados, así como su conexión.

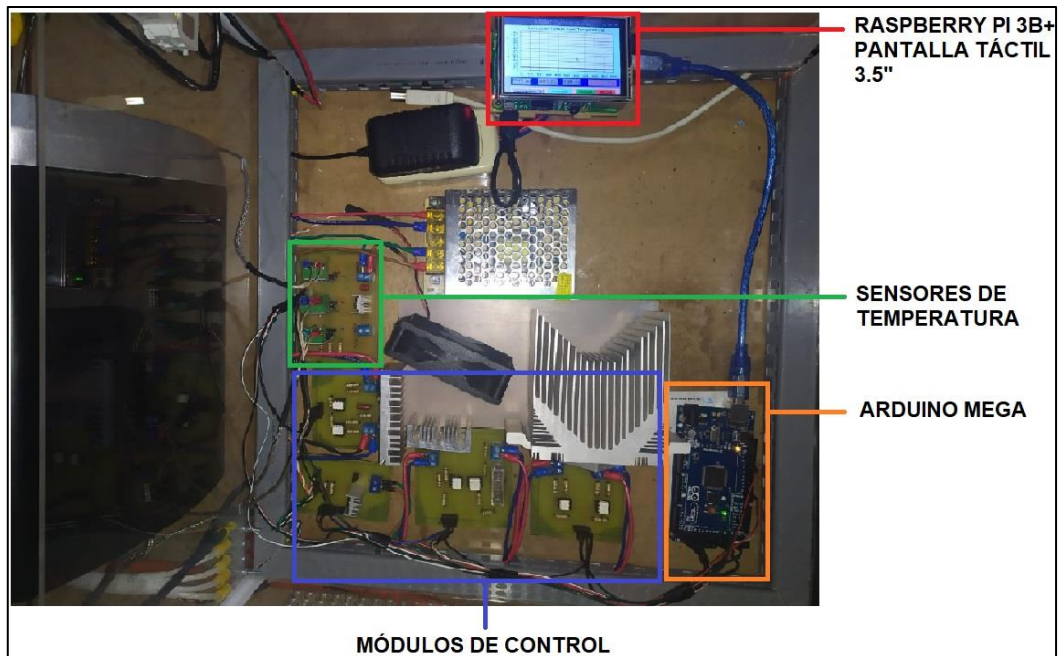


Fig. 0.36. Conexión para la prueba de funcionamiento.

Fuente: (El Autor, 2020)

En la Figura 2.37 se puede observar el funcionamiento de la interfaz gráfica, como la adquisición de datos a través del Arduino y el envío de información a la Raspberry PI por medio del puerto USB, cumpliendo la función de controlador maestro-esclavo.

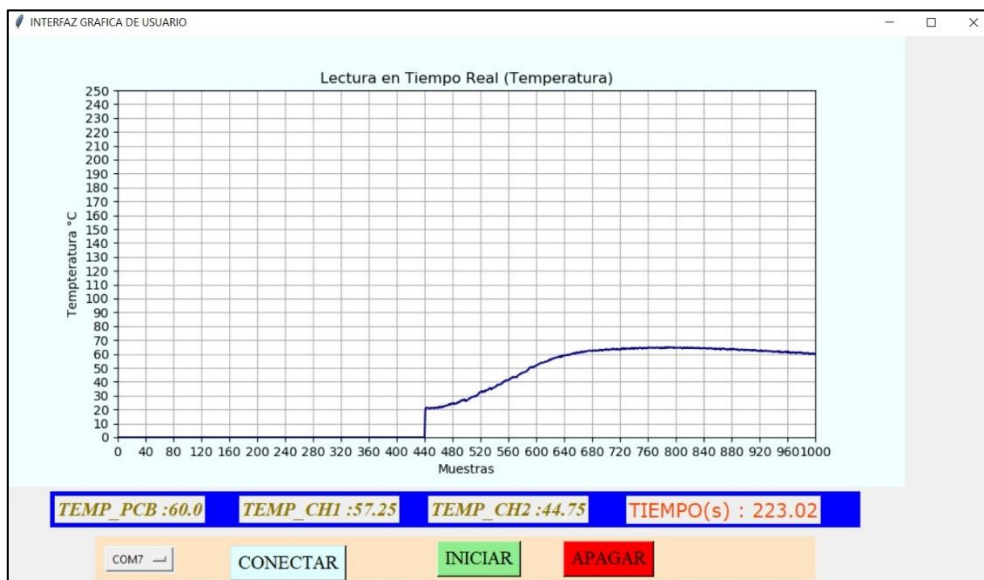


Fig. 0.37. Funcionamiento de la interfaz gráfica de usuario.

Fuente: (El Autor, 2020)

La ventana de interfaz gráfica permite iniciar y apagar el proceso mediante botones virtuales, así como la lectura de los sensores de temperatura en diferentes puntos del horno a través de etiquetas “labels” virtuales, los cuales se explica a continuación:

- TEMP\_PCB. Es la indicación de la temperatura en la placa de circuito impreso a soldar.
- TEMP\_CH1. Es la indicación de la temperatura en la cámara del horno derecha.
- TEMP\_CH2. Es la indicación de la temperatura en la cámara del horno izquierda.

Además, cuenta con una etiqueta extra para mostrar al usuario el tiempo en segundos desde que inicia el proceso de soldadura y una gráfica en tiempo real de la temperatura de la PCB que es la de mayor interés (TEMP\_PCB).

# CAPÍTULO 3

## Implementación del Equipo de Soldadura

### 3.1. Modificación del Horno

Para este proyecto se utilizó un Horno Hometech Modelo TO-162H-1 como el que se muestra en la Figura 3.1 cuyas dimensiones al interior del horno son de 30cm de ancho x 20cm alto x 23cm de profundidad, en el cual es posible soldar una placa de circuito impreso de hasta 20x25 cm.



Fig. 0.1. Horno Hometech TO-162H-1

Fuente: (El Autor, 2020)

Debido a que únicamente se va a utilizar la estructura metálica, se procede a retirar todos aquellos elementos incorporados que no aportan a las características del proceso. La modificación del horno se lleva a cabo mediante 2 etapas las cuales son:

#### 3.1.1. Extracción de elementos innecesarios

Se parte por la extracción de las resistencias eléctricas debido a que almacenan mucho calor aún después del corte del suministro eléctrico, así como la desconexión de perillas selectoras del termostato y temporizador del horno.



Fig. 0.2. Extracción de elementos y modificación del Horno.

Fuente: (El Autor, 2020)

### 3.1.2. Incorporación de elementos necesarios

En este apartado se detalla la instalación de elementos calefactores, enfriamiento y protección térmica.

#### 3.1.2.1. Elementos calefactores

Debido a la poca potencia e inercia térmica que presentan los resistores eléctricos incluidos en el horno se decide reemplazarlos por lámparas halógenas de cuarzo lineal de 1500 W (Figura 3.3), se instala 4 lámparas en la parte superior con una conexión en paralelo.



Fig. 0.3. Elementos calefactores: a) Resistencias. b) Lámparas halógenas de cuarzo.

Fuente: (El Autor, 2020)

Como precaución no se debe tener contacto directo con las lámparas que no tengan un protector o un reflector, ya que las huellas dactilares en la ampolla de cristal de cuarzo, facilitan la creación de verrugas en esa zona al tener contenido grasoso, que llegan a disminuir la vida útil o incluso destruir el cuarzo.

#### 3.1.2.2. Incorporación de ventiladores: convección y enfriamiento

El horno tiene una característica que es la transmisión de calor por medio de convección, como se indicó en el análisis bibliográfico, esto es posible por medio de un ventilador con aspas metálicas ya que va ir incorporado en el interior de la cámara del horno; el motor universal es ubicado en la parte posterior externa del horno.

En la Figura 3.4 se muestra la instalación del ventilador utilizado para la convección.



Fig. 0.4. Instalación de ventilador de convección.

Fuente: (El Autor, 2020)

Para el enfriamiento del proceso se instala un ventilador Brushless en la parte derecha de la tapa externa del horno y mediante un conducto con una compuerta permite el paso del aire frío al interior de la cámara del horno.

En la Figura 3.5 se muestra la instalación del ventilador utilizado para la etapa de enfriamiento.

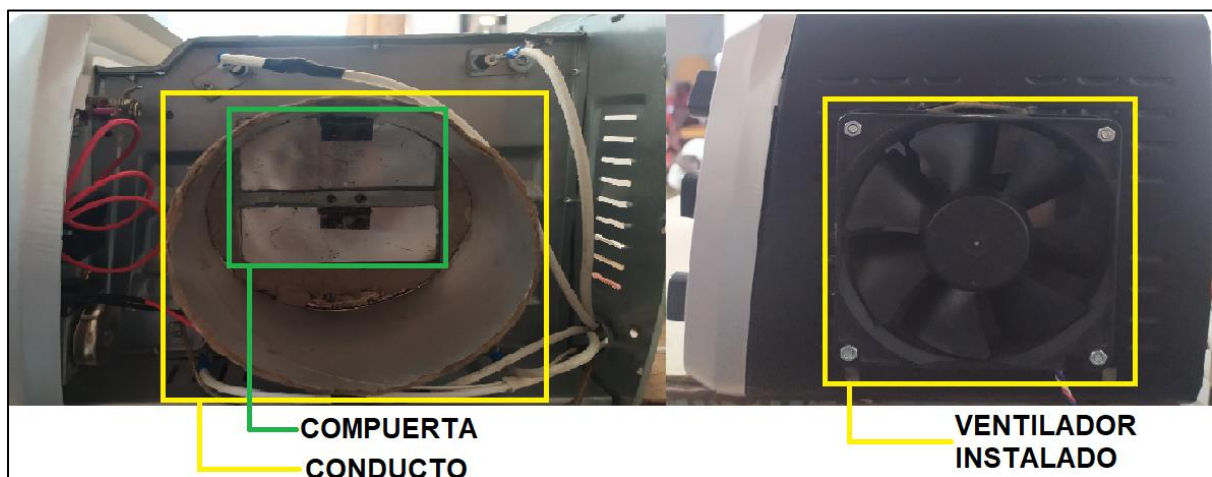


Fig. 0.5. Instalación de ventilador de enfriamiento.

Fuente: (El Autor, 2020)

### 3.1.2.3. Incorporación de sensores de temperatura

Para un mejor seguimiento del proceso térmico se instala 3 sensores de temperatura ubicados en 3 puntos diferentes de la cámara del horno, 2 de ellos se encuentran en las paredes internas derecha e izquierda y el tercer sensor irá ubicado encima de la placa de circuito impreso a soldar.

En la Figura 3.6 se muestra la instalación de los sensores de temperatura.

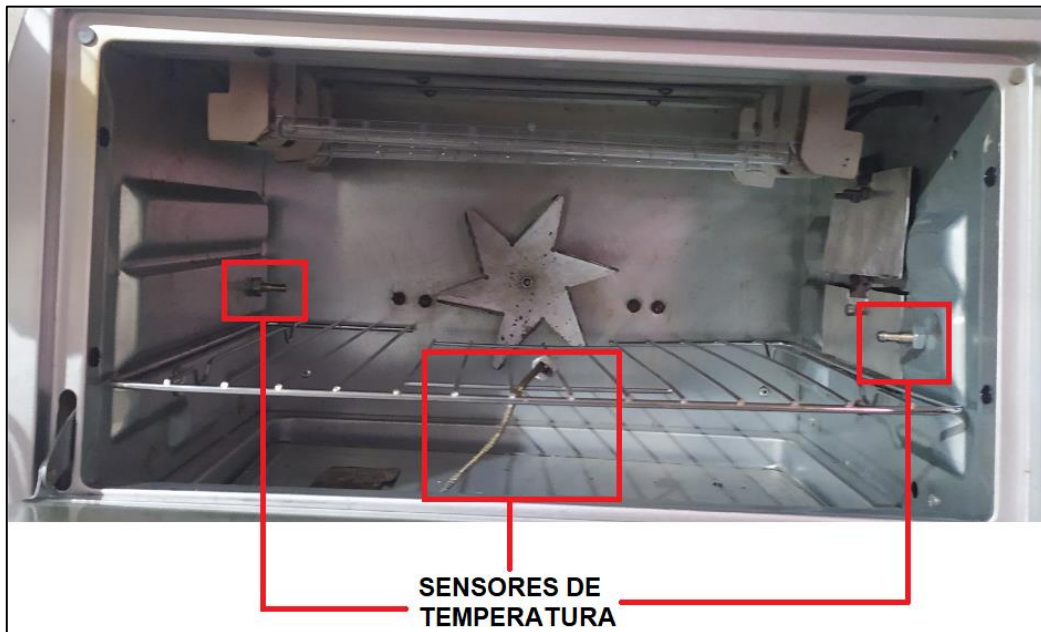


Fig. 0.6. Instalación de sensores de temperatura.

Fuente: (El Autor, 2020)

#### 3.1.2.4. Aislamiento térmico

El aislamiento térmico se define como un material o combinación de materiales de baja conductividad térmica que tienen la función de retardar la transferencia de calor al actuar como una barrera en la trayectoria del flujo de calor, dichos materiales son versátiles ya que pueden ser adaptados a cualquier tamaño, forma y superficie (Deshmukh, Birwal, Datir, & Patel, 2017). Por lo tanto, el aislamiento es el resultado de realizar el proceso para aislar térmicamente el sistema utilizando materiales aislantes para disminuir considerablemente la tasa de transferencia de calor entre el sistema y el cuerpo adyacente o el medio ambiente.

La Figura 3.7 ilustra de mejor manera esta situación.

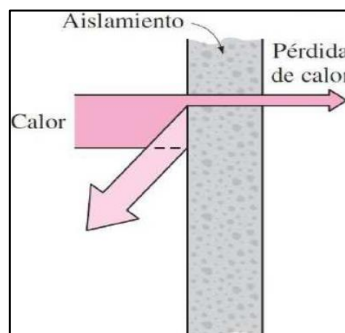


Fig. 0.7. Barrera que retarda la transferencia de calor.

Fuente: (CONUEE (Comisión Nacional Para El Uso Eficiente de la Energía), 2009)



Las propiedades térmicas de los aislantes son diversas y deben ser analizadas dependiendo de su uso, en esta ocasión se analizan materiales térmicos enfocados al uso en hornos industriales detallando la densidad, temperatura que son capaces de soportar y su coeficiente de conductividad térmica (k). En la Tabla 3.1 se describen las propiedades térmicas de los materiales aislantes más comerciales.

TABLA 0.1  
PROPIEDADES DE MATERIALES TERMOAISLANTES

TIPO DE MATERIAL	DENSIDAD ( $kg/m^3$ )	COEFICIENTE DE CONDUCTIVIDAD TÉRMICA ( $W/m \cdot ^\circ C$ )	TEMPERATURA MÁXIMA ( $^\circ C$ )
Lámina mineral de vidrio	10 a 80	0.031 a 0.042	450
Placa de silicato de calcio	240	0.054	1000
Manta de fibra cerámica	64 a 192	0.030 a 0.079	1400
Perlita expandida	80	0.057	1000
Lana mineral de roca	80	0.033	900
Espuma de poliuretano	30	0.026	175

Fuente: (Palomo, 2017)

Los materiales aislantes detallados son usados en hornos y aplicaciones refractarias, es decir son materiales termoaislantes. En la Figura 3.8 se puede apreciar tales materiales.

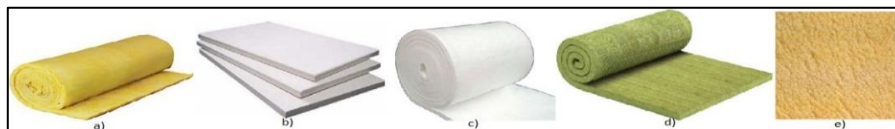


Fig. 0.8. Materiales termoaislantes: a) Fibra de vidrio. b) Silicato de calcio. c) Fibra cerámica. d) Lana de roca. e) Poliuretano.

Fuente: (Palomo, 2017)

Para poder aislar térmicamente el Horno de Reflujo se decide utilizar lana de vidrio, logrando así la protección contra el contacto accidental hacia el horno y evitar las pérdidas de calor por posibles fugas o grietas en la estructura.

En la Figura 3.9 se puede observar cómo se recubre por completo el Horno con lana de vidrio tomando las medidas de precaución de no manipular directamente con las manos descubiertas.



Fig. 0.9. Aislamiento térmico del Horno.

Fuente: (El Autor, 2020)

Una vez realizado todas las modificaciones al horno, la planta térmica queda lista para poder ser controlada a través de un módulo de control tal como lo muestra la Figura 3.10.



Fig. 0.10. Planta térmica terminada.

Fuente: (El Autor, 2020)

### 3.2. Implementación del módulo de control

Para poder implementar el módulo de control se requiere primero construir las placas de control para luego proceder con el conexionado de todos los elementos. A continuación, se detalla cada una.

#### 3.2.1. Construcción de placas de control

Una vez diseñadas las placas de circuito impreso, se procede a realizar la fabricación de las mismas en baquelitas de fibra de vidrio. El procedimiento se lleva a cabo por el método de planchado y uso de ácido.

Después de haber colocado cada uno de los componentes electrónicos en cada placa se realiza el soldado. La Figura 3.11 muestra las placas de control terminadas en la que se tiene: circuito detector de cruce por cero, módulos de disparo con Triacs, módulo para control de ventiladores y módulo de sensores.

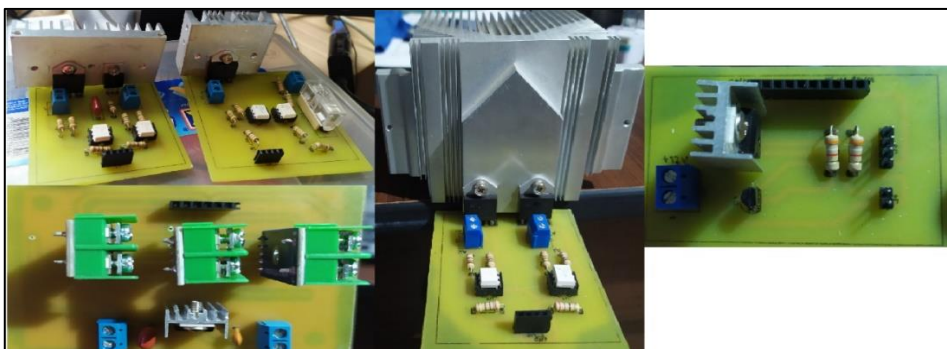


Fig. 0.11. Placas de control.

Fuente: (El Autor, 2020)

### 3.2.2. Instalación del sistema eléctrico y circuitos de control

Para proceder con el conexionado eléctrico, circuitos electrónicos, fuentes de alimentación, elementos de protección y en si la agrupación de todos los elementos se decide construir una caja de acrílico de 3mm de dimensiones (40x24x42) cm (Figura 3.12). En la caja se incluirá todos los elementos de control, accionamiento y protección además de una canaleta para cubrir el cableado eléctrico.



Fig. 0.12. Caja de acrílico para alojar toda la circuitería.

Fuente: (El Autor, 2020)

En la Figura 3.13 se puede apreciar la caja con todos los elementos situados en su lugar.

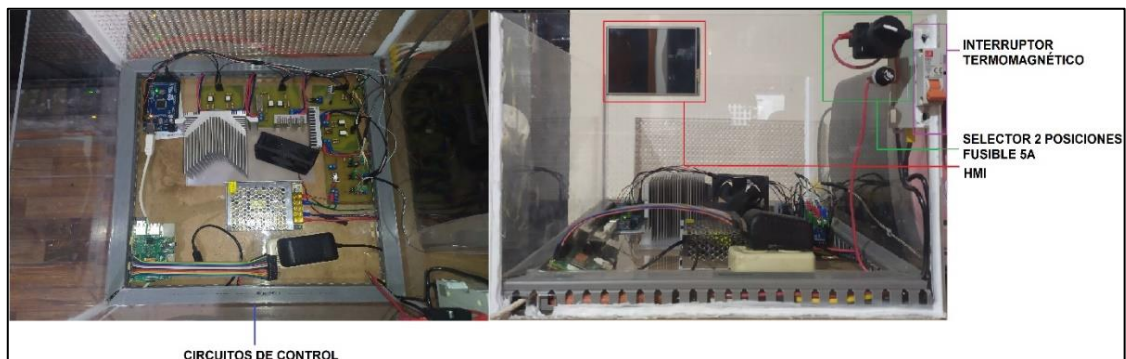


Fig. 0.13. Instalación del sistema eléctrico y circuitos de control.

Fuente: (El Autor, 2020)

### 3.3. Implementación final del equipo

Una vez realizada la modificación del horno y la construcción del módulo de control, el equipo finalizado se muestra en la Figura 3.14.

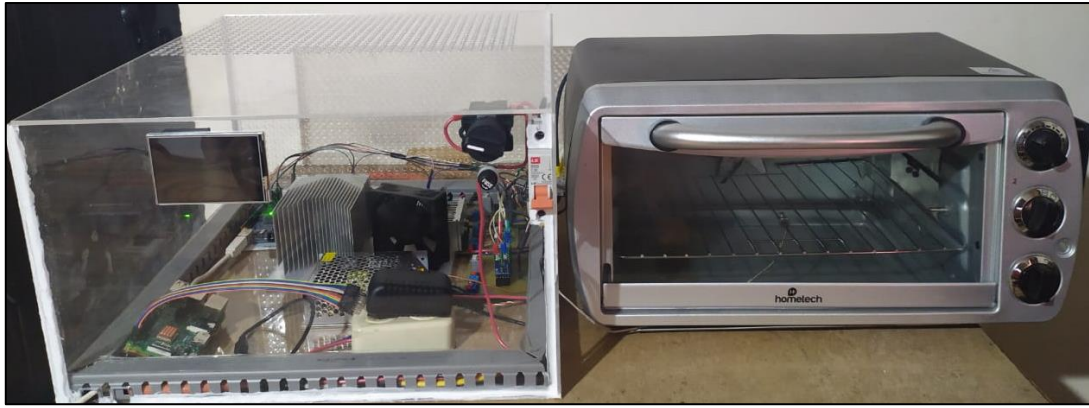


Fig. 0.14. Equipo de soldadura SMD finalizado.

Fuente: (El Autor, 2020)

El módulo de control (circuitería) como la planta térmica (Horno de Reflujo) son implementados siguiendo los requerimientos de diseño.

### 3.4. Implementación del Control PID digital

En la Figura 3.15 se muestra el diagrama de bloques del sistema de temperatura con un control PID digital. Un valor de error es calculado cuando se realiza la diferencia entre el punto de consigna o setpoint y el valor real medido por el sensor, en base a este error el microcontrolador calcula la variable de control  $u(k)$ , que va hacia el actuador para controlar la planta.

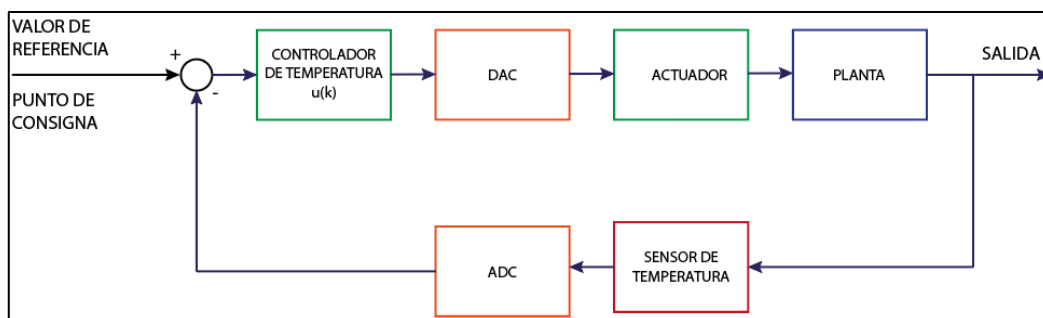


Fig. 0.15. Diagrama de bloques del control de temperatura digital.

Fuente: (El Autor, 2020)

Para llevar a cabo la implementación del control PID digital se utiliza el lenguaje de programación propio de Arduino a través de la plataforma Arduino IDE. La expresión 6 describe la salida de un control PID en su versión discreta, por lo que será la expresión que se implementará.

La Figura 3.16 muestra la parte más importante de la programación en la cual es implementado el control PID digital.

```
error = setpoint - processValue[0];
error_pasado += error*(tiempoMuestreo/1000);
int error_D = (error - error_anterior)/(tiempoMuestreo/1000);
if(error > 30) ITerm = 0;

PTerm = Kp * error;    //PTerm=kp*error
ITerm = (Kp/Ti) * (error_pasado);
DTerm = Kp*Td*error_D;

controlValue = PTerm + ITerm + DTerm;
```

Fig. 0.16. Implementación del control PID digital.

Fuente: (El Autor, 2020)

Se parte por la creación de variables que componen la ecuación 6 del control PID digital de la cual se explica: término proporcional (PTerm), término integral (ITerm), término derivativo (DTerm), estos tres términos son sumados y enviados a la variable de control (controlValue).

Después de haber implementado el algoritmo de control PID es necesario ingresar las constantes de sintonía en el microcontrolador de acuerdo a las constantes encontradas en la etapa de diseño del controlador (Figura 3.17).

```
76   Kp = 69.1853*6;
77   Ti = 30;
78   Td = 7.5;
```

Fig. 0.17. Implementación de las constantes de sintonía en Arduino.

Fuente: (El Autor, 2020)

### 3.5. Pruebas y ajustes necesarios del control PID de temperatura

Como se mencionó en el capítulo anterior se va utilizar el método de Ziegler and Nichols para sintonizar el controlador, a continuación, se muestran los resultados. Cabe mencionar que las pruebas se realizan sin ninguna PCB para evitar estropear en la etapa de pruebas.

#### 3.5.1. PID sintonizado con el criterio de Ziegler and Nichols (sin sintonía fina)

En la Figura 3.18 se puede observar el comportamiento del sistema con el control PID sintonizado con el criterio de Ziegler and Nichols, donde se representa el setpoint (SP-rojo), variable de proceso (PV-amarillo) y variable de control (CV-negro).

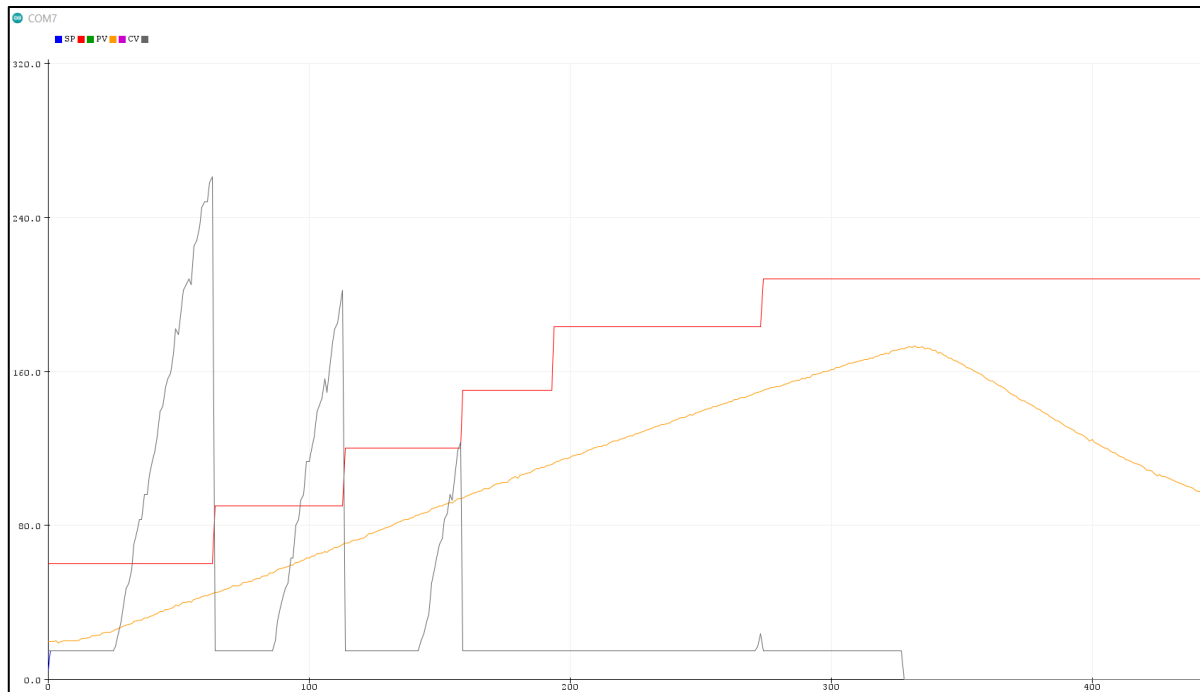


Fig. 0.18. Respuesta del control PID sin sintonía fina.

Fuente: (El Autor, 2020)

Se puede notar que el sistema responde de manera lenta, no logra llegar al valor de referencia en el tiempo establecido por lo que se deberá modificar las ganancias del controlador mediante un ajuste fino.

### 3.5.2. PID sintonizado con el criterio de Ziegler and Nichols (con sintonía fina)

Se debe tomar en cuenta que los requerimientos del proceso térmico no son fijados por un único setpoint estabilizado en el tiempo, por el contrario, se tiene diferentes valores de referencia o setpoint variables en instantes cortos de tiempo con la finalidad de seguir la curva de temperatura recomendada por el fabricante del tipo de suelda.

Por esta razón, se necesita que el sistema reaccione más rápido, por lo que se aumenta la ganancia proporcional (ajuste fino) sin que ello afecte la estabilidad del sistema, tal como lo muestra la respuesta simulada en Simulink representada en la Figura 3.19.

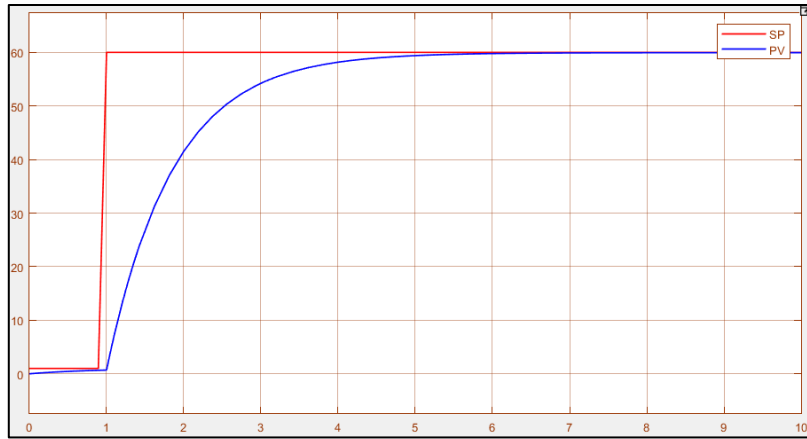


Fig. 0.19. Respuesta del sistema con un aumento de ganancia proporcional.

Fuente: (El Autor, 2020)

En la Figura 3.20 se puede observar el comportamiento del sistema con el control PID sintonizado con el criterio de Ziegler and Nichols con un ajuste fino, donde se representa el setpoint (SP-rojo), variable de proceso (PV-morado) y variable de control (CV-negro).

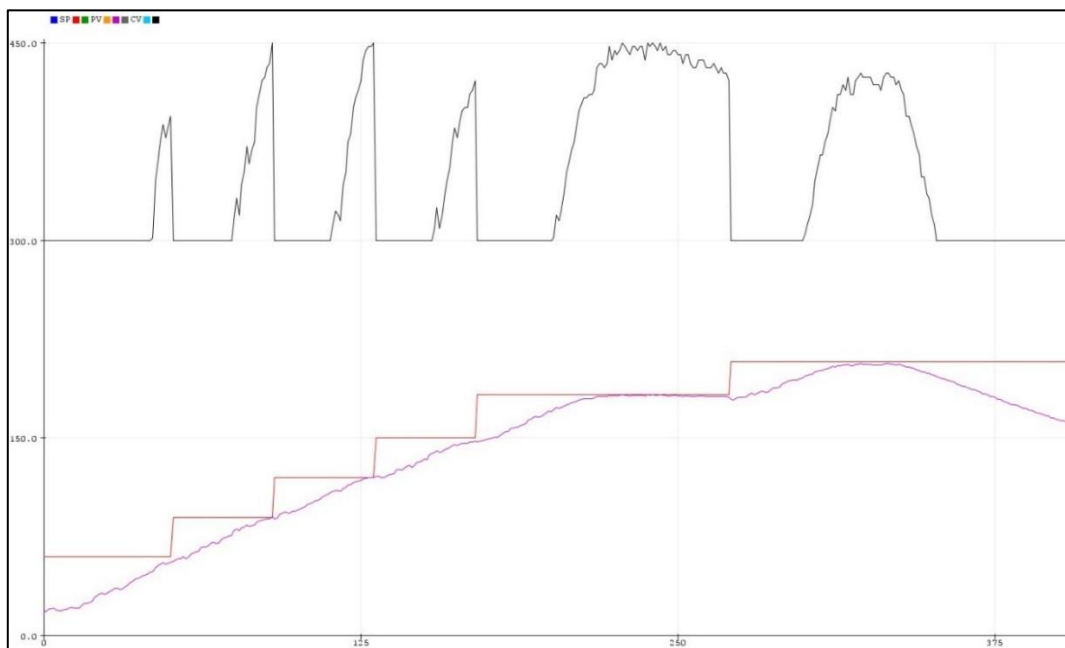


Fig. 0.20. Respuesta del control PID con sintonía fina.

Fuente: (El Autor, 2020)

En esta prueba se puede observar claramente que la variable de proceso alcanza a los valores de referencia en los tiempos programados, sin presentar sobreimpulsos lo que se concluye con un sistema de control estable.

### 3.5.3. Filtrado de ruido en sensores aplicado al control PID

Hay ocasiones en la que la señal de los sensores pudiera presentar interferencias o ruido, lo que conllevaría a problemas de sintonización y desestabilización del algoritmo de control en lazo cerrado, por lo que se ve en la necesidad de implementar un filtro para mejorar la medición de los sensores de temperatura. El filtro a utilizar es conocido como EMA por sus siglas en inglés (Exponential Moving Average) o media móvil exponencial y para su implementación se parte por su fórmula matemática (Llamas, 2017b).

$$S(k) \begin{cases} Y(0) & k = 0 \\ \alpha Y(k) + (1-\alpha)S(k-1) & k > 0 \end{cases}$$

Donde:

$\alpha$ : Coeficiente de suavizado comprendido entre 0 y 1.

$Y(k)$ : Valor de la señal a filtrar en un período de tiempo  $k$  (actual).

$S(k)$ : Valor de la EMA en cualquier período de tiempo  $k$ .

$S(k-1)$ : Valor de la EMA en el período de tiempo  $k-1$  (anterior).

En la Figura 3.21 se puede observar la medición de temperatura sin filtro (color azul) y la señal filtrada (color rojo) en la cual se puede apreciar lo útil que resulta la implementación de este tipo de filtro y lo sencillo de hacerlo mediante una sola línea de código en Arduino.

```
filtroSensor = alpha*processValue[0]+(1.0-alpha)*filtroSensor;
```

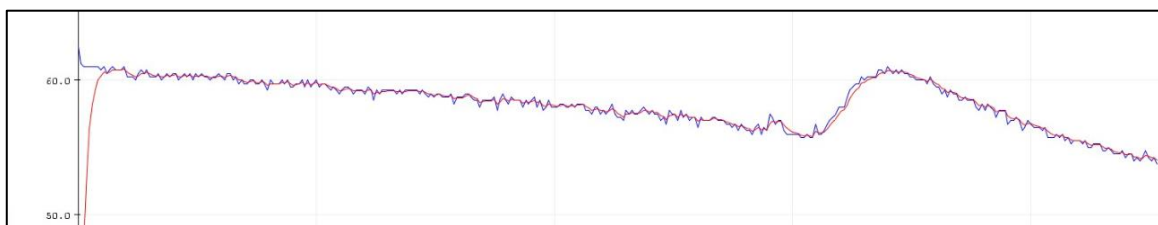


Fig. 0.21. Señal filtrada del sensor de temperatura.

Fuente: (El Autor, 2020)

### 3.6. Implementación de la Interfaz Gráfica de Usuario

Para poder implementar un HMI a través de una pantalla táctil es necesario primero configurar el dispositivo el cual permitirá el funcionamiento de la pantalla y dotar de funcionalidad a la interfaz gráfica a través de software.



### 3.6.1. Configuración de la Raspberry Pi

Para poder usar la pantalla táctil es necesario configurar primero el miniordenador Raspberry pi. A continuación, se enlista los procedimientos necesarios para empezar a utilizar el miniordenador.

- Descargar el sistema operativo Raspbian desde la página oficial.
- En una memoria microSD de mínimo 8GB clase 10 se monta la imagen iso descargada de la página oficial.
- En la raíz de la memoria microSD crear un archivo sin formato con el nombre “ssh” para realizar la comunicación con el miniordenador desde un pc (debido a que la Raspberry Pi no cuenta con teclado ni pantalla de visualización).
- Para conectar remotamente desde el computador es necesario conectar el cable de red al miniordenador y buscar la dirección IP asociada a la red conectada.

Una vez realizado los pasos anteriormente descritos ya se puede utilizar el protocolo SSH para conectar de forma remota y segura al Raspberry Pi a través del software Bitwise como lo muestra la Figura 3.22. De esta forma se puede programar y configurar la Raspberry Pi (con sistema operativo basado en Linux) a través de un computador con Windows.

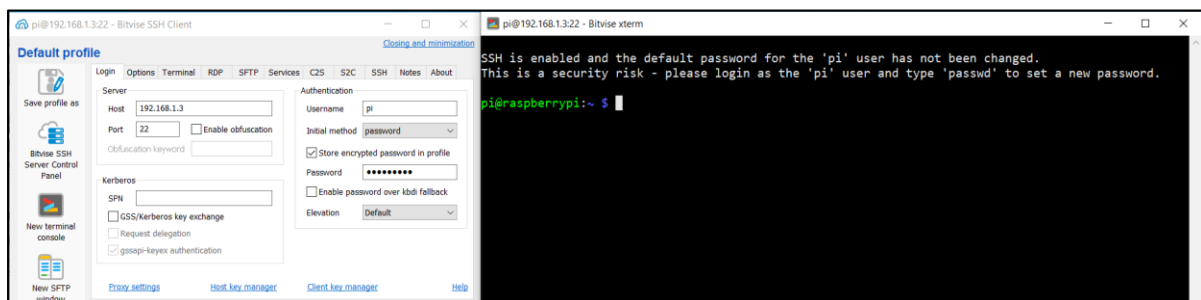


Fig. 0.22. Conexión remota de la Raspberry Pi.

Fuente: (El Autor, 2020)

Una vez habilitada la línea de comandos de la Raspberry Pi se procede a instalar todo lo necesario para poder crear una aplicación de interfaz gráfica. El primer paso es configurar la pantalla táctil para posteriormente instalar el software con el que se programará la GUI.

El fabricante para poder configurar la pantalla recomienda abrir la terminal del Raspberry Pi y ejecutar los siguientes comandos:

```
sudo rm -rf LCD-show
```

```
sudo apt install git
```

```
git clone https://github.com/goodtft/LCD-show.git
```

```
chmod -R 755 LCD-show
cd LCD-show/
sudo ./LCD35-show
```

Si no se encuentra algún error, el miniordenador se reiniciará y quedará habilitada la pantalla táctil como se puede observar en la Figura 3.23.



Fig. 0.23. Pantalla táctil configurada.

Fuente: (El Autor, 2020)

A partir de aquí únicamente queda instalar el software con las librerías a utilizar para programar la interfaz gráfica. Se enlista los comandos principales:

- sudo apt-get update
- sudo apt-get upgrade
- sudo apt-get install matchbox-keyboard
- sudo apt-get install idle3
- sudo apt-get install python3-matplotlib
- sudo apt-get install python3-numpy
- sudo pip3 install pyserial

### 3.6.2. Adecuación de la Interfaz Gráfica

En el capítulo anterior se había diseñado la interfaz gráfica en Python, creación de botones, etiquetas, ubicación de elementos, gráficas, etc., pero necesita una programación extra para que pueda enviar y recibir información del Arduino para la toma de datos en tiempo real.

En la Figura 3.24 se detalla las instrucciones necesarias para que la interfaz gráfica pueda tener funcionalidad para lo requerido en el proyecto.

```
if serialConnection.inWaiting() > 0:
    for i in range(numData):
        value[i] = float(serialConnection.readline().strip()) #leer sensor
        data[i].append(value[i]) #Guarda lectura en la última posición
```

```

def plotData(self, Samples, lines):
    end = time.time();
    #data.append(value) #Guarda lectura en la última posición / #Save reading
    lines.set_data(range(Samples), data[0]) # Dibujar nueva línea del dato en
    temperatura1.set("TEMP_PCB :"+str(value[0])) # Mostrar valor del sensor /
    temperatura2.set("TEMP_CH1 :"+str(value[1])) # Mostrar valor del sensor /
    temperatura3.set("TEMP_CH2 :"+str(value[2])) # Mostrar valor del sensor /
    tiempo1.set("TIEMPO(s) : "+str(round(totalTime,2)))

```

Fig. 0.24. Programación para la transmisión de datos Raspberry Pi - Arduino.

Fuente: (El Autor, 2020)

Con estas instrucciones almacena en variables los datos recibidos por los sensores conectado al Arduino y se los puede usar en el resto del programa para poder visualizarlos tanto de manera numérica como gráfica.

### 3.7. Prueba real de soldado

Luego de haber realizado todas las modificaciones del Horno y verificado el funcionamiento del controlador se realiza una prueba de soldadura.

Para realizar la prueba se utiliza una soldadura con Plomo cuya composición es Sn63Pb37, es decir cuenta con 63% de Estaño y 37% de Plomo cuyo perfil de soldadura que recomienda el fabricante es la que indica en la Figura 3.25.

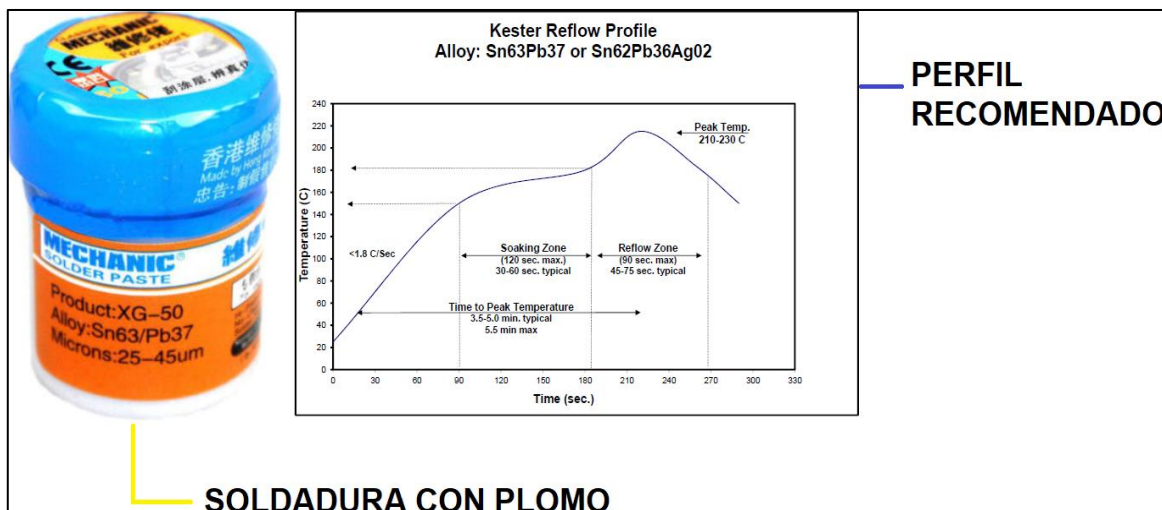


Fig. 0.25. Soldadura Sn63Pb37 y perfil de temperatura.

Fuente: (El Autor, 2020)

El proceso de soldadura se la realiza siguiendo los siguientes pasos:

- 1) Poseer una PCB en la cual se va soldar los componentes SMD.

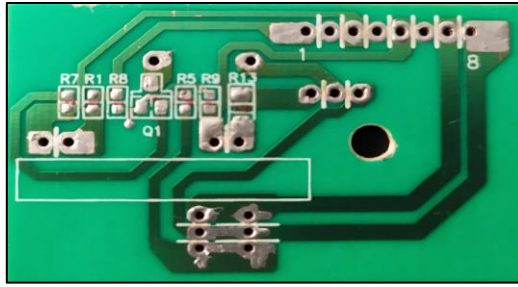


Fig. 0.26. PCB diseñada para componentes SMD.

Fuente: (El Autor, 2020)

- 2) Poner la soldadura en pasta (Sn63Pb37) cuidadosamente en cada uno de los pads<sup>5</sup> en donde se asentarán los componentes SMD.



Fig. 0.27. Colocación de soldadura en pasta (Sn63Pb37) en pads.

Fuente: (El Autor, 2020)

- 3) Ubicar cuidadosamente los componentes SMD sobre la soldadura en pasta que fue aplicada.

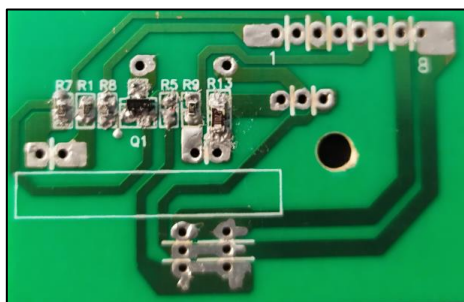


Fig. 0.28. Ubicación de componentes SMD en pads.

Fuente: (El Autor, 2020)

---

<sup>5</sup> La superficie de cobre de un circuito impreso en donde se suelda y fija el componente a la placa se conoce como pad.

4) Colocar la PCB dentro del Equipo de Reflujo e iniciar el proceso de soldadura.



Fig. 0.29. Colocación de PCB en equipo de soldadura.

Fuente: (El Autor, 2020)

La medición de temperatura se la puede visualizar mediante la interfaz gráfica mostrado en la Figura 3.30. En esta prueba se puede notar que el valor de los dos sensores es casi similar y la medición del tercer sensor presenta un valor más bajo debido a que se encuentra al lado de la compuerta que permite el ingreso de aire frío. Con esta consideración se comprueba la uniformidad de calor en la cámara del horno.

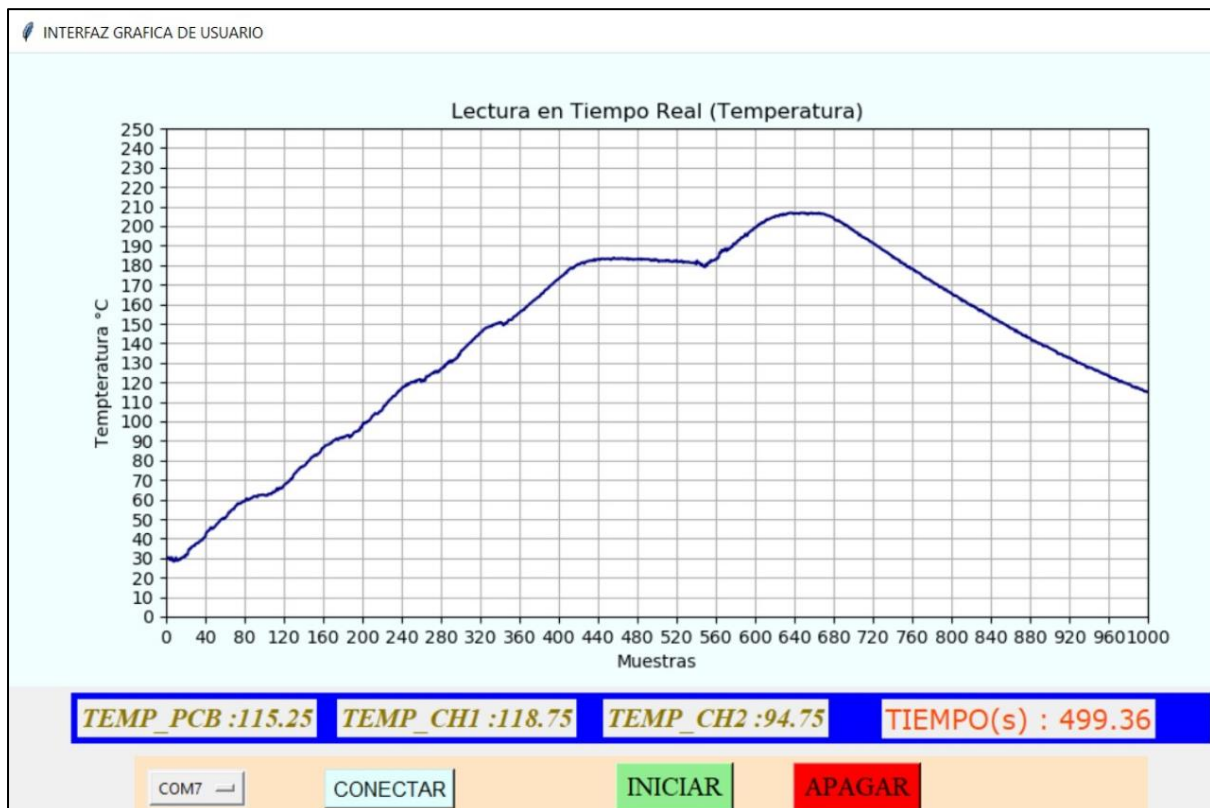


Fig. 0.30. Medición de temperatura.

Fuente: (El Autor, 2020)

En la Figura 3.31 se muestra el resultado final en la cual se puede observar que los componentes quedan completamente soldados y el grano de soldadura presenta una estructura fina consiguiendo una unión más fuerte y fiable.

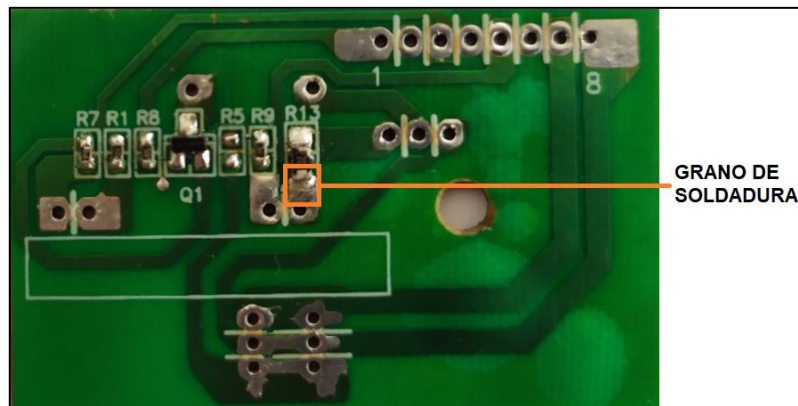


Fig. 0.31. Prueba de soldadura.

Fuente: (El Autor, 2020)

### 3.8. Presupuesto para la construcción del equipo de soldadura

En el presupuesto se incluye el costo de todos los materiales necesarios para realizar la construcción del equipo a nivel de prototipo. En la Tabla 3.2 se detalla cada componente con su costo en dólares americanos.

TABLA 0.2  
PRESUPUESTO DEL EQUIPO DE SOLDADURA

CANTIDAD	UNIDAD	ESPECIFICACIÓN	COSTO UNITARIO (\$)	COSTO TOTAL (\$)
1	u	Horno Tostador HomeTech	30	30
1	m <sup>2</sup>	Lana de vidrio	5	5
4	u	Soporte cerámico para lámpara halógena 1500W	2,5	10
3	m	Cable de asbesto para alta temperatura N° 12 AWG	2	6
1	m	Cable de asbesto para alta temperatura N° 10 AWG	3	3
4	u	Lámpara Halógena de cuarzo lineal 1500W	3	12
2	u	Bisagras metálicas pequeñas	0,15	0,30
1	u	Ventilador Brushless 12VDC – 1.30A 4 terminales	5	5
1	u	Ventilador Brushless 12VDC – 0.12A 2 terminales	3	2
1	u	Motor universal de 120VAC - 25W + rodete	15	15
1	u	Arduino Mega 2560	20	20
1	u	Raspberry Pi Model 3B+	60	50
1	u	Pantalla táctil 3.5" para Raspberry Pi 3B+	25	25
1	u	Tarjeta de memoria microSD 16GB	5	5
1	u	Cable dupont macho-hembra	3,5	3,5
1	u	Adaptador de voltaje 5V 3A	3	3
1	u	Fuente de voltaje 12V 5A	7	7
3	u	Módulo Sensor Max 6675 con termopar	8.35	25.05
5	u	Fabricación de PCB	3	15
21	u	Resistencia eléctrica de 1/2W de diferentes valores	0,05	1,05
3	u	Capacitores cerámicos diferentes valores	0,6	1,8
1	u	Optoacoplador bipolar H11AA1	2	2
4	u	Optoacoplador moc3021	0,4	1,6
5	u	Zocalo Socket DIP DIL 4 pines	0,2	1
1	u	Portafusible para pcb y fusible de 1A	1	1
1	u	Portafusible y fusible de 5A	1,5	1,5
4	u	Triac BTA41	1,9	7,6
1	u	Triac BT136-600D	1	1
1	u	Transistor NPN TIP122	1	1

1	u	Transistor NPN 2N3904	0,8	0,8
10	u	Bornera doble Kf301-2p	0,35	3,5
1	u	Bornera plástica 15A	1	1
2	u	Espadines header hembra 2.54mm	1	2
2	u	Espadines header macho 2.54mm	0,75	1,5
3	u	Disipadores de calor	3	9
1	u	Selector 2 posiciones	1,45	1,45
1	u	Tomacorriente doble	2,1	2,1
1	u	Interruptor termomagnético 32A	6	6
1	u	Caja acrílica 3mm	50	50
6	m	Conductor eléctrico N° 14 AWG	0,34	2,04
2	m	Conductor eléctrico N° 3x12 AWG	3	6
1	u	Enchufe 3 patas 15A	1	1
1	u	Canaleta ranurada	2,8	2,8
8	u	Terminales tipo U	0,04	0,32
24	u	Terminales tipo pin	0,12	2,88
TOTAL				353,79

Fuente: (El Autor, 2020)

### 3.9. Comparación de costos de equipos de soldadura para componentes SMD.

Para la comparación de costos de equipos de soldadura se basa en el análisis bibliográfico en el cual se tienen los precios de diferentes modelos de hornos comerciales con el costo del horno desarrollado. Hay que aclarar que los precios de los diferentes modelos de hornos comerciales se los encuentran a ese precio en sus países de origen, y considerando que no existe producción nacional, el importar un equipo de estos implicaría en un aumento de precio debido a costos de envío, pago de impuestos y tasas arancelarias. Por esta razón se considera el precio de venta como costo de producción en el Ecuador. En la Tabla 3.3 se indica tales cantidades.

TABLA 0.3  
COSTOS DE PRODUCCIÓN DE DIFERENTES MODELOS DE HORNO

MODELO DE HORNO	COSTO (\$)
EQUIPO CONSTRUIDO	353.79
T962A	700
T937M	700
SUNY-SMT100	1000
ZB630RF	1600
T200A	2000
TYR108	2750
QHL320A	2800
SMD-2007	2900
NEODEN IN6	3000

Fuente: (El Autor, 2020)

Para poder observar la comparación de costos de producción de diferentes modelos de Hornos, la siguiente figura nos ilustra claramente que el equipo construido presenta un valor menor.

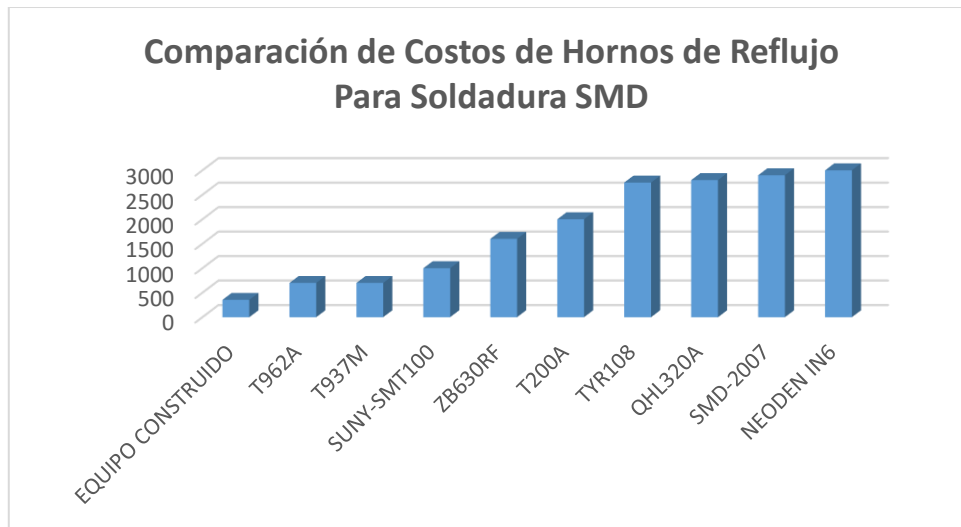


Fig. 0.32. Comparación de costos de producción de diferentes modelos de Hornos.

Fuente: (El Autor, 2020)

Esta comparación se realiza considerando el equipo construido a nivel de prototipo por lo que de llegar a ser comercializado se podrá analizar el precio del equipo (costo + utilidad) en futuras investigaciones.

### 3.10. Manual de operación y mantenimiento del equipo automatizado

En este apartado se detalla un manual de operación y mantenimiento del equipo de soldadura con la finalidad de poder utilizarlo sin inconvenientes.

Para poder utilizar el equipo de manera correcta es necesario seguir las siguientes instrucciones:

1. Conectar el enchufe de alimentación de manera que se respete la conexión Fase – Neutro – Tierra.
2. Ingresar la PCB a soldar en el horno ubicándole lo más cerca posible al sensor de temperatura evitando tocar los elementos calefactores.
3. Encender el miniordenador Raspberry Pi a través del selector de 2 posiciones.
4. Ejecutar la aplicación de interfaz gráfica que se encuentra en el escritorio. Para ello primero se debe abrir IDLE Python y desde ahí abrir la interfaz gráfica.
5. Establecer la conexión al puerto detectado.
6. Permitir el paso de corriente a los elementos calefactores a través del interruptor termomagnético.
7. Iniciar el proceso de soldadura (El foco rojo será la indicación del funcionamiento de los elementos calefactores).



8. Cuando el ventilador de enfriamiento se haya detenido será la indicación del fin del proceso de soldadura, presionar el botón “APAGAR” y por seguridad bajar el interruptor termomagnético.
9. Extraer la PCB con cuidado ya que podría seguir caliente.

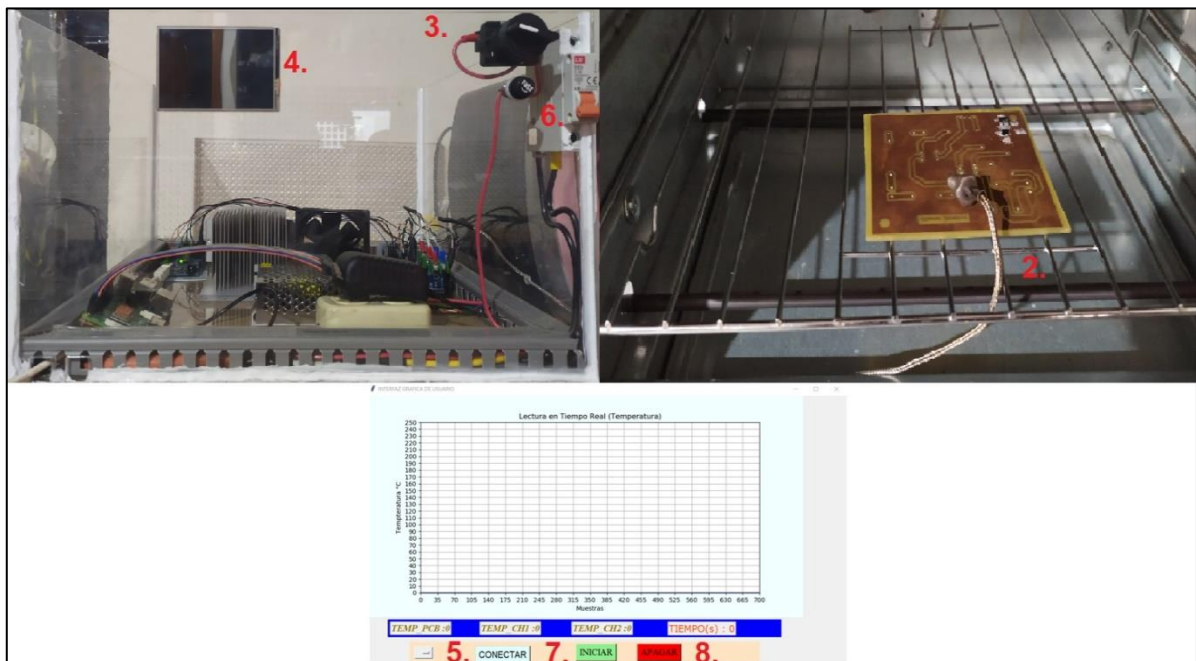


Fig. 0.33. Operación del equipo de soldadura.

Fuente: (El Autor, 2020)

Las partes que pueden sufrir un mayor desgaste son las lámparas halógenas ya que son las que proveen de calor al horno, por esta razón se debe limpiar los extremos de contacto cada cierto tiempo de uso tanto de los elementos calefactores como de las bases cerámicas en donde van instaladas dichas lámparas, ya que puede generarse una especie de hollín (Figura 3.34).



Fig. 0.34. Hollín generado en lámpara halógena.

Fuente: (El Autor, 2020)

En caso de que no se encienda alguna lámpara halógena revisar los bornes de conexión de la base cerámica (Figura 3.35).



Fig. 0.35. Bornes de conexión en base cerámica.

Fuente: (El Autor, 2020)

El ventilador a sufrir mayor desgaste es el utilizado para realizar la circulación de aire caliente, por lo que se deberá revisar y lubricar las partes móviles ya que al estar expuesto al calor del horno podría resecarse y trabarse (Figura 3.36). De igual forma se debe revisar antes y después de usar el equipo que la tuerca que une el rodete con el eje del motor se encuentre bien ajustado.



Fig. 0.36. Revisión de partes móviles del motor universal.

Fuente: (El Autor, 2020)

## CONCLUSIONES

- Con el análisis bibliográfico realizado se logró conocer las diferentes tecnologías de soldadura para circuitos impresos, y los parámetros necesarios que se necesita para llevar a cabo una adecuada soldadura a través de un equipo automatizado. Esta información recopilada permite definir qué elementos se requieren incluir en el equipo a automatizar y como deben operar para asegurar su funcionamiento.
- La utilización de software libre como Python y Arduino ayuda a reducir los costos por pagos de licencias, además de ser softwares muy livianos compatibles en cualquier dispositivo o sistema operativo, y presentan características de funcionalidad parecidas a software como Labview para aplicaciones pequeñas de interfaz gráfica y transmisión de datos.
- Cada una de las placas de circuito impreso cumplen satisfactoriamente la función para lo cual fueron diseñadas, logrando controlar cargas de corriente alterna por medio de Triacs, toma de datos de los sensores y activación de cargas de corriente continua a través de transistores.
- El realizar un control PID discreto con el microcontrolador Arduino resulta eficiente siempre y cuando se realice una correcta obtención del modelo matemático de la planta, con tiempos de muestreo correctos y adecuado manejo de temporizadores para que ejecute las tareas de control en el microcontrolador.
- El incluir un ventilador para la circulación de aire caliente permite una distribución más homogénea de calor en la cámara del horno, aunque al ser un ventilador improvisado debido al costo de un ventilador propiamente destinado para tal finalidad, presenta una ligera disminución de funcionalidad, pero a pesar de este inconveniente logra mantener la uniformidad de calor en la cámara del horno.
- Los resultados de soldadura obtenidos son muy satisfactorios al tener un grano de soldadura fina y una adecuada sujeción de los componentes electrónicos a la placa de circuito impreso, esto se logra al respetar cada etapa de soldado con un buen control de temperatura.
- El costo de elaboración del equipo de soldadura representa un 46.7% más económico que el modelo de horno más barato encontrado en el mercado y que logra brindar características similares que uno industrial, lo que lo vuelve un equipo de bajo costo, pero confiable, funcional y práctico.
- Después de haber realizado el proyecto y pruebas necesarias se concluye que el objetivo general planteado al comienzo fue resuelto satisfactoriamente a pesar de los problemas que se presentaron al transcurso de su implementación.

## RECOMENDACIONES

- Para mejorar el proceso y hacer aún más eficiente y exacto el seguimiento del perfil de temperatura se recomienda utilizar un controlador avanzado, pudiendo ser un controlador en cascada, control basado en lógica difusa, algoritmos genéticos o incluso un controlador basado en inteligencia artificial.
- Se recomienda para futuras investigaciones se utilice infrarrojos en lugar de lámparas halógenas como elementos calefactores debido a que, al ser lámparas fabricadas para usos de iluminación, están paulatinamente desapareciendo del mercado debido a la tecnología led.
- Para realizar una mejor soldadura es recomendable ubicar la bandeja en donde se asienta la PCB lo más cerca posible de los elementos calefactores, lo que permite una transmisión de calor más directa.
- Para poder realizar el proceso de soldadura con acabados más profesionales se recomienda investigar un método de aplicación de la pasta de soldadura como el de serigrafía, ya que al aplicar la soldadura sin seguir un protocolo podría quedar restos de estaño en áreas no deseadas lo que pudiera generar algún cortocircuito entre pines.
- El presente equipo de soldadura está especialmente diseñado para la soldadura de componentes electrónicos de montaje superficial utilizando una pasta de soldar a base de plomo – estaño con composición Sn63Pb37 por ser mayormente utilizado en el Ecuador, en caso de requerir seguir otros perfiles de soldadura previa configuración en el controlador se podría incorporar un menú en la interfaz gráfica en el cual permita seleccionar el tipo de soldadura.

## REFERENCIAS

- Acuña, P., Chica, G., & Marquez, C. (2011). *Componentes , herramientas y equipos de Laboratorio*. Retrieved from [https://s3.amazonaws.com/academia.edu.documents/39473732/Practica1-11.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1521757643&Signature=nEkbWtnlWFHSfD3DspS%2FjqMdrik%3D&response-content-disposition=inline%3B filename%3DPractica1-11\\_electronica\\_analoga](https://s3.amazonaws.com/academia.edu.documents/39473732/Practica1-11.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1521757643&Signature=nEkbWtnlWFHSfD3DspS%2FjqMdrik%3D&response-content-disposition=inline%3B filename%3DPractica1-11_electronica_analoga).
- Alzate, E., Montes, J., & Silva, C. (2016). Medición de Temperatura: Sensores Termoeléctricos. *ResearchGate*, 1–7. Retrieved from <https://www.researchgate.net/publication/26544249>
- AV Electronics. (2020). 3.5 TFT Touch for Raspberry Pi 2/3. Retrieved August 5, 2020, from <https://avelectronics.cc/producto/3-5-tft-touch-for-raspberry-pi-2-3/>
- Carrasco, F. (2018). *Led con tecnología “SMD” (Dispositivos de Montaje Superficial)*. Retrieved from [http://www.fcarrasco.es/img\\_productos/archivos/NECESARIA LECTURA LED SMD COB\\_2886649.pdf](http://www.fcarrasco.es/img_productos/archivos/NECESARIA LECTURA LED SMD COB_2886649.pdf)
- Carrillo, A. (2011). *Sistemas Automáticos de Control Fundamentos Básicos de Análisis y Modelado* (2da ed.; UNERMB, Ed.). Retrieved from [http://150.185.9.18/fondo\\_editorial/images/PDF/CUPUL/SISTEMA DE CONTROL 1.pdf](http://150.185.9.18/fondo_editorial/images/PDF/CUPUL/SISTEMA DE CONTROL 1.pdf)
- Cervantes, Al., & García, H. (2013). *Horno Automático de Precisión para Componentes Electrónicos de Montaje Superficial. (Tesis de pregrado)*. (Instituto Politécnico Nacional, México). Retrieved from <https://tesis.ipn.mx/jspui/handle/123456789/12749>
- CONUEE (Comisión Nacional Para El Uso Eficiente de la Energía). *Beneficios del aislamiento térmico en la industria.* , (2009).
- Corporación-Peregrine. (2016). *Soldering Guidelines for Mounting Bottom-terminated Components*. California.
- Creus, A. (2011). *Instrumentación Industrial 8va edición* (Afaomega, Ed.). México D.F.
- De Blas, E. A. (2018). *Diseño y construcción de horno para soldadura SMD*. Retrieved from [http://oa.upm.es/52383/1/PFC\\_EDUARDO\\_ALONSO\\_DE\\_BLAS.pdf](http://oa.upm.es/52383/1/PFC_EDUARDO_ALONSO_DE_BLAS.pdf)

- Del Busto, R. (2013). *Análisis y Diseño de Sistemas de Control Digital* (Primera). México: McGRAW-HILL/INTERAMERICANA EDITORES, S.A. DE C.V.
- Deshmukh, G., Birwal, P., Datir, R., & Patel, S. (2017). Thermal Insulation Materials: A Tool for Energy Conservation. *Journal of Food Processing & Technology*, 08(04), 1–5. <https://doi.org/10.4172/2157-7110.1000670>
- Dpto. Electrónica. (2015). La teoría | Soldadura SMD con tecnología BGA. Retrieved September 22, 2019, from <http://bga.blog.tartanga.eus/cuestiones-teoricas/>
- Dpto. Ing. Electrónica. (2011). *Práctica 8. Fabricación de circuito impreso 1*. Retrieved from [http://www.dinel.us.es/ASIGN/CE\\_2T/pracs/2010-2011/p2/practica8.pdf](http://www.dinel.us.es/ASIGN/CE_2T/pracs/2010-2011/p2/practica8.pdf)
- Entrena, F. (2016). *Instalación de equipos y elementos de sistemas de automatización industrial*. (Primera; IC, Ed.). Andalucía.
- FCTA-México. (2018). Soldadura en Pasta de Baja Temperatura. Retrieved March 11, 2020, from <https://fctamexico.mx/soldadura-en-pasta-de-baja-temperatura/>
- Fernández, P. (2009). *Compresores*. Universidad de Cantabria. Departamento De Ingeniería Eléctrica Y Energética. Retrieved from [es.pfernandezdiez.es/libro/?id=9](http://es.pfernandezdiez.es/libro/?id=9)
- García, L. (2009). *Control Digital Teoría y Práctica* (Segunda). Medellín: Politécnico Colombiano JIC.
- Gouvea, M. (2014). El potencial de la tecnología de montaje en superficie (SMT) en un proyecto. Retrieved October 3, 2019, from PRODUZA website: <https://produza.ind.br/tecnologia/surface-mount-technology-smt/>
- Gurmukh, A. (2014). *SOLDER JOINT RELIABILITY. A UNIFIED THERMO-MECHANICAL MODEL APPROACH* (Tesis Postgrado). North Dakota State University of Agriculture and Applied Science.
- Gutiérrez, M., & Iturralde, S. (2017). *Fundamentos Basicos De Instrumentacion y Control* (UPSE, Ed.). Retrieved from [www.upse.edu.ec](http://www.upse.edu.ec)
- Heredia, A. (2009). *ANÁLISIS DIDÁCTICO DE LOS DIFERENTES TIPOS DE LÁMPARAS INCANDESCENTES Y DESCARGA*. 1–13. Retrieved from [https://archivos.csif.es/archivos/andalucia/ensenanza/revistas/csicsif/revista/pdf/Numero\\_21/ANTONIO JOSE\\_ HEREDIA SOTO\\_1.pdf](https://archivos.csif.es/archivos/andalucia/ensenanza/revistas/csicsif/revista/pdf/Numero_21/ANTONIO JOSE_ HEREDIA SOTO_1.pdf)

- Hernández, R. (2010). *Introducción a los sistemas de control: Conceptos, aplicaciones y simulación con MATLAB*. (1ra ed.). México: PEARSON EDUCATION.
- Holman, J. P. (2010). Heat Transfer. In Mcgraw-Hill (Ed.), *HEAT TRANSFER, TENTH EDITION* (Tenth). <https://doi.org/10.1016/b978-1-933762-24-1.50019-x>
- Humanoc. (2010). Perfil de soldadura THT donde se aprecian los conos de soldadura. Retrieved October 3, 2019, from [https://upload.wikimedia.org/wikipedia/commons/d/d4/Cono\\_de\\_soldadura.png](https://upload.wikimedia.org/wikipedia/commons/d/d4/Cono_de_soldadura.png)
- JMN Electronics. (2018). Diseño, programación y fabricación de productos y prototipos electrónicos. Cómo soldar SMD: parte 1. Retrieved October 2, 2019, from Alicante, España website: <http://jmnelectronics.com/archives/1595>
- Kester. (2009). *Easy Profile 256 No-Clean Solderpaste* (p. 2). p. 2. Retrieved from [www.kester.com](http://www.kester.com)
- Lee, N. C. (2002). *Reflow Soldering Processes and Troubleshooting: SMT, BGA, CSP and Flip Chip Technologies*. New York: Newnes.
- Llamas, L. (2017a). Detector de cruce por cero con Arduino y optoacoplador H11AA1. Retrieved August 5, 2020, from Ingeniería, informática y diseño website: <https://www.luisllamas.es/arduino-cruce-por-cero-h11aa1/>
- Llamas, L. (2017b). Filtro paso bajo y paso alto exponencial (EMA) en Arduino. Retrieved October 3, 2020, from Ingeniería, informática y diseño website: <https://www.luisllamas.es/arduino-paso-bajo-exponencial/>
- Lund, L. (2004). Revista ElectroIndustria. Lo que hay que saber de la Soldadura en Electrónica. Retrieved March 11, 2020, from <http://www.emb.cl/electroindustria/articulo.mvc?xid=169>
- Machado, E., & Coto, H. (2017). Sistema de adquisición de datos con Python y Arduino. *Revista Ciencia, Ingeniería y Desarrollo Tec Lerdo*, 1–6. Retrieved from [http://revistacid.itslerdo.edu.mx/coninci2017/05 Sistema de adquisición de datos con Python y Arduino.pdf](http://revistacid.itslerdo.edu.mx/coninci2017/05_Sistema_de_adquisici%3Fon_de_datos_con_Python_y_Arduino.pdf)
- Maloney, T. (2006). *Electrónica Industrial Moderna* (5a ed.). México: PEARSON EDUCATION.
- Manual\_Ersa. (2015). *Manual de soldadura ERSA. Haciendo fácil la soldadura*. Retrieved

from [https://www.tch.es/wp-content/uploads/manual-de-soldadura\\_E\\_def.-1.pdf](https://www.tch.es/wp-content/uploads/manual-de-soldadura_E_def.-1.pdf)

Martín, Á., & De Soto Gorroño, A. (2005). *Desarrollo y construcción de prototipos electrónicos: tutoriales OrCAD 10 Y LPKF 5 de ayuda al diseño*. MARCOMBO S.A.

Masabanda, S., & Yauripoma, M. (2009). *ELABORACIÓN DE UN MANUAL PARA PRÁCTICAS DE LA MATERIA TEORÍA DE PROCESOS DE SOLDADURA (Tesis Pregrado)*. (Escuela Politécnica Nacional). Retrieved from <http://bibdigital.epn.edu.ec/handle/15000/1030>

Maxim Integrated Products. (2014). *MAX6675. Cold-Junction-Compensated K-Thermocouple-to-Digital Converter (0°C to +1024°C)*. Retrieved from <https://datasheets.maximintegrated.com/en/ds/MAX6675.pdf>

Méndez, A. (2018). *Soldadura manual de componentes electrónicos*. España.

Merchan, F. (2018). *TECNOLOGÍA DE MONTAJE SUPERFICIAL*. Retrieved October 4, 2019, from <https://tecnologiademontajesuperficial.es.tl/>

Ñeco, R., Reinoso, O., García, N., & Aracil, R. (2013). *Apuntes de sistemas de control*. Retrieved from <https://elibro.net/es/ereader/utnorte/62263?page=NaN>

Norman, R. (2001). *Solder Reflow Technology*. Research International Solder Reflow Technology Handbook.

Ogata, K. (2010). *Ingeniería de control moderna (5ta ed.)*. Madrid: PEARSON EDUCATION S.A.

Oshiro, M. (2014). *FICHA ESTÁNDAR DE FAMILIA DEL CATÁLOGO DE BIENES, SERVICIOS Y OBRAS DEL MEF FAMILIA 28540043 LÁMPARAS HALÓGENAS*. 1–7. Retrieved from [https://www.mef.gob.pe/contenidos/doc\\_siga/catalogo/ctlogo\\_familias\\_lampara\\_halogenea.pdf](https://www.mef.gob.pe/contenidos/doc_siga/catalogo/ctlogo_familias_lampara_halogenea.pdf)

Pallás Areny, R. (2003). *Sensores y acondicionadores de señal (Cuarta ed.)*. Retrieved from [https://books.google.com.ec/books?id=Eevyk28\\_fVkc&pg=PR12&lpg=PR12&dq=sensores+y+acondicionadores+de+señal+4a+ramón+pallás+areny+fecha+de+publicación+original&source=bl&ots=JXmG59Emgj&sig=ACfU3U3OA2TGz1NzAMumXPawWEiqTUSsUQ&hl=es-419&sa=X&ved=2ahUKEwiyx96r9](https://books.google.com.ec/books?id=Eevyk28_fVkc&pg=PR12&lpg=PR12&dq=sensores+y+acondicionadores+de+señal+4a+ramón+pallás+areny+fecha+de+publicación+original&source=bl&ots=JXmG59Emgj&sig=ACfU3U3OA2TGz1NzAMumXPawWEiqTUSsUQ&hl=es-419&sa=X&ved=2ahUKEwiyx96r9)

Palomo, M. (2017). *Aislantes Térmicos Criterios De Selección Por Requisitos Energéticos*.



- (*Tesis de pregrado*). (Universidad Politécnica de Madrid). Retrieved from [http://oa.upm.es/47071/1/TFG\\_Palomo\\_Cano\\_Marta.pdf](http://oa.upm.es/47071/1/TFG_Palomo_Cano_Marta.pdf)
- Paredes, A. (2020). Explicación Control de fase directo | AX Desarrollos. Retrieved September 28, 2020, from <http://axdesarrollos.blogspot.com/2013/09/explicacion-control-de-fase-directo.html>
- Pineda, A. (2013). *INSTRUMENTACIÓN VIRTUAL. Fundamentos de Programación Gráfica con Labview*. Retrieved from <http://prod77ms.itesm.mx/podcast/EDTM/ID355.pdf>
- Placeres, J. (2002). *DESARROLLO DE PROTOTIPOS PARA TECNOLOGIA DE MONTAJE SUPERFICIAL (SMT) (Tesis Postgrado)*. Instituto Tecnológico y de Estudios Superiores de Monterrey.
- Polosaari, K. (2006). Circuito integrado con encapsulado THT del tipo DIP. Retrieved October 3, 2019, from ST Microelectronics website: [https://es.wikipedia.org/wiki/Tecnología\\_de\\_agujeros\\_pasantes#/media/Archivo:Three\\_IC\\_circuit\\_chips.JPG](https://es.wikipedia.org/wiki/Tecnología_de_agujeros_pasantes#/media/Archivo:Three_IC_circuit_chips.JPG)
- Prasad, R. (1998). Surface mount technology: Principles and practice (2nd edn). In *Microelectronics Journal* (Vol. 29). [https://doi.org/10.1016/s0026-2692\(97\)00072-4](https://doi.org/10.1016/s0026-2692(97)00072-4)
- Ramos Lara, R. (2007). *SISTEMAS DIGITALES DE INSTRUMENTACIÓN Y CONTROL*.
- Rashid, M. (2004). *Electrónica De Potencia* (Tercera Ed). México: PEARSON EDUCATION.
- Reyes Cortés, F., & Cid Monjaraz, J. (2015). *ARDUINO APLICACIONES EN ROBÓTICA, MECATRÓNICA E INGENIERÍAS* (Primera). México: Alfaomega Grupo Editor, S.A. de C.V.
- RFICs, P. (2016). *Solder Reflow Recommendations Sn / Pb and Pb Free*. 1–3.
- Rodríguez, J. (2015). MÓDULO DE INSTRUMENTACIÓN INDUSTRIAL. Medición De Temperatura. Retrieved March 12, 2020, from Corporación Universitaria De La Costa website: <https://es.slideshare.net/LucasMosquera1/6-mediciondetemperatura>
- Salt, J., Cuenca, Á., Casanova, V., & Correcher, A. (2015). *CONTROL AUTOMÁTICO Tiempo Continuo y Tiempo Discreto* (Primera). Barcelona: EDITORIAL REVERTÉ, S.A.
- Salvador Escoda S.A. (2014). *Manual práctico de ventilación: Catálogo técnico* (Vol. 2). Retrieved from <http://www.salvadorescoda.com/tecnico/VE/tec-ve.htm>

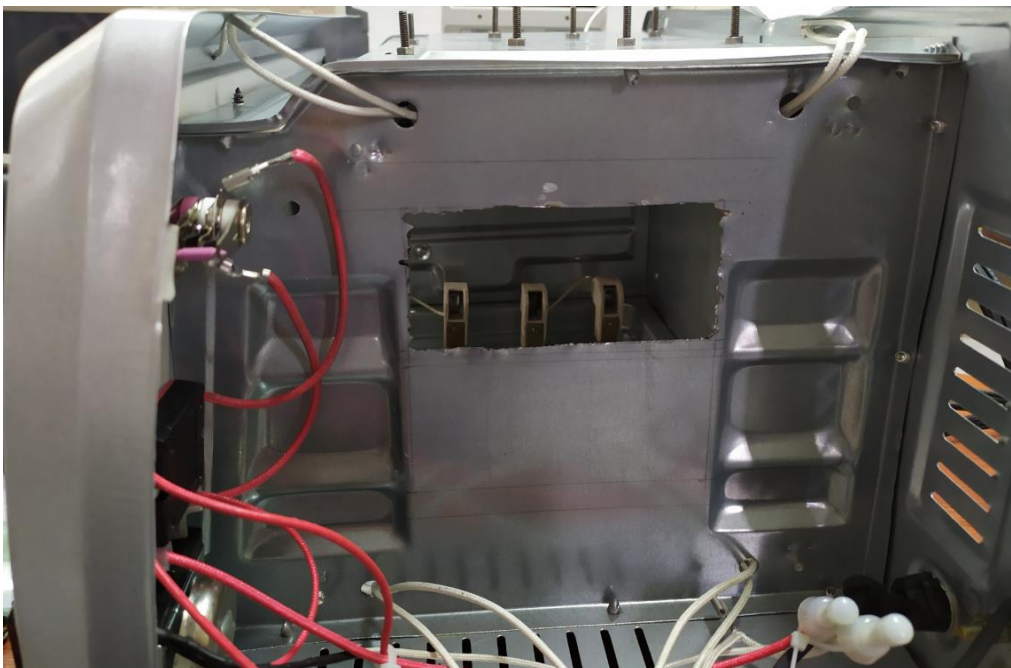
- Soler, & Palau. (2012). *Manual práctico de ventilación* (Vol. 4). Retrieved from [https://cdn2.hubspot.net/hubfs/2009592/Ebooks/SPL - Manual práctico de ventilación/SPA\\_Manual\\_practico\\_ventilacion.pdf](https://cdn2.hubspot.net/hubfs/2009592/Ebooks/SPL - Manual práctico de ventilación/SPA_Manual_practico_ventilacion.pdf)
- Strauss, R. (1998). SMT Soldering Handbook. In Second (Ed.), *SMT Soldering Handbook* (Newnes). <https://doi.org/10.1016/b978-0-7506-3589-9.x5000-0>
- Tamari, A. (2017). Why Use Through-Hole Technology in PCB Design? | Altium. Retrieved October 4, 2019, from Altium website: <https://resources.altium.com/pcb-design-blog/why-use-through-hole-technology-in-pcb-design>
- Tojeiro, G. (2014). *TALLER DE ARDUINO Un enfoque práctico para principiantes* (Primera Ed). Barcelona: MARCOMBO S.A.
- Tojeiro, G. (2016). *Raspberry Pi2 para electrónicos* (1ra ed.). México D.F.: Alfaomega Grupo Editor, S.A. de C.V.
- Torrente, Ó. (2016). *El mundo GENUINO-ARDUINO. Curso práctico de formación* (Primera Ed). México: Alfaomega Grupo Editor, S.A. de C.V.
- Tsai, T. N. (2012). Thermal parameters optimization of a reflow soldering profile in printed circuit board assembly: A comparative study. *Applied Soft Computing Journal*, 12(8), 2601–2613. <https://doi.org/10.1016/j.asoc.2012.03.066>
- Verrone, E. F. (2006). *Soldadura SMT por Convección Forzada. I*, 1–6. Retrieved from [http://picmania.garcia-cuervo.net/recursos/redpictutorials/fabricacion\\_pcb/soldadura\\_smt\\_por\\_convecci%F3n\\_forzada\\_trate05-009.pdf](http://picmania.garcia-cuervo.net/recursos/redpictutorials/fabricacion_pcb/soldadura_smt_por_convecci%F3n_forzada_trate05-009.pdf)
- Whalley, D. C. (2004). A simplified reflow soldering process model. *Journal of Materials Processing Technology*, 150(1–2), 134–144. <https://doi.org/10.1016/j.jmatprotec.2004.01.029>
- Yunus, A., & Afshin, J. (2015). *HEAT AND MASS TRANSFER FUNDAMENTALS & APPLICATIONS* (Fifth; McGraw-Hill, Ed.). New York.

## ANEXOS

Anexo A: Medición de temperatura por medio de una cámara termográfica al horno con la incorporación de lámparas halógenas.



Anexo B: Corte en estructura del horno para compuerta



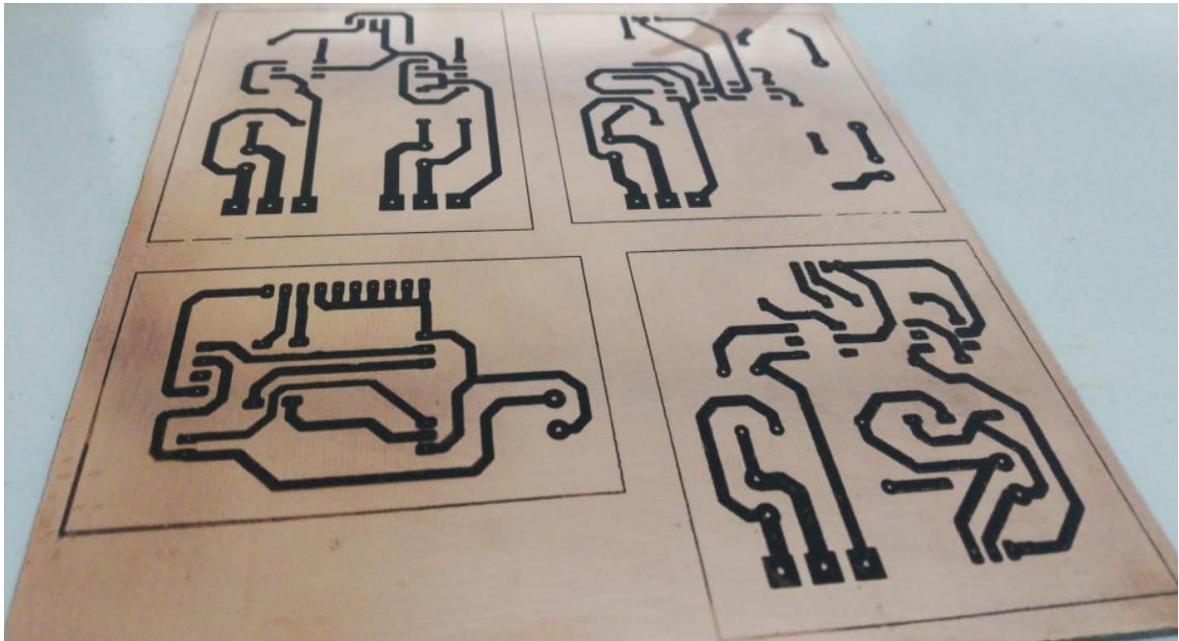
Anexo C: Conducto para ventilación y aislamiento térmico



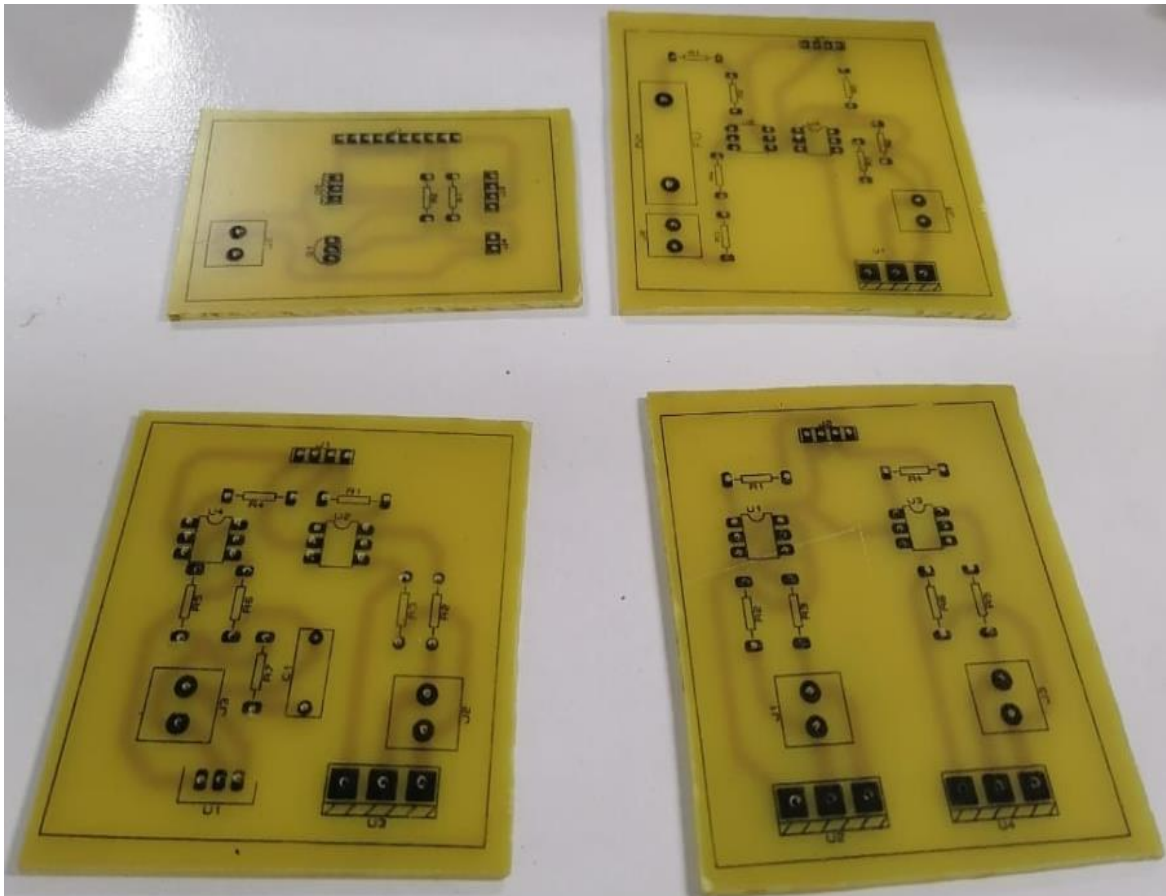
Anexo D: Corte en tapa externa del horno para instalación de ventilador



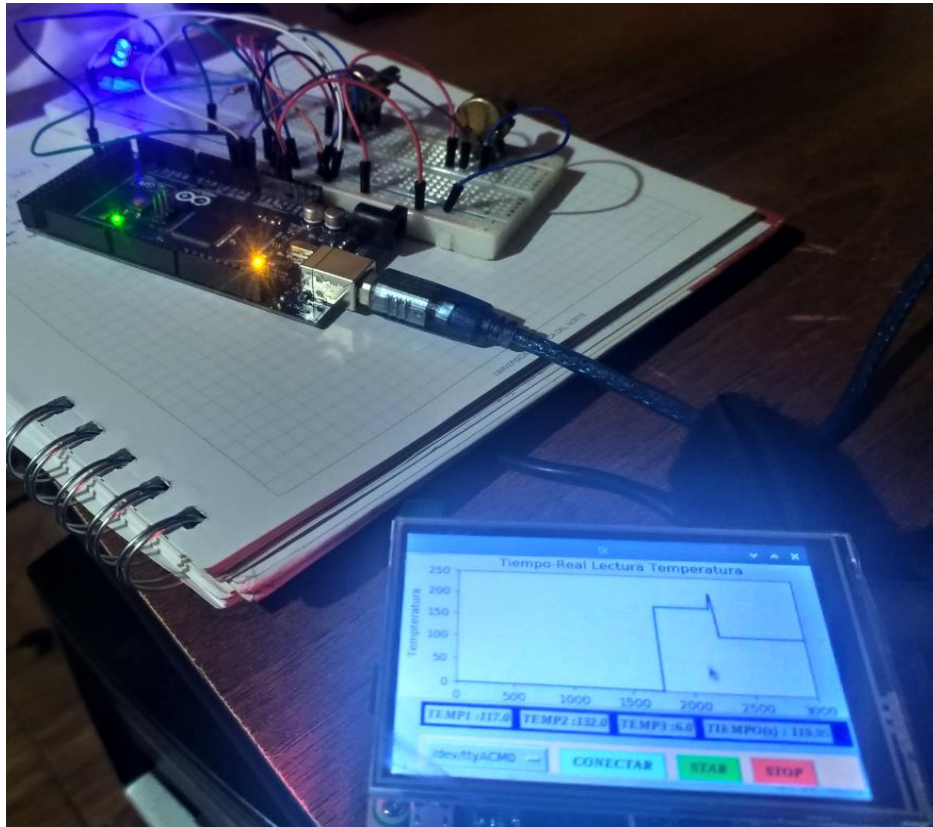
### Anexo E: Elaboración de PCB



### Anexo F: Tarjetas de circuito impreso terminadas



## Anexo G: Prueba comunicación Raspberry Pi - Arduino





## 6-Pin DIP Random-Phase Optoisolators Triac Driver Output (400 Volts Peak)

The MOC3020 Series consists of gallium arsenide infrared emitting diodes, optically coupled to a silicon bilateral switch.

- To order devices that are tested and marked per VDE 0884 requirements, the suffix "V" must be included at end of part number. VDE 0884 is a test option. They are designed for applications requiring isolated triac triggering.

Recommended for 115/240 Vac(rms) Applications:

- Solenoid/Valve Controls
- Lamp Ballasts
- Interfacing Microprocessors to 115 Vac Peripherals
- Motor Controls
- Static ac Power Switch
- Solid State Relays
- Incandescent Lamp Dimmers

**MOC3021**  
**MOC3022**  
**MOC3023**



STANDARD THRU HOLE

### MAXIMUM RATINGS (T<sub>A</sub> = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
--------	--------	-------	------

#### INFRARED EMITTING DIODE

Reverse Voltage	V <sub>R</sub>	3	Volts
Forward Current — Continuous	I <sub>F</sub>	60	mA
Total Power Dissipation @ T <sub>A</sub> = 25°C Negligible Power in Triac Driver Derate above 25°C	P <sub>D</sub>	100	mW
		1.33	mW/°C

#### OUTPUT DRIVER

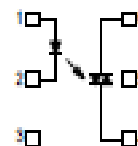
Off-State Output Terminal Voltage	V <sub>ORM</sub>	400	Volts
Peak Repetitive Surge Current (PW = 1 ms, 120 pps)	I <sub>TSM</sub>	1	A
Total Power Dissipation @ T <sub>A</sub> = 25°C Derate above 25°C	P <sub>D</sub>	300	mW
		4	mW/°C

#### TOTAL DEVICE

Isolation Surge Voltage <sup>(1)</sup> (Peak ac Voltage, 60 Hz, 1 Second Duration)	V <sub>ISO</sub>	7500	Vac(pk)
Total Power Dissipation @ T <sub>A</sub> = 25°C Derate above 25°C	P <sub>D</sub>	330	mW
		4.4	mW/°C
Junction Temperature Range	T <sub>J</sub>	-40 to +100	°C
Ambient Operating Temperature Range	T <sub>A</sub>	-40 to +85	°C
Storage Temperature Range	T <sub>stg</sub>	-40 to +150	°C
Soldering Temperature (10 s)	T <sub>L</sub>	260	°C

1. Isolation surge voltage, V<sub>ISO</sub>, is an internal device dielectric breakdown rating. For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.

### SCHEMATIC



- ANODE
- CATHODE
- NC
- MAIN TERMINAL
- SUBSTRATE  
DO NOT CONNECT
- MAIN TERMINAL

# Anexo I: Hoja de especificaciones TRIAC BTA41



## BTA40, BTA41, BTB41

40 A standard TRIACs

### Features

- High current TRIAC
- Low thermal resistance with clip bonding
- High commutation capability
- BTA series UL1557 certified (File ref: 81734)
- Packages are RoHS (2002/95/EC) compliant

### Applications

- On/off function in static relays, heating regulation, induction motor starting circuits
- Phase control operations in light dimmers, motor speed controllers, and similar

### Description

Available in high power packages, the BTA/BTB40-41 series is suitable for general purpose AC switching.

The BTA series provides an insulated tab (rated at 2500 V rms).

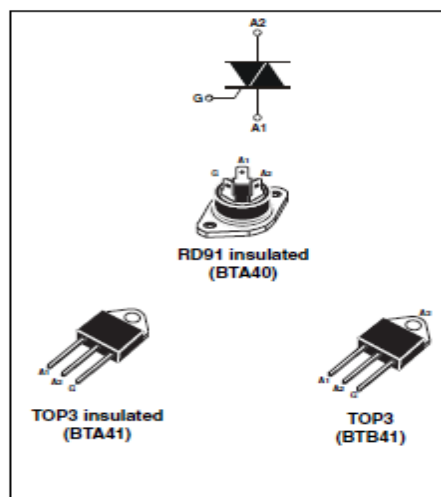


Table 1. Device summary

Symbol	Parameter	BTA40 <sup>(1)</sup>	BTA41 <sup>(1)</sup>	BTB41	Unit
$I_{T(RMS)}$	On-state rms current	40	41	41	A
$V_{DRM}/V_{RRM}$	Repetitive peak off-state voltage	600 and 800	600 and 800	600 and 800	V
$I_{GT}$	Triggering gate current	50	50	50	mA

1. Insulated package

### Characteristics

BTA40, BTA41, BTB41

## 1 Characteristics

Table 2. Absolute maximum ratings

Symbol	Parameter	Value	Unit		
$I_{T(RMS)}$	On-state rms current (full sine wave)	TOP3	40	A	
		RD91 / TOP ins.			
$I_{TSM}$	Non repetitive surge peak on-state current (full cycle, $T_j$ initial = 25 °C)	F = 50 Hz	400	A	
		F = 60 Hz	420		
$I_t^2$	$I_t^2$ Value for fusing	$t_p = 10$ ms	1000	A <sup>2</sup> s	
dI/dt	Critical rate of rise of on-state current $I_G = 2 \times I_{GT}$ , $t_r \leq 100$ ns	F = 120 Hz	$T_j = 125$ °C	50	A/ $\mu$ s
$V_{DSM}/V_{RSM}$	Non repetitive surge peak off-state voltage	$t_p = 10$ ms	$T_j = 25$ °C	$V_{DSM}/V_{RSM} + 100$	V
$I_{GM}$	Peak gate current	$t_p = 20$ $\mu$ s	$T_j = 125$ °C	8	A
$P_{G(AV)}$	Average gate power dissipation		$T_j = 125$ °C	1	W
$T_{stg}$ $T_j$	Storage junction temperature range Operating junction temperature range			- 40 to + 150 - 40 to + 125	°C

Table 3. Electrical characteristics ( $T_j = 25$  °C, unless otherwise specified)

Symbol	Parameter	Value	Unit		
$I_{GT}^{(1)}$	$V_D = 12$ V $R_L = 33$ $\Omega$	I - II - III IV	50 100	mA	
		ALL	1.3		
$V_{GT}$		MIN.	0.2	V	
$V_{GD}$	$V_D = V_{DRM}$ $R_L = 3.3$ k $\Omega$ $T_j = 125$ °C	ALL	80	mA	
$I_H^{(2)}$	$I_T = 500$ mA		70 160	mA	
$I_L$	$I_G = 1.2 I_{GT}$	I - III - IV II	500 10		
dV/dt <sup>(2)</sup>	$V_D = 67\%$ $V_{DRM}$ gate open	$T_j = 125$ °C	MIN.	500	V/ $\mu$ s
(dV/dt) <sub>c</sub> <sup>(2)</sup>	(dI/dt) <sub>c</sub> = 20 A/ms	$T_j = 125$ °C	MIN.	10	V/ $\mu$ s

1. Minimum  $I_{GT}$  is guaranteed at 5% of  $I_{GT}$  max.
2. for both polarities of A2 referenced to A1



## 9. Characteristics

Table 6. Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
<b>Static characteristics</b>						
$I_{GT}$	gate trigger current	$V_D = 12\text{ V}$ ; $I_T = 0.1\text{ A}$ ; T2+ G+; $T_J = 25\text{ }^\circ\text{C}$ ; <a href="#">Fig. 7</a>	-	2	5	mA
		$V_D = 12\text{ V}$ ; $I_T = 0.1\text{ A}$ ; T2+ G-; $T_J = 25\text{ }^\circ\text{C}$ ; <a href="#">Fig. 7</a>	-	2.5	5	mA
		$V_D = 12\text{ V}$ ; $I_T = 0.1\text{ A}$ ; T2- G-; $T_J = 25\text{ }^\circ\text{C}$ ; <a href="#">Fig. 7</a>	-	2.5	5	mA
		$V_D = 12\text{ V}$ ; $I_T = 0.1\text{ A}$ ; T2- G+; $T_J = 25\text{ }^\circ\text{C}$ ; <a href="#">Fig. 7</a>	-	5	10	mA
$I_L$	latching current	$V_D = 12\text{ V}$ ; $I_G = 0.1\text{ A}$ ; T2+ G+; $T_J = 25\text{ }^\circ\text{C}$ ; <a href="#">Fig. 8</a>	-	1.6	10	mA
		$V_D = 12\text{ V}$ ; $I_G = 0.1\text{ A}$ ; T2+ G-; $T_J = 25\text{ }^\circ\text{C}$ ; <a href="#">Fig. 8</a>	-	4.5	15	mA
		$V_D = 12\text{ V}$ ; $I_G = 0.1\text{ A}$ ; T2- G-; $T_J = 25\text{ }^\circ\text{C}$ ; <a href="#">Fig. 8</a>	-	1.2	10	mA
		$V_D = 12\text{ V}$ ; $I_G = 0.1\text{ A}$ ; T2- G+; $T_J = 25\text{ }^\circ\text{C}$ ; <a href="#">Fig. 8</a>	-	2.2	15	mA
$I_H$	holding current	$V_D = 12\text{ V}$ ; $T_J = 25\text{ }^\circ\text{C}$ ; <a href="#">Fig. 9</a>	-	1.2	10	mA
$V_T$	on-state voltage	$I_T = 5\text{ A}$ ; $T_J = 25\text{ }^\circ\text{C}$ ; <a href="#">Fig. 10</a>	-	1.4	1.7	V
$V_{GT}$	gate trigger voltage	$V_D = 12\text{ V}$ ; $I_T = 0.1\text{ A}$ ; $T_J = 25\text{ }^\circ\text{C}$ ; <a href="#">Fig. 11</a>	-	0.7	1	V
		$V_D = 400\text{ V}$ ; $I_T = 0.1\text{ A}$ ; $T_J = 125\text{ }^\circ\text{C}$ ; <a href="#">Fig. 11</a>	0.25	0.4	-	V
$I_O$	off-state current	$V_D = 600\text{ V}$ ; $T_J = 125\text{ }^\circ\text{C}$	-	0.1	0.5	mA
<b>Dynamic characteristics</b>						
$dV_D/dt$	rate of rise of off-state voltage	$V_{DM} = 402\text{ V}$ ; $T_J = 125\text{ }^\circ\text{C}$ ; $R_{GT1} = 1\text{ k}\Omega$ ; ( $V_{DM} = 67\%$ of $V_{DRM}$ ); exponential waveform	-	5	-	V/ $\mu\text{s}$
$t_{gt}$	gate-controlled turn-on time	$I_{TM} = 6\text{ A}$ ; $V_D = 600\text{ V}$ ; $I_G = 0.1\text{ A}$ ; $di_G/dt = 5\text{ A}/\mu\text{s}$	-	2	-	$\mu\text{s}$

Anexo K: Hoja de especificaciones Transistor TIP122

TIP120, TIP121, TIP122 (NPN); TIP125, TIP126, TIP127 (PNP)

MAXIMUM RATINGS

Rating	Symbol	TIP120, TIP125	TIP121, TIP126	TIP122, TIP127	Unit
Collector-Emitter Voltage	$V_{CE0}$	60	80	100	Vdc
Collector-Base Voltage	$V_{CB}$	60	80	100	Vdc
Emitter-Base Voltage	$V_{EB}$	5.0			Vdc
Collector Current - Continuous - Peak	$I_C$	5.0 8.0			Adc
Base Current	$I_B$	120			mAdc
Total Power Dissipation @ $T_C = 25^\circ\text{C}$ Derate above $25^\circ\text{C}$	$P_D$	63 0.52			W W/°C
Total Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above $25^\circ\text{C}$	$P_D$	2.0 0.016			W W/°C
Unclamped Inductive Load Energy (Note 1)	E	50			mJ
Operating and Storage Junction, Temperature Range	$T_J, T_{stg}$	-65 to +150			°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Max	Unit
Thermal Resistance, Junction-to-Case	$R_{\theta JC}$	1.92	°C/W
Thermal Resistance, Junction-to-Ambient	$R_{\theta JA}$	62.5	°C/W

Stresses exceeding those listed in the Maximum Ratings table may damage the device. If any of these limits are exceeded, device functionality should not be assumed, damage may occur and reliability may be affected.

1.  $I_C = 1\text{ A}$ ,  $L = 100\text{ mH}$ , P.R.F. = 10 Hz,  $V_{CC} = 20\text{ V}$ ,  $R_{BE} = 100\ \Omega$

ELECTRICAL CHARACTERISTICS ( $T_C = 25^\circ\text{C}$  unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
----------------	--------	-----	-----	------

OFF CHARACTERISTICS

Collector-Emitter Sustaining Voltage (Note 2) ( $I_C = 100\text{ mAdc}$ , $I_B = 0$ )	TIP120, TIP125 TIP121, TIP126 TIP122, TIP127	$V_{CE0(sus)}$	60 80 100	- - -	Vdc
Collector Cutoff Current ( $V_{CE} = 80\text{ Vdc}$ , $I_B = 0$ ) ( $V_{CE} = 40\text{ Vdc}$ , $I_B = 0$ ) ( $V_{CE} = 50\text{ Vdc}$ , $I_B = 0$ )	TIP120, TIP125 TIP121, TIP126 TIP122, TIP127	$I_{CE0}$	- - -	0.5 0.5 0.5	mAdc
Collector Cutoff Current ( $V_{CB} = 60\text{ Vdc}$ , $I_E = 0$ ) ( $V_{CB} = 80\text{ Vdc}$ , $I_E = 0$ ) ( $V_{CB} = 100\text{ Vdc}$ , $I_E = 0$ )	TIP120, TIP125 TIP121, TIP126 TIP122, TIP127	$I_{CBO}$	- - -	0.2 0.2 0.2	mAdc
Emitter Cutoff Current ( $V_{BE} = 5.0\text{ Vdc}$ , $I_C = 0$ )		$I_{EBO}$	-	2.0	mAdc

ON CHARACTERISTICS (Note 2)

DC Current Gain ( $I_C = 0.5\text{ Adc}$ , $V_{CE} = 3.0\text{ Vdc}$ ) ( $I_C = 3.0\text{ Adc}$ , $V_{CE} = 3.0\text{ Vdc}$ )		$h_{FE}$	1000 1000	- -	-
Collector-Emitter Saturation Voltage ( $I_C = 3.0\text{ Adc}$ , $I_B = 12\text{ mAdc}$ ) ( $I_C = 5.0\text{ Adc}$ , $I_B = 20\text{ mAdc}$ )		$V_{CE(sat)}$	- -	2.0 4.0	Vdc
Base-Emitter On Voltage ( $I_C = 3.0\text{ Adc}$ , $V_{CE} = 3.0\text{ Vdc}$ )		$V_{BE(on)}$	-	2.5	Vdc

Anexo L: Hoja de especificaciones Transistor 2N3904

2N3903, 2N3904

ELECTRICAL CHARACTERISTICS ( $T_A = 25^\circ\text{C}$  unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
<b>OFF CHARACTERISTICS</b>				
Collector - Emitter Breakdown Voltage (Note 2) ( $I_C = 1.0\text{ mAdc}$ , $I_B = 0$ )	$V_{(BR)CEO}$	40	-	Vdc
Collector - Base Breakdown Voltage ( $I_C = 10\ \mu\text{Adc}$ , $I_E = 0$ )	$V_{(BR)CBO}$	60	-	Vdc
Emitter - Base Breakdown Voltage ( $I_E = 10\ \mu\text{Adc}$ , $I_C = 0$ )	$V_{(BR)EBO}$	6.0	-	Vdc
Base Cutoff Current ( $V_{CE} = 30\text{ Vdc}$ , $V_{EB} = 3.0\text{ Vdc}$ )	$I_{BL}$	-	50	nAdc
Collector Cutoff Current ( $V_{CE} = 30\text{ Vdc}$ , $V_{EB} = 3.0\text{ Vdc}$ )	$I_{CEX}$	-	50	nAdc
<b>ON CHARACTERISTICS</b>				
DC Current Gain (Note 2) ( $I_C = 0.1\text{ mAdc}$ , $V_{CE} = 1.0\text{ Vdc}$ )	2N3903 2N3904	$h_{FE}$	20 40	- -
( $I_C = 1.0\text{ mAdc}$ , $V_{CE} = 1.0\text{ Vdc}$ )	2N3903 2N3904		35 70	- -
( $I_C = 10\text{ mAdc}$ , $V_{CE} = 1.0\text{ Vdc}$ )	2N3903 2N3904		50 100	150 300
( $I_C = 50\text{ mAdc}$ , $V_{CE} = 1.0\text{ Vdc}$ )	2N3903 2N3904		30 60	- -
( $I_C = 100\text{ mAdc}$ , $V_{CE} = 1.0\text{ Vdc}$ )	2N3903 2N3904		15 30	- -
Collector - Emitter Saturation Voltage (Note 2) ( $I_C = 10\text{ mAdc}$ , $I_B = 1.0\text{ mAdc}$ ) ( $I_C = 50\text{ mAdc}$ , $I_B = 5.0\text{ mAdc}$ )		$V_{CE(sat)}$	- -	0.2 0.3
Base - Emitter Saturation Voltage (Note 2) ( $I_C = 10\text{ mAdc}$ , $I_B = 1.0\text{ mAdc}$ ) ( $I_C = 50\text{ mAdc}$ , $I_B = 5.0\text{ mAdc}$ )		$V_{BE(sat)}$	0.65 -	0.85 0.95
<b>SMALL-SIGNAL CHARACTERISTICS</b>				
Current - Gain - Bandwidth Product ( $I_C = 10\text{ mAdc}$ , $V_{CE} = 20\text{ Vdc}$ , $f = 100\text{ MHz}$ )	2N3903 2N3904	$f_T$	250 300	- -
Output Capacitance ( $V_{CB} = 5.0\text{ Vdc}$ , $I_E = 0$ , $f = 1.0\text{ MHz}$ )		$C_{obo}$	-	4.0
Input Capacitance ( $V_{EB} = 0.5\text{ Vdc}$ , $I_C = 0$ , $f = 1.0\text{ MHz}$ )		$C_{ibo}$	-	8.0
Input Impedance ( $I_C = 1.0\text{ mAdc}$ , $V_{CE} = 10\text{ Vdc}$ , $f = 1.0\text{ kHz}$ )	2N3903 2N3904	$h_{ie}$	1.0 1.0	8.0 10
Voltage Feedback Ratio ( $I_C = 1.0\text{ mAdc}$ , $V_{CE} = 10\text{ Vdc}$ , $f = 1.0\text{ kHz}$ )	2N3903 2N3904	$h_{re}$	0.1 0.5	5.0 8.0
Small-Signal Current Gain ( $I_C = 1.0\text{ mAdc}$ , $V_{CE} = 10\text{ Vdc}$ , $f = 1.0\text{ kHz}$ )	2N3903 2N3904	$h_{fe}$	50 100	200 400
Output Admittance ( $I_C = 1.0\text{ mAdc}$ , $V_{CE} = 10\text{ Vdc}$ , $f = 1.0\text{ kHz}$ )		$h_{oe}$	1.0	40
Noise Figure ( $I_C = 100\ \mu\text{Adc}$ , $V_{CE} = 5.0\text{ Vdc}$ , $R_B = 1.0\text{ k}\Omega$ , $f = 1.0\text{ kHz}$ )	2N3903 2N3904	NF	- -	6.0 5.0
<b>SWITCHING CHARACTERISTICS</b>				
Delay Time	$(V_{CC} = 3.0\text{ Vdc}$ , $V_{BE} = 0.5\text{ Vdc}$ , $I_C = 10\text{ mAdc}$ , $I_{B1} = 1.0\text{ mAdc}$ )	$t_d$	-	35
Rise Time		$t_r$	-	35
Storage Time	$(V_{CC} = 3.0\text{ Vdc}$ , $I_C = 10\text{ mAdc}$ , $I_{B1} = I_{B2} = 1.0\text{ mAdc}$ )	2N3903	$t_s$	-
Fall Time		2N3904		175 200
			$t_f$	-

Anexo M: Hoja de especificaciones Optoacoplador H11AA1

**H11AA1**

Vishay Semiconductors Optocoupler, Phototransistor Output,  
AC Input, with Base Connection



<b>ABSOLUTE MAXIMUM RATINGS</b> ( $T_{amb} = 25\text{ }^{\circ}\text{C}$ , unless otherwise specified)				
PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
<b>INPUT</b>				
Forward continuous current		$I_F$	$\pm 60$	mA
Power dissipation		$P_{dex}$	100	mW
Derate linearly from 25 °C			1.3	mW/°C
<b>OUTPUT</b>				
Power dissipation		$P_{dex}$	200	mW
Derate linearly from 25 °C			2.6	mW/°C
Collector emitter breakdown voltage		$BV_{CEO}$	30	V
Emitter base breakdown voltage		$BV_{EB0}$	5	V
Collector base breakdown voltage		$BV_{CB0}$	70	V
<b>COUPLER</b>				
Isolation test voltage (RMS)	Between emitter and detector, referred to standard climate 23 °C/50% RH, DIN 50014	$V_{ISO}$	5300	$V_{RMS}$
Creepage distance			$\geq 7$	mm
Clearance distance			$\geq 7$	mm
Comparative tracking index	per DIN IEC 112/VDE 0903, part 1	CTI	175	
Isolation resistance	$V_{IO} = 500\text{ V}$ , $T_{amb} = 25\text{ }^{\circ}\text{C}$	$R_{IO}$	$\geq 10^{12}$	$\Omega$
	$V_{IO} = 500\text{ V}$ , $T_{amb} = 100\text{ }^{\circ}\text{C}$	$R_{IO}$	$\geq 10^{11}$	$\Omega$
Storage temperature range		$T_{stg}$	- 55 to + 150	°C
Operating temperature range		$T_{amb}$	- 55 to + 100	°C
Lead soldering time at 260 °C		$T_{sld}$	10	s

**Note**

- Stresses in excess of the absolute maximum ratings can cause permanent damage to the device. Functional operation of the device is not implied at these or any other conditions in excess of those given in the operational sections of this document. Exposure to absolute maximum ratings for extended periods of the time can adversely affect reliability.

<b>ELECTRICAL CHARACTERISTICS</b> ( $T_{amb} = 25\text{ }^{\circ}\text{C}$ , unless otherwise specified)						
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
<b>INPUT</b>						
Forward voltage	$I_F = \pm 10\text{ mA}$	$V_F$		1.2	1.5	V
<b>OUTPUT</b>						
Collector emitter breakdown voltage	$I_C = 1\text{ mA}$	$BV_{CEO}$	30			V
Emitter base breakdown voltage	$I_C = 100\text{ }\mu\text{A}$	$BV_{EB0}$	5			V
Collector base breakdown voltage	$I_C = 100\text{ }\mu\text{A}$	$BV_{CB0}$	70			V
Collector emitter leakage current	$V_{CE} = 10\text{ V}$	$I_{CEO}$		5	100	nA
<b>COUPLER</b>						
Collector emitter saturation voltage	$I_F = \pm 10\text{ mA}$ , $I_C = 0.5\text{ mA}$	$V_{CEsat}$			0.4	V

**Note**

- Minimum and maximum values were tested requirements. Typical values are characteristics of the device and are the result of engineering evaluations. Typical values are for information only and are not part of the testing requirements.

<b>CURRENT TRANSFER RATIO</b> ( $T_{amb} = 25\text{ }^{\circ}\text{C}$ , unless otherwise specified)						
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
DC current transfer ratio	$I_F = \pm 10\text{ mA}$ , $V_{CE} = 10\text{ V}$	$CTR_{DC}$	20			%
Symmetry (CTR at + 10 mA)/(CTR at - 10 mA)			0.33	1	3	

## Anexo N: Código para identificación de sistemas Matlab – Arduino

```
1 /*UNIVERSIDAD TÉCNICA DEL NORTE
2  * FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
3  * INGENIERÍA EN MANTENIMIENTO ELÉCTRICO
4  * AUTOR: ROTMAN QUIMBITA
5  * TEMA: INTERFAZ MATLAB - ARDUINO PARA IDENTIFICACIÓN DE SISTEMAS
6  */
7 #include <max6675.h>
8 #include <avr/io.h>
9 #include <avr/interrupt.h>
10 #define pasoCero 2
11 #define pulso 4
12 int disparo1 = 30;
13 int disparo2 = 31;
14 int disparo3 = 34;
15 int disparo4 = 35;
16 //Variables para almacenar datos a enviar y recibir en matlab:
17 int controlValue = 0;
18 int datos=0;
19 float processValue = 0.0;
20 float temperatura = 0.0;
21 //Configura Pines para módulo Max6675:
22 int ktcSO = 50;
23 int ktcCS = 53;
24 int ktcCLK = 52;
25 //Variables para tiempo de muestreo:
26 unsigned long muestreoActual = 0;
27 unsigned long muestreoAnterior = 0;
28 unsigned long deltaMuestreo = 0;
29 int tiempoMuestreo = 400;
30 //Comunicacion serial:
31 String dataString = "";
32 bool dataComplete = false;
33 //Realiza un scalamiento de valores parecida a la función map pero con valores decimales
34 float scaling(float x, float in_min, float in_max, float out_min, float out_max)
35 {
36     return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
37 }
38 //Inicializa una comunicación MAX6675 con los pines seleccionados
39 MAX6675 ktc(ktcCLK, ktcCS, ktcSO);
40 void setup(){
41     Serial.begin(9600);
42     pinMode(pasoCero, INPUT_PULLUP);
43     pinMode(disparo1, OUTPUT);
44     pinMode(disparo2, OUTPUT);
45     pinMode(disparo3, OUTPUT);
46     pinMode(disparo4, OUTPUT);
47     digitalWrite(disparo1,LOW);
48     digitalWrite(disparo2,LOW);
49     digitalWrite(disparo3,LOW);
50     digitalWrite(disparo4,LOW);
51     // Configuración Timer1
52     OCR1A = 100;
53     TIMSK1 = 0x03;
54     TCCR1A = 0x00;
55     TCCR1B = 0x00;
56     // Interrupción de cruce por cero
57     attachInterrupt(0,zeroCrossingInterrupt, RISING);
58 }
59 //Rutina de interrupcion:
60 void zeroCrossingInterrupt(){
61     TCCR1B=0x04;
62     TCNT1 = 0;
63 }
64 //Configuración de Timer 1:
65 ISR(TIMER1_COMPA_vect){
66     digitalWrite(disparo1,HIGH);
67     digitalWrite(disparo2,HIGH);
```

```

68 digitalWrite(disparo3,HIGH);
69 digitalWrite(disparo4,HIGH);
70 TCNT1 = 65536-pulso;
71 }
72 ISR(TIMER1_OVF_vect){
73 digitalWrite(disparo1,LOW);
74 digitalWrite(disparo2,LOW);
75 digitalWrite(disparo3,LOW);
76 digitalWrite(disparo4,LOW);
77 TCCR1B = 0x00;
78 }
79
80 void loop(){
81   if (dataComplete){
82     datos = dataString.toInt();
83     controlValue = scaling(datos,0,100,465,290);
84     Serial.println(processValue);
85     dataString = "";
86     dataComplete = false;
87   }
88   muestreoActual = millis();
89   deltaMuestreo = (double)muestreoActual - muestreoAnterior;
90   if(deltaMuestreo >= tiempoMuestreo){
91     muestreoAnterior += tiempoMuestreo;
92     temperatura = ktc.readCelsius();
93     processValue=scaling(temperatura,20,230,0,100);
94     OCR1A = controlValue;
95   }
96 }
97
98 void serialEvent(){
99   while (Serial.available()){
100     char inChar = (char)Serial.read();
101     dataString += inChar;
102
103     if (inChar == '\n'){
104       dataComplete = true;
105     }
106 }

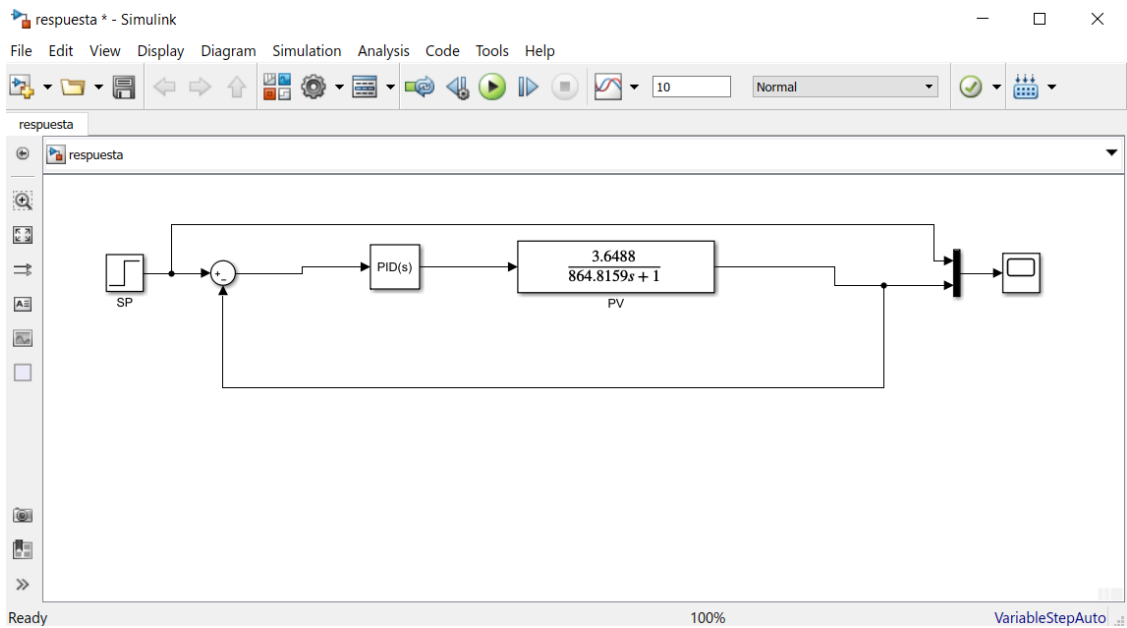
```

```

1  %*UNIVERSIDAD TÉCNICA DEL NORTE
2  %* FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
3  %* INGENIERÍA EN MANTENIMIENTO ELÉCTRICO
4  %* AUTOR: ROTMAN QUIMBITA
5  %* CÓDIGO MATLAB PARA TOMA DE DATOS A TRAVÉS DE ARDUINO
6  %/
7  - clear
8  - close all
9  - clc
10 - tf = 1800;
11 - ts = 0.1;
12 - t = 0:ts:tf;
13 - Q = length(t);
14 - controlValue = zeros(1,Q);
15 - processValue = zeros(1,Q);
16 - scene = figure;
17 - tam = get(0,'ScreenSize');
18 - set(scene,'position',tam);
19 - plot(t,processValue,'b','LineWidth',2);hold on;
20 - plot(t,controlValue,'r','LineWidth',2); legend('PV','CV');grid on;
21 - xlabel('Tiempo (s)');
22 - ylabel('Amplitud(%)');
23 - COM = 'COM7';
24 - delete(instrfind({'Port'},{COM}));
25 - arduinoSerial = serial(COM,'Baudrate',9600);
26 - fopen(arduinoSerial);
27 - pause(3);
28 - for k=1:Q
29 -     tic
30 -
31 -     if t(k)>=15
32 -         controlValue(k)=25;
33 -         end
34 -         fprintf(arduinoSerial,'%s\n',num2str(controlValue(k)),'sync');
35 -         processValue(k) = fscanf(arduinoSerial,'%f\n');
36 -         clf
37 -         plot(t(1:k),processValue(1:k),'b','LineWidth',2);hold on;
38 -         plot(t(1:k),controlValue(1:k),'r','LineWidth',2); legend('PV','CV');grid on;
39 -         xlabel('Tiempo (s)');
40 -         ylabel('Temperatura(C)');
41 -         drawnow
42 -         while toc<ts
43 -             end
44 -         end
45 -         fprintf(arduinoSerial,'%s\n','0','sync');
46 -         fclose(arduinoSerial);
47 -         save('Data','controlValue','processValue');

```

## Anexo O: Programa en Simulink para ver la respuesta del sistema con control PID



## Anexo P: Código de programación del sistema automatizado

```
1 /*UNIVERSIDAD TÉCNICA DEL NORTE
2  * FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
3  * INGENIERÍA EN MANTENIMIENTO ELÉCTRICO
4  * AUTOR: ROTMAN QUIMBITA
5  * AUTOMATIZACIÓN DEL EQUIPO DE SOLDADURA
6  */
7 //Declaración de variables y librerías/////
8 #include <max6675.h>
9 #include <avr/io.h>
10 #include <avr/interrupt.h>
11 #define pasoCero 2 //interrupción (detector cruce por cero)
12 #define pulsos 4 //ancho de pulso de disparo (cuenta)
13 word salidaPwm = 9; //regular velocidad ventilador PWM
14 int disparo1 = 30; //señal de disparo al optoacoplador moc3021
15 int disparo2 = 31;
16 int disparo3 = 34;
17 int disparo4 = 35;
18 int disparo5 = 26;
19 int ventilador_p = 41; //ventilador enfriamiento proceso
20 int ventilador_e = 45; //ventilador enfriamiento placas
21 ///Variables para módulo max6675///
22 int ktcSO = 50;
23 int ktcCS1 = 53;
24 int ktcCS2 = 48;
25 int ktcCS3 = 46;
26 int ktcCLK = 52;
27 ///Inicializa comunicación MAX6675 con los pines seleccionados
28 MAX6675 ktc1(ktcCLK, ktcCS1, ktcSO);
29 MAX6675 ktc2(ktcCLK, ktcCS2, ktcSO);
30 MAX6675 ktc3(ktcCLK, ktcCS3, ktcSO);
31 ///Variables para tiempo de muestreo///
32 unsigned long muestreoActual, muestreoAnterior, deltaMuestreo;
33 unsigned long tiempoAnterior, Tiempo, deltaTiempo =0;
34 unsigned long tiempoMuestreo = 400, tiempoTranscurrido=1000; //tiempo de muestreo en milisegundos
```



```

35 //////////////////////////////////////////////////VARIABLES COMUNICACION SERIAL////////////////////////////////////
36 String inputString = ""; //datos de tipo string
37 bool stringComplete = false; //bandera para verificar que el dato a llegado de manera correcta
38 int dato = 0; //variable que almacena el dato
39 //////////////////////////////////////////////////
40 //////Variables para control PID/////
41 int controlValue = 0;
42 float processValue[3]; //Y(0)
43 float filtroSensor = processValue[0]; //S(0)=Y(0)
44 float alpha=0.4;//Coeficiente de filtrado 0 - 1
45 float error = 0.0;//almacena el error actual
46 float error_anterior = 0;//error anterior para la derivada
47 float error_pasado = 0.0;//suma del error para la parte integral
48 int outMin, outMax = 0;
49 float Kp, Ti, Td = 0.0;
50 float PTerm, ITerm, DTerm = 0.0;
51 int setpoint=60;//inicia proceso con primer valor de referencia
52 int contador = 0;
53 int i=465;//valor que define el ángulo de disparo para calefactores
54 int j=0;//valor que define el ángulo de disparo para ventilador convección
55
56 void setup(){
57   Serial.begin(9600); //Velocidad del puerto serial
58   pinMode(pasoCero, INPUT_PULLUP); //zero cross detect
59   //digitalWrite(pasoCero, HIGH); //enable pull-up resistor
60   pinMode(disparo1, OUTPUT); //TRIAC gate control
61   pinMode(disparo2, OUTPUT); //TRIAC gate control
62   pinMode(disparo3, OUTPUT);
63   pinMode(disparo4, OUTPUT);
64   pinMode(disparo5, OUTPUT);
65   pinMode(ventilador_p, OUTPUT);
66   pinMode(ventilador_e, OUTPUT);
67   digitalWrite(ventilador_p, LOW);
68   digitalWrite(ventilador_e, LOW);
69
70   digitalWrite(disparo1, LOW);
71   digitalWrite(disparo2, LOW);
72   digitalWrite(disparo3, LOW);
73   digitalWrite(disparo4, LOW);
74   digitalWrite(disparo5, LOW);
75   outMin = 465;
76   outMax = 290;
77   ///Constantes de sintonia por ziegler y nichols
78   Kp = 69.1853*6;
79   Ti= 30;
80   Td = 7.5;
81   // configuracion Timer1(mirar datasheet ATMEGA 2560)
82   OCR1A = 100; //inicializa el comparador (256 = 100 en hexadecimal)
83   TIMSK1 = 0x03; //habilita el comparador A y las interrupciones de desbordamiento
84   TCCR1A = 0x00; //registros de control del temporizador configurados para
85   TCCR1B = 0x00; //operacion normal, temporizador desactivado
86   //-----Configurar Timer 3 para ventilador conveccion-----//
87   OCR3A = 100;
88   TIMSK3 = 0x03; //Registro de configuración de interrupción
89   TCCR3A = 0x00;
90   TCCR3B = 0x00;
91   //-----//
92   // configurar interrupción de cruce por cero en el pin 2
93   attachInterrupt(digitalPinToInterrupt(2), zeroCrossingInterrupt, RISING);
94   //Interrupcion 0 es en el pin 2
95   //Llama a la funcion zeroCrossingInterrupt cada que pasa de LOW a HIGH(flanco de subida)
96   //----Configuracion PWM ventilador----//
97   pinMode(salidaPwm, OUTPUT);
98   Fwm25Khz();
99   //-----//
100 }
101 //----Rutinas de interrupción-----//
102 void zeroCrossingInterrupt(){ //funcion cuando detecta cruce por cero
103   TCCR1B=0x04; //inicio temporizador 1 con división por 256 entradas (prescaler)

```

```

103 //Para controlar la velocidad del incremento del temporizador utilizamos el prescaler
104 //el cual establece la velocidad de temporizacion con la siguiente ecuacion:
105 //(velocidad del temporizador (Hz))=(velocidad del reloj Arduino(16MHz))/prescaler
106 //El prescaler puede ser igual a 1,8,64,256,1024
107 TCNT1 = 0; //Registro contador, reinicia el temporizador (cuenta desde cero)
108 TCCR3B=0x04; //inicio temporizador 3 con división por 256 entradas (prescaler)
109 TCNT3 = 0;
110 }
111 ISR(TIMER1_COMPA_vect){ //comparador1 (ejecuta la interrupcion)
112 digitalWrite(disparo1,HIGH); //configurar la puerta TRIAC a alto
113 digitalWrite(disparo2,HIGH);
114 digitalWrite(disparo3,HIGH);
115 digitalWrite(disparo4,HIGH);
116 TCNT1 = 65536-pulsos; //ancho de pulso de disparo
117 //Timer 1 es un temporizador de 16 bits, lo que significa que puede almacenar un valor de contador
118 //maximo de 65536. Una vez que un contador alcanza su maximo valor, este volvera a cero.(Esto se
119 //llama desbordamiento)
120 }
121 ISR(TIMER1_OVF_vect){ //temporizador1 desbordamiento
122 digitalWrite(disparo1,LOW); //apaga la puerta TRIAC
123 digitalWrite(disparo2,LOW);
124 digitalWrite(disparo3,LOW);
125 digitalWrite(disparo4,LOW);
126 TCCR1B = 0x00; //desactiva temporizador deteniendo disparadores involuntarios
127 }
128 ISR(TIMER3_COMPA_vect){ //comparador3
129 digitalWrite(disparo5,HIGH); //configurar la puerta TRIAC a alto
130 TCNT3 = 65536-pulsos; //ancho de pulso de disparo
131 }
132 ISR(TIMER3_OVF_vect){ //temporizador3 desbordamiento
133 digitalWrite(disparo5,LOW); //apaga la puerta TRIAC
134 TCCR3B = 0x00;
135 }

136 //-----//
137
138 void loop(){
139 //----DATOS RECIBIDOS-----//
140 if (stringComplete) //Se verifica que un dato a llegado correctamente
141 {
142 dato = inputString.toInt(); // si es verdadero convierte el dato de tipo string en dato de tipo entero
143 inputString = ""; //reinicia las variables de la
144 stringComplete = false; //comunicacion serial
145 }
146 if (dato == 1){
147 digitalWrite(ventilador_e, HIGH); //Enciende ventilador enfriamiento placas
148 //-----Ejecución control PID-----//
149 muestreoActual = millis(); //Guarda el valor del tiempo antes del ciclo currentMillis
150 deltaMuestreo = (double)muestreoActual - muestreoAnterior;
151 if(deltaMuestreo >= tiempoMuestreo){
152 processValue [0] = ktc1.readCelsius();
153 filtroSensor = alpha*processValue[0]+(1.0-alpha)*filtroSensor; //Filtro EMA reducción de ruido en sensor
154 processValue [1] = ktc2.readCelsius();
155 processValue [2] = ktc3.readCelsius();
156 if((contador >=0)&&(contador <= 40)){
157 setpoint=60;
158 digitalWrite(disparo5,LOW);
159 j = 0; //sin convección etapa calentamiento
160 }
161 else if((contador >40)&&(contador <= 70)){
162 setpoint = 90;
163 digitalWrite(disparo5,LOW);
164 j = 0; //sin convección etapa calentamiento
165 }
166 else if((contador >70)&&(contador <= 100)){
167 setpoint = 120;
168 digitalWrite(disparo5,HIGH);
169 j = 275; //convección baja

```

```

170     }
171     else if((contador >100)&&(contador <= 130)){
172         setpoint = 150;
173         digitalWrite(disparo5,HIGH);
174         j = 250;//convección intermedia
175     }
176     else if((contador >130)&&(contador <= 230)){
177         setpoint = 185;
178         digitalWrite(disparo5,HIGH);
179         j = 225;//convección moderada
180     }
181     else if((contador >230)&&(contador <= 285)){
182         setpoint = 215;
183         digitalWrite(disparo5,HIGH);
184         j = 200;//convección alta
185     }
186     error = setpoint - filtroSensor;//Calculamos el ERROR pid
187     error_pasado += error*(tiempoMuestreo/1000);//calculo de aproximación de area bajo la curva
188     int error_D = (error - error_anterior)/(tiempoMuestreo/1000);
189     if(error > 30) ITerm = 0;//la constante integral solo afectará los errores por debajo de 30°C
190     //---Acciones de control PID---//
191     PTerm = Kp * error;//Acción Proporcional
192     ITerm = (Kp/Ti) * (error_pasado);//Acción Integral
193     DTerm = Kp*Td*error_D;//Acción Derivativa
194     //-----//
195     controlValue = PTerm + ITerm + DTerm;//envia señal de control PID
196     //Saturacion de la accion de control a un tope maximo
197     if(controlValue <= 0) controlValue = 0; else if(controlValue >= 7700) controlValue = 7700;
198     error_anterior = error; //Guarda el error anterior.
199     i=map(controlValue,0,7700,outMin,outMax);
200     OCR1A = i; //Registro comparador de salida (control de potencia en lámparas halógenas)
201     //j=200;
202     OCR3A = j; //control de potencia ventilador de convección
203     //Apaga Triacs por completo:

204     if (contador >= 285){
205         digitalWrite(disparo1,LOW);
206         digitalWrite(disparo2,LOW);
207         digitalWrite(disparo3,LOW);
208         digitalWrite(disparo4,LOW);
209         digitalWrite(disparo5,LOW);
210         i = 0;
211         OCR1A = 0;
212         j = 0;
213         OCR3A = 0;
214         digitalWrite(ventilador_p, HIGH);//Enciende ventilador enfriamiento proceso
215         cicloPwm(65);//0 velocidad minima, 79 velocidad maxima
216         digitalWrite(ventilador_e, LOW); //Apaga ventilador enfriamiento placas
217     }
218     if (contador >= 600){
219         digitalWrite(ventilador_p, LOW);//Apaga ventilador enfriamiento proceso
220     }
221     muestreoAnterior = muestreoActual; //Actualización del tiempoAnterior
222     //Serial.print(processValue[0]);//envia datos del sensor a través del puerto serial
223     Serial.println(filtroSensor);
224     Serial.println(processValue[1]);
225     Serial.println(processValue[2]);
226
227 }
228 Tiempo = millis();
229 deltaTiempo = muestreoActual - tiempoAnterior;
230 if(deltaTiempo >= tiempoTranscurrido){
231     contador=contador+1;
232     //Serial.println(processValue[0]);
233     //Serial.println(processValue[1]);
234     //Serial.println(processValue[2]);
235     tiempoAnterior = Tiempo;
236 }
237 }

```

```

238 else{
239     digitalWrite(ventilador_p,LOW);
240     digitalWrite(ventilador_e, LOW);
241     digitalWrite(disparo1, LOW);
242     digitalWrite(disparo2, LOW);
243     digitalWrite(disparo3, LOW);
244     digitalWrite(disparo4, LOW);
245     digitalWrite(disparo5, LOW);
246     OCR1A = 0;
247     OCR3A = 0;
248 }
249 }
250
251 //CONFIGURACION DE TIMER 2//
252 void Pwm25Khz() {
253     TCCR2A = 0;// TC2 Registro de control A
254     TCCR2B = 0;// TC2 Registro de control B
255     TIMSK2 = 0;// TC2 Interrupción del registro de máscara
256     TIFR2 = 0; // TC2 Registro de bandera de interrupción
257     TCCR2A |= (1 << COM2B1) | (1 << WGM21) | (1 << WGM20);
258     TCCR2B |= (1 << WGM22) | (1 << CS21);// prescaler 8
259     OCR2A = 79;// valor de desbordamiento maximo (Hz)
260     OCR2B = 0;
261 }
262
263 void cicloPwm(byte ocrb) {
264     OCR2B = ocrb;// Ancho PWM (trabajo)
265 }
266
267 void serialEvent(){
268     while (Serial.available()){//concatenamos cada uno de los caracteres que recibimos del puerto serial
269         char inChar = (char)Serial.read(); // y almacenamos en inputString
270         inputString += inChar;
271         if (inChar == '\n'){ //si uno de los caracteres es un salto de linea
272             stringComplete = true;//levantamos la bandera para verificar que los datos han llegado de manera correcta
273         }
274     }
275 }

```

## Anexo Q: Código de programación de la interfaz gráfica

```
#!/UNIVERSIDAD TÉCNICA DEL NORTE
#* FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
#* INGENIERÍA EN MANTENIMIENTO ELÉCTRICO
#* AUTOR: ROTMAN QUIMBITA
#* INTERFAZ GRÁFICA PARA EL EQUIPO DE SOLDADURA
#
import serial #modulo para la comunicacion serial
import time #manejo de tiempos similar al delay en arduino
import collections
import matplotlib.pyplot as plt
import glob
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import matplotlib.animation as animation
import numpy as np
from tkinter import * #Incluimos la libreria para interfaz grafica HMI
#El Threading es una tecnica de programacion que permite que una aplicacion ejecute
#simultaneamente varias operaciones en el mismo espacio de proceso
#El objetivo es leer el puerto del arduino y simultaneamente graficar los datos en un GUI
from threading import Thread #Hilo
from tkinter import messagebox

#####VARIABLES#####
isRun = True # bandera para recibir datos
isTime = False
serialConnection = type('serial', (object,), {})()
thread = type('Thread', (object,), {})()

Samples = 1000 #Muestras que deseamos observar en la gráfica
data = []
numData = 3 #Número de sensores
sampleTime = 50 #Tiempo de muestreo 50
value=[] #vector para almacenar datos
totalTime = 0

#Funcion para la deteccion automatica del puerto conectado
def serial_ports():
    """ Lists serial port names
    :raises EnvironmentError:
        On unsupported or unknown platforms
    :returns:
        A list of the serial ports available on the system
    """
    if sys.platform.startswith('win'):
        ports = ['COM%s' % (i + 1) for i in range(256)]
    elif sys.platform.startswith('linux') or sys.platform.startswith('cygwin'):
        # this excludes your current terminal "/dev/tty"
        ports = glob.glob('/dev/tty[A-Za-z]*')
    elif sys.platform.startswith('darwin'):
        ports = glob.glob('/dev/tty.*')
    else:
        raise EnvironmentError('Unsupported platform')
    result = []
    for port in ports:
        try:
            s = serial.Serial(port)
            s.close()
            result.append(port)
        except (OSError, serial.SerialException):
            pass
    return result

def getData():
    global serialConnection
    baudRate = 9600 # Baudios
    try:
        serialConnection = serial.Serial(varPort.get(), baudRate) # Instanciar objeto Serial
    except:
        print('No se puede conectar al puerto')
        time.sleep(1) # dar tiempo para comenzar a recibir datos
    global totalTime
    totalTime = 0
    while (isRun): #leer datos
        global value
        start = time.time()
        if serialConnection.inWaiting() > 0:
            for i in range(numData):
                value[i] = float(serialConnection.readline().strip()) #leer sensor
                data[i].append(value[i]) #Guarda lectura en la última posición
        if isTime:
            end = time.time()
            totalTime +=(end - start)

def on_closing():
    if messagebox.askokcancel("Salir", "Salir del programa?"):
        global isRun
        isRun = False
        try:
            thread.join() #tiempo de espera, espera hasta que el hilo termine
        except:
            pass
        try:
            serialConnection.close() #cerrar el puero serial
        except:
            pass
```

```

        root.destroy()          #Cierra la ventana creada
        root.quit()           #finaliza este programa

def conectar():
    global thread
    thread = Thread(target=getData) # Crear objeto de la clase Thread
    thread.start() #Iniciar subprocesso

def iniciar():
    global isTime
    isTime = True #bandera para iniciar conteo del tiempo
    global totalTime
    totalTime = 0
    serialConnection.write(('1\n').encode()) #envia el dato con salto de linea a arduino y lo codificamos
                                           # para que lo envíe de manera correcta

def apagar():
    global isTime
    isTime = False
    serialConnection.write(('0\n').encode())

def plotData(self, Samples, lines):
    end = time.time();
    #Guarda lectura en la última posición
    lines.set_data(range(Samples), data[0]) # Dibujar nueva línea del dato en la posición del vector
    temperatur1.set("TEMP_PCB :"+str(value[0])) # Mostrar valor del sensor
    temperatura2.set("TEMP_CH1 :"+str(value[1])) # Mostrar valor del sensor
    temperatura3.set("TEMP_CH2 :"+str(value[2])) # Mostrar valor del sensor
    tiempo1.set("TIEMPO(s) : "+str(round(totalTime,2)))

for i in range(numData):
    #Contenedor tipo lista donde almacenamos la lectura del sensor
    data.append(collections.deque([0] * Samples, maxlen=Samples))
    value.append(0)

# Limites de los ejes
xmin = 0
xmax = Samples #Número de muestras
ymin = 0
ymax = 250 #Unidades del sensor

fig = plt.figure(facecolor='azure', figsize=(10,5))#Crea una nueva figura
ax = plt.axes(xlim=(xmin, xmax), ylim=(ymin, ymax))
plt.title("Lectura en Tiempo Real (Temperatura)") #Titulo de la figura
plt.xticks(range(0,1040, 40))
plt.yticks(range(0,260, 10))
plt.grid()

ax.set_xlabel("Muestras")
ax.set_ylabel("Temperatura °C")
lines = ax.plot([], [], 'b', color='navy')[0]#Grafica datos iniciales y
#retorna líneas que representan la gráfica
root = Tk()#Inicializar el root de Tkinter o ventana principal
root.protocol("WM_DELETE_WINDOW", on_closing) #Protocolo a usar al pulsar la X de cierre
root.title('INTERFAZ GRAFICA DE USUARIO') #Titulo de ventana

canvas = FigureCanvasTkAgg(fig, master=root)
canvas._tkcanvas.grid(row=0)

# Un frame es simplemente un widget que nos permite contener a otros elementos;
# y este frame a su vez puede estar dentro de una ventana principal
frame = Frame(root,width=900, height=40, bg="blue")
frame.grid(row=1,column=0, padx=10, pady=5)
frame.grid_propagate(False) #Congela tamaño del frame

temperatur1 = StringVar(root,"TEMP_PCB : 0.00")
labelTemp1 = Label(frame, textvariable = temperatur1,fg='gold4',font="Times 16 italic bold")
#labelTemp1.grid(row = 0, column = 0, padx = 20, pady=20)
labelTemp1.place(x=5, y=5)

temperatura2 = StringVar(root,"TEMP_CH1 : 0.00")
labelTemp2 = Label(frame, textvariable = temperatura2,fg='gold4',font="Times 16 italic bold")
#labelTemp2.grid(row = 0, column = 1, padx = 20, pady=20)
labelTemp2.place(x=210, y=5)

temperatura3 = StringVar(root,"TEMP_CH2 : 0.00")
labelTemp3 = Label(frame, textvariable = temperatura3,fg='gold4',font="Times 16 italic bold")
#labelTemp3.grid(row = 0, column = 2, padx = 20, pady=20)
labelTemp3.place(x=420, y=5)

```

```

tiempo1 = StringVar(root,"TIEMPO(s) : 0.00")
labelTiempo = Label(frame, textvariable = tiempo1,fg='orange red',font=("Verdana",16))
#labelTiempo.grid(row = 0,column = 3,padx = 20,pady=20)
labelTiempo.place(x=640, y=5)

#Creamos un frame dentro de la ventana principal
frame1 = Frame(root,width=800, height=50, bg="bisque")
frame1.grid(row=2,column=0, padx=10, pady=5)#lo integramos a la ventana principal
frame1.grid_propagate(False) #Congela tamaño del frame

ports = serial_ports()
varPort = StringVar(root)
#varPort.set(ports[0]) //selecciona el puerto detectado

#Cracion de un menu despegable para elegir el puerto a conectar
portSelect = OptionMenu(frame1, varPort, *ports)
#portSelect.grid(row=0, column=0, padx = 20, pady=20, ipadx = 20, ipady=5)
portSelect.place(x=10, y=10)

button1 = Button(frame1,text="CONECTAR",command=conectar, bg="LightCyan",font=("Times",16))
#button1.grid(row=0,column=1,padx=20,pady=20)
button1.place(x=150, y=10)

button2 = Button(frame1, text="INICIAR",command=iniciar, bg="LightGreen",font=("Times",16))
#button2.grid(row=0,column=2,padx=20,pady=20)
button2.place(x=380, y=5)

button3 = Button(frame1,text="APAGAR",command=apagar, bg="red",font=("Times",16))
#button3.grid(row=0,column=3,padx=20,pady=20)
button3.place(x=520, y=5)

anim = animation.FuncAnimation(fig,plotData, fargs=(Samples,lines), interval=sampleTime)

root.geometry('1100x650') #tamaño de ventana en pixeles
root.mainloop() #Inicializar GUI o ciclo principal

```