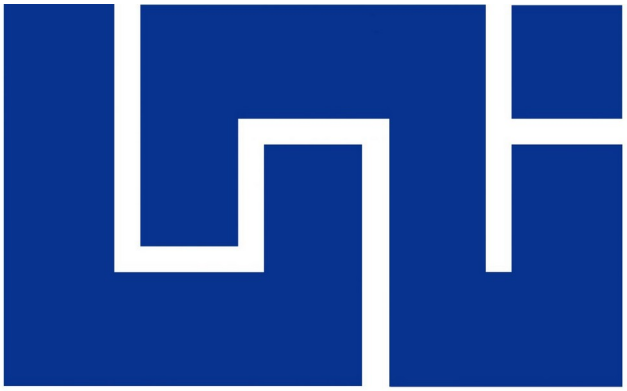
Universidad Nacional de Ingeniería



Recinto Universitario Simón Bolívar (UNI-RUSB)

Faculta de Electrotecnia y Computación (FEC)

Trabajo Monográfico para Optar al Titulo de

Ingeniero Electrónico

Prototipo de un Sistema de Control Electrónico para Horno Industrial a Gas Dirigido a la Industria Panificadora

Autor: Br. Royblan Sánchez Castillo

Tutor: Ing. Álvaro Gaitán

2020

Managua, Nicaragua

Resumen

Los centros de producción de alimentos son uno de tantos ambientes donde se implementan sistemas de control eléctricos y electrónicos (casi cualquier lugar donde allá producción), siempre se necesitara dar mantenimiento a la maquinaria de producción, en este caso en los centros de producción de pan hay diferentes tipos de maquinaria siendo utilizada, pero de la que nos enfocaremos sera el horno a gas, al momento de dar mantenimiento a dichos hornos es probable que no se encuentren repuestos originales y se deba recurrir a sistemas mas sencillos que el técnico pueda adaptar para poner en marcha el horno de nuevo, algunos métodos que eligen es aplicar módulos independiente para cada función, como también realizar circuitos con contactores, bi-metalicos y ampolletas de mercurio.

El sistema de control del horno a gas realiza las siguientes funciones: encendido de soplete a gas, desconexión de gas en caso de falla de encendido, termostato y cronometro, en el siguiente documento se desea explicar el diseño un prototipo de control de horno a gas que realice las funciones ya mencionadas anteriormente para poder tener una base de donde se pueda ofrecer un sistema que sea de carácter “genérico” y que cumpla con las funciones básicas e indispensable para el control de un horno a gas, con la finalidad de poder ser adquirido por técnicos que puedan implementarlo en el mantenimiento de dichos hornos.

Abstract

Food production centers are one of many environments where electrical and electronic control systems are implemented (almost anywhere where there is production), it will always be necessary to maintain the production machinery, in this case in the bread production centers there are different types of machinery being used, but the one we will focus on will be the gas oven, when maintaining these ovens it is probable that there are no original spare parts and you must resort to simpler systems that the technician can adapt to To start the furnace again, some methods they choose is to apply separate modules for each function, as well as to make circuits with contactors, bi-metals and mercury bulbs.

The gas oven control system performs the following functions: gas torch ignition, gas disconnection in case of misfire, thermostat and stopwatch, in the following document you want to explain the design of a gas oven control prototype to perform the functions already mentioned above in order to have a base where a system can be offered that is “generic” in nature and that fulfills the basic and indispensable functions for the control of a gas oven, in order to be able to be acquired by technicians who can implement it in the maintenance of said ovens.

Índice de contenido

Índice de figuras

Introducción

La industria panadera así como cualquier otra que se dedica al mercado alimenticio está en constante demanda de sus producto, dicho esto, se ve claro la importancia de la automatización de sus procesos de producción, este proceso lleva varias etapas de mucha importancia una de ellas es la parte de horneado del pan, antiguamente este horneado se realizaba utilizando como combustible la leña que debía ser vigilada por el operario para poder mantener un fuego constante, esto a cambiado al implementar quemadores y sopletes que utilizan otros combustibles como el gas y el diésel, estos ayudan ademas de mantener un fuego constante es mas fácil de controlar para encender y apagar.

Pero el trabajo del operario no termina ahí, si no que este debía también medir el tiempo de horneado en dependencia de la temperatura alcanzada en el horno con lo que este debía vigilar constantemente el progreso del horneado para luego tener que decidir bajo experiencia cuando el producto estaba listo, este proceso igual se debe realizar teniendo un horno con soplete o quemador, los fabricantes de hornos ya resuelven este problema agregando a sus productos controladores propios que funcionan bajo sus propios criterios para poder automatizar el proceso como utilizar electro-válvulas graduadas, tipos de sonda de temperatura (tipo K, RTD o termómetros de mercurio), combustible que se utiliza en el sistema entre otros, esto genera dificultad en su mantenimiento al momento de necesitar un repuesto como también en al momento de necesitar uno de estos sistemas para implementarlo en un horno que no este automatizado.

Por esta razón se propone el siguiente tema monográfico para el diseño de un prototipo de un controlador de horno de bajo coste y de buena disponibilidad de actuadores que sea sencillo para su uso por operadores sin conocimiento como también de técnicos que deseen implementarlo, con las ventaja de poder automatizar el proceso de horneado sin muchas complicaciones.

Este proyecto se realiza enfocándose en tres partes fundamentales que son un controlador PID digital para el control de temperatura, detectores de llama para el proceso de encendido de un horno a gas y como seguridad ante posibles fugas de gas y como ultima parte realizar una interfaz mediante un microcontrolador y sus respectivos periféricos (teclados, pantalla) para la configuración de los parámetros de temperatura y tiempo de horneado, el microcontrolador se encarga de leer los datos y de realizar las funciones necesarias mediante los actuadores para el control de horno.

Este documento contiene la teoría necesaria a utilizar en el diseño del controlador de horno a gas donde se implementara técnicas tales como el PWM que es la acción directa del controlador PID, las características de la llama que se han utilizado para su detección y los criterios a tener en cuenta al realizar mediante un controlador el encendido de un quemador a gas por esto motivamos la lectura completa del documento para así entender la automatización del horneado de pan.

Justificación

La importancia de la automatización de un horno en la industria de alimento conlleva varias ventajas, al realizar el proceso de horneado de forma mas rápida y precisa, crea un ahorro de los recursos utilizados en el horneado, tales como el combustible utilizado para calentar el producto (Gas, Diésel, Electricidad, entre otros), ademas de evitar errores en el proceso de horneado, de esta forma reduciendo los casos de producto derrochado, esto al controlar adecuadamente la temperatura y el tiempo de horneado, ademas ayuda a que el producto de pan pueda crecer a su punto máximo y dar una buena presentación y calidad.

Otra necesidad de crear este sistema es debido a los costos de importación y compatibilidad de este sistema como producto que se comercializa actualmente, de esta forma se pretende desarrollar de forma “genérica”, es decir, que se puede adaptar a cualquier horno industrial, con el fin de que pueda realizar todas las funciones básicas de un horno automatizado.

Otro punto es, los controladores que se comercializan poseen funciones más avanzadas, por lo que aumenta su coste, que al final no se utilizan o el obrero no sabe utilizar; además este control se podría implementar en un horno en el cual originalmente no era automatizado, pudiéndolo automatizar más fácilmente con el conocimiento necesario.

Los técnicos encargados de dar mantenimiento y reparación a los hornos ya exportados con sus propios sistemas de control encuentran dificultad al momento de dar mantenimiento a estos, debido a que necesitan ser capacitados para conocer y estudiar el sistema de control original del horno, por lo que deciden realizar el control desde cero, aquí intento ofrecer un producto sencillo y básico que permita realizar ese control iniciado desde cero mas eficiente, ya que el control elegido por estos técnicos es implementar módulos independiente para cada función (control de temperatura, modulo de encendido, entre otros) y en otros casos mediante contactores, bi-metalicos y ampolletas de mercurio.

Objetivos

Objetivo general

Diseñar un prototipo de un sistema de control digital con PID, para el control de temperatura, tiempo de horneado y encendido de un horno a gas.

Objetivos específicos

- Diseñar un circuito detector de llama mediante electrónica analógica que permita determinar la presencia de esta en los quemadores.
- Implementar todos los circuitos necesarios que permita la interacción del microcontrolador con los actuadores y el operador (Relés, Teclado, Pantalla LCD, etc).
- Diseñar controlador para el proceso de encendido del horno utilizando un circuito generador de chispa y el detector de llama.
- Diseñar un controlador PID digital, utilizando el método de Ziegler-Nichols, que permita regular la temperatura del horno controlando el tiempo de encendido del quemador mediante modulación PWM.
- Evaluar las pruebas de funcionamiento mediante la implementación del prototipo en un horno industrial a gas de 5 bandejas.

Capitulo 1: Marco teórico

Para este sistema se tiene que investigar las siguientes partes fundamentales: Medición de temperatura, detección de llama, controladores PID y control de los actuadores utilizando PWM, cada uno de estos temas de investigación se abordaran de manera global y resumida en el siguiente marco teórico, ademas se abordaran otros temas de interés de menor prioridad a los ya mencionados.

1.1 Control de un horno a gas

La producción pan a nivel industrial exige la implementación de sistemas automatizados para cubrir la demanda del mercado, por esta razón ya los hornos a demás de otras maquinarias utilizadas en la industria se pueden encontrar con sistemas de control digitales, estos son ofrecidos por diferentes marcas, algunos implementan sus sistemas propios mientras que otros hacen usos de dispositivos que encuentren a la mano en su país de origen, cuando estos sistemas originales fallan se ve la necesidad de buscar repuestos compatibles, algunas veces esto no sera del todo posible debido a los diseños del fabricante (un ejemplo es el uso de válvulas capases de graduar el flujo de gas en un soplete), por lo que se debe improvisar con el material disponible en el mercado local ya que la espera por envíos desde el extranjero puede mermar la producción.

Pero primero deberemos saber las partes que constituyen un horno ademas del funcionamiento básico del control de un horno, en la siguiente imagen se muestra el diagrama de tuberías e instrumentos (Diagrama DTI) de un horno a gas.(Norma ISA S5.1, 2009)

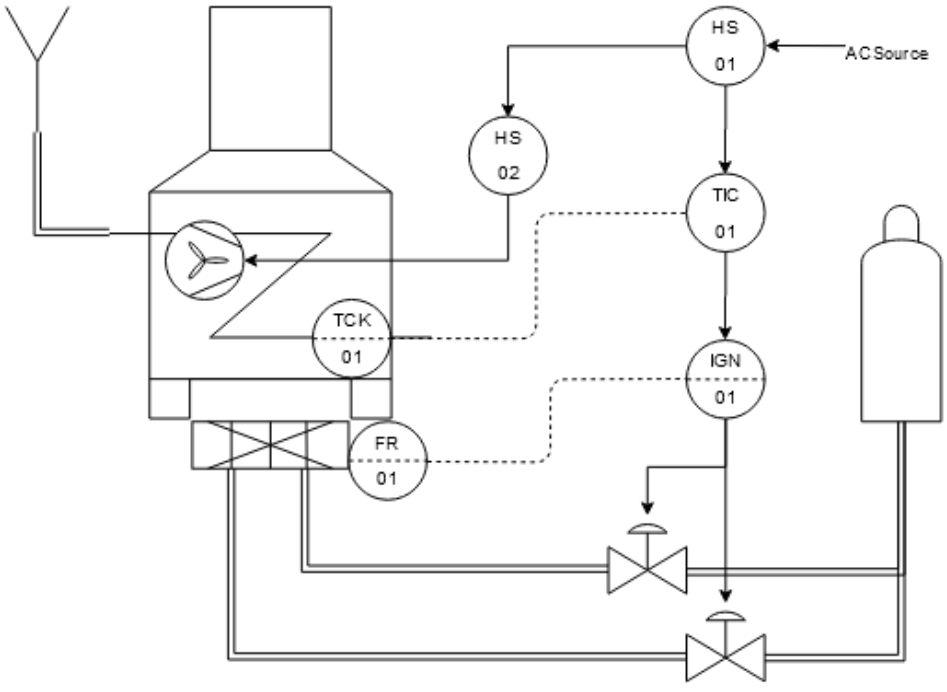


Fig.1 Diagrama DTI sencillo de un horno a gas.

El diagrama DTI muestra los dispositivos básicos utilizados en el control del horno, acá se lista cada uno.(Norma ISA S5.1, 2009)

- HS-01: Interruptor principal, puede ser un selector o un circuito “paro-marcha” con contactor.
- HS-02: Interruptor o selector de velocidad para la turbina interior.
- TIC-01: Controlador de temperatura, generalmente termostato.
- TCK-01: Termopar tipo K.
- IGN-01: Modulo de ignición.
- FR-01: Sonda detectora de llama.

El resto de simbología corresponde a la estructura del horno tales como el propio horno, el quemador o soplete, tubería de gas, electro-válvulas de gas, botella de gas, turbina o fan y chimenea, el diagrama muestra como se realiza la conexión del suministro de gas del tanque hasta el soplete pasando por las electro-válvulas, dichas electro-válvulas son controladas por el modulo de ignición, que a la ves es controlado por el termostato, siendo el termostato alimentado por el interruptor principal cuando se ponga en marcha el horno, cuando el termostato se pone en funcionamiento este activa al modulo de ignición cada ves que se necesite calor, para luego desconectar la alimentación del modulo cuando se ha llegado a la temperatura, apagando así el modulo y las electro-válvulas, por lo que cada ves que se necesite calor se tendrá que hacer el procedimiento de encendido del soplete.

Claramente que el diagrama explicado aquí carece de alarmas de seguridad en caso de tener problemas en el encendido, ni tampoco el uso de un controlador preciso como lo es un PID, ya que funciona como introducción. Para realizar la implementación del control por medio de termostatos y módulos, se aplica el circuito de la siguiente imagen que es un diagrama de mando y fuerza.

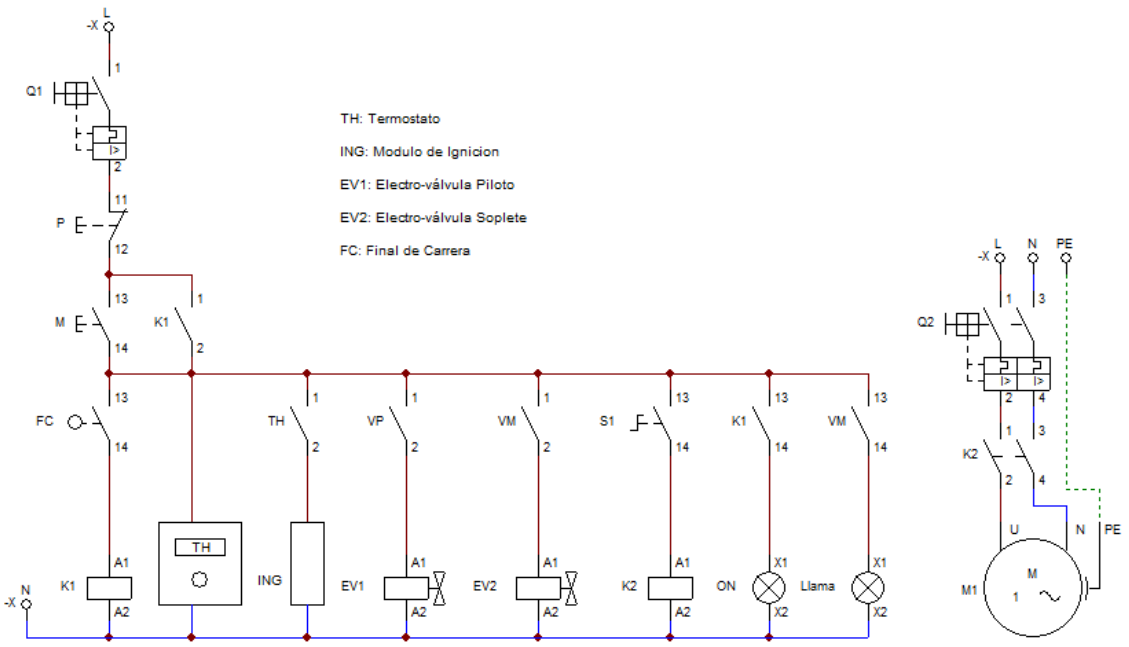


Fig.2 Circuito sencillo para el control de un horno a gas en un diagrama de mando y fuerza.

En este tipo de control se utiliza dos tipos de sondas para poder implementar como retroalimentación, una es el sensor de temperatura (Ej: termopar), el cual no es complicado de situar en el horno ya que solo debe estar asegurado dentro del horno para que reciba el calor que se le entrega al producto, en el caso de un horno para pan, este trae una turbina en su interior que realiza la tarea de hacer circular el aire en su interior, por lo que se puede suponer que la temperatura sera la misma en cada parte del horno, por lo que no sera necesario posicionar la sonda en un lugar en especifico, solo un lugar donde no pueda estorbar.

Otra sonda es la varilla detectora de llama, esta se debe posicionar al costado del piloto, algunos módulos tiene la característica que esta varilla ademas de ser detectora también es la salida de un generador de chispa utilizado para encender el piloto, por lo que el modulo tiene que estar aterrizado a la carcasa del horno y la varilla estar bien posicionada para cuando las chispas salten a la boca del piloto permita su encendido, a la ves que la llama puede tocar en todo momento a la varilla detectora, acá una imagen de su posicionamiento en el piloto.(Baso Guide, 2018)

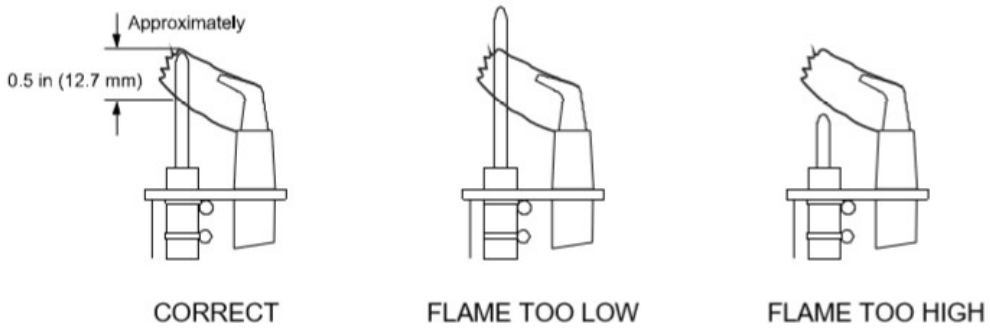


Fig.3 Posicionamiento de la varilla detectora.
Fuente: 24VAC Direct Spark Gas Ignition Control - Reference Guide.

El sistema de tuberías es muy sencillo ya que lo único que se hace es trasladar el gas desde el tanque al soplete siendo lo mas complejo el acople con las electro-válvulas y las boquillas del piloto y soplete, esto en si es un trabajo mas de mecánica o de plomero que eléctrico por lo que no lo abordaremos, la construcción del horno generalmente es de hierro en su interior recubierto de fibra de vidrio del tipo en fibras, después es recubierto por acero inoxidable para protegerlo del medio en que se trabaja ya que se producen alimentos, el soplete expulsa las llamas generadas en una cavidad situada en el piso del horno, el aire movilizado por la turbina se encargara de trasportar el calor obtenido en el piso hacia todas las partes internas del horno.

El horno también posee una chimenea donde se expulsaran los gases producidos por la combustión como en algunos casos también parte del aire caliente que se encuentra en el interior del horno, esto ultimo puede ser regulado dependiendo del gusto del operador.

1.2 Medición de temperatura

Para la medición de la temperatura, en la industria panificadora se hace uso comúnmente de dos tipos de sensores estos son el termopar (llamado también Termocupla) y las RTD.

Las RTD por sus siglas “Resistance Temperature Detection”, es un sensor de temperatura el cual su resistencia depende de la temperatura con un coeficiente positivo (PTC – Positive Temperature Coefficient), es decir, su resistencia especifica aumenta a medida que aumenta la temperatura; esto se debe a que al calentarse un metal sus átomos experimentan una agitación térmica, produciendo que los electrones que circulan por el, tengan mayor probabilidad de que choquen con sus átomos aumentando su resistencia al paso de estos electrones, a mayor temperatura, mayor agitación y mayor resistencia.(Park & Mackay, 2003)

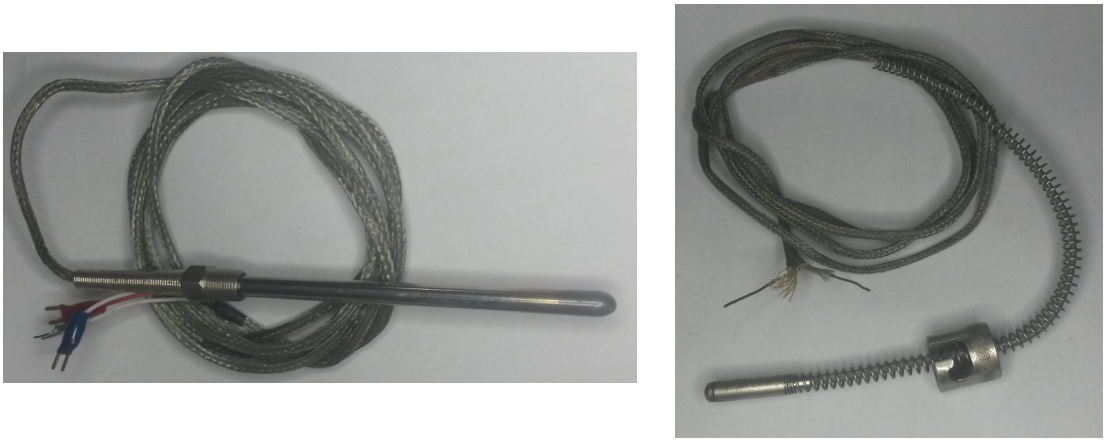


Fig.4 Comparación grafica entre la RTD PT100 (izquierda) y un termopar tipo K (derecha), se puede ver que que no tiene mucha diferencia en el exterior.

El sensor PT100 es una RTD que a $0^{O}C$ tiene una resistencia de 100Ω, esta resistencia aumenta a medida que aumenta la temperatura, este sensor no posee una respuesta resistiva totalmente lineal, pero sí muy parecida por lo que si se quiere una buena precisión se puede emplear tablas donde aparece su temperatura en base a su resistencia, para poder utilizar una RTD en la medición de temperatura no es necesario medir su resistencia directamente, usualmente se mide la caída de tensión producida por su resistencia o bien medir la corriente que permita que circule por él.

El sensor termopar se basa en el efecto, de la circulación de una pequeña corriente en un circuito cerrado formado por dos metales diferentes cuyas uniones (unión de medida o caliente y unión de referencia o fría) se mantienen a distinta temperatura, esta circulación de corriente obedece a dos efectos termoeléctricos combinados, el efecto Peltier (año 1834) que provoca la liberación o absorción de calor en la unión de los metales distintos cuando una corriente circula a través de la unión y el efecto Thomson (año 1854), que consiste en la liberación o absorción de calor cuando una corriente circula a través de un metal homogéneo en el que existe un gradiente temperaturas. (Creus, 2010)

Los termopares generan una diferencia de potencial muy pequeña entre sus terminales, del orden de los micro-voltios hasta los mili-voltios, esta tensión continua es proporcional a la temperatura medida siempre que haya una diferencia de temperaturas con la unión de referencia, estas características varían dependiendo del tipo de metales utilizados, por lo que existen diferentes tipos (Tipo E, J, K, N, R, y mas) siendo las mas utilizadas en hornos para pan las tipo J y K.(Creus, 2010)

Para la lectura de estos sensores de temperatura existen circuitos ideales para su implementación, en el caso de las RTD se puede aplicar el puente de Wheatstone en conectar cuatro resistencias entre si, dos en serie que a su ves en paralelo con la serie de las otras dos incluyendo la RTD y midiendo la tensión resultante en los puntos donde se unen ambas series, en el caso de los termopar, estos basta con amplificar la tensión obtenida para ser leída por cualquier instrumento y microcontrolador, ambos casos de sensores se utilizan tablas de referencia para obtener el valor de temperatura esto con el fin de obtener una mayor precisión en la lectura, ya que no tienen una respuesta totalmente proporcional. (Park & Mackay, 2003)

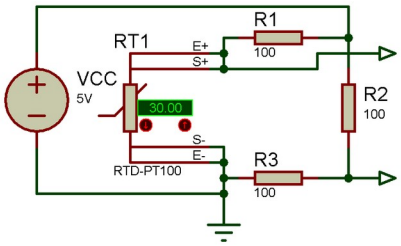


Fig.5 Ejemplo de la configuración del puente de Wheatstone, los dos terminales es donde se leerá señal.

Pero para mayor comodidad y precisión se pueden implementar módulos convertidores disponibles en el mercado tales como el MAX6675 para los termopares y el MAX31865 para las RTD, estos convertidores poseen varias ventajas en un solo integrado de ocho pines del tipo SOP8 incluyendo ADC de 12Bit para la lectura, compensación en la medición y comunicación simple mediante SPI, por lo que se obtendría mayor precisión que la que resultaría realizando los circuitos convertidores por uno mismo.

1.3 Detección de llama

La detección de llama en un horno o estufa industrial permite saber en qué momento se ha encendido el soplete o dado caso se apaga debido a factores externos, esto es importante porque permite la liberación del gas de forma segura.

Comercialmente existen varios sistemas que permiten la detección de la llama, en donde, se podrían mencionar tres (Creus, 2010):

Detección mediante el uso de sensores térmicos

Estos utilizan bi-metales, varillas de dilatación o expansión de metales líquidos como el mercurio, esto permite realizar un control relativamente aceptable, la desventaja de este tipo de sensor es que son lentos al momento de actuar cuando la llama ya se ha encendido o apagado, ya que,


Fig.6 Contacto NC activado de forma térmica mediante una sonda de mercurio.

mediante el calor producido en la llama, esta debe transferirse al dispositivo detector para que realiza un cambio físico en él, esto genera histéresis en la respuesta del sensor.(Park & Mackay, 2003)

Detectores basados en radiación de luz y calor

Estos sensores se basan en el espectro electromagnético que irradia la llama tales como luz visible (10%), luz infrarroja (90%) y ultravioleta (1%). Entre los inconvenientes que tienen estos tipos de sensores es que su mantenimiento debe hacerse con mayor frecuencia, ya que el hollín producido por la llama y las partículas de suciedad pueden cubrir la célula detectora del sensor, perdiendo así la detección de la llama, otro punto a tomar en cuenta, en el caso de los sensores basados en luz visible e infrarroja la luz proveniente de otras fuentes podrían perjudicar la detección de la llama, teniendo que adecuar el circuito de control para que pueda diferenciar la luz producida por la llama de las fuentes externas.(Creus, 2010)

Una ventaja de estos sensores es que pueden ser usados de forma mixta, ósea que pueden ser empleados en sopletes a gas o diésel.

Detectores de ionización

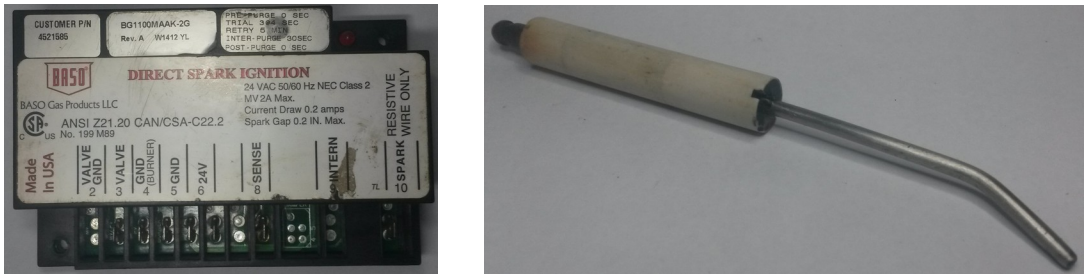


Fig.7 Izquierda: Modulo de encendido por ionización, derecha: electrodo utilizado para la detección de llama por ionización.

Los detectores por ionización se basan en hacer circular una pequeñísima corriente por la llama mediante la introducción de una varilla que es ionizada por la llama, este sistema intenta detectar la llama mediante una conexión de alta impedancia producido por la llama hacia el chasis del horno, dependiendo del tipo de varilla el mantenimiento no será necesario, ya que al utilizar varillas que mantengan sus propiedades sin importar la temperatura garantiza su funcionamiento correcto prolongado (Kanthal o Glober). Una desventaja de este sistema es el no poder utilizarse en llamas producidas con diésel ya que los compuestos que combustionan no permitan una buena ionización.(Creus, 2010)

1.4 Microcontrolador

El control de estos sistemas generalmente es realizado por un dispositivo llamado “Microcontrolador”. Este dispositivo recibe los datos provenientes de los sensores, y en base a estos datos toma decisiones previamente programadas por el diseñador del sistema.

En el mercado existe diferentes marcas y modelos de este dispositivo entre los que se pueden mencionar: Los PIC de Microchip Technology INC. y los Atmega de Atmel (Actualmente unificados). Esto permite elegir el microcontrolador con todos los aspectos necesarios (pines digitales, analógicos, entre otros) para nuestro sistema, aparte de esto, existen versiones de estos microcontroladores acondicionados para su uso en prototipos, donde traen todo lo necesario para que permita ser programados y acoplados a sensores, actuadores, entre otros, como por ejemplo Arduino utilizando microcontroladores Atmel y las placas pingüino utilizando PIC.

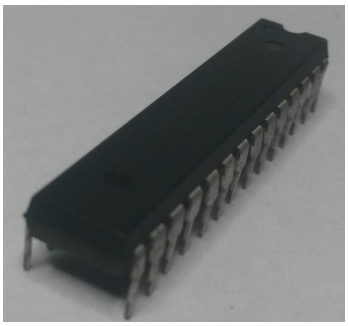


Fig.8 Microcontrolador Atmega8 de 28 pines.

De forma sencilla estos dispositivos son comparados con un ordenador, solo que pequeño y de recursos reducidos (En realidad solo lo justo), posee todo lo que constituye a un ordenador, consta de memoria RAM, memoria ROM, una ALU, bus de comunicación, entre otros mas, lo que diferencia a uno de otros microcontroladores es su capacidad y periféricos disponibles, estos pueden ser, puertos series, puertos SPI, ADC, salidas analógicas (PWM), salidas digitales, entre otros. Aunque se comparen con un ordenador, estos no son de solo enchufar y usar (plug and play) porque carecen de regulación en la alimentación, los puertos de los periféricos necesitan componentes adicionales para poder tomar y enviar datos, no poseen salidas visuales, algunos necesitan de un oscilador de cuarzo para establecer su frecuencia de trabajo, otros no la necesitan por contener uno interno, sabiendo esto es que se crean y comercializan versiones de estos microcontroladores en placas que incorporan todo estos circuitos adicionales para su uso mas cómodo, estas son llamadas placas de desarrollo.

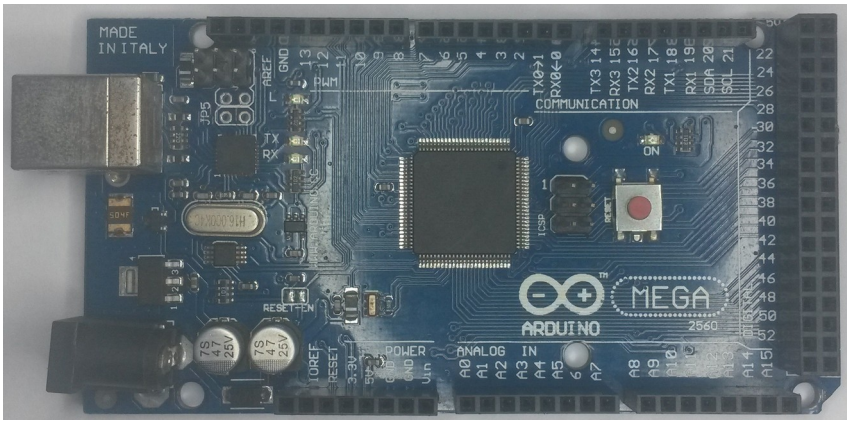


Fig.9 Placa de desarrollo Arduino Mega 2560, es basada en un microcontrolador ATmega2560.

Las placas de desarrollo como ya se dijo añaden al microcontrolador todo lo necesario para que este pueda funcionar por su cuenta (poseer cristal de cuarzo, resistencias pull-up y pull-down en sus salidas y entradas, LED's indicadores entre otros), ademas que le permita comunicarse con un ordenador (permitir la comunicación serie por un puerto USB), regular su alimentación y

realizar conexiones cómodas con sus pines de salida y entrada, todo esto sin necesidad de agregar alguna otra cosa a la placa de desarrollo, son muy versátiles cuando se trata de experimentar con prototipos y poder concretar ideas, pero no son lo mas indicado cuando se trata de implementar o comercializar un producto final.

Para elegir una placa de desarrollo es preciso saber las características del prototipo o proyecto que se pretende realizar, estas pueden ser, tipos de salidas y entrada, si se utilizara algún tipo de puertos de comunicación, cantidad de memoria del programa entre muchos mas, son variadas las características que se pueden citar, con esto uno podría centrarse en la comodidad y compatibilidad que ofrece el fabricante de la placa, estos podrían ser el lenguaje a utilizar en la programación, documentación y librerías, costos entre otros.

1.5 Interfaz de usuario

Para que el operador pueda interactuar con los sistemas de control sin la necesidad de tener muchos conocimientos sobre el funcionamiento de cualquier control industrial es necesario realizar algunos circuitos que permitan de manera intuitiva el envió de ordenes asía el microcontrolador, para esto se pueden utilizar dos periféricos muy importantes, uno para la visualización de datos y otro para el ingreso de datos, acá se menciona algunos de estos periféricos.

Pantalla LCD

Las pantallas LCD son dispositivos que se comercializan a un costo relativamente bajo, estas nos permiten imprimir caracteres alfanuméricos en ellos, por lo que es posible imprimir texto y datos numéricos, son una buena opción para toda clase de proyectos y claramente nunca faltan en prototipos, estos dispositivos poseen 16 pines de los cuales 3 son pines de instrucciones para su manejo, 8 son para el envió de de datos que sean caracteres del tipo ACSII, y el resto de pines para la configuración de la retroalimentación de la pantalla como su contraste, la siguiente imagen muestra una pantalla LCD de 16 columnas de caracteres por 2 filas (16X2).

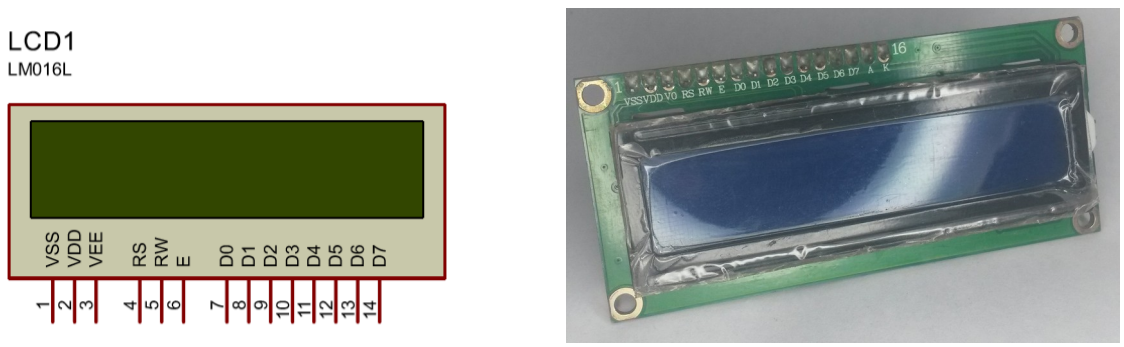


Fig.10 Pantalla LCD 16X2, se aprecia los pines de conexión.

Existen librerías ya destinadas para el manejo de este dispositivo, estas permiten utilizarlo con solo 4 pines de datos en ves de 8, reduciendo así un poco las salidas necesarias para su uso, en la siguiente tabla se puede aprecias los pines que dispone.

Tabla 1
Descripción de los pines de una LCD 16X2.

# Pin	Nombre	Descripción	# Pin	Nombre	Descripción
1	VSS	Negativo o Ground.	9	D2	Pin 3 datos.
2	VDD	+5V DC.	10	D3	Pin 4 datos.
3	V0	Voltaje regulador de contraste.	11	D4	Pin 5 datos.
4	RS	Selector de registro.	12	D5	Pin 6 datos.
5	RW	Lectura o escritura.	13	D6	Pin 7 datos.
6	E	Habilitar.	14	D7	Pin 8 datos.
7	D0	Pin 1 datos.	15	A	Positivo LED.
8	D1	Pin 2 datos.	16	K	Negativo LED.

Existen otras tecnologías que se pueden utilizar como periférico para la visualización de datos, tales como los display 7 segmentos, estos son mas económicos pero resultan mas limitados debido a que están diseñados principalmente para la visualización de números unicamente, hasta el punto de ser posible de no dispones de caracteres como ":" y ".", estos resultarían mas adecuado en el momento de la decisión de un producto final donde reducir costos conviene al crear una interfaz de visualización de datos mas simple pero a la ves mas complicada y menos intuitiva.

Teclado alfanumérico 4X4

Como periféricos de entradas de datos en la interfaz de usuario de un sistema de control industrial, lo primero que se tomaría en cuenta son el uso de botones que permitan la configuración del controlador, pero cuando nos planteamos la necesidad de ingresar números mediante botones nos damos cuenta que necesitaremos 10 botones, de manera convencional esto supondría tener que usar 10 pines del microcontrolador para poder leer sus estados mediante resistencia pull-down, y si se desea agregar otros botones que permitan funciones básicas como "inicio" o "cancelar" la cantidad de pines necesarios aumenta demasiado, debido a esto es necesario poder construir un teclado sin la necesidad de utilizar tantos pines del microcontrolador, acá es donde se encontramos una solución que ya se ha tomado como estándar en la construcción de prototipos y es el uso de teclado matriciales, este tipo de teclado ordena los botones en filas y columnas para forma una matriz con el fin de reducir los pines necesarios para el uso de todos los botones de un teclado, en la imagen siguiente se puede apreciar esta configuración y como están conectados todos los botones entre si.

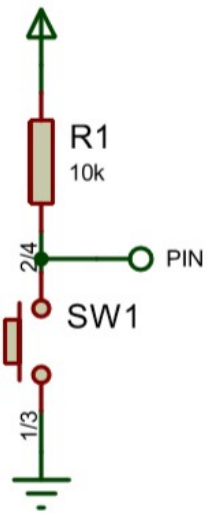


Fig.11 Botón en configuración pull-down.

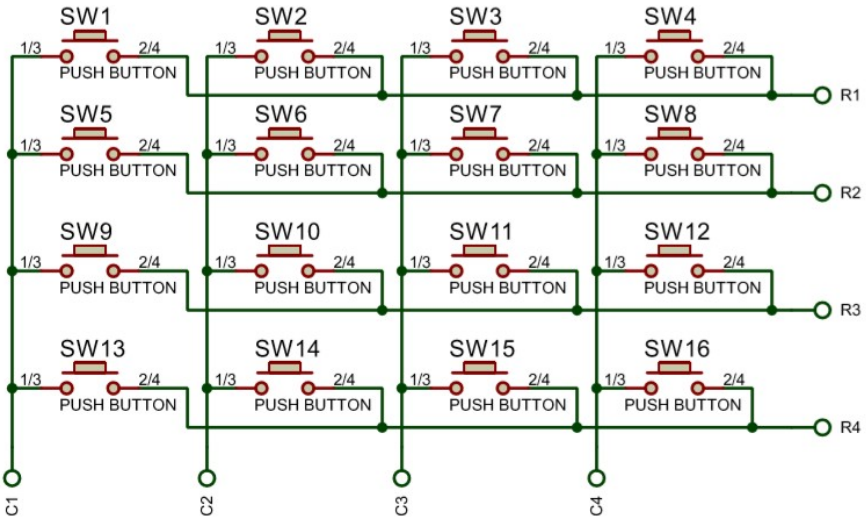


Fig.12 Estructura matricial de un teclado 4X4.

Estos teclados se ofrecen como alfanuméricos pero en realidad cada botón se puede asignar como el carácter o función que uno desee, esto debido a que existen librerías que permiten utilizarlos fácilmente como también configuran que carácter se recibirá por cada botón presionado, estos son los mas utilizados en prototipos de sistemas por lo que se pueden obtener a un bajo precio, nos son la única opción ya que existen otros métodos para poder leer barios botones con pocos pines, algunas opciones mas son el uso de divisores de voltajes que se generaran al presionar cada botón con el inconveniente de que al aumentar la cantidad de botones, cada divisor con respecto a otro tiene muy poca diferencia por lo que podrían aparecer falsas pulsaciones, ademas que se necesitaría un pin con ADC para poder utilizar esta opción.

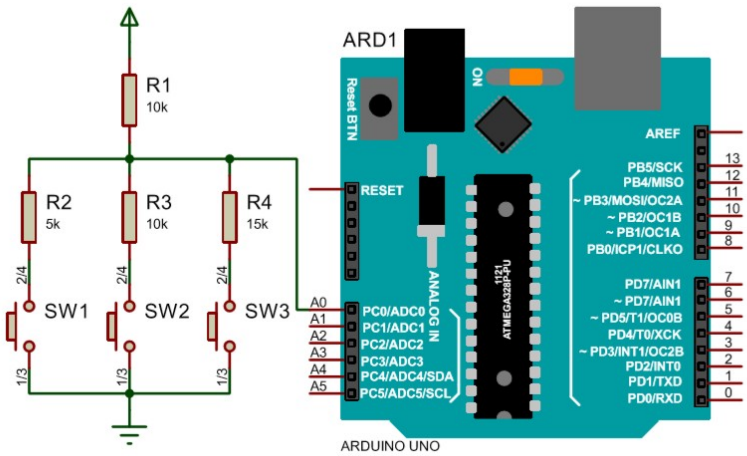


Fig.13 Conexión al ADC de un Arduino uno un teclado por divisor de voltaje.

Otra opción es el uso de algún integrado independiente al microcontrolador que se encargue solamente de leer los botones, este se puede elegir de forma que sea muy limitado en otras opciones pero que disponga de suficientes pines, este dicho integrado independiente se comunicaría de manera serial mediante algún protocolo que se prefiera con el microcontrolador principal, ya existen opciones en el mercado de dispositivos destinas a realizar esta tarea.

1.6 Ruido eléctrico

Casi cualquier dispositivo de control industrial estará sometido a un ambiente propenso al ruido eléctrico, esto es debido a la gran cantidad de cargar inductivas que se pueden encontrar en las zonas de producción, estas pueden ser motores AC, electro-válvulas, contactores, fuentes conmutadas, generadores de chispas para encendido de sopletes, entre otros. Este tipo de ambiente afectara directamente a los microcontroladores por esta razón se debe tomar acciones para proteger todo el circuito del dispositivo de control que emplea el microcontrolador, a continuación se explica algunas acciones que se pueden tomar.

Filtro EMI

El filtro EMI (ElectroMagnetic Interference – Interferencia Electromagnética) es utilizado pera eliminar el ruido eléctrico proveniente del suministro eléctrico de donde se alimenta el dispositivo a proteger, la alimentación es el primer punto donde se debe empezar a proteger un circuito de control, esto debido a que el cableado de alimentación funciona como punto de conexión con otros dispositivos que se alimentan de la mismo suministro eléctrico, si alguno de estos produce ruido eléctrico pueden contaminar al circuito de control ya que pueden viajar por el cableado de alimentación, generalmente producirán este ruido las fuentes conmutadas debido a que utilizan frecuencias altas en su circuitería, conmutación de cargas inductivas y cualquier dispositivo que genere chispas o arcos eléctricos. El circuito correspondiente a un filtro EMI es el siguiente.

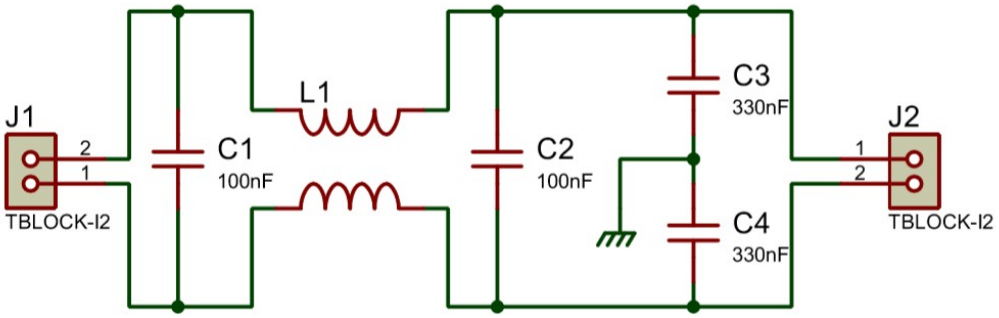


Fig.14 Circuito correspondiente a un filtro EMI.

Como se puede apreciar el filtro EMI consta de condensadores y un inductor, este se debe ubicar antes del dispositivo a proteger por lo que la corriente demandada por la carga pasara por el filtro, con esto en cuenta se debe saber que todos los componentes utilizados debe soportar el consumo de la carga. La alimentación de 120V AC se ingresa por J1 y la carga a proteger se conecta por J2, los condensadores C1 y C2 se encargan de filtrar cualquier ruido de alta frecuencia que aparezca antes y después del inductor, el inductor L1 en realidad son dos inductores a la ves, estos son devanados sobre un mismo núcleo de ferrita en forma de tiroideo, por el se hacen pasar ambas lineas de alimentación fase y neutro, el inductor ademas de realizar su efecto al retrasar el aumento de corriente en el encendido, al ser devanados a la ves, cuando el

ruido eléctrico intenta pasar por el inductor lo hace por ambas lineas por lo que genera un campo magnético idéntico en cada linea, al estar estos juntos y estar presente el núcleo de ferrita permite que el ruido se cancele al chocar ambos campos magnéticos idénticos, por ultimo los dos condensadores C3 y C4 tienen una de sus terminales conectadas a tierra física, por lo que permite que el ruido eléctrico que este referenciado a tierra pasen por ambos condensadores evitando que circulen hacia el dispositivo de control, por lo que ambos condensadores deben conectarse a una puesta tierra real y no ser solamente conectados al chasis o el borne negativo de la fuente del dispositivo, esto para que su funcionamiento sea efectivo.

Protección de pines de entrada

Al momento de utilizar pines del microcontrolador como entradas que deban leer el estado de algún dispositivo externo se puede presentar que dicho dispositivo genere ruido eléctrico o este se encuentre en un ambiente con ruido, por lo que podemos contaminar al microcontrolador con dicho ruido, para esto se debe proteger la entradas digitales que estén en uso, lo mas básico es aplicar un filtro en la entrada digital del microcontrolador.

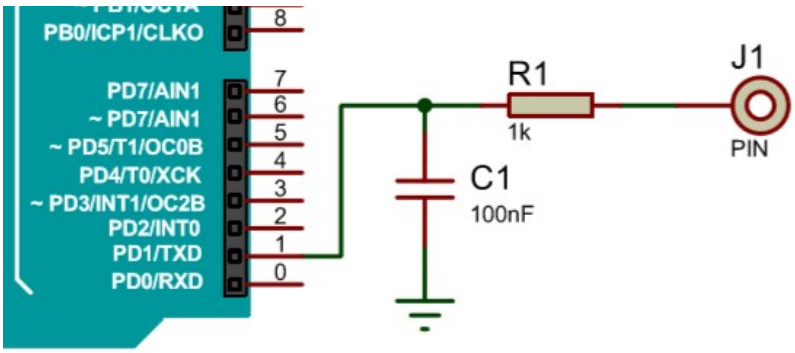


Fig.15 Filtro RC aplicado en la entrada digital de un microcontrolador.

El filtro que se puede apreciar en la imagen anterior es un filtro RC, este simplemente manda a tierra cualquier ruido de alta frecuencia proveniente desde el exterior ademas se intentara cargar por lo que absorberá cualquier ruido si este no tiene un comportamiento alterno, aunque esto puede solucionar los problemas de ruido puede crear otro inconveniente ya que el condensador utilizado se debe cargar primero con la señal de entrante para que el microcontrolador pueda leer el estado perfectamente ya crea un retraso debido a la carga del condensador, por lo que se debe elegir un condensador adecuado que permita proteger a la ves que permite leer el estado, aunque si dicho pin se utiliza en aplicaciones de respuesta lenta como por ejemplo leer un detector de llama, un capacitor de valor algo alto no afectara en la lectura.

Ademas se puede tomar en cuenta un par de cosas mas para aumentar la protección, una es agregar un diodo zener de 5V en anti-paralelo con el condensador de forma que este limitara la tensión de entrada a 5V que son los soportados por el microcontrolador, por lo que cualquier sobre impulso puede ser detenido para evitar daños. Lo segundo a agregar es invertir el estado a

leer a como comúnmente se hace, si normalmente decimos que al leer un estado en alto desde un detector de llama equivale a presencia de llama, se puede hacer al revés, cuando se lee un estado en bajo equivale a presencia de llama, esto con la finalidad de que cuando aparezcan tensiones ajenas al dispositivo de control, estas se ignoren ya que el microcontrolador estará esperando un estado en bajo que es muy poco probable que aparezca por error.

Protección al ruido en la conmutación

Cuando se esta trabajando con relés, estos emplean bobinas para su funcionamiento, dichas bobinas pueden almacenar energía que al momento de crearse un cambio brusco como el de corte en la conmutación del relé, dicha energía almacenada en la bobina puede generar ruido eléctrico de alta tensión, no solo el relé en si puede producir este ruido, también lo producen los dispositivos que se conmutan con dicho relé como por ejemplo si se desea activar una electro-válvula mediante un relé, la electro-válvula puede ser de mayor potencia por lo que puede producir un mayor ruido que llegue a transmitirse mediante el propio relé hacia el microcontrolador.

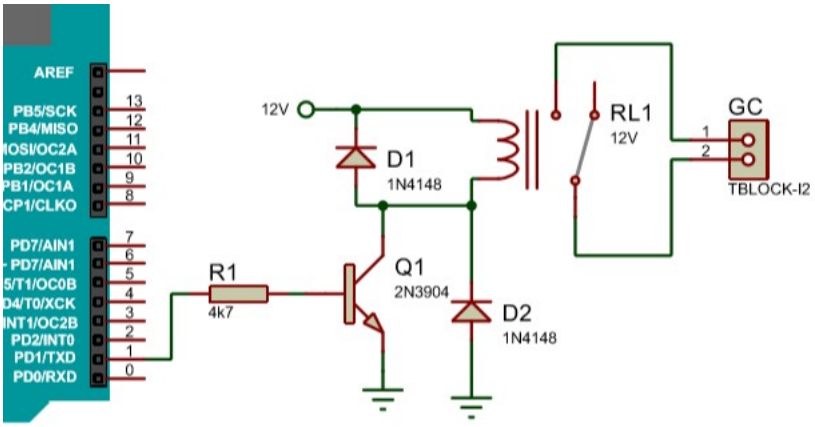


Fig.16 Protección en la conmutación utilizando diodos.

En la imagen anterior se puede apreciar como se posicionan los diodos D1 y D2 para prevenir el ruido producido por el relé RL1, como se sabe una bobina al ser alimentada, esta retrasara el paso de la corriente mientras se carga, una ves cargada permitirá el pase de corriente sin oposición, si dicha corriente se corta la bobina intentara que la corriente no se detenga por lo que producirá por si misma una corriente con polaridad contraria que empezara a circular, dicha corriente producida por la bobina aparece debido a la auto-inducción del campo magnético que produjo al ser alimentada, pero dicha corriente producida por la auto-inducción sera de una tensión mayor, para evitar que dicha tensión afecte a otros componentes se implementa en anti-paralelo un diodo con la bobina del relé, este diodo se encargara de realizar un corto en la bobina cuando esta produzca esa dicha tensión con polaridad inversa, este diodo es D1, el diodo D2 se encargara de proteger al transistor Q1 sobre posibles sobre-tensiones que aparezcan entre su colector y emisor.

Una ves solucionados los problemas del ruido que puede producir el propio relé, también puede ocurrir que el responsable de la interferencia sea la carga que se controla con dicho relé, siempre y cuando sea una carga inductiva de potencia considerable, podemos tener el problema que el ruido que produce dicha carga sea muy alto, como no siempre tendremos la oportunidad de poder instalar por nosotros la carga se puede integra en el controlador una protección ante el ruido como se muestra en la imagen siguiente directamente en los contactos del relé.

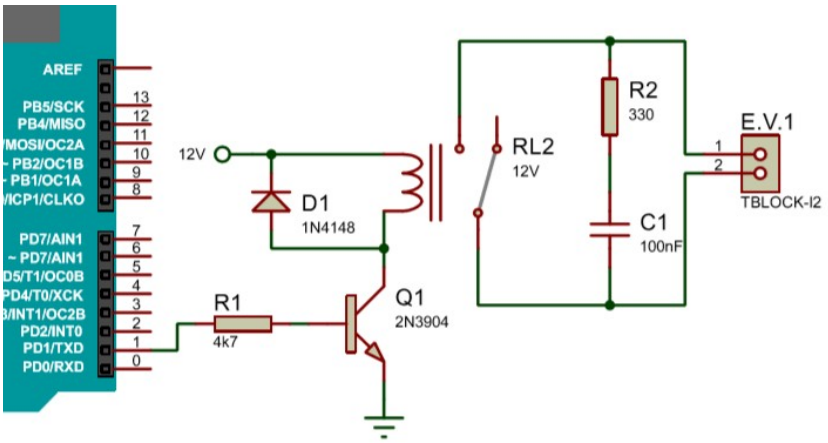


Fig.17 Protección en la conmutación de una carga inductiva.

Esta protección en realidad es destinada a detener las chispas originadas en los contactos de un relé o contactor al momento de la desconexión de cargas grandes, pero como mencionamos anteriormente, este salto de chispas o electrones pueden producir ruido eléctrico, ademas permite que las bobinas que posee la carga se puedan descargar por medio de esta protección. Esta protección consta de poner en paralelo a los contactos del relé una serie de una resistencia y un condensador, la función de esta es simple, como sabemos el comportamiento de una bobina, el problema ocurre en la desconexión de la alimentación hacia la bobina, así que al poner la serie en paralelo con los contactos del relé ocurre que al momento de que estos contactos estén cerrados la corriente no circula por la serie sino por los contactos debido a que el inicio y final de la serie RC estarán cortocircuitadas, por lo que la corriente de alimentación salta la seria y circula por los contactos hacia la carga, al momento de abrirse ocurrirá un salto de corriente en los contactos del relé produciendo el ruido eléctrico, pero al estar presente la seria y estando descargado el condensador debido a que al momento que estuvo cerrado cortocircuito el condensador con la resistencia, estos dos servirán como un pase libre para la corriente almacenada en la carga hasta que el condensador se carga abriendo el circuito, así evitamos saltos de corriente que produzcan ruido eléctrico proveniente de una carga inductiva.

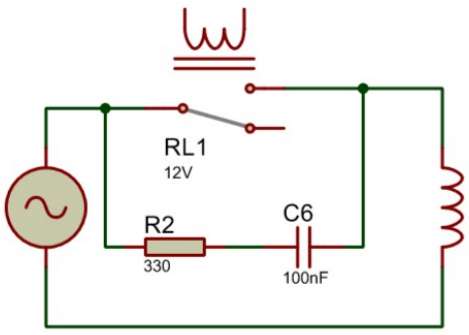


Fig.18 Circuito equivalente de la protección y la carga inductiva.

1.7 Programación del Microcontrolador

1.7.1 Tipos de Control en la Regulación

En el mundo de los controles se han desarrollado muchas formas de lograr la regulación que se desea en un sistema de control, sabemos que el objetivo es lograr el punto de operación que se desea o también llamado setpoint, a continuación se dará una repasado sobre los tipos de control que comúnmente se utilizan y los que mas se conocerán en el ambiente de trabajo.

1.7.1.1 Control Todo o Nada (On/Off)

Este tipo de control es el primero que uno se le viene a la mente al momento de tener la intención de controlar una variable, es implementado en sistema con control sencillo ya que se puede construir de forma eléctrica, mecánica o térmica, como se intuye este tipo de control solo tiene dos posibles valores de salida todo o nada, el estado de esta salida se decide en base a que si se logro el setpoin o no, en palabras sencillas, si no se ha logrado el setpoin poner en alto la salida, si se ha logrado poner en bajo la salida, como se puede ver es muy simple por lo que no sera muy preciso, su aplicación dependerá de la planta a controlar, hay casos donde es suficiente y en otros aplicarlo sera un desastre, generalmente su aplicación sera donde no se requiera mucha precisión y la respuesta de la variable a controlar sea lenta, comúnmente se utiliza en sistemas simples de calefacción y hornos ya que los termostatos utilizan este principio.(Creus, 2010)

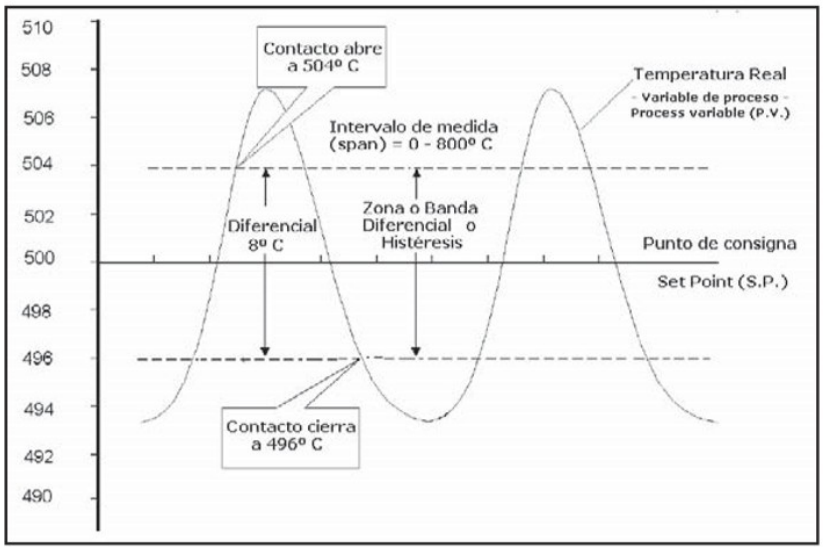


Fig.19 Ejemplo grafico de la salida de un control On/Off.
Fuente: Instrumentación Industrial - Creus 8va edición.

El comportamiento de este tipo de control genera una oscilación de la salida superpuesta sobre el setpoin debido a que este tipo de control necesita algo llamado “zona neutra”, poniendo de ejemplo el caso de controlar temperatura, al momento de llegar a tocar el setpoin, la temperatura alcanzada no sera

estable y podría variar en pequeños instantes con pequeñas variaciones de temperatura que podrían ser de 1Cº o menos, esto produce un estado donde se ha llegado al setpoin e inmediatamente se pierde, este cambio rápido genera que la salida oscile entre alto y bajo de una forma muy brusca y rápida hasta que la temperatura sobrepase el setpoin, si se utiliza actuadores tales como relés y electro-válvulas, estos podrían recibir daño o producirlo a otros elementos, acá se implementa la zona neutra que consta de que la salida mantendrá su estado anterior al momento de llegar al setpoin, por lo que si se llega a sobrepasar unos cuantos grados automáticamente la salida se apaga y se mantendrá así aunque se pierda el setpoin, hasta que la variable de temperatura se disminuya lo suficiente para que el sistema necesite otro impulso en la salida, estos puntos se decide en base al diseño, un ejemplo 5% por debajo del setpoin para encender la salida y 5% por encima del setpoin para apagar la salida.

1.7.1.2 Control Proporcional de Tiempo Variable

Este tipo de control es implementado solo eléctricamente, podría decirse que es una mejora del control anterior ya que realiza la misma función y conserva los mismos inconvenientes pero mas reducidos, su diferencia es que agrega algunos detalles interesantes, lo que agrega es en ves de simplemente desconectar la salida este control la varia en un ciclo de trabajo, es proporcional debido a que la variación de este ciclo de trabajo sera en base a que tan cerca esta el setpoin, quedando de esta forma, al estar lejos del setpoin la salida se mantendrá en alto constante, cuando este sobrepasa el limite inferior podría variar el ciclo de trabajo en un 50%, cuando logra el setpoin, cambia el ciclo de trabajo a; 25%, al tocar el limite superior la salida se desconecta.(Creus, 2010)

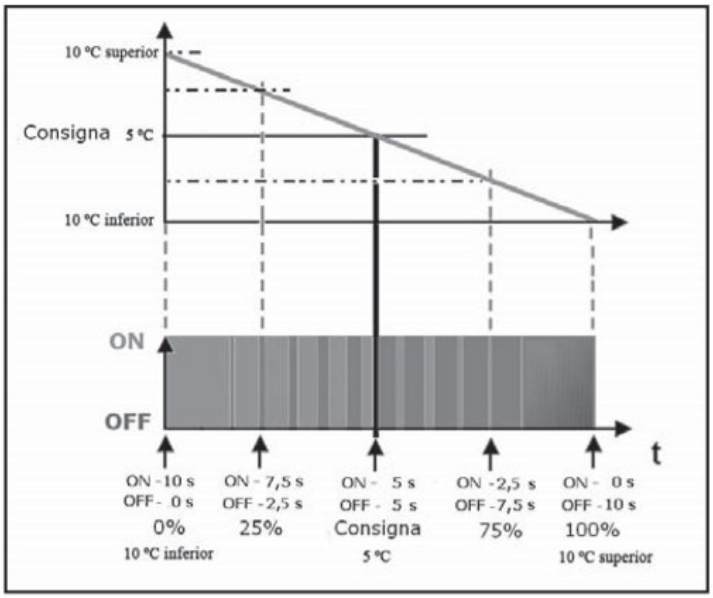


Fig.20 Ejemplo grafico de la salida de un control proporcional de tiempo variable. Fuente: Instrumentación Industrial - Creus 8va edición.

Este tipo de control permite que la salida tenga una oscilación un poco reducida pero siempre existente, ademas como se dijo al principio no se puede implementar (o es muy complejo) de forma mecánica o térmica (usando ampolletas de mercurio).

1.7.1.3 Controlador PID

El control PID (Control Proporcional, Integral y Derivativo) es una técnica utilizada en el control automático de procesos, esta se aplica mediante una retroalimentación (sensor), este control consiste en calcular con que proporción y rapidez cambia la salida del controlador a la ves que se mide el estado del proceso, esto con el objetivo de alcanzar el punto de referencia o Setpoint programado en el controlador, la ecuación que representa a este tipo de controlador es la siguiente.

$$EC.1 \qquad u(t) = K_P e(t) + \frac{K_P}{T_i} \int_0^t e(t) \partial t + K_P T_d \frac{\partial e(t)}{\partial t}$$

*Donde: $e(t)$ es el es el error en el proceso, $u(t)$ es la salida del controlador, K_P es la constante o ganancia proporcional, T_i es la constante de tiempo integral y T_d es la constante de tiempo derivativo.(Adonayt Ruge)

Como su nombre lo indica proporcional, integral y derivativo, este tipo de controlador utiliza tres parámetros importantes; la primera, la constante proporcional que no es mas que un valor fijo el cual se multiplica por el error para poder generar la salida del controlador, esta parte crea una respuesta proporcional al error en la salida del controlador por lo que los cambios no son muy bruscos pero si pudiendo sobrepasar al setpoint por lo que se crea una oscilación constante en el proceso sobre el setpoint.

La segunda es la constante integral la cual realiza una suma de todos los errores pasados desde el inicio del proceso mediante una integral multiplicado por la constante proporcional sobre el tiempo integral, esta parte acelera la respuesta de la parte proporcional por lo que con facilidad se produce inestabilidad del controlador, algunos procesos solo ameritan de estas dos partes para funcionar, por lo que solo basta programar la parte integral de forma que no genere inestabilidad, también dado a que la precisión no lo amerita se puede ignorar las oscilaciones generadas en la salida, y la ultima parte la constante derivativa, que intenta predecir los errores futuros mediante la derivada del error por la constante proporcional y el tiempo derivativo, esta parte termina amortiguando a la parte integral reduciendo la rapidez con que reacciona a medida que se acerca al setpoint por lo que genera una respuesta mas suave en la salida.

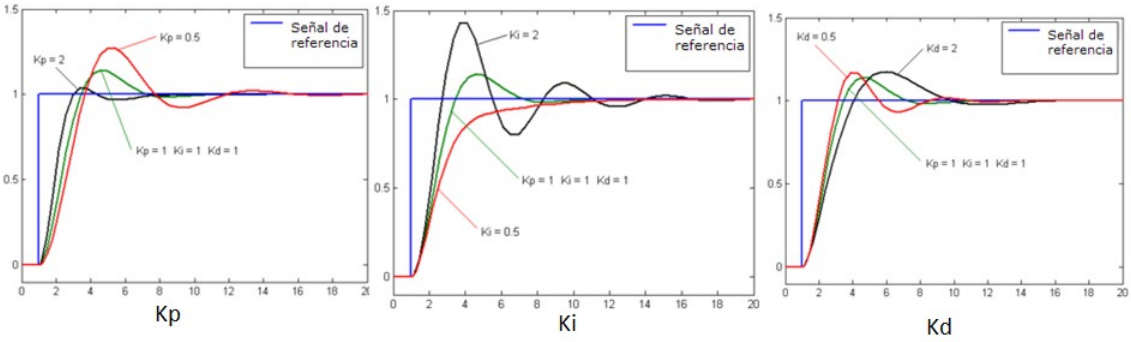


Fig.21 Respuesta de las diferentes partes del control PID.
Fuente: wikipedia.org/wiki/Controlador_PID.

Para que la implementación de los controladores PID en un proceso predeterminado sea satisfactorio este debe ser sintonizado con valores específicos para las variables K_P, K_I y K_D, para esto están disponibles varios métodos de sintonía, tales como el método de Ziegler y Nichols, Cohen y Coon, Kaya y Sheib, entre otros. Ademas estos métodos de sintonía pueden ser en lazo cerrado o abierto por lo que solo basta estudiar y elegir el método mas adecuado para el proceso a controlar.

1.7.1.3.1 Sintonización de Controladores PID

En la sintonización de este tipo de controlador la primera opción son los métodos propuestos por Ziegler y Nichols, ellos propusieron dos en lazo abierto y en lazo cerrado, son de los preferidos debido a que no ameritan un modelo matemático del proceso a controlar o también llamado planta, mas bien se centrar en obtener las ganancias del controlador en base a como se comporta la salida del sistema al variar su entrada, fijándose en esta salida se intenta obtener las ganancias.

Método de Ziegler y Nichols en lazo cerrado (ganancia limite)

El método en lazo cerrado consiste en armar el proceso completo como debería trabajar solo que poniendo las ganancias a “0”, a la ves que se permita tomar datos de la salida de forma que se puedan graficar, una ves armado se procede a aumentar la ganancia k_p hasta que la salida del proceso comience a oscilar, pero dicha oscilación no debe ser cualquiera, esta debe ser estable, osea su amplitud y periodo debe ser constante y no variar en el tiempo, de esto nos daremos cuanta al momento de tomar la salida y gratificarla, desde acá ya nos damos cuenta que este tipo de sintonización no se puede aplicar a cualquier proceso, funciona solo en procesos donde la variable a controlar cambie de forma rápida, un proceso rápido donde se permita la oscilación, por lo que en un control de temperatura no se podrá usar para sintonizar sistemas lentos como el de un horno gas, las ecuaciones utilizadas para la sintonización son las siguientes.(Universidad de Tucuman, 2017)

$$EC.2 \quad (1)\,K_P = \frac{K_C}{1.7} \quad (2)\,T_I = \frac{P_u}{2} \quad , \quad (3)\,T_D = \frac{P_u}{8}$$

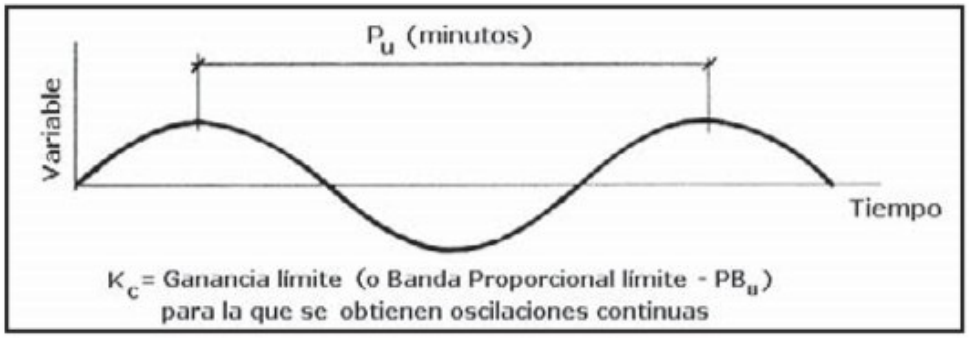


Fig.22 Ejemplo grafico de la oscilación que se busca.
Fuente: Instrumentación Industrial - Creus 8va edición.

Lo que se busca de la grafica es la el tiempo del periodo de la señal P_u y la amplitud K_C.(Creus, 2010)

Método de Ziegler y Nichols en lazo abierto (curva de respuesta)

El método en lazo abierto consiste en realizar una prueba al proceso sin controlador a la ves que se registra los valores del elemento a controlar hasta llegar a un punto determinado donde la salida se estanque para luego generar una grafica de lo ocurrido. En el caso de querer controlar la temperatura de un horno a gas, la grafica representa la respuesta transitoria al calentar el horno y como este se comporta, en un horno a gas esta prueba consistiría en calentar el horno mientras se mide la temperatura interior tomando el tiempo transcurrido, no es necesario que el horno esta su máxima capacidad de producción de calor en esta prueba, con aplicar una potencia del 50% de producción de calor seria suficiente, ya que el método de Ziegler y Nichols intenta generar una salida del controlador PID que siga la pendiente de la curva de respuesta del proceso a controlar para luego generar pequeñas oscilaciones cercanas al setpoint que deberían reducirse a medida que pasa el tiempo.

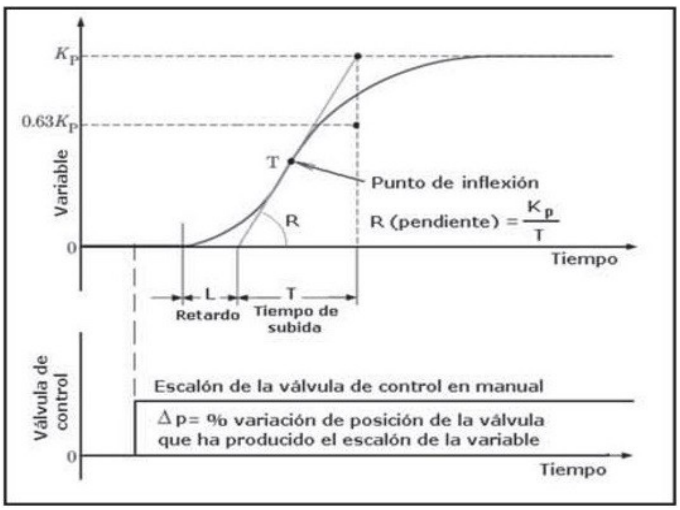


Fig.23 Ejemplo grafico de la curva de respuesta.
Fuente: Instrumentación Industrial - Creus 8va edición.

Ya obtenida la curva de respuesta del proceso a controlar se utilizan las siguientes ecuaciones[1] proporcionadas por Ziegler y Nichols para obtener los valores de sintonía del PID.(Adonayt Ruge)

$$EC.3 \quad (1)K_P=1.2\frac{\gamma_0}{k_0\tau_0} \quad , \quad (2)T_I=2\tau_0 \quad , \quad (3)T_D=0.5\tau_0$$

Estos valores nuevos se obtienen de la grafía de la siguiente forma.

$$EC.4 \quad (1)\tau_0=t_1-t_0 \quad , \quad (2)\gamma_0=t_2-t_1 \quad \text{y} \quad (3)k_0=\frac{y_1-y_0}{u_1-u_0}$$

1.7.2 Implementación de un PID en un microcontrolador

Para la implementación de un PID en un microcontrolador se necesita que dicho microcontrolador lea la temperatura, realice el calculo del PID, traduzca el valor numérico del PID a una que se pueda aplicar al actuador y por ultimo controlar dicho actuador, como se ve el microcontrolador debe realizar una lista de tareas determinadas y dichas tareas tomaran tiempo por lo que claramente si se aplica el PID en un microcontrolador este deberá ser digital, por ser digital la medición de temperatura se realizara con una frecuencia de muestreo que dicho de forma simple cada cuanto se tomara la medida y se procederá a realizar el calculo PID, el método de sintonización PID de Ziegler y Nichols en lazo abierto nos muestra los cálculos finales a realizar para un controlador PID digital, de acá obtendremos las constantes[1] de sintonización.(Adonayt Ruge)

$$EC.5 \quad (1)k_p=K_P \quad , \quad (2)k_i=\frac{K_P(T)}{T_I} \quad \text{y} \quad (3)k_d=\frac{k_P(T_D)}{T}$$

Como se puede apreciar en las formulas se agrega un tiempo de muestreo T , este sera el periodo de la frecuencia de muestreo, dicha frecuencia se decidirá dependiendo de la planta a controlar, para el resto de las tarea sera necesario realizar una versión de la formula del PID que se puede calcular con el código de programación del microcontrolador y por ultimo adecuar el valor resultante para controlar el actuador, todo esto se debe hacer cada cierto tiempo que es dictado por la frecuencia de muestreo, acá una imagen que muestra el algoritmo a realizar.

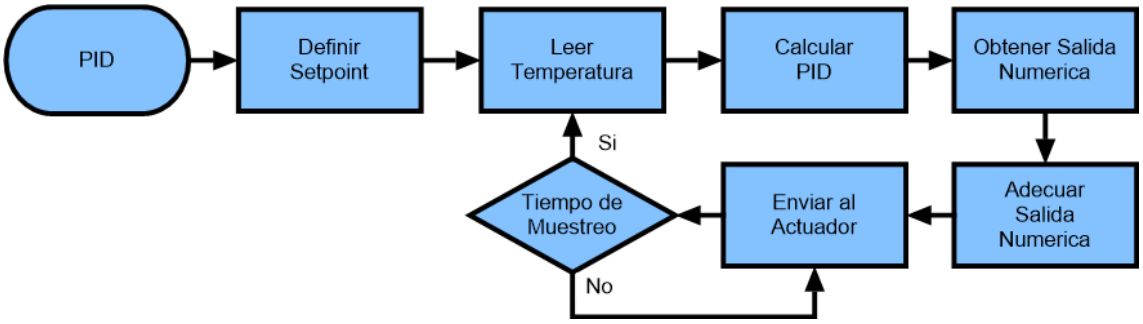


Fig.24 Algoritmo ejemplo al aplicar un PID en un microcontrolador.

1 Estas ecuaciones serán las empleadas en el prototipo.

Como se muestra el tiempo de muestreo se puede aplicar mediante una condición que cuente cada ves que se deba realizar todas las tareas relacionadas al PID, en esta sección nos enfocaremos en como realizar el calculo del PID en un microcontrolador, pero las demás partes se podrán indagar mas adelante.

Calculo del PID mediante la formula característica

Como se pudo ver anteriormente la formula característica del PID utiliza tres partes, una proporcional que es una simple multiplicación fácil de realizar en un microcontrolador, las siguiente dos son una parte integral y una derivativa que ya serán mas complejas de realizar, ademas de verificar si el microcontrolador tiene la capacidad de realizar dichos cálculos, aunque existe la posibilidad de realizar estos cálculos en su forma mas sencilla, reemplazando la integral por una suma y la derivada por una resta, la siguiente imagen muestra el algoritmo a realizar solo para el calculo PID.(Adonayt Ruge)

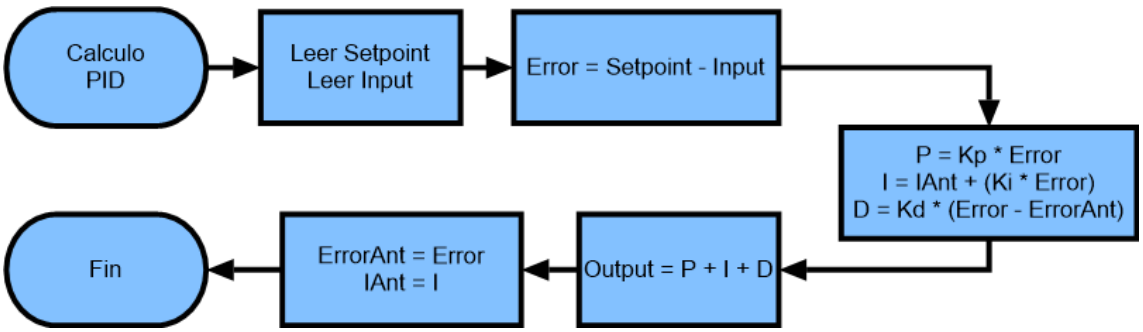


Fig.25 Algoritmo ejemplo para el calculo del PID en un microcontrolador.

Todas las variables utilizadas en el algoritmo son las siguientes:(Adonayt Ruge)

- Setpoint: El valor del estado que se desea.
- Input: Debe tomar el valor del sensor a leer.
- Error: Que tan lejos esta la planta del valor deseado, se obtiene de restar el Setpoint menos el valor leído del sensor.
- P: Es el resultado de la parte proporcional que consiste en multiplicar la constante proporcional por el error.
- I: Es el resultado de la parte integral que consiste en multiplicar el error por la constante integral mas su valor anterior.
- D: Es el resultado de la parte derivativa que consiste en multiplicar la constante derivativa con la diferencia entre el error y el error anterior.
- Output: Contendrá la suma de todas las partes del PID.
- ErrorAnt: Es donde se almacenara el error utilizado en un calculo anterior.
- IAnt: Es donde se almacenara el valor de la parte integral utilizado en un calculo anterior.

Como se aprecia en la imagen anterior, se realiza el calculo de cada parte del PID por separado para luego sumar sus resultados, en la parte proporcional solamente multiplicamos su constante proporcional por el error, en la parte integral se realiza lo mismo multiplicar el error por la constante integral pero

con la diferencia de que sumaremos en cada calculo el valor anterior que tuvo la parte integral, al aplicar esta suma realizamos el objetivo de la integral en la formula original del PID, con la parte derivativa realizaremos una diferencia entre el error y el error anterior, este resultado lo multiplicaremos con la constante derivativa, la forma como se vera en un código de programación es la siguiente.(Adonayt Ruge)

```
Input = thermocouple.readCelsius();
Error = Setpoint - Input;
P = Kp * Error;
I = IAnt + (Ki * Error);
D = Kd * (Error - ErrorAnt);
Output = P + I + D;
ErrorAnt = Error;
IAnt = I;
```

Fig.26 Código ejemplo para la implementación del PID en un microcontrolador.

Calculo del PID mediante una librería

El control PID se puede realizarse también desde una librería ya disponible por la comunidad o por los propios distribuidores de las placas de desarrollo, el IDE de Arduino como también los foros de Arduino se encuentran disponible algunas de estas librerías PID.(Brett Beauregard, 2018)

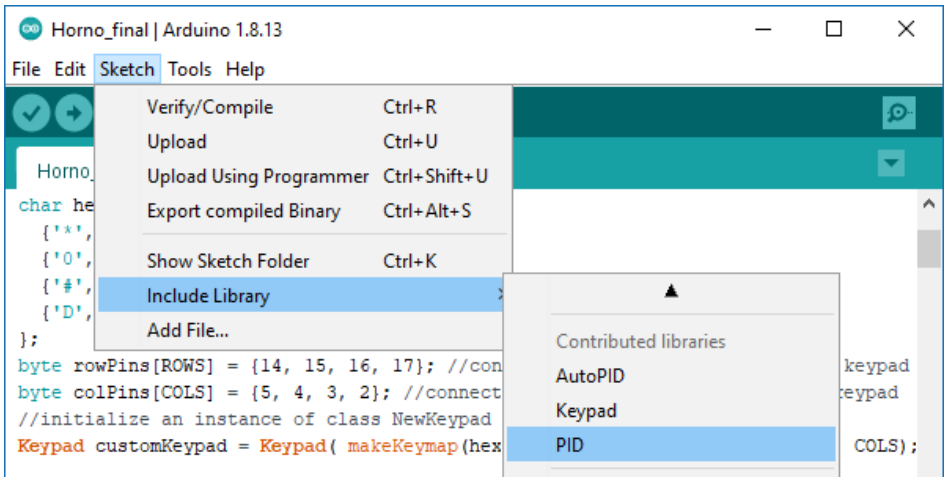


Fig.27 Incluyendo librería PID desde Arduino IDE.

Una librería muy popular en los foros Arduino es la llamada “PID_v1.h” esta solo nos pide algunas variables relacionadas al PID para utilizarla, estos son: (Brett Beauregard, 2018)

- Input: Valor que recibimos del sensor a utilizar.
- Output: Salida del PID.
- Setpoint: Valor deseado (temperatura, posición, etc.).
- Kp: Constante proporcional.
- Ki: Constante integral.
- Kd: Constante derivativa.

Dicha librería se debe configurar algunas funciones que para poder adecuarla con la planta a controlar, estas son:(Brett Beauregard, 2018)

- PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT); Esta función simplemente asocia las variables descritas anteriormente con la librería del PID de forma que la librearía sepa donde debe leer y poner los datos. El único detalle de mas es la sección donde dice “DIRECT”; Puede ser “DIRECT” o “REVERSE”, este determina en qué dirección se moverá la salida ante un error determinado. “DIRECT” es lo más común.
- myPID.SetOutputLimits(0, 100); En esta función se define los limites del controlador PID, este está diseñado para variar su salida dentro de un rango determinado. Por defecto, este rango es 0-255: el rango Arduino PWM.
- myPID.SetMode(AUTOMATIC); Especifica si el PID debe estar activado (AUTOMATIC) o desactivado (MANUAL). El PID se establece por defecto en la posición desactivada cuando se crea. Por lo que esta funcion solo funciona para encender el PID.
- myPID.Compute(); Contiene el algoritmo PID, al llamarse una vez en cada ciclo realiza el calculo y lo almacena en la variable designada como salida.

En ambos casos, ya sea calculando el PID de manera manual o utilizando librerías, la implementación del control sera el mismo, por lo que se necesitara de una frecuencia de muestreo ademas de adecuar la salida para poder controlar los actuadores.

1.7.3 Modulación PWM

El PWM o Pulse-Width Modulation (Modulación de ancho de pulso) es una técnica de modulación de una señal (Generalmente de onda cuadrada) en que consiste en modificar el ciclo de trabajo pero sin modificar la frecuencia de la señal ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se entrega a una carga.

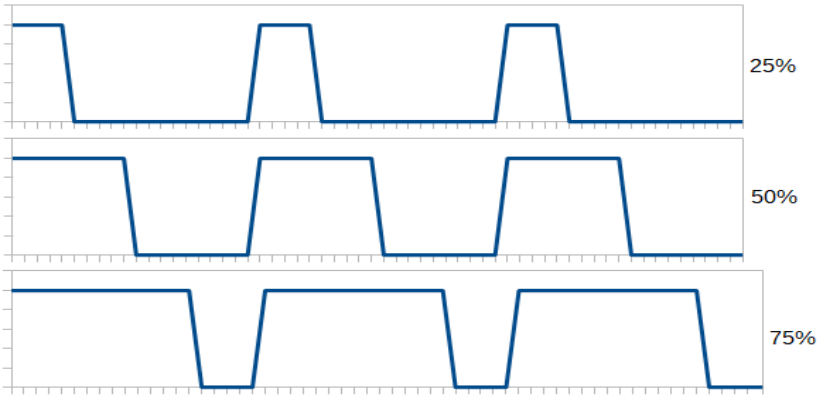


Fig.28 Ejemplo grafico de la modulación PWM.

En el caso de controlar la energía entregada a una carga, esta se efectuá mediante la reducción o aumento de los tiempos en alto de una señal cuadrada que se utilizara para controlar mediante amplificación o alimentando la carga si

es posible, el voltaje promedio que radica sobre la carga es proporcional al ciclo de trabajo de la señal por lo que es posible controlar la potencia consumida en la carga con simplemente dos estados on-off (alto o en bajo).

En el caso de controlar sistemas grandes como de un horno a gas la modulación PWM es implementada sobre una electro-válvula de pase de gas, con lo que se puede controlar la cantidad de calor generado en el horno, pero esto debe ser realizado a frecuencias muy bajas ya que los periodos en alto están en tiempos en unidades de segundos, ademas de que un dispositivo mecánico como una electro-válvula no puede trabajar a frecuencias altas. Un pequeño ejemplo sobre lo que se sugiere es el siguiente, se quiere generar una señal PWM con un ciclo de trabajo del 60% con un periodo de la señal de 20 segundos, la frecuencia de la señal seria la siguiente.

$$EC.6 \qquad (1)\, f = \frac{1}{P} = \frac{1}{20\,Seg.} = 0.05\,Hz \qquad (2)\, T_{on} = 20\,Seg.\left(\frac{60}{100}\right) = 12\,Seg.$$

Acá se puede ver que la frecuencia que al final se usara sera menor que 1 debido a los largos tiempos en el periodo de la señal, también se ve que en un 60% del ciclo de trabajo se traduce a 12 segundos de tiempo en alto, simplemente restando se sabe que serán 8 segundos de tiempo en bajo.

1.8 Rebobinado de un transformador

Generalmente cuando se trabaja en circuitos de alta impedancia y se quiere obtener suficiente corriente para poder medir, sera necesario utilizar una tensión elevada, una llama podría conducir corriente eléctrica, esto se puede utilizar para poder detectar la llama, pero al ser de muy alta resistencia la corriente seria poca, en un sistema de control no se puede encontrar una fuente de alta tensión a no ser que se construya una fuente elevadora complicando el circuito, la fuente mas cercana de una tensión suficientemente alta es el propio suministro eléctrico del circuito, ya que para alimentar un sistema de control se tiene reducir los 120V AC del suministro eléctrico a una tensión de 12V DC, esta podría ser una solución, pero podría llegar a ser peligroso utilizar la fase y el neutro del suministro eléctrico directamente, por lo que se debe emplear un transformador para aislar la alimentación.

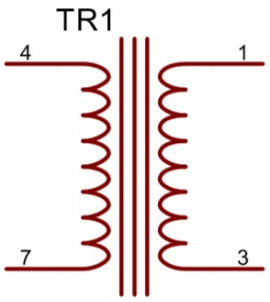


Fig.29 Símbolo de un transformador sencillo.

La idea de rebobinar un transformador es obtener los voltajes alternos que uno desee con la ventaja de poder aislar la alimentación, el procedimiento de rebobinado consiste en poder obtener el valor de potencia del núcleo disponible, ya obtenido este valor podemos calcular las vueltas necesarias para obtener los voltajes ademas de elegir los calibres, para esto se muestra una serie de formulas[2] muy resumidas y simplificadas, que en la practica han demostrado ser útiles y no dar problemas, para obtener el calculo de la potencia se mide el área del núcleo y se realizan las siguientes formulas.(CVR, 2020)

$$EC.7 \qquad (1)\, S_n = A(B) = cm^2 \qquad (2)\, P_{VA} = (S_n)^2 = VA$$

Como se puede ver se mide el área S del núcleo y luego se calcula la potencia al elevarlo al cuadrado, las medidas A y B son en centímetros y corresponden a las siguientes que se pueden ver en la imagen.

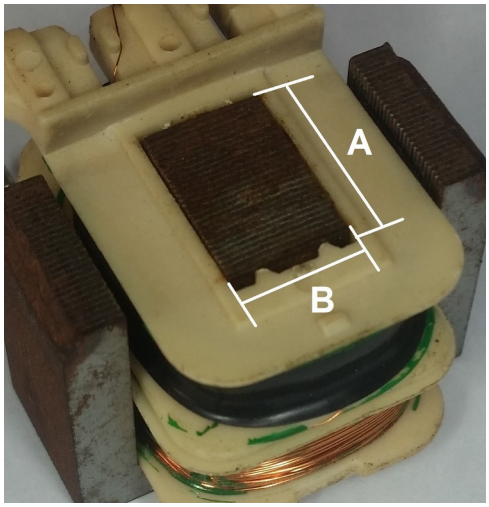


Fig.30 Medidas para obtener el área del núcleo en un transformador.

El calculo de potencia nos sirve para descartar el núcleo en caso de necesitemos otra potencia distinta al que disponemos, cuando ya decidimos que núcleo utilizar debemos ahora obtener la relación de espira por voltio característica del núcleo elegido, osea relación que existe en cuantas vueltas se necesita realizar para obtener un voltio, para esto se hace uso de una constante la cual es producto de resumir varios parámetros de las ecuaciones originales tomando en cuenta que casi siempre se devanaran núcleos de hierro al silicio, el calculo es el siguiente.(CVR, 2020)

$$EC.8 \qquad \alpha = \frac{42}{S_n}$$

Ya obtenido el valor de las espira por voltio sera fácilmente obtener la cantidad de vueltas necesarias en el primario y el secundario, para esto se utiliza los voltajes deseados, analizando un poco no damos cuenta que en si la relación de espiras por voltio nos dice cuantas espiras se necesita para un voltio, por lo que solo habrá que multiplicar el resultado para los voltios que se necesitan, esto se hace de la siguiente forma.(CVR, 2020)

$$EC.9 \qquad \omega = E(\alpha)$$

Este calculo se realiza con cada tensión deseada en cada devanado del transformador incluyendo el primario, por ultimo que daría elegir que calibre para poder empezar a devanar, esto es tan sencillo como solo elegir el calibre de alambre que soporte la corriente que circulara por dicho devanado, para

2 Estas formulas serán las empleadas en sección 2.2 Circuito detector de llama.

saber que corriente circulara simplemente se despeja utilizando la potencia del núcleo, como sabemos la potencia de un transformador es constante y no cambia, por lo que la potencia que entra es la misma que sale, por lo que podemos hacer lo siguiente.(CVR, 2020)

$$EC.10 \qquad I = \frac{P_{VA}}{E} = Amp.$$

Esto de nuevo se realiza en cada devanado, ya obtenida la corriente de cada devanado simplemente se busca en una tabla AWG de calibres de alambre esmaltado o magnético el diámetro de alambre que soporte cada corriente de cada devanado.

AWG	Diámetro mm	Sección mm^2	espiras por cm	Kg por Km	Resistencia Ω por Km	Cap. De Corriente Amperios
21	0.7230	0.41	12.8	3.64	41.46	1.20
22	0.6438	0.33	14.4	2.89	51.50	0.92
23	0.5733	0.26	16.0	2.29	56.40	0.73
24	0.5106	0.20	18.0	1.82	85.00	0.58
25	0.4547	0.16	20.0	1.44	106.20	0.46

Fig.31 Ejemplo de una tabla AWG de calibres de alambre esmaltado.

Ya una ves elegido el alambre y obtenido los valores de cantidad de vueltas lo que queda es devanar el transformador, para esto se debe tomar el carrete o formaleta del transformador y empezar a enrollar el alambre esmaltado, se recomienda realizar el enrollado en un solo sentido como por ejemplo en sentido orario o hacia la derecha, se debe realizar de forma ordenada dejando una espira al lado de la otra, esto con el objetivo de que alcance la mayor cantidad de alambre posible y no tener dificultad al armar el núcleo de nuevo, cada ves que se termine un devanado se debe aislar con papel con parafina o bien cinta de pintor o masking tape, se debe cubrir el devanado anterior para empezar el siguiente, por ultimo al terminar el devanado y armado del núcleo se puede pintar el transformador completo con barniz di-eléctrico para evitar humedad y dejar rígidos los devanados evitando así vibraciones producidas por el campo magnético.(CVR, 2020)

Capitulo 2: Análisis y presentación de resultados

En esta parte se define los requerimientos del sistema, esto basándose en las exigencias del abitad del problema a resolver como del mismo problema, esto se logra escuchando las necesidades de los operadores y técnicos que realizan su trabajo en las plantas de producción con su experiencia manipulando equipo industrial, en el caso de la producción de pan el objetivo primordial es la automatización del horneado y su seguridad en el encendido de sopletes a gas.

La metodología a utilizar es de tipo aplicada o tecnológica. Las investigaciones se realizarán a la ves se aplica el conocimiento adquirido con el fin de eliminar errores en cada parte del proyecto, ya que se tiene como propósito principal la unión de varias funciones en una sola solución con el fin claro de que sea competente.

2.1 Etapa de análisis

En esta etapa se explica sobre la problemática encontrada en la industria panificadora en Nicaragua que motiva al autor a desarrollar este proyecto, además de aclarar cuales son los requerimientos que debe cumplir el sistema en base a las entrevistas realizadas a personal técnico de la industria.

Oferta de dispositivos de control industrial

En el proceso de producción del pan el horneado podría ser la parte mas importante debido a que en este punto se define el producto final el cual sera vendido, por esta razón este proceso debe ser preciso en cuanto a temperatura y tiempo de horneado, la cantidad de producción evita que el operador pueda dedicarse solo en una parte del proceso, de ahí la necesidad de automatizar el horneado del pan que permita encender y apagar el horno con solo presionar un botón de inicio luego de configurados los valores de temperatura y tiempo.

Estos sistemas ya se ofrecen comercialmente o bien están integrados de fabrica en el horno industrial, al comprarlo nuevo, un inconveniente relacionado al problema a resolver radica al momento de dar mantenimiento o reparación a dichos hornos, ya que estos necesitan repuestos compatibles con el modelo y marca, haciendo el trabajo de mantenimiento mas difícil ya que el país no posee un flujo constante de importación de este tipo de equipos y sus respectivos repuestos, esto genera que algunos técnicos opten por desechar el sistema de control original e instalar uno genérico o diseñar el sistema en base a

Fig.32 Controlador Omron E5EN con PID para hornos, no posee funciones de detector de llama.

contactores y ampolletas de mercurio, esto termina reduciendo las capacidades o cualidades del sistema original.

Dichos controles que se comercializan pueden llegar a ser muy sofisticados o costosos, debido a que las funciones mas buscadas son las de control de temperatura, cuenta regresiva, encendido de soplete y seguridad, por lo que se desea ofrecer un control que conste de un dispositivo único, costo admisible y simple.

2.1.1 Criterios del sistema

Para definir los criterios del diseño para el prototipo, se recurrió a realizar entrevistas[3] a técnicos que se dedican al mantenimiento y reparación de hornos para pan a gas, dichos técnicos proporcionaron toda la información sobre las características que debe poseer el controlador de horno, osea nos dicen que debe cumplir para cubrir las necesidades de los productores de pan.

Mantenimiento y reparación del horno

Se recurrió a preguntarles cuales son las soluciones que comúnmente aplican al momento de necesitar implementar el control de un horno a gas, nos responden que se intenta encontrar los repuestos originales o por lo menos parecidos, algunos casos es posible remitiéndose a las casas exportadoras pero quedando sujeto a los costos, en otros casos quedan con la consecuencia de no poder encontrar exactamente el mismo componente, por lo que inmediatamente se recurre a instalar controladores que se encuentren con facilidad en el mercado local, siempre y cuando el dueño de los medios de producción tenga el capital para invertir en dicho equipo, si no es el caso de poseer el capital se decide instalar módulos separados que permitan el encendido del horno (módulos de encendido) y control del tiempo de horneado (termostato), algunas veces siendo módulos usados.

En la instalación de un control de horno por parte de dichos técnicos toman como prioridad la detección de llama como medio de protección mas que la necesidad de encendido del horno ya que es común que los operadores se les encomiende realizar la tarea de encendido, preferirán invertir en un control completo si disponen de esta cualidad, acá queda claro que un detector de llama es un sensor a utilizar en el prototipo.

En el momento de operar el horno

Se les pregunto de manera oral los procedimientos en los que un hipoteco control permite al operador utilizar el horno y así poder definir que debe hacer el sistema de control, primero control de temperatura, como se puede suponer el sensor de temperatura es el termostato debe poder leer la temperatura y

3 Ver entrevista en sección 1 Anexos.

cortar el suministro de calor una ves alcanzado la temperatura deseada, acá se puede integrar el PID que remplazara al termostato.

El detector de llama generalmente viene integrado en el modulo de encendido que se utilizara para el encendido del soplete, ellos comentan que este debe hacerse mediante un proceso, esto debido a que se trabaja con gas circulando por tuberías, el encendido es de esta forma, primero debe abrirse una electro-válvula que permita el pase de gas hacia un piloto, como es un piloto la llama es pequeña por ende el pase de gas también es pequeño, el modulo de encendido intentara encender el piloto primero, por esta razón debe situarse al lado del piloto el electrodo sensor (Spark), este proviene del propio modulo, este electrodo funciona como salida de un generador de chispa y a la ves sonda del detector de llama.

De ultimo comentan que es prescindible un contacto de puerta, este se puede tomar como sensor ya que su única función es detectar cuando la puerta este cerrada. Debido a que el horno posee una turbina interna, ellos mencionan que esta se encarga de hacer circular el aire dentro del horno con la finalidad de poder tener una temperatura uniforme en su interior, pero conlleva un peligro, si por alguna razón se tiene que abrir la puerta del horno a la mitad del proceso de horneado, esta turbina podría arrojar aire caliente sobre el operador, he aquí que se debe saber si la puerta esta abierta, para poder encender el motor de la turbina solo cuando el horno este cerrado. El encendido de la turbina se realizara mediante un contactor gobernado por el contacto de puerta, por lo que el circuito del motor es independiente.

Inspeccionando en una visita

Se realizo unas visitas a pequeñas empresas productoras de pan para saber el entorno en que se trabaja y los tipos de horno que generalmente se utilizan, en esta visita se descubrió que la mayoría de los hornos utilizados poseen las mismas características, utilizan sopletes que implementan detectores de llama con módulos de encendido, implementan una turbina en su interior para hacer circular el aire en su interior, utilizan varios hornos de las mismas capacidades del orden de 5 bandejas, algunas excepciones es el uso de hornos rotatorios donde se ingresa un carro completo de 16 bandejas y los utilizados con combustible diésel, aunque estos son implementados mas en empresas de mayor capital.

Concluyendo en base a las entrevistas

Como conclusión de las explicaciones recibidas de forma oral en las entrevista los sensores a utilizar son 3:

- Sensor de temperatura.
- Detector de llama.
- Contacto de puerta.

El sensor de temperatura se utilizara para realizar un control PID, este remplazara el control utilizado por los técnicos que es un termostato osea control on-off, el PID controlara el soplete para mantener la temperatura configurada lo mas estática posible dentro del horno, los rangos de temperatura que se necesita es entre 100Cº a 300Cº, el soplete se activara cerrando y abriendo el pase de gas mediante una electro-válvula como en el termostato pero con la diferencia de que se aplicara con un PWM modificado que permita mantener la temperatura mas estable sobre la temperatura deseada.

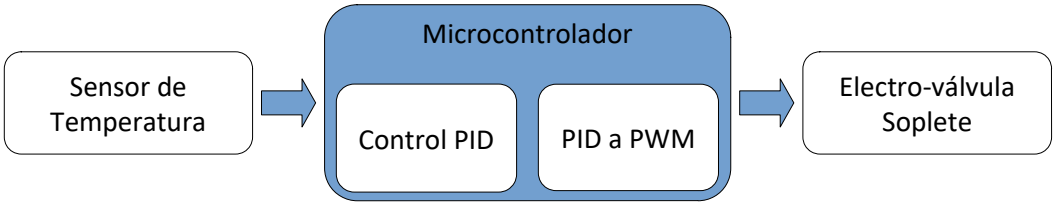


Fig.33 Proceso del control de temperatura utilizado en el campo.

En base a la información obtenido por los técnicos el proceso de encendido de un horno en un hipotético control de horno seria el siguiente.

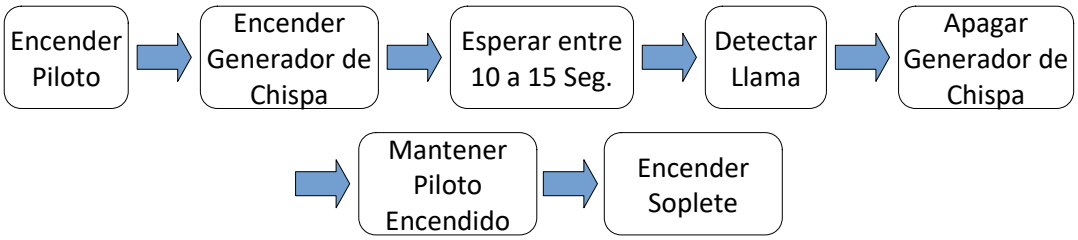


Fig.34 Proceso de encendido del soplete a gas.

El generador de chispa se encenderá luego de abierta la electro-válvula del piloto, este deberá permanecer encendido hasta que el gas llegue desde el tanque y salga por la boquilla del piloto, una ves encendido el piloto, el detector de llama confirma la presencia de llama, el controlador de horno apagara el generador de chispa y activara la segunda electro-válvula correspondiente pase de gas del soplete, este sera de un flojo mayor de gas, ambas electro-válvulas permanecerán encendidas mientras se necesite calor, si no se detecta la llama en el piloto por algún tipo de error el controlador apagara todas las electro-válvulas e indicara error, mientras tanto cuando se llegue a la temperatura configurada y allá que apagar el soplete, solamente se desactiva la segunda electro-válvula que corresponde al soplete, dejando el piloto encendido para permitir encender el soplete de nuevo de forma mas rápida.

De ultimo el sensor del contacto de puerta sera implementado de forma independiente al prototipo del control de horno para que gobierne el apagado automático de la turbina en el momento que se quiera abrir la puerta del horno a mitad de horneado a como lo han implementado los técnicos.

Para el diseño del prototipo se utilizara un microcontrolador aprovechando esto se intentara poder controlar el tiempo de horneado y realizar el encendido del

soplete a gas de forma automática sin necesidad de asistencia de un operador, ademas se agregara una interfaz de usuario que permita visualizar la temperatura actual y un teclado para ingresar los datos de configuración.

2.1.2 Requerimientos del sistema

Ya definida todas las cualidades y características que deberá llevar el prototipo de control de horno se puede listar los requerimientos del sistema los requisitos mínimos del controlador de horno industrial se describen a continuación:

- Funcional en hornos a gas, se garantizara su funcionamiento con hornos a gas, no se garantiza con diésel.
- De carácter genérico, cubriendo las necesidades básicas del control automatizado del horneado de pan.
- Pensado para encendido del horno por chispa de alto voltaje.
- Control de temperatura mediante el uso de un control PID.
- Que realice una cuenta regresiva del tiempo de horneado para su apagado automático una ves terminado el tiempo.
- Panel interactivo con el usuario, este consta de una LCD 16X2 donde se reflejara la temperatura actual y el tiempo restante, ademas junto a un teclado donde le permitirá al usuario ingresar los valores de temperatura y tiempo.
- Monitoreo de la presencia de llama.
- Sistemas de seguridad básicos, esto sera la des-conexión del suministro de gas al momento que se detecte el apagado repentino de la llama.

2.2 Etapa de diseño del sistema

Se define el proceso que se realizó para el diseño del sistema, empezando desde un esquema general y luego describiendo cada uno de las partes del sistema de control; Se realizara explicaciones mas detalladas de como se pretende utilizar cada dispositivo, y ademas de explicar porque se decidió utilizar dichos dispositivos .

Esquema general del sistema

Para el diseño del sistema control electrónico del horno industrial, se divide el proceso de diseño en diferentes partes importantes descritas en el esquema general del sistema, cada una de estas sera diseñada, construida y probada de forma independiente para luego al final unificar estas partes en un solo sistema listo para realizar las pruebas necesarias.

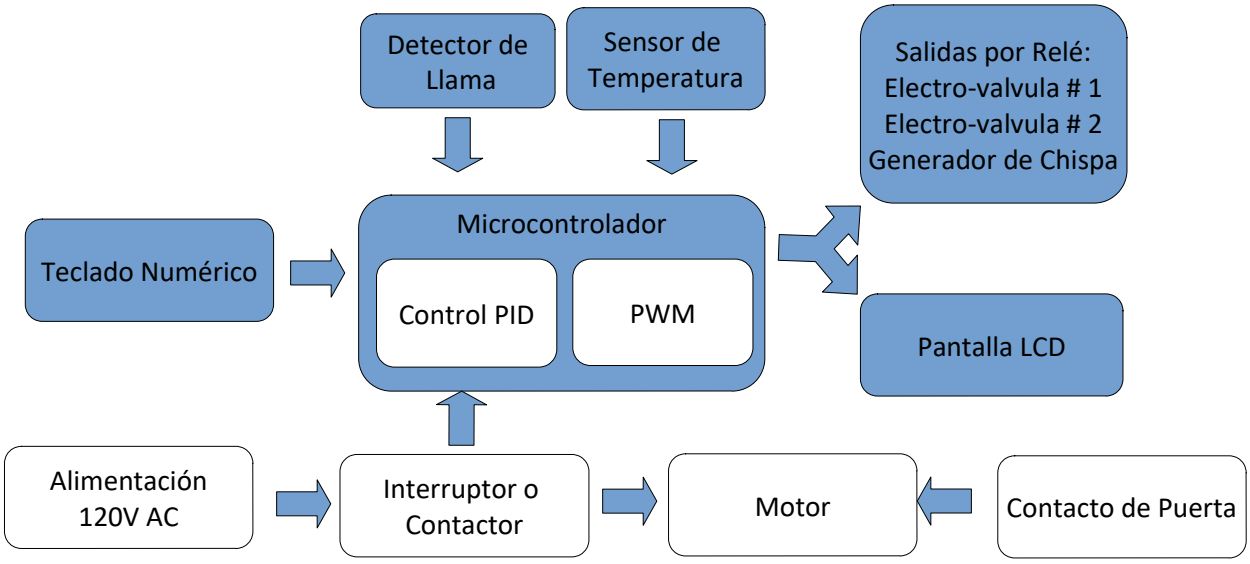


Fig.35 Esquema general del sistema.

En el esquema se puede apreciar la distribución de las partes del prototipo de control de horno indicando las salidas y entradas, cada uno de estas partes serán diseñados y construido por separado en base a su respectiva investigación, esto como se ha dicho antes es por razones de poder poner a prueba cada parte y así poder descartar errores, también se muestra las dos técnicas de control que se encargara de usar el microcontrolador, claramente el microcontrolador no solo se encargara de realizar estas 2 tareas, también realizara los procesos de encendido del soplete o flauta a gas y las correspondientes acciones a realizar en base a los sensores, esto se explico anteriormente pero se aclara a mas detalle en el desarrollo de cada elemento.

2.2.1 Medición de temperatura

Para la medición de la temperatura se opto por usar una sonda termopar tipo K, ya que como se definió en el criterio del sistema, el rango esperado de operación es entre los 100Cº a 300Cº, la comparativa realizada para tomar esta decisión se baso en el costo del sensor a implementar y que tan conocido es por los técnicos a los cuales se les pregunto[4] sobre la problemática en el horno a gas, esto se puede ver en el siguiente cuadro.

Tabla 2
Elección del sensor de temperatura.

Sensor	Costo	Rango	Familiaridad	Elección
Termopar tipo K	$20	-40 °C - 1.100 °C	Si	X
RTD PT100	$35	-200 °C - 850 °C	Poco	

4 Ver entrevista en sección 1 Anexos.

Las sondas termopar tipo K son comúnmente usadas en los sistemas de control, pero también en la industria del pan, su costo es accesible y rango de operación es el adecuado, es por eso que se decide por este tipo de sonda. El principio de funcionamiento para poder determinar la temperatura utilizando este tipo de sonda se basa en medir la variación de voltaje de la misma, este valor esta en el orden de µV a mV, mediante una formula que también es proporcionada por el fabricante, se puede determinar la temperatura, para poder eliminar errores y aumentar la precisión se opto por utilizar un modulo convertidor de termopar a valores digitales, este es el MAX6675, este enviara al microcontrolador los datos de temperatura por un puerto serial.

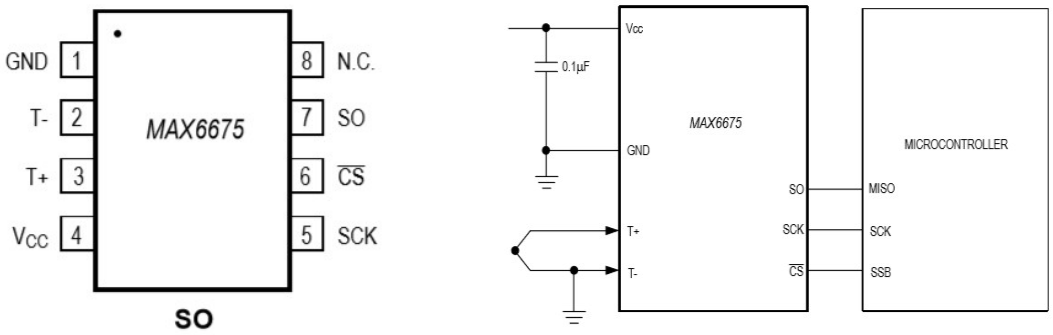


Fig.36 Modulo MAX6675. Fuente: DataSheet MAX6675.

Este modulo es capas de comunicarse por medio del protocolo serial SPI, es alimentado con 5V DC, las conexiones correspondientes de este modulo y el Arduino son las siguientes.

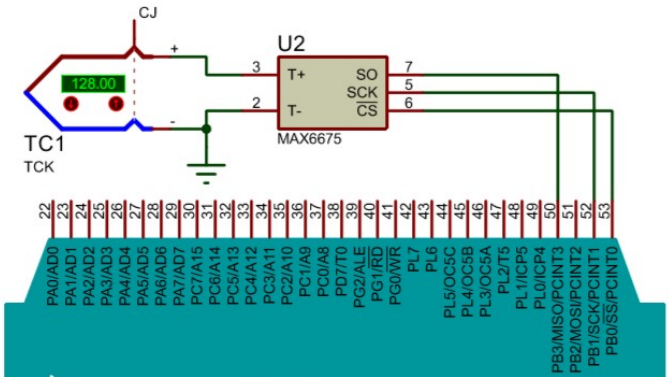


Fig.37 Conexión del modulo MAX6675 a la placa de desarrollo Arduino Mega 2560.

Como se puede ver el uso de este modulo simplifica el circuito de temperatura y aumenta la precisión gracias a las prestaciones del IC MAX6675. La forma de utilizar este modulo es sencilla, simplemente se configura los pines de conexión del puerto SPI para luego solo mandar a llamar la función encargada de leer el sensor, acá un diagrama de flujo que lo explica gráficamente.

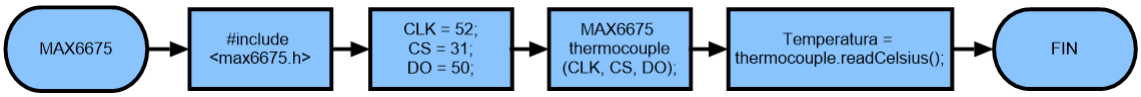


Fig.38 Diagrama de flujo sobre el uso del modulo MAX6675.

2.2.2 Circuito detector de llama

Para la detección de llama se decide por utilizar un detector por ionización, este tiene varias ventajas en cuanto a lo que se pretende implementar, se resume en que no hay elementos que puedan producir interferencias y crear falsos positivos, su respuesta es instantánea al momento de no haber llama y no es muy sofisticado como para poder ser construido en base a componentes discretos, esta elección se puede ver en la siguiente tabla.

Tabla 3
Elección del detector de llama.

Tipo Detector	Posible Falso Positivo	Respuesta	Costo	Elección
Térmico	No	Lento	$80	
Luz y Calor	Si	Rápida	$20 - $200	
Ionización	No	Rápida	$164.11	
Ionización (construido)	No	Rápida	$26.06	X

El detector por ionización consiste en detectar la llama aprovechando los iones liberados durante la combustión del gas, este libera electrones libres que se pueden utilizar para hacer circular una reducida cantidad de corriente a través de la llama.

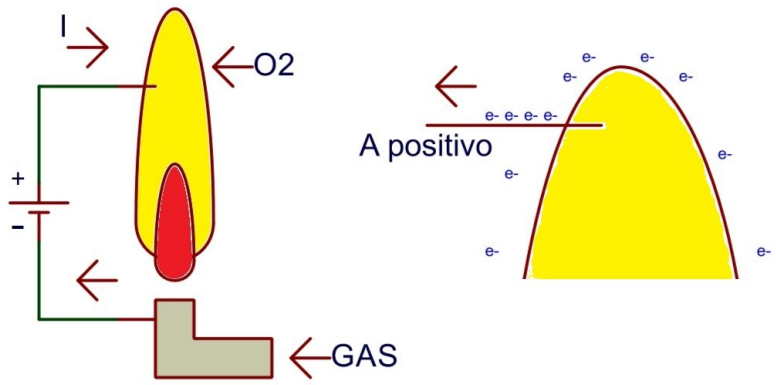


Fig.39 Se puede apreciar donde aparecen los electrones libre al momento que combustionan la mezcla de oxigeno y gas.

Al momento que la llama libera electrones nosotros podemos aplicarle una tensión, en este caso debe ser alta, ya que la llama se vuelve un conductor pero de muy alta resistencia, del orden de MΩ (alrededor de 80MΩ), hay dos opciones, la primera consiste en aplicar la tensión de 120V AC del suministro eléctrico, el problema de utilizar esta opción es que sera obligatorio tener una buena conexión a tierra en el chasis del horno, o menos indicado conectar el toma de neutro (N) al chasis; La segunda opción que es la que utilizaremos,

consiste en aislar la tensión alterna a utilizar mediante un transformador, en este caso utilizaremos una tensión de 80V AC, una ventaja mas de utilizar esta tensión aislada es que no entrara en conflicto con otros dispositivos que pudieran ser instalados junto al control de horno (Generador de chispa).

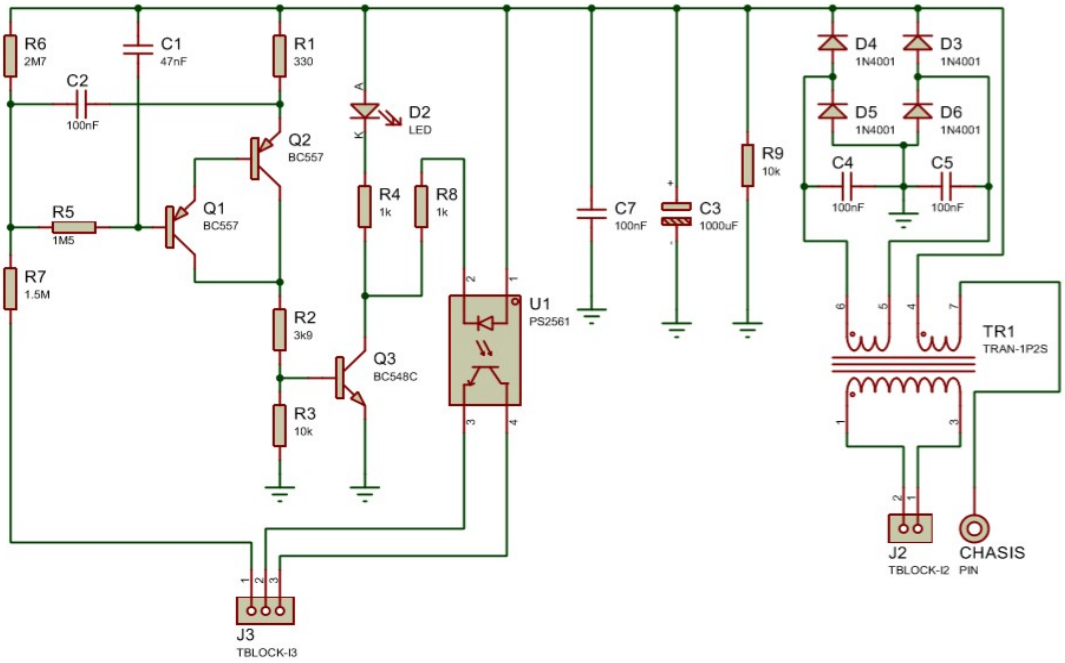


Fig.38 Circuito eléctrico del detector de llama.

En la imagen anterior se puede apreciar el esquemático completo del circuito detector de llama, como se puede apreciar el transformador posee dos devanados secundarios, uno es de 9.5V AC y el otro de 80V AC, el de 9.5V AC se utiliza rectificado para alcanzar 12V DC que se utilizaran para alimentar el circuito, el de 80V AC sera el que se aplique a la llama, este devanado se conecta un terminal con los 12V DC y el otro directamente al chasis del horno, este devanado no debe suministrar mucha corriente ya que la llama solo permitirá que circulen décimas de µA, cuando estemos en presencia de llama estos 80V serán aplicados sobre la llama creando un divisor de voltaje entre la llama y las resistencias que se encuentran en la base del transistor Q1, luego inmediatamente circula una pequeña corriente por la llama, esta sera amplificada por el transistor Q1 que esta en configuración Darlinton con el transistor Q2, estos previamente polarizados.(Foros Electronica, 2018)

Estos transistores son del tipo PNP , porque?, la llama no se comporta simplemente como un conductor, tiene otro comportamiento, este se puede interpretar de varias formas, una que resulta para entender el circuito es ver a la llama como una pequeña batería que esta polarizada en inversa, osea el positivo conectado al chasis del horno y el negativo conectado al electrodo detector, esta genera un poca corriente por lo que es mas conveniente ponerla en serie con otra fuente, en

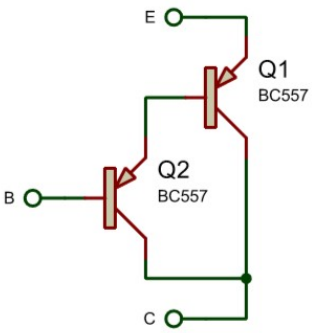


Fig.39 Configuración Darlinton de transistores PNP.

este caso la tensión de 80V AC, con esto se detecta la corriente que circula por las dos fuentes tomando en cuenta sus tensiones; Otra forma de verlo es pretender que la llama es un diodo muy malo, pero diodo, como esta libera electrones estos se pueden tomar por un extremo y reponerlos por el otro, pero no en sentido contrario, muy parecido al efecto rectificador de las bulbos al vació.

Al amplificar ese pequeña corriente resultante la podemos utilizar para poner en saturación el transistor Q3, al medir el voltaje desde el colector de Q3 nos permite obtener como salida un estado en alto cuando no hay llama y uno en bajo cuando este presente, el opto-acoplador es necesario debido a que la tensión de alimentación del circuito detector esta mezclada con una AC de 80V, para evitar que esta AC circule hacia el microcontrolador se implementa dicho opto-acoplador que permitirá aislar eléctricamente al microcontrolador con el detector de llama ademas de poder leer el estado del transistor Q3.

Rebobinado del transformador

El transformador utilizado en el detector de llama necesita un voltaje en especifico en el secundario pero encontrar un transformador de estas características es muy poco probable por lo que se decidió rebobinar un transformador disponible, el transformador que se necesita posee las siguientes características.

- Devanado primario: 120V AC.
- Primer devanado secundario: 9.5V AC.
- Segundo devanado secundario: 80V AC.
- Potencia de núcleo: mínimo 2 VA.

Las medidas del transformador disponible son los siguiente:

$$A=1.4\,cm \qquad B=1\,cm \qquad E_1=120\,V\,AC \qquad E_2=12.5\,V\,AC$$

Se procede a calcular[5] la potencia y el área del transformador.

$$EC.11 \qquad S_n=A(B)=1.4\,cm(1cm)=1.4\,cm^2$$

$$EC.12 \qquad (2)P_{VA}=(S_n)^2=(1.4\,cm^2)^2=1.96\,VA\approx 2\,VA$$

Como el núcleo del transformador posee la característica mas importante, que es la potencia, se decide re-devanar el transformador, en este caso solamente se necesitara cambiar su secundario y se dejara intacto su primario ya que este ya es de 120V AC, para el calculo del secundario se realiza lo siguientes.

$$EC.13 \qquad \alpha=\frac{42}{S_n}=\frac{42}{1.4\,cm^2}=30\,espiras/V$$

$$EC.14 \qquad \omega_{2-1}=E_{2-1}(\alpha)=9.5V\,(30\,espiras/V)=285\,espiras$$

$$EC.15 \qquad \omega_{2-2}=E_{2-2}(\alpha)=80V\,(30\,espiras/V)=2400\,espiras$$

5 Una explicación mas detallada se encuentra en la sección 8 del marco teórico.

Ya obtenido la cantidad de vueltas queda decidir el calibre del alambre al obtener la corriente máxima del secundario mediante la siguiente formula.

$$EC.16 \qquad I = \frac{P_{VA}}{E_{2-1}} = \frac{1.96\,VA}{9.5\,V} = 0.21\,A$$

Ya obtenido el valor de la corriente máxima se procede a buscar en la tabla AWG el calibre del alambre que soporte esta corriente, el elegido es el siguiente.

AWG	Diámetro mm	Sección mm^2	espiras por cm	Kg por Km	Resistencia Ω por Km	Cap. De Corriente Amperios
28	0.3211	0.08	28.4	0.72	212.50	0.23
29	0.2859	0.064	32.4	0.57	265.60	0.18
30	0.2546	0.051	35.6	0.45	333.30	0.15

Fig.40 Calibre de alambre elegido desde una tabla AWG.

Ya que las corrientes son muy bajas y no se esta seguro que se termine demandando la potencia máxima al transformador, se decidió realizar ambos devanados del secundario con el mismo calibre de alambre numero 28 AWG, ya que realizando los cálculos del devanado de 80V AC este sera de un calibre mucho mas pequeño debido a que entregara mucho menor corriente por ser de mayor voltaje y no es fácil conseguir calibres de alambre tan delgados.

2.2.4 Circuito para la interfaz de usuario

La interfaz de usuario consta de dos elementos, una pantalla LCD 16X2 para mostrar los datos importantes para el usuario y un teclado matricial 4X4 para el ingreso de datos e instrucciones, aunque sea un dispositivo adquirido la LCD debe realizarse un pequeño circuito que permite reducir la cantidad de lineas de datos que utiliza, mientras que el teclado se decidió construirlo desde cero ya que es un dispositivo muy sencillo y permitirá hacerlo a gusto del autor, a continuación se explica cada uno de estos circuitos.

Base para LCD 16X2

La pantalla LCD es un dispositivo que se comercializa actualmente y su costo no es elevado, es una excelente opción para mostrar los datos que nos entereza mostrar al usuario que son valores numéricos cortos de Temperatura y tiempo a como también mostrar un menú sencillo, dicho dispositivo posee 16 pines de conexión. No todos estos pines se necesitaran conectar al microcontrolador ya que se utilizara una librería para el uso del LCD, esta librería solo necesita los pines de datos del D4 al D7 mas los pines RS y E.

También están los pines del LED para retroalimentación de la pantalla, estos se pueden conectar directamente a los pines de alimentación con su respectiva resistencia limitadora para que este se ilumine en el momento en que se alimente la LCD, ademas también esta el pin V0 para el contraste de los caracteres a mostrar que se puede configurar de forma permanente mediante un trimpot, todo esto estará ya configurado en el circuito diseñado para la base del LCD por lo que solo se dejara los pines necesarios de comunicación con el microcontrolador y alimentación de la LCD, circuito es el siguiente.

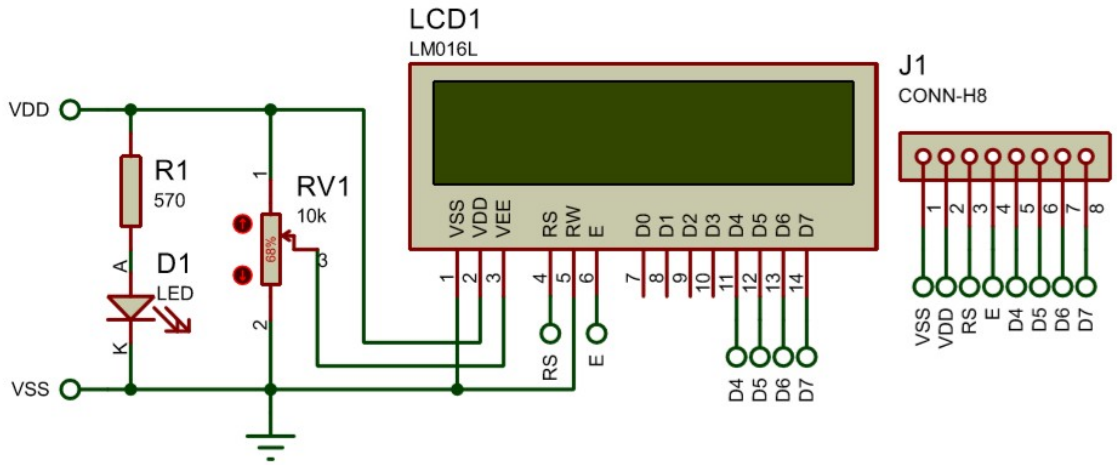


Fig.41 Circuito eléctrico para la base de la LCD 16X2.

Implementando este circuito para la LCD 16X2 se logra disminuir la cantidad de pines de 16 a directamente 8 pines.

Teclado matricial 4X4

En el prototipo se necesita una cantidad considerable de botones para el ingreso de datos de forma cómoda por parte del usuario, de forma tradicional un microcontrolador puede leer un botón utilizando solo una de sus entradas digitales, en el caso del prototipo de control de horno este necesitara por lo menos 10 botones solo para tomar lectura de números en un teclado numérico, esto supondría tener que utilizar 10 pines para poder leer un teclado de solo números, esto supone un uso excesivo de pines que se dedicaran solamente a leer botones por lo que se debe usar alguna técnica para disminuir la cantidad de pines necesarios, esta técnica es el teclado matricial.

Este tipo de teclado como lo indica su nombre esta basado en un arreglo de botones en forma de matriz de 4 filas y 4 columnas siendo en total 16 botones, al acomodarles de esta forma se reduce la cantidad de pines necesarios para tener 16 botones en solamente 8 pines, de forma tradicional tendrían que ser 16 pines necesarios.

El principio de funcionamiento de este teclado se basa en que al momento de presionar cualquier botón el microcontrolador detectara que los pines correspondientes a una columna y fila en especifico se pondrán en corto, cada

botón permitirá el paso de corriente de una fila a una columna en especifico de esta forma se puede saber que botón se ha presionado, acá el circuito del teclado matricial.

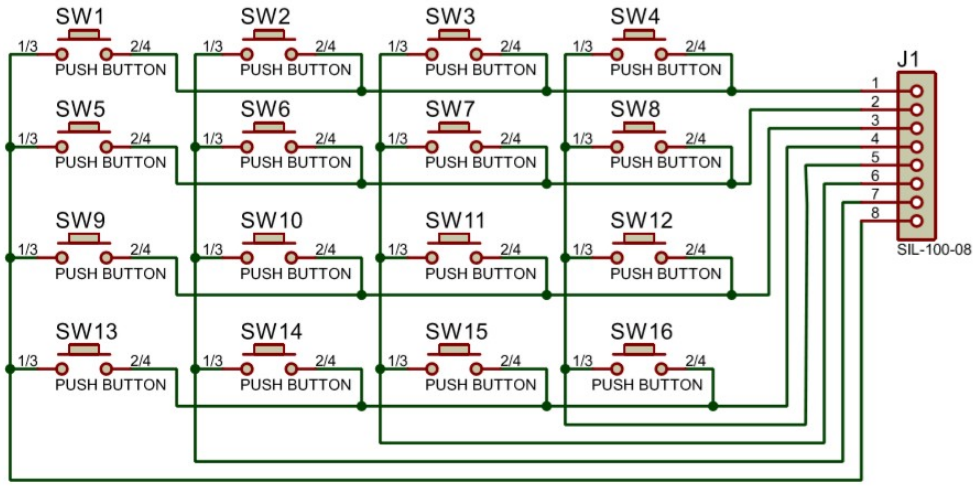


Fig.42 Circuito correspondiente al arreglo matricial del teclado.

2.2.3 Circuito para salidas con relé

En el horno industrial para pan como se ha mencionado se utilizan electro-válvulas de pase de gas ademas de un generador de chispa externo, estos dispositivos se comercializan generalmente para ser alimentados al voltaje del del suministro eléctrico comercial que es de 120V AC a 60 Hz en el caso de Nicaragua, por lo que el microcontrolador debe poder manipularlos con solo los 5 V DC que puede proporcionar a su salida, por esto es necesario utilizar los ya conocidos y ampliamente utilizados relés, los mas comunes son los de bobina a 12V con contactos que soportan 10A a 120V AC, acá una imagen de estos relés.

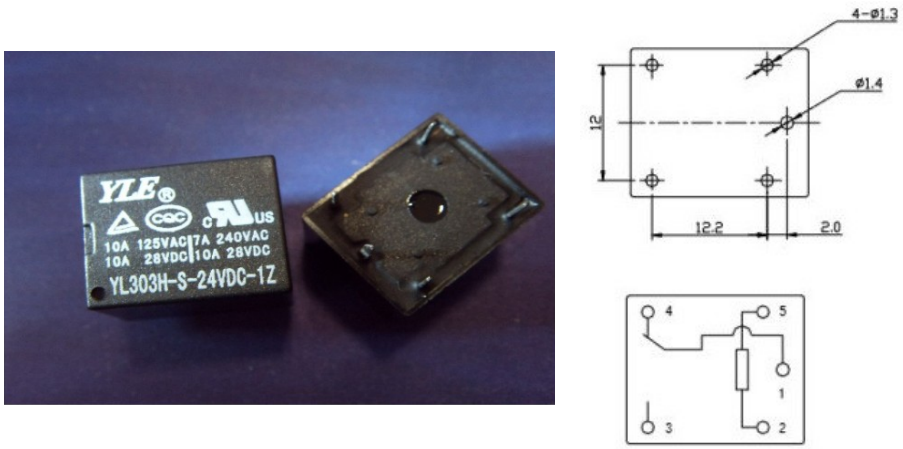


Fig.43 Relé 12V DC. Fuente: DataSheet YL303H-S-12VDC-1Z.

Ya que el microcontrolador trabaja con 5V DC se debe usar una fuente de 12V DC para el proyecto con lo que la alimentación del microcontrolador sera regulada a 5V DC, esto porque los relés consumen mucha mas corriente que el microcontrolador, para que el microcontrolador puede alimentar a voluntad los relés 12V se utiliza el ya conocido transistor en su configuración corte-saturación, el circuito final de las salidas de relés es el siguiente.

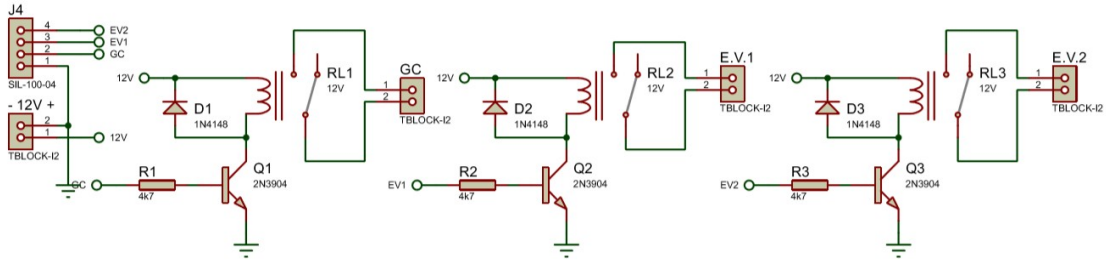


Fig.44 Circuito con transistores que permite controlar tres relés.

El microcontrolador solo tendrá que utilizar 3 pines para activar los relés, si las fuentes son distintas se debe unir ambos negativos para que se permita una circulación de la corriente desde el micro hasta el transistor. El circuito es sencillo, el relé se pone en serie con el colector-emisor del transistor, se agrega un diodo en anti-paralelo con el relé verificando que el cátodo quede conectado al positivo de la alimentación del circuito, el pin de salida del microcontrolador se conecta a una resistencia limitadora para luego conectarla a la base del transistor, para elegir la resistencia limitadora adecuada se debe saber el consumo del relé a controlar con el transistor, se puede calcular la corriente que consume el relé mediante la ley de ohm, por lo que se mide la resistencia de la bobina del relé que resulta ser de 440Ω.

$$EC.17 \qquad I_C = \frac{V_{CC}}{R_{RL}} = \frac{12V}{440\Omega} = 0.027\,A$$

Sabiendo la corriente de consumo del relé se calcula la resistencia limitadora utilizando la ganancia del transistor, en este caso se utiliza el C9014 con β = 400, este transistor es muy parecido a los conocidos 2N2222 y BC548, se eligió el C9014 debido a su disponibilidad y bajo precio pero como realiza una tarea no muy exigente se puede utilizar cualquiera de uso general tipo NPN, acá el calculo de la resistencia limitadora.(Boylestad & Nashelsky, 2009)

$$EC.18 \qquad I_B = \frac{I_C}{\beta + 1} = \frac{27\,mA}{440+1} = 61\,\mu A \rightarrow (I_B)10 = 0.61\,mA$$

Algunos autores de libros de diseño con transistores advierten de la dependencia que posee la ganancia de un transistor con la temperatura, el transistor disminuye su ganancia al aumentar la temperatura, por esta razón a veces recomiendan aumentar la corriente de base del transistor para poder prever este problema, esta corriente se puede aumentar ya que el transistor puede soportar una corriente de hasta 50 mA máximo en la base, he aquí de

que al calcular la corriente de base se decide multiplicarla por 10 para tomar en cuenta estos efectos.

$$EC.19 \qquad R_B = \frac{V_{CC} - V_{BE}}{I_B} = \frac{12\,V - 0.7}{0.61\,mA} = 18.5\,k\,\Omega \rightarrow 4.7\,k\,\Omega$$

$$EC.20 \qquad I_{B \rightarrow 4.7\,k\,\Omega} = \frac{V_{CC} - V_{BE}}{R_B} = \frac{12\,V - 0.7\,V}{4.7\,k\,\Omega} = 2.4\,mA$$

Al realizar el calculo nos damos cuenta que la resistencia limitadora teórica es de casi 20 kΩ por lo que se considera que se puede aumentar la corriente de base aun mas por lo que reduciremos esta resistencia, se decide usar una resistencia de 4.7 kΩ esta es muy común y permitirá asegurar que el efecto de la temperatura no afecte, mas si se planea utilizar el prototipo en un horno, la corriente de base que obtendremos con la nueva resistencia a aumentado hasta 2.4 mA, esta corriente no podrá dañar al transistor a largo plazo.

2.2.5 Fuente de alimentación

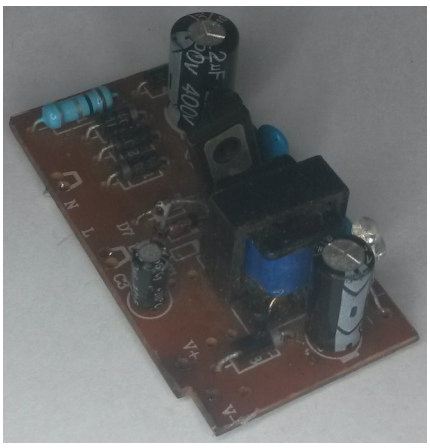


Fig.45 Electrónica de una fuente conmutada utilizada para cargar teléfonos móviles.

El prototipo del control de horno necesita una alimentación de 12V DC y una corriente máxima de 150 mA, como se aprecia no es mucho el consumo por lo que la fuente puede ser pequeña, entre las fuentes de alimentación a elegir hay dos opciones, las lineales o las conmutadas ambas se pueden utilizar en el prototipo, pero hay algunos inconvenientes que nos permite elegir la adecuada.

En el caso de las fuentes lineales, estas se basan en un transformador con núcleo de hierro al silicio, estos transformadores solo se pueden encontrar con dificultad como mínimo de 5VA que es mas de lo que se necesita (alrededor de 2W) con un coste elevado, ademas los transformadores pierden eficiencia al sud-utilizarlos, ademas ya se esta utilizando un transformador de este tipo en el circuito detector de llama, en este circuito la alimentación esta mezclada con una alterna de 80V AC debido al diseño del detector por lo que no se puede alimentar al microcontrolador con dicho transformador, ademas de ocupar mas espacio un transformador pesa, por lo que es inviable utilizar otro mas solo para alimentar el microcontrolador por lo que realizar una fuente lineal para alimentar al microcontrolador queda descartada.

Visto lo anterior se decide utilizar una fuente conmutada típica utilizada para cargar teléfonos móviles, estas se encuentran comercialmente fácilmente y pueden entregar de 500 a 2000 mA dependiendo de cual se compre, son

ligeras y aunque mas compleja que una fuente lineal, gracias a la tecnología SMD pueden llegar a ocupar menos espacio, estos cargadores para teléfonos móviles son un producto muy demandado por lo que su costo disminuye bastante debido a la competencia existente en el mercado, ademas estas suelen ser construidas en base a un único integrado mas algunos componentes discretos por lo que podrían llegar a integrarse con el resto del circuito, las empresas ponen a disposición en el datasheet del integrado en que se basan estas fuentes el circuito típico necesario para poder construir este tipo de cargadores por lo que hay la posibilidad de construir la fuente por uno mismo.

Tabla 4
Elección del tipo de fuente a utilizar.

Tipo de Fuente	Disponibilidad	Peso	Costo	Elección
Transformador 5VA (Solo Transformador)	Poco	±100 gr	$20	
Cargador de teléfono (Solo Electrónica)	Si	±9.8 gr	$5	X

Aun así el utilizar un cargador de teléfono como alimentación trae un inconveniente, este es que su salida es de 5V ya que esta tensión es la que utiliza un teléfono y se necesita 12V en el prototipo, pero esto se puede resolver ya que las fuentes conmutadas utilizan voltajes de referencia para poder configurar y mantener su salida por lo que solo hay que modificar esta referencia, en los cargadores de teléfono es muy sencillo ya que utilizan un diodo zener para obtener esta referencia, por lo que solo hay que cambiar este diodo por uno de un valor mayor, ya que generalmente utilizan uno de 4.1V.

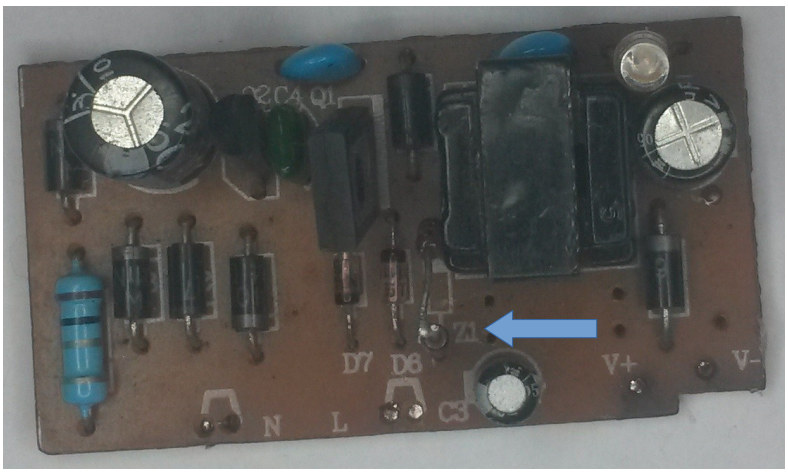


Fig.46 Ubicación del diodo zener utilizado como referencia en una fuente de teléfono móvil.

En la imagen se aprecia donde se encuentra dicho diodo zener en la placa de la fuente, solo bastara remplazarlo por uno de 11V respetando la mascara de componentes del diodo nombrado como “Z1”, ¿pero no modificara en algo el comportamiento de la fuente al realizar este cambio ademas del voltaje de salida?, en cuanto al comportamiento de la fuente no abra ningún cambio pero si se reducirá la corriente máxima que puede entregar la fuente, pero como el prototipo no demandara mas de 150 mA, este cambio no nos perjudica ya que la fuente en si podía entregar 500 mA y la reducción lo dejara en una corriente máxima de entrega de 200 mA.

$$EC.21 \qquad P = V(I_{Max}) = 5\,V(500\,mA) = 2.5\,W$$

$$EC.22 \qquad I_{Max\ Nueva} = \frac{P}{V_{Nuevo}} = \frac{2.5\,W}{12\,V} = 208\,mA$$

2.2.6 Protección al ruido eléctrico del prototipo

Realizando pruebas entre algunas de las partes del prototipo pudimos notar que el microcontrolador fue afectado por el funcionamiento del generador de chispa, cada ves que se producía una chispa el microcontrolador actuaba errático, claramente el salto de la chispa de alto voltaje producía el ruido eléctrico, el único medio por el cual ambos el microcontrolador y el generador de chispa están relacionados es el compartir el suministro eléctrico de 120V AC, viendo esto se decidió agaragar un filtro EMI en la alimentación total del prototipo, el circuito correspondiente a este filtro es el siguiente.

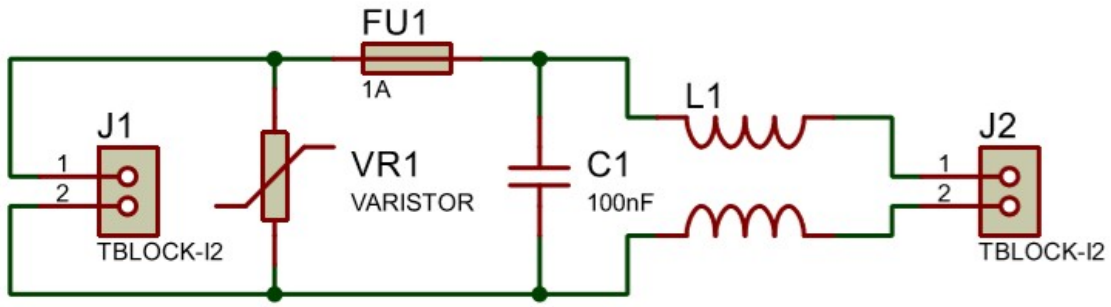


Fig.47 Filtro utilizado para proteger al prototipo contra el ruido eléctrico.

Como se puede ver el filtro no esta completo para poder ser un filtro EMI, esto debido a que mediante prueba se descubrió que solo el inductor era necesario para eliminar las interferencia provenientes del generador de chispa, pero para aumentar la protección se decidió agregar un varistor de 200V, un fusible de 1A y un condensador se 100 nF del tipo X2, el varistor protegerá en caso de sobre-impulsos o subidas de tensión, estas también las puede producir las electro-válvulas, el fusible es del tipo rápido, se agrega en caso de ocurrir un corto dentro del prototipo, y por ultimo el condensador como un filtro capacitivo para filtrar ruido de alta frecuencia que pueda venir del exterior.

Para el funcionamiento del prototipo con esta protección es suficiente, pero al momento de utilizar el prototipo para tomar datos, conectando este a un ordenador ocurrió otro problema, cada ves que se desconectaba la electro-válvula del soplete en su conmutación con el relé el ordenador podía perder la conexión serial por medio de USB con el prototipo, se pudo deducir que el ruido proviene de la corriente inversa producida por la bobina de la electro-válvula al momento de ser desconectada, esta debió ser de alto voltaje por lo que viajo por el relé al microcontrolador y luego al ordenador, se decidió agregar a la electro-válvula un circuito que se encargara de descargar esta corriente inversa para evitar la desconexión y permitir tomar los datos cómodamente, la modificación fue la siguiente.

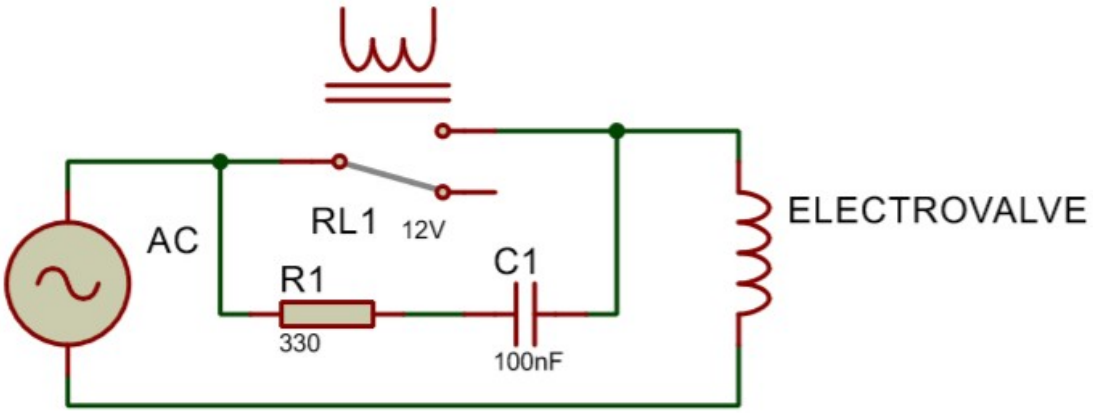


Fig.48 Circuito para descargar la corriente inversa proveniente de la electro-válvula.

Como se puede ver en le circuito se acoplo una serie de un condensador y resistor en paralelo con los contactos de relé utilizado en la conmutación de la electro-válvula, esta seria es saltada al momento de estar cerrado los contactos del relé por lo que la electro-válvula es alimentada sin problemas, al momento de abrirse los contactos el circuito queda en serie con la electro-válvula, acá es donde el condensador y la resistencia entran en marcha adsorbiendo la energía contenida en la bobina evitando así que se genere el ruido eléctrico.

Este circuito solo fue necesario utilizarlo para poder tomar datos con el ordenador, ya que al microcontrolador no le afecta el uso de la electro-válvula, por lo que se retira una ves tomados los datos de interés.

2.2.7 Unificación de las partes en un circuito

Una ves que se han montado y probado las partes del prototipo ha llegado la hora de unificarlas en un solo circuito y seguido realizar pruebas de funcionamiento, el circuito resultante de unir todas las parte se puede ver continuación.

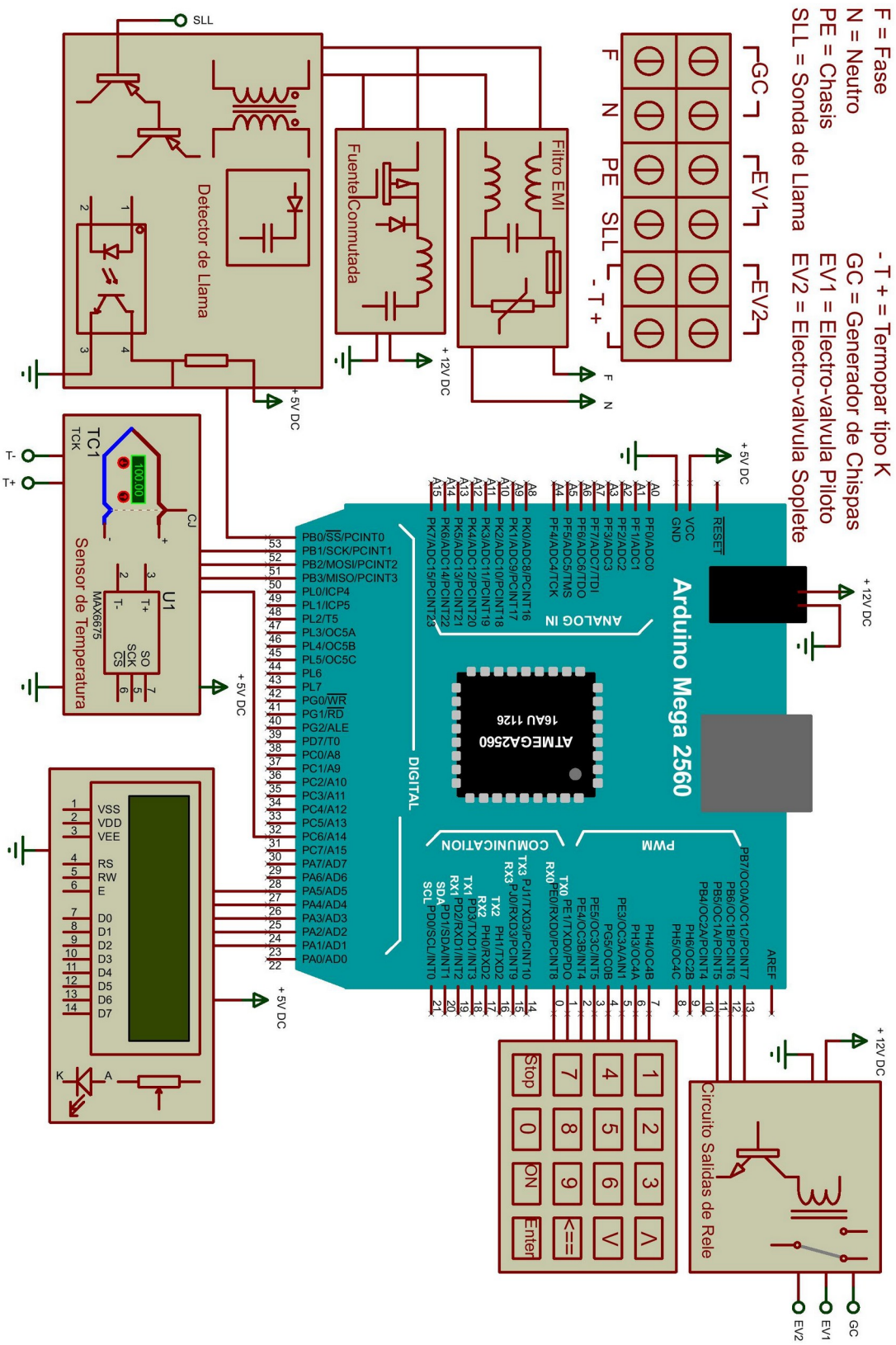


Fig.48 Circuito correspondiente a la unión total de las partes del prototipo.

Como se puede apreciar el prototipo contara con 12 bornes para realizar las conexiones con el resto del circuito del horno, en estos bornes solo sera necesario conectar lo que se lista a continuación.

- F: Fase correspondiente al suministro eléctrico 120V AC.
- N: Neutro correspondiente al suministro eléctrico 120V AC.
- PE: Conexión al chasis del horno que generalmente debe estar aterrizado.
- SLL: Acá es donde se debe conectar la sonda que se utilizara para detectar la llama.
- - T +: En estos dos bornes se debe conectar la termopar tipo K respetando la polaridad.
- GC: Entre estos bornes se debe conectar en serie la alimentación del generador de chispa para que sea controlada por el control de horno.
- EV1:Entre estos bornes se encuentra los contactos para controlar la electro-válvula de pase de gas del piloto.
- EV2:Entre estos bornes se encuentra los contactos para controlar la electro-válvula de pase de gas del soplete.

2.2.8 Programación del microcontrolador

2.2.8.1 Control PID

El control PID se puede realizar desde dos puntos, se puede ingresar la formula característica del PID en el código a programar o bien mas sencillo y menos complicado mandar a llamar a realizar su calculo desde una librería ya disponible, el IDE de Arduino como también los foros de Arduino se encuentran disponible estas librerías PID.(Brett Beauregard, 2018)

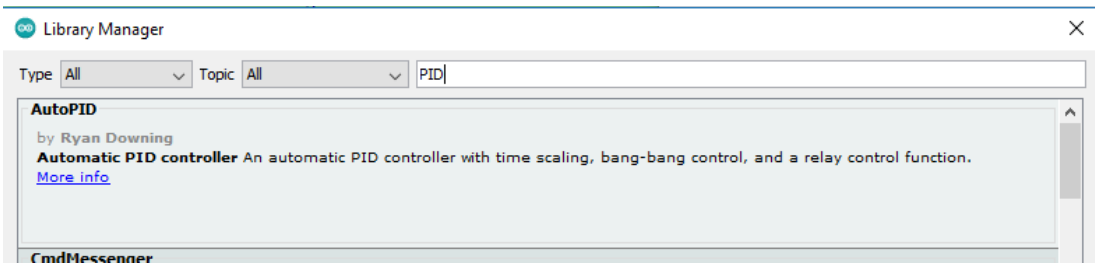


Fig.49 Buscando librerías desde el Arduino IDE.

La librería a utilizar es la llamada “PID_v1.h” esta solo nos pide algunas variables relacionadas al PID para utilizarla, estos son:

- Input: La igualaremos al valor que recibimos del sensor a utilizar.
- Output: Sera la salida del calculo del PID.
- Setpoint: Lo igualaremos al valor deseado que se quiere alcanzar (temperatura, posición, etc.).
- Kp: Sera la constante proporcional.
- Ki: Sera la constante integral.
- Kd: Sera la constante derivativa.

Igualmente se debe configurar algunas funciones de la librería “PID_v1.h” que nos interesan, estas son:(Brett Beauregard, 2018)

- PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT); Esta función simplemente asocia las variables antes creadas con la librería del PID de forma que la librearía sepa donde debe leer y poner los datos. El único detalle de mas es la sección donde dice “DIRECT”; Puede ser “DIRECT” o “REVERSE”, este determina en qué dirección se moverá la salida ante un error determinado. “DIRECT” es lo más común.
- myPID.SetOutputLimits(0, 100); En esta función se define los limites del controlador PID, este está diseñado para variar su salida dentro de un rango determinado. Por defecto, este rango es 0-255: el rango Arduino PWM. No sirve de nada enviar 300, 400 o 500 al PWM. Sin embargo, dependiendo de la aplicación, se puede desear un rango diferente, en el caso de querer definir el porcentaje del ciclo de trabajo lo definimos así “(0, 100)”.
- myPID.SetMode(AUTOMATIC); Especifica si el PID debe estar activado (AUTOMATIC) o desactivado (MANUAL). El PID se establece por defecto en la posición desactivada cuando se crea. Por lo que esta funcion solo funciona para encender el PID.
- myPID.Compute(); Contiene el algoritmo PID, al llamarse una vez en cada ciclo realiza el calculo y lo almacena en la variable designada como salida.

Como se puede ver en las descripción de las variables usadas por la librería PID, esta no nos resuelve el paso de encontrar las constantes “kp”, “ki” y “kd” estas se deben de ingresar dependiendo del sistema, para eso se debe de realizar un proceso de calculo sintonizador para encontrar los valores de estas constantes.

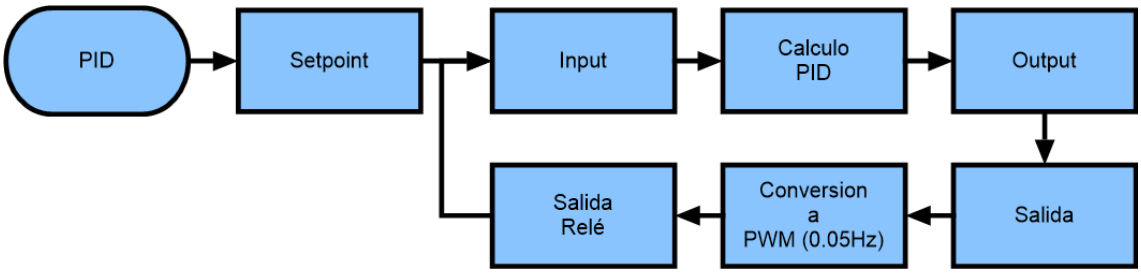


Fig.50 Algoritmo para el uso del PID en el prototipo.

Este sera algoritmo que realizara el microcontrolador para el calculo del PID se pretende utilizar una frecuencia de 1 Hz, osea medir cada segundo, por lo que el microcontrolador realizara todo este recorrido cada segundo.

Sintonización PID: Método de Ziegler y Nichols en lazo abierto

Para sintonizar el controlador PID se eligió por el método de Ziegler-Nichols en lazo abierto o también conocido como de la curva de respuesta, esta decisión se debe a que este método utiliza una curva de respuesta de la planta a controlar, osea registra el comportamiento de la planta, ya que la respuesta transitoria de la temperatura en el horno no es tan rápida es mas viable sintonizar el PID por este método, ademas de que este método es el mas

comúnmente usado en el control de hornos; El método intenta generar una salida que siga la pendiente de la curva de respuesta del proceso a controlar, ya registrado el comportamiento de la planta, se debe generar una grafica de la respuesta transitoria de la planta, desde esta grafica se calculara las constantes que se deben utilizar en el control PID. La planta como se sabe en este caso es el horno a gas, en lazo abierto muchos procesos pueden definirse según la siguiente función de transferencia de primer orden más tiempo muerto, esta se ve así.(Adonayt Ruge)

$$EC.23 \qquad G(s)=\frac{K_0 e^{-s\tau_0}}{1+\gamma_0 s}$$

En dicho método de Ziegler-Nichols en lazo abierto, no es necesario tener que realizar el modelo matemático del horno, sino podemos basarnos en la grafica del comportamiento del horno, para obtener la grafica de la curva de respuesta se debe realizar un pequeño DataLogger, se puede ocupar el mismo microcontrolador que se utilizara en el sistema de control con sus sensores para poder leer y almacenar los datos o enviar los datos por serial, para realizar la lectura y generar la grafica lo que se debe hacer es abrir el lazo de retroalimentación del sistema para aplicar una orden fija en la entrada del microcontrolador, esto generara un escalón en la salida, el cual es el que debemos graficar, se decidió tomar lecturas de la temperatura hasta que el escalón en la salida se estabilice ingresando una orden de un ciclo de trabajo del 20% en la entrada, teniendo una frecuencia de muestreo de 1Hz en las lecturas, osea tomar la una medición cada segundo, para esto se creo un pequeño código que lee la temperatura y marca el tiempo transcurrido para luego ser enviado por el puerto serial, en la siguiente imagen se puede apreciar el diagrama de flujo.

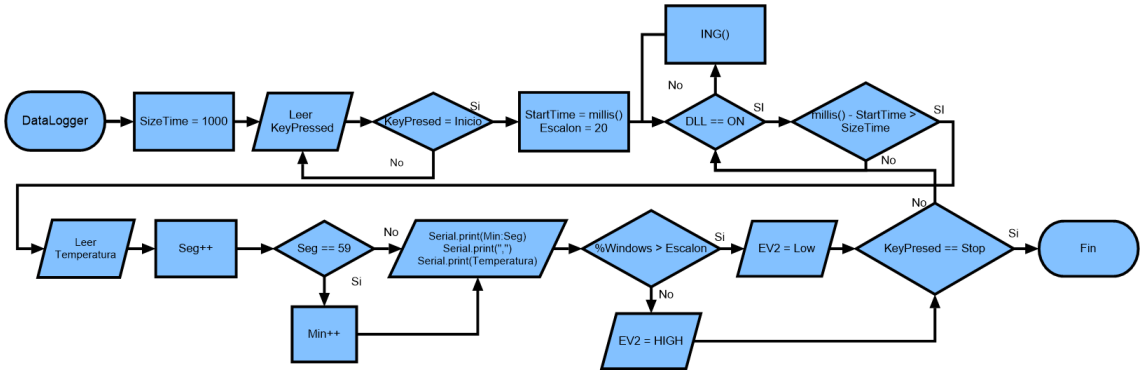


Fig.51 Diagrama de flujo para la toma de datos, curva de respuesta horno a gas.

Como se puede apreciar en el diagrama de flujo los datos leídos son enviados por el puerto serie, estos datos serán capturado desde el puerto serie para ser almacenados en un documento de texto plano, también se aprecia en el diagrama que primero se enviá el tiempo transcurrido (Minuto, segundos), luego “,” y por ultimo la temperatura, la coma es enviada simplemente para separar los datos de tiempo con los de temperatura, una ves guardados en un

archivo .txt, para lograr capturar los datos enviados por el microcontrolador se decidió realizar un pequeño programa[6] en Python 3.8.3 que lea los datos en el puerto serial e inmediatamente los almacene en un archivo llamada "DatosTT.txt", el funcionamiento de este script de python es muy sencillo se puede ver en la siguiente imagen la lógica de ejecución.

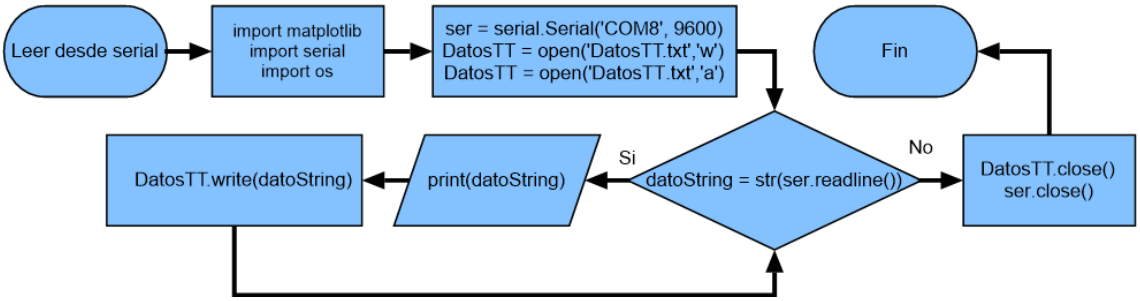


Fig.52 Diagrama de flujo que explica el script hecho en python 3.8.3.

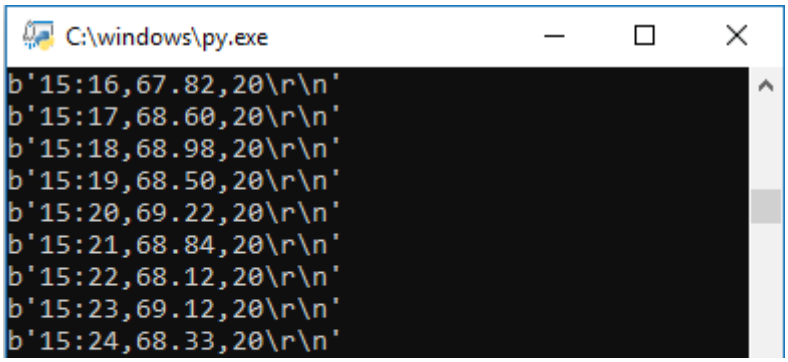
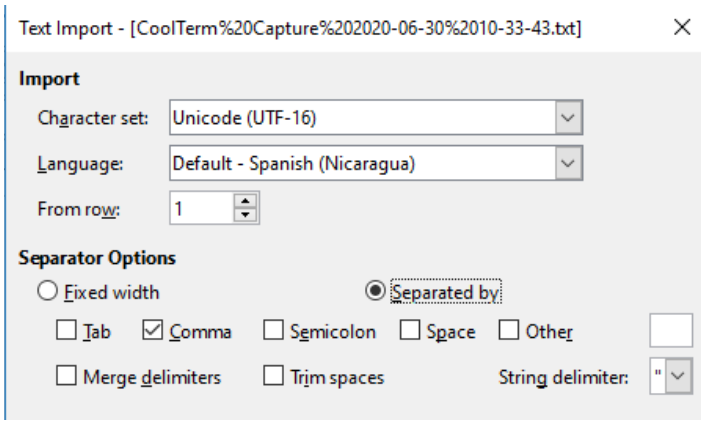


Fig.53 Los datos recibidos desde el serial.

Una ves terminado y obtenido el archivo, este se arrastra sobre una hoja de calculo donde nos da la opción de ubicar los datos separados por una coma en dos columnas.



	A	B
1	0:00	37.9
2	0:00	38.07
3	0:00	36.82
4	0:00	38.82
5	0:00	38.72
6	0:00	37.63
7	0:00	38.65
8	0:00	39.8
9	0:02	39.19
10	0:04	38.62
11	0:06	38.92

Fig.54 Los datos obtenidos sobre una hoja de calculo, LibreOffice Calc.

Con los datos en una hoja de calculo solo falta crear la grafica en base a las opciones que nos ofrece dicho programa de hoja de calculo, la grafica obtenida es la siguiente.

6 Programa en Python 3.8.3 sección 4 Anexos

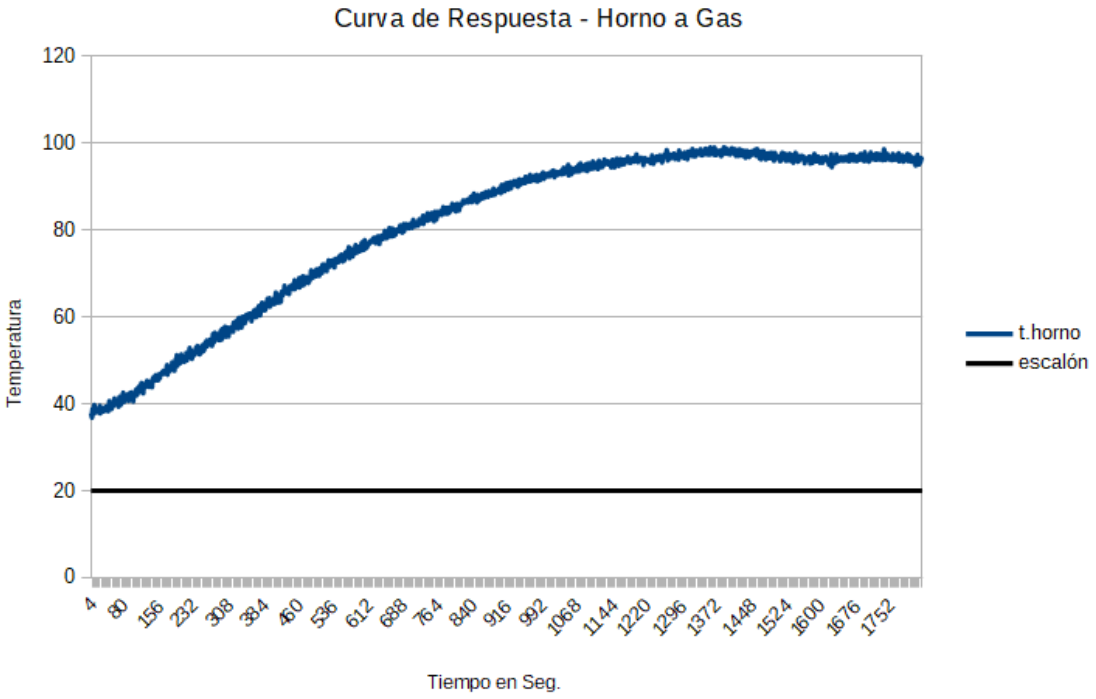


Fig.55 Curva de respuesta transitoria del horno a gas.

Ya obtenida la grafica de la curva de respuesta del horno se puede sacar los parámetros necesarios que se utilizaran en el calculo de sintonización del PID, los datos que se necesitan extraer de la grafica son los siguientes.

- t_0 : Tiempo exacto donde se inyecto el escalón en el sistema en lazo abierto.
- t_1 : Tiempo exacto donde se produce un cambio en la salida.
- t_2 : Tiempo donde se intercepta la recta que pasa por el punto de inflexión y la recta de la temperatura alcanzada con el escalón.
- y_0 : Valor de inicio del estado del sistema, en nuestro caso temperatura de inicio.
- y_1 : Temperatura alcanzada.
- u_0 : Escalón inicial.
- u_1 : Escalón ingresado o final.

Para obtener estos datos lo importante es crear una recta que pase por dos puntos en especifico, uno es el punto donde produce el cambio en la salida y el segundo es el punto de inflexión de la curva de respuesta, este punto llamado de inflexión es aquel donde la curva cambia su sentido, en este caso el punto donde la curva deja de ser ascendente para empezar a decender gradualmente, se pueden realizar cálculos matemáticos para obtener esta recta o trazarlo encima de la grafica de manera manual, en nuestro caso se decidió realizarlo de manera manual, la siguiente imagen muestra el trazado y los datos obtenidos de la grafica.(Adonayt Ruge)

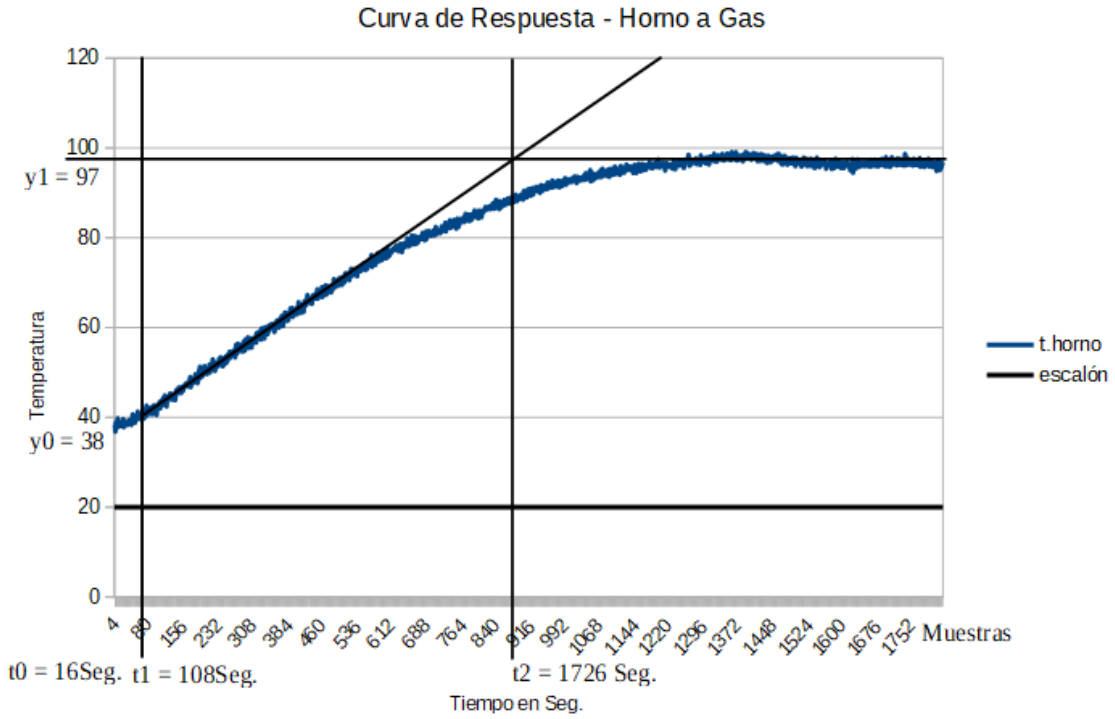


Fig.56 Trazado realizado sobre curva de respuesta del horno a gas.

Una vez obtenidos los datos se utilizan las siguientes ecuaciones proporcionadas por Ziegler-Nichols.

$$EC.24 \quad (1)\tau_0 = t_1 - t_0 = 108 - 16 = 92 \quad (2)y_0 = t_2 - t_1 = 1726 - 108 = 1618$$

$$(3)k_0 = \frac{y_1 - y_0}{u_1 - u_0} = \frac{97 - 38}{20 - 0} = 2.95$$

Según Ziegler-Nichols, la relación de estos coeficientes con los parámetros del controlador son:

$$EC.25 \quad (1)K_P = 1.2\frac{y_0}{k_0\tau_0} = 1.2\frac{1618}{(2.95)(92)} = 7.15 \quad (2)T_I = 2\tau_0 = 2(92) = 184$$

$$(3)T_D = 0.5\tau_0 = 0.5(92) = 46$$

En el método de Ziegler-Nichols se toma un tiempo de muestreo (T), en el modelo ellos aconsejan un valor que cumpla esta condición $T < \tau_0/4$, el controlador que se pretende diseñar tendrá un tiempo de muestreo de 1 segundo. Ya obtenido el resultado de estos datos es donde podemos calcular los valores finales que utilizaremos para sintonizar el PID.(Adonayt Ruge)

$$EC.26 \quad (1)k_P = K_p = 7.15 \quad (2)k_i = \frac{K_p(T)}{T_I} = \frac{(7.15)(1)}{184} = 0.04$$

$$(3)k_d = \frac{K_p(T_D)}{T} = \frac{(7.15)(46)}{1} = 328.9$$

Por ultimo estos son los valores de las contantes que utilizaremos en el controlador PID.

2.2.8.2 Adecuar modulación PWM

Una ves obtenido el calculo del control PID ahora nos toca convertirlo en una salida física del controlador mediante un actuador, pero aquí se genera un problema, debido a que el resultado del PID es un valor que varia, este espera que se aplique a una salida regulable, osea que pueda adoptar varios valores entre dos limites, en cuanto al soplete del horno, este solo puede tomar dos valores, encendido o apagado, por lo que hay que traducir y adecuar la salida ya calculada del PID.

Originalmente el PID se puede aplicar a una salida PWM, como se sabe el PWM es simplemente el variar del ciclo de trabajo de una señal cuadrada, pero dicha señal cuadrada originalmente se utiliza con una frecuencia alta, el cambiar el ciclo de trabajo produce que se pueda controlar el voltaje promedio de la señal que se aplica, podemos crear un PWM utilizando una frecuencia muy baja de manera que pueda trabajar con un relé, así ya no importaría el voltaje promedio de la señal, es mas ya no tendría sentido medir voltaje promedio, sino que nos enfocamos en el tiempo que pasara en alto o encendido el relé.

Es tan baja la frecuencia a utilizar que es mejor verlo desde el punto de vista del periodo de la señal que de la frecuencia, la pregunta ahora es saber que periodo utilizar, tomando como herramienta un criterio de ingeniería podemos basarnos en el comportamiento del soplete, como el conjunto del soplete y la electro-válvula son un actuador con respuesta lenta podemos utilizar el tiempo mínimo que necesita nuestro actuador para generar calor como la mínima salida del PID, realizando pruebas se estima que en promedio el soplete necesita un segundo para poder comenzar a generar calor, esto debido a la acción de la electro-válvula que al accionarse permite el pase de gas para luego este tener que encenderse con la llama del piloto, como la frecuencia de muestreo del prototipo es de 1 Hz, cada segundo se leerá la temperatura y se generara la salida del PID que inmediatamente se enviá al actuador (electro-válvula y soplete), así que el PID tendrá que esperar otro segundo para poder cambiar la salida, si la salida estaba en alto y se ha alcanzado le setpoin el PID tendrá que esperar a que empiece el siguiente segundo para poder poner la salida en bajo, de esta forma se asegura que el soplete le dio tiempo de calentar sin ser interrumpido.

Sabiendo lo anterior este segundo mínimo supondremos que correspondería a una salida del PID del 5%, dividimos 5 entre 100% nos daría que en nuestro caso el periodo seria de 20 Segundos. Entonces lo siguiente a realizar es como pasar la salida cambiante del PID a un valor ente 0 y 100 que se tomara como porcentajes, en realidad esto ya se realizo anteriormente, al utilizar una librería que realizara el calculo del PID con solo mandarla a llamar se puede configurar un limite de la salida del PID, este se definió como 0 a 100 por lo que lo único que queda es pasar los valores del PID a ciclos de trabajo, a continuación se

muestra como los porcentajes de salida del PID se traducen en un ciclo de trabajo del PWM.

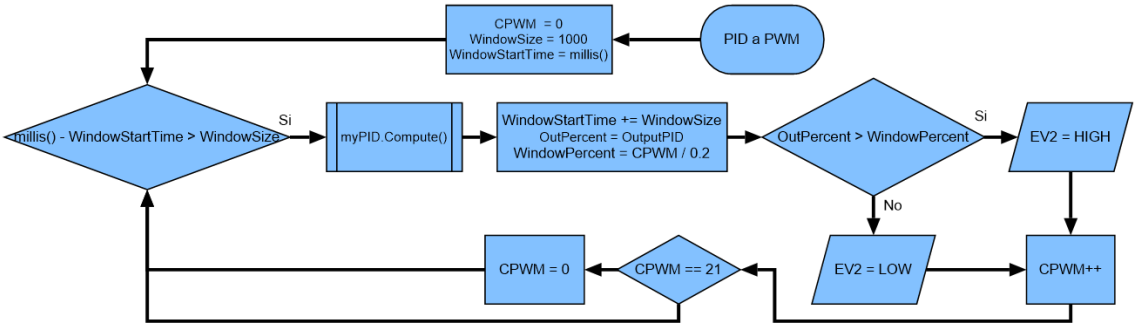


Fig.57 Diagrama de flujo que explica como se traduce la salida PID a una salida para relé.

En el diagrama de flujo anterior muestra como se traduce la salida del PID a dos estados posibles en la salida del prototipo, primero se pregunta si ya ha pasado un segundo con “millis() - WindowStartTime > WindowSize”, si es asi se calcula el PID e inmediatamente se toma en forma de porcentajes de la salida con “OutPercent = OutputPID, luego se calcula el porcentaje del tiempo transcurrido actualmente del periodo de la señal del PWM con “WindowPercent = CPWM / 0.2” (20 / 100 = 0.2 del periodo), “CPWM” es la cuente del periodo en segundos, para luego inmediatamente hacer la comparación de estos dos con “OutPercent > WindowPercent”, lo que se quiere aplicar es que si el PID manda una salida del 50% del periodo en alto, el microcontrolador contara el periodo de 20 segundos (“CPWM”), en cada segundo calcula el porcentaje de tiempo transcurrido hasta entonces, cuando el tiempo este por debajo del 50% lo pondrá en alto la salida del relé con “EV2 = HIGH” y cuando este supere el 50% del periodo lo pondrá en bajo la salida del relé con “EV2 = LOW”.

2.2.8.3 Código de programación

Para tener una mejor comprensión del programa utilizado en el prototipo se recomienda ver el código de programación[7] a la ves que se lee esta sección. El código de programación final se dividió en 5 funciones que se encargaran de realizar una tarea especializada cada vez que se les mande a llamar, se listan así:

- Loop(): Función principal del programa, se encargara de mostrar la pantalla de espera.
- Menu(): Se encarga de mostrar el menú para poder navegar por el.
- GetValue(): Dicha función se encarga de tomar los datos del teclado, mostrarlo en pantalla de forma cómoda y almacenarlos en la memoria ROM del microcontrolador.
- Inicio(): Esta función se encargara de realizar el proceso de horneado, en esta parte se realiza el calculo del PID, se traduce la salida PID a una

7 Código de programación sección 3 Anexos

salida PWM para el relé, se realiza la cuenta regresiva del tiempo de horneado y se detecta la ausencia de llama si es el caso.

- ING(): Esta función se encarga de realizar el encendido del soplete, osea el proceso de ignición.

En el siguiente diagrama de flujo se puede apreciar como se distribuyen estas funciones en el código, esta es una vista sirve como un mapa del código a implementar.

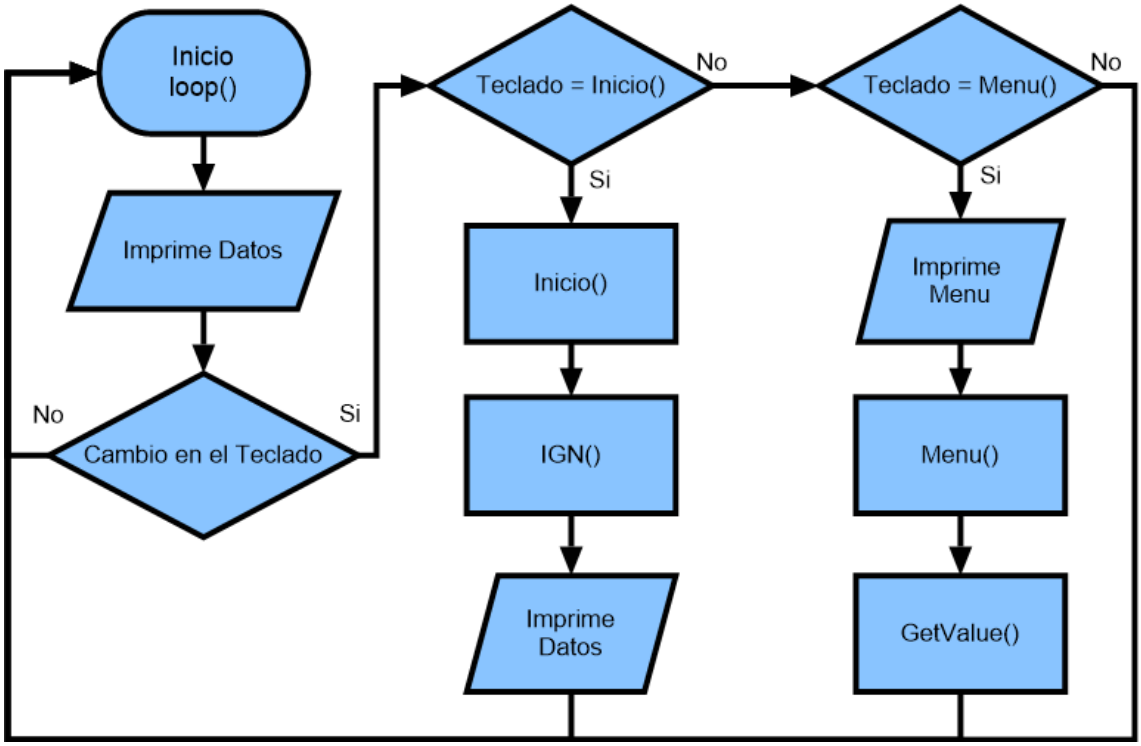


Fig.58 Diagrama de flujo que resume al código de programación del control de horno.

Acá esta todo el recorrido que se podrá realizar en el menú del control de horno, este solo es una vista global de funcionamiento del código, continuación se explicara lo que realiza cada función a mas detalle por separado para mejorar la comprensión del código en su totalidad.

Función principal loop()

Como se menciona en el diagrama de flujo anterior el controlador empieza en la función principal que se encargara de llamar a las demás, esta tendrá una pantalla de espera en donde se mostrara los siguientes datos: temperatura seleccionada, temperatura actual y tiempo seleccionado, a como se podía suponer por su nombre, el microcontrolador imprimirá estos datos en pantalla que serán refrescados cada segundo mientras espera que haya un cambio en el teclado, en el teclado habrá dos botones asignados correspondientes al inicio del horneado (Teclado = Inicio()) y para mostrar el menú de configuración (Teclado = Menu()), estos botones lo que harán es mandar a llamar a su función correspondiente para que realice su tarea en base a lo que decida el usuario, acá se puede ver la lógica de la función “loop()”.

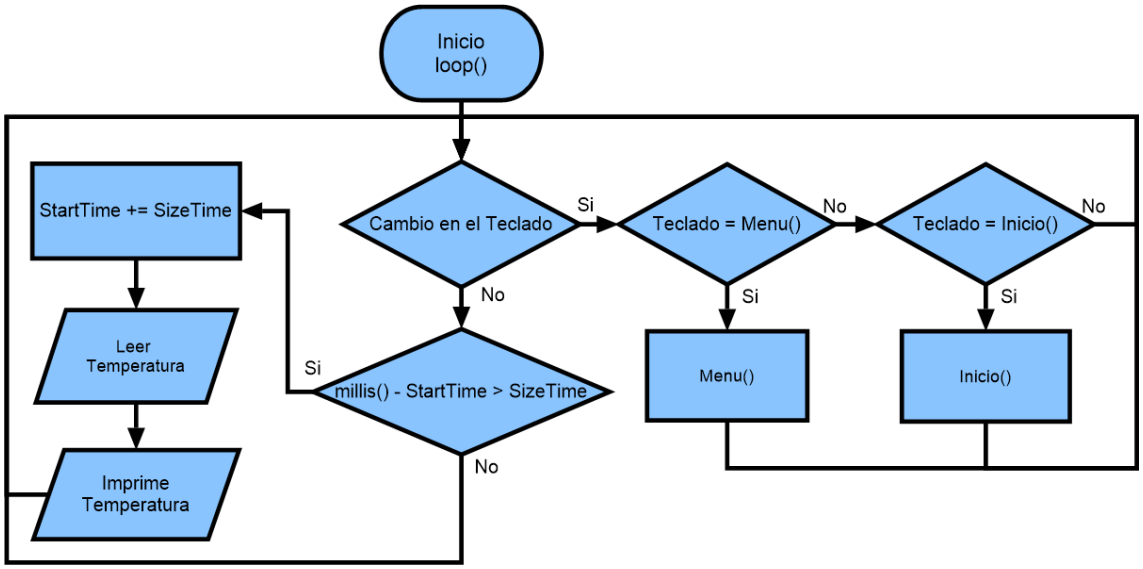


Fig.59 Diagrama de flujo correspondiente a la función principal loop().

La condición “if(millis()-StartTime > SizeTime)” es donde se pregunta si ya ha pasado un segundo, si es verdadero se procede a impremir los datos en pantalla, el resto del tiempo se pregunta si hay algún cambio en el teclado, si se presiona algún botón que no corresponda con los dos únicas opciones, en este caso simplemente se ignora y vuelve a esperar otro cambio del teclado.

Función Menu()

Esta función se encarga de mostrar el menú, que en este caso solo hay dos opciones a ofrecer ‘Set. Temp.” para ingresar los datos de temperatura y “Set. Time.” para ingresar los datos de tiempo, al seleccionar cualquiera de ellas se procederá a tomar los datos llamando a la función GetValue() que los acomoda y almacena en memoria, acá esta el diagrama de flujo de esta función Manu().

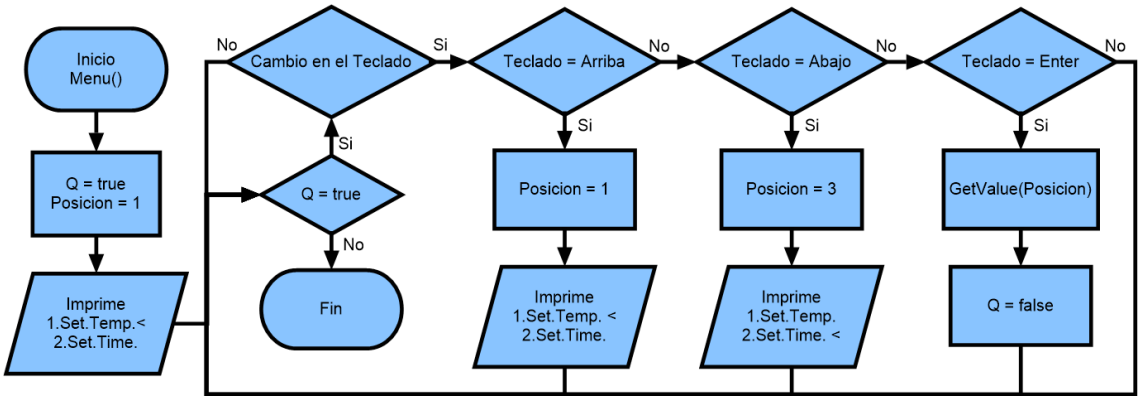


Fig.60 Diagrama de flujo correspondiente a la función Menu().

Como se puede ver en el diagrama al llamar la función se imprime el menú y automáticamente se selecciona la posición 1 en el menú a la ves que la variable “Q” se le asigna el valor de “true” para luego entrar en un bucle donde entran en juego dos condiciones, se pregunta si la variable “Q” sigue en estado

verdadero y si hay algún cambio en el teclado se procede a preguntar si es alguna de las teclas que nos interesan, esta teclas son:

- ❖ Arriba: seleccionara la primera opción al imprimir el cursor en pantalla en la opción uno y pondrá la variable posición igual a 1.
- ❖ Abajo: seleccionara la segunda opción al imprimir el cursor en pantalla en la opción dos y pondrá la variable posición iguala a 2.
- ❖ Enter: Llamara a la función GetValue() para que tome los datos y tomando el valor de posición.

El resto de teclas serán ignoradas ya que no tiene funciones asignadas, al presionar “enter” se manda a llamar a la función que tomara los datos pero a la vez se le proporcionara que posición a quedado en el menú mediante el envió de un valor numérico de tipo entero, la función GetValue() tomara el valor y sabrá en que posición de memoria debe almacenarlo para así poder diferenciar que valor esta ingresando el operador, solo cuando los datos ya se hallan ingresado es cuando la variable “Q” se le asigna el valor de ”false”, esto hará que el bucle termine al momento de no cumplir la primera condición que decía “Q = True”, así dando por terminada la tarea de la función Menu().

Función GetValue()

Como ya se dijo “GetValue()” es la función encargada de tomar los datos de temperatura y tiempo desde el teclado y almacenarlos en memoria para que estén disponibles en cualquier momento, aunque sea simple la tarea que debe realizar se complica si es que se quiere tener una buena experiencia del usuario al momento de ingresar los datos de forma que se vea intuitiva, en el siguiente diagrama de flujo se puede apreciar lo que realiza esta función.

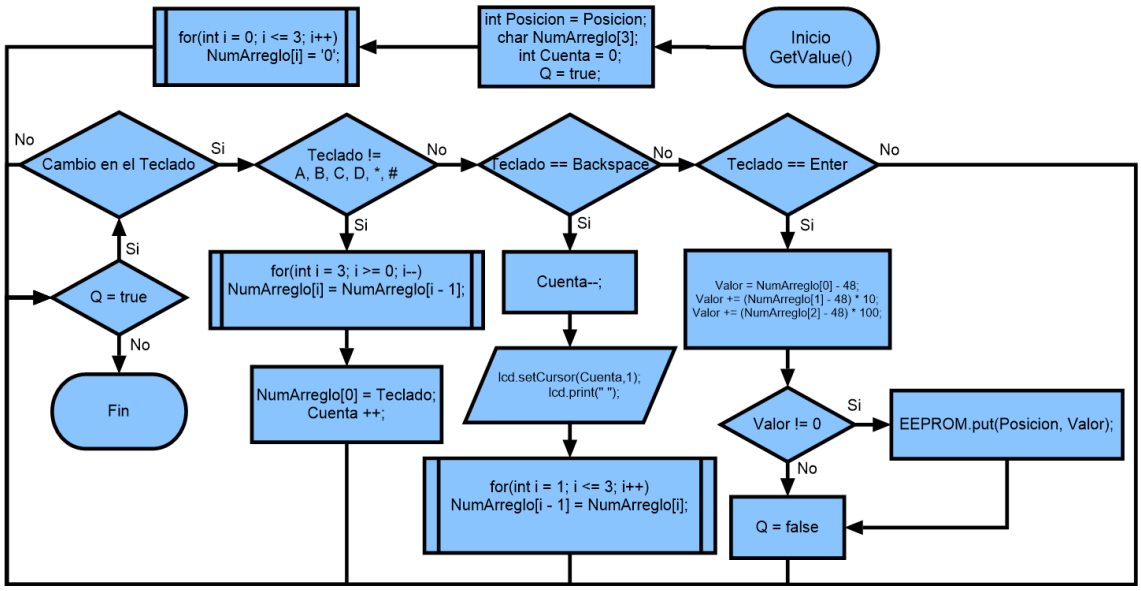


Fig.61 Diagrama de flujo correspondiente a la función encargada de tomar los datos desde el teclado GetValue().

Cada ves que es llamada a la función “GetValue()” se le debe enviar a la ves un valor tipo entero, este valor corresponde a la posición o la opción que se

tomo en el menú, este sera el diferenciador para luego de tomar el dato, donde guardarlo, ya sea en la posición de memoria “1” que le corresponde al valor de temperatura o la posición de memoria “3” que le corresponde al valor del tiempo de horneado, lo primero que hace la función es crear las variables que necesita para la tomas de datos estas son:

- int Posicion = Posicion: acá se almacena el valor que diferenciara entre temperatura o tiempo.
- char NumArreglo[3]: un arreglo tipo carácter donde se guardaran temporalmente los datos ingresados desde el teclado.
- int Cuenta = 0: una variable que sera utilizada para contar la cantidad de dígitos ingresados.
- bool Q = true: variable tipo boleana utilizada para terminar la función al cambiar su valor.

Lo siguiente a realizar es poner a cero el arreglo tipo carácter para evitar inconvenientes de cualquier valor residual, después volvemos a preguntar si hay algún cambio en el teclado, esperando que lo haya, para luego preguntar lo siguiente:

Teclado != A, B, C, D, *, #: esta condiciones lo que hace es discriminar cualquier letra o signo del teclado, ya que lo que interesa son los números, si llega a ser un numero se procede a desplazar un digito al arreglo donde de almacenara los datos temporalmente, esto es debido a que se quiere almacenar cada nuevo valor ingresado en la posición “0” del arreglo, se aumenta uno la cuenta y luego se vuelve a espera otro cambio del teclado.

Teclado == Backspace: si se presiona la tecla asignada como “Backspace” se procede a disminuir la cuenta en uno, se elimina el valor anterior ingresa de la pantalla y se desplaza en sentido contrario el arreglo donde se almacena los datos temporalmente para eliminar el valor anterior, esto dejara el valor anterior al que se quiere eliminar en la posición “0” del arreglo.

Teclado == Enter: al ser presionada la tecla asignada como “Enter” se procede a convertir el arreglo tipo carácter a un solo numero entero, esto se logra restando 48 a cada numero del arreglo, esto se hace ya que un valor de tipo carácter se almacena en código ACSII de 8 bit, aunque los números enteros también utilizan 8 bits, estos se cuentan igual que en binario por lo que con 8 bits se puede contar hasta 255 utilizando una variable tipo entero, en el código ACSII no, ya que en ACSII se le asigna un numero binario de 8 bits a cada carácter de números, letras y signos, los números en ACSII se pueden leer como binario de 8 bits pero nos daremos cuenta que estarán sumados con el valor 48, por lo que solo basta restarlo y ya se convertirán a un valor de tipo entero.

DEC	4
BIN(8BITS)	00000100
ACSII	00110100

Fig.62 Comparación del numero 4 en decimal, binario y Código ACSII.

No solo se debe convertir los datos del arreglo tipo carácter a enteros si no también se debe almacenar en una solo variable tipo entero, por esta razón se

quería tener los dígitos separados, ya que solo bastara multiplicar el valor ingresados por su posición de digito que ocupa en el valor deseado por el usuario, osea si es una unidad, decena o centena, cada resultado se suma a una solo variable tipo entero llamada valor, inmediatamente se realiza otra condición que pregunta si al final de todo el valor ingresado es distinto de cero, si es el caso permite la escritura del valor en memoria con "EEPROM.put(Posicion, Valor)", si es al contrario y el valor ingresado es cero, ignora la parte del almacenamiento y cambia el valor de Q = false, por que ignorarlo cuando es cero?, no tiene sentido configurar una temperatura de 0Cº o un tiempo de 0 Min. Por lo que no se toma en cuenta y al ignorar se conserva el valor almacenado anteriormente por lo que la configuración previa no es afectada.

Función Inicio()

Esta función es la mas importante, podría decirse el corazón del código del sistema, ademas es la que pasara mas tiempo en uso, en esta función es donde al final se controlara el horno por lo que tendrá que realizar todas las tareas importantes del control, como se listan acá.

- Vigilar la presencia de llama.
- Llamar a la función de encendido del soplete "IGN()" si la llama se apaga.
- Realizar calculo del PID.
- Realizar la cuenta regresiva del tiempo de horneado.
- Esperar cualquier orden del usuario desde el teclado si desea pausar o cancelar el horneado.

La función deberá realizar todo lo listado anteriormente a la vez; En realidad no es a la vez ya que el código se ejecuta secuenciado (una instrucción a la vez) pero gracias a que el microcontrolador es muy rápido, se puede hacer parecer que es a la vez.

Primeramente se debe inicializar las variables a ocupar, las cuales son las siguientes.

- int CT2 = 0: Variable que sera utilizada para realizar un conteo de las veces que se intentara encender el soplete cada vez que se detecte la ausencia de llama, con ella se limitara hasta un máximo de 3 intentos para encender el soplete.
- int WindowPercent = 0: Se utilizara para almacenar el porcentaje de tiempo transcurrido con respecto al periodo.
- int OutPercent = 0: Se utilizara para almacenar el porcentaje de la salida del controlador.
- Int CPWM = 0: se utilizara para contar el periodo que se pretende utilizar en el PWM, en este caso 20 segundos, por lo que aumentara en uno a cada segundo que pase y se devolverá a cero una ves llegue a 20, esta

variable se utilizara para calcular el valor de la variable “WindowPercent”.

- ❖ int SEG = 0: Se utiliza para almacenar los segundos restantes del proceso de horneado.
- ❖ int MIN = 0: Se utiliza para almacenar los minutos restantes del proceso de horneado.

El diagrama de flujo completo de la función “Inicio()” es el siguiente.

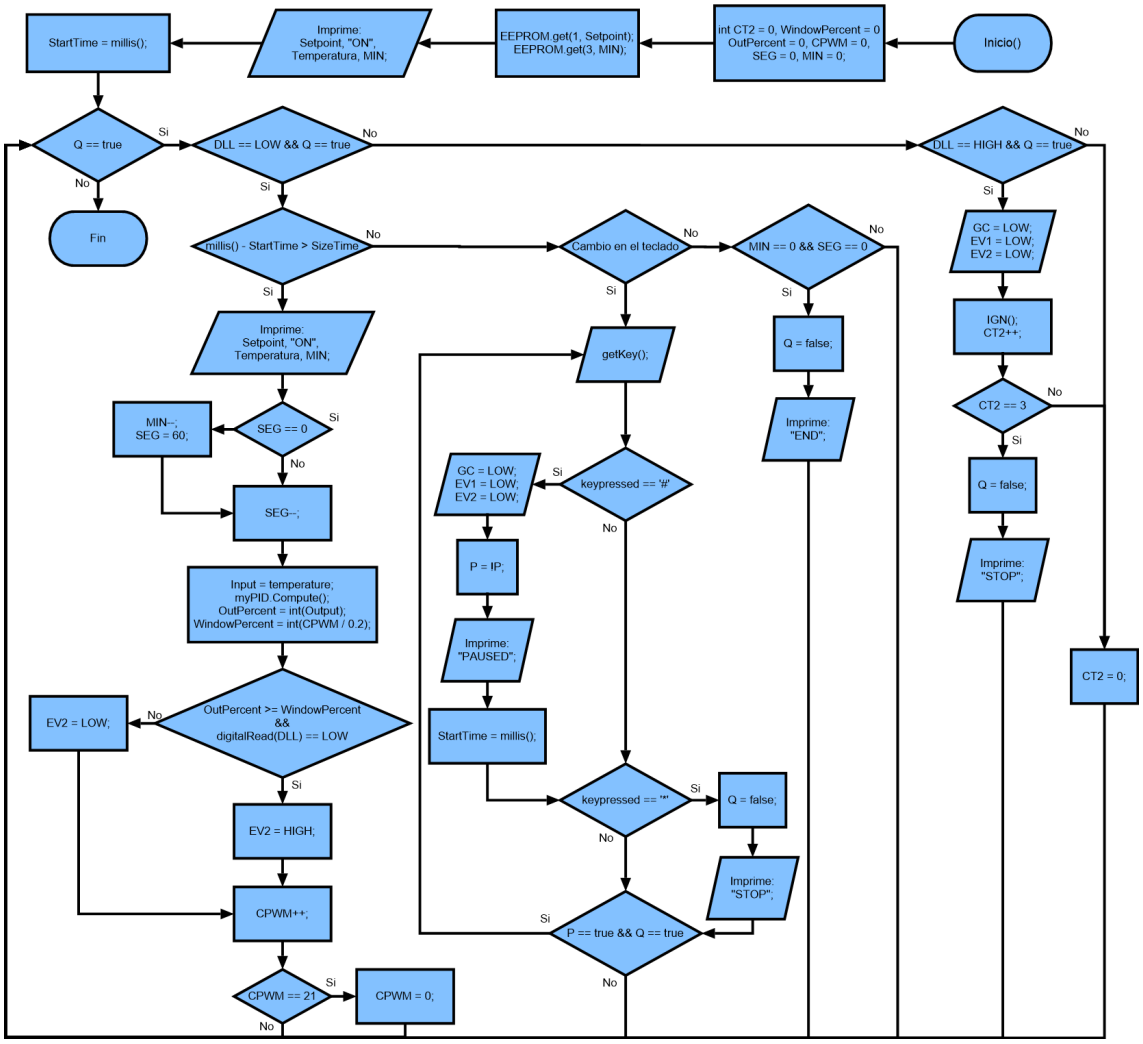


Fig.63 Diagrama de flujo correspondiente a la función Inicio().

Inmediatamente se procede a tomar los valores almacenados en memoria que corresponden a los parámetros configurados con “EEPROM.get(1, Setpoint), EEPROM.get(3, MIN);” que son “posición 1 a Setpoint” que sera la temperatura seleccionada y “posición 3 a MIN” que sera el tiempo de horneado seleccionado, luego imprime los datos “Setpoint”, el mensaje "ON", temperatura actual y “MIN” o tiempo de horneado restante en pantalla, para luego tomar el tiempo de inicio de la tarea “StartTime = millis()”. De acá en adelante la función se mantendrá en un bucle constante gracias a la condición “Q == true” que se encargara de terminar la tarea una ves que la variable Q tome el valor de falso; Dentro de este bucle entran en juego dos bucles mas que se encargaran de

realizar instrucciones diferentes para dos situaciones posibles, cuando hay llama y cuando no.

La condición que actuara cuando haya llama tiene el siguiente argumento “DLL == LOW && Q == true”, dentro de esta rama se realizaran la mayoría de las tareas, dentro de esta condición hay tres condiciones mas, la primera es “millis() - StartTime > SizeTime”, esta simplemente ejecutara las lineas de codigo que contiene cada segundo, primero refrescara los datos de pantalla volviendo a imprimir en la LCD los datos “Setpoint”, el mensaje "ON", temperatura actual y “MIN”, luego pregunta si la variable “SEG” es igual a cero, osea si los segundos ya se han acabado, si es el caso reducira la variable que contiene los minutos en uno e igualara “SEG” a 60, lo que esta haciendo es realizar la cuenta regresiva de los minutos cada ves que hayan pasado 60 segundo, luego se reduce en uno la variable “SEG” por lo que se imprime en pantalla los minutos menos 1 y 59 segundos, esta primera parte es para la cuenta regresiva y la visualización de los datos, seguido se calculo el PID, se iguala la variable “Input” al valor de la temperatura altual, luego se llama a la librería de PID con “myPID.Compute()”, la librería pondra el resultado en la variable “Output” pero de tipo doble por lo que se combierte a entero y se almacena en “OutPercent” de esta forma “OutPercent = int(Output)”, se calculo el porcentaje de la ventana de la siguiente forma “WindowPercent = int(CPWM / 0.2)”, “CPWM” es la cuenta de cuantos segundos han pasado del periodo de 20 segundos, “CPWM” se divide entre “0.2” que es una constante obtenida de una regla de tres para saber que porcentaje de tiempo es si el 100% es 20 segundos ya que este es el periodo que se ha decidido usar y quedara fijo, acá se puede ver un ejemplo de donde sale esta constante.

$$EC.27 \qquad WindowPercent = \frac{CPWM}{Periodo \Big/ 100\,\%} = \frac{7\,Seg.}{20\,Seg. \Big/ 100\,\%} = \frac{7\,Seg.}{0.2\,Seg/\%} = 35\,\%$$

Seguido habrá una condición de argumento “OutPercent >= WindowPercent && digitalRead(DLL) == LOW” acá es donde se adecuá la salida del PID a valores utilizables con un rele, “OutPercent >= WindowPercent” decide cuando se pone en alto la salida del microcontrolador que va hacia el relé que controla la electro-válvula del soplete, la lógica es la siguiente, cuando se genera un valor en porcentaje del PID este literalmente debe ser el mismo porcentaje de tiempo de que el soplete debe estar encendido, pero como el periodo corre constantemente el porcentaje de tiempo transcurrido varia, así que simplemente se comprueba que el porcentaje de tiempo en alto que dicta el PID todavía no ha terminado, si es así se pone en alto la salida del relé, si al contrario el porcentaje de tiempo que dicta el PID es menor que el transcurrido por el periodo se pone en bajo la salida del relé, por lo que el valor numérico de porcentaje que proporciona el PID es el mismo tiempo del porcentaje del periodo que estará en alto; Por ultimo para la condición de que haya pasado un segundo “millis() - StartTime > SizeTime”, se aumenta la variable “CPWM” en

uno y se pregunta si esta ya ha llegado a 21, si es el caso se reinicia a cero para volver a contar el próximo periodo, todo esto se realizara cada segundo.

Siguiendo en la condición de que haya llama faltan dos condiciones mas, estas 2 ocurren todo el tiempo por lo que no hay condición de tiempo, uno es cuando hay algún cambio proveniente del teclado, si es el caso tomara el carácter que se pulso “getKey()” y preguntara solo por dos teclas que nos interesan estas son, “keypressed == '#'” este se tomara como “pausa”, si es el caso procederá a poner en bajo todas las salidas de los relés para luego cambiar el valor la variable buleana “P”, lo que se le hará a esta variable solamente es aplicarle una operación de negación con “P = !P”, osea invertir su valor, imprime “PAUSED” en la pantalla y tomara el tiempo de inicio, la segunda tecla que nos interesa es “keypressed == '*'”, esta se tomara como “Paro”, acá simplemente se cambia el valor de “Q” a falso “Q = false”, ya que sabemos que se utiliza para terminar el bucle de la función “Inicio()”, luego se imprime el mensaje “STOP”, las demás teclas se ignoran ya que no hay condiciones para ellas, luego que se realizan las respectivas acciones solo cuando se presiona una tecla que nos interese viene otra condición que funciona como un “do while” con argumento “P == true && Q == true”, esta condición hace que cuando se presione el botón pausa se quede en un bucle infinito donde solo se permanecerá leyendo el teclado, por eso cuando se cumple esta condición regresa hasta “getKey()”, solo cuando se vuelva a presionar el botón de pause se cambia el valor de “P” y volverá al bucle del proceso de horneado, también esta “ Q == true”, esto es para cuando se pausa el proceso de horneado y en ves de querer volver al proceso de horneado se procede directamente a cancelar el proceso y permitir terminar la función “Inicio()”. La ultima condición que se realiza cuando hay llama es con el argmento “MIN == 0 && SEG == 0”, se puede intuir que es cuando el tiempo de horneado haya acabado, si se cumple simplemente se cambia el valor de “Q” a falso para terminar el proceso y se imprime en la pantalla “END”.

Por ultimo para la función “Inicio()” esta la condición cuando no hay llama, esta se encargara de mandar a llamar a la función “IGN()” para que intente encender el soplete ademas de verificar que solo se intente realizar el encendido un máximo de tres veces, cuando se cumple se manda apagar las salidas de los relés con “GC = LOW, EV1 = LOW, EV2 = LOW”, luego se manda a llamar a la función “IGN()”, luego de que termine el proceso de encendido de “IGN()” automáticamente se aumento la variable “CT2” en uno para empezar a contar cuantas veces se a mandado a llamar a dicha función, después se pregunta si dicha variable “CT2” es igual a 3. si se cumple es que ya se intento encender el soplete 3 veces y no se ha conseguido por lo que se cambia la variable “Q” a falso para que termine la función “Inicio()” y luego imprime el mensaje “STOP”, esto seria todo lo que realiza la función “Inicio()” durante el proceso de horneado.

Función IGN()

Esta función es muy sencilla pero a la vez muy importante ya que se encargara de encender el soplete a gas, esta función realizara un intento de encendido del soplete por cada vez que sea llamada, como se explico anteriormente, es la función “Inicio()” la que se encargara de velar si se apaga el soplete, por lo que llamara a la función “IGN()” un máximo de tres veces cada vez que se inicie el proceso de horneado, lo que hará dicha función se puede ver en el siguiente diagrama de flujo.

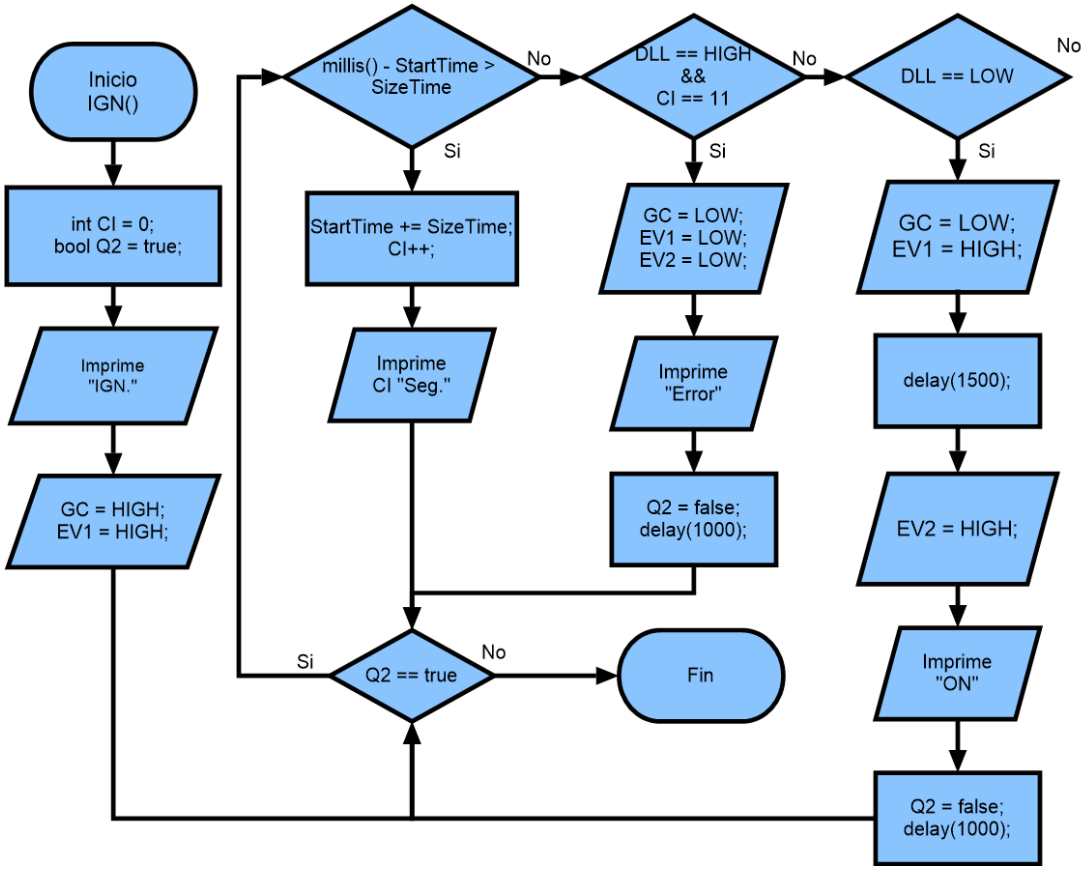


Fig.64 Diagrama de flujo correspondiente a la función encargada de realizar el proceso de encendido del soplete IGN().

Como se puede ver en el imagen la función iniciara con dos variables, una es “CI” de tipo entero que se utilizara para contar los segundos durante el intento de encendido del soplete, y “Q2” de tipo buleano que se utilizara para decidir cuando se terminara la función, inmediatamente se imprime un mensaje en la pantalla indicando que se procederá a encender el soplete “IGN” y se ponen en alto las salidas correspondientes a los relés, “GC” es el relé del generador de chispa y “EV1” es la electro-válvula del piloto, una ves encendidos los dos dispositivos simplemente se espera hasta un máximo de 10 segundos a que el soplete encienda, esto se logra gracias a la condición “millis() - StartTime > SizeTime” que se cumplirá cada ves que haya pasado un segundo o 1000 mili-segundos por lo que la variable “CI” es aumentada en 1 cada segundo, lo que detendrá la espera es la siguiente condición “DLL == HIGH && CI == 11”, como

seta sera verdadera solo cuando la cuenta allá llegado a 11 y “DLL” que es el pin de entrada del detector de llama este en alto.

Como se menciono en la sección del detector de llama, para aumentar la inmunidad al ruido en el microcontrolador, dicho ruido que es producido por el generador de chispa, se decidió que el detector de llama envié un flanco negativo al momento de detectar la llama, esto permite que los picos que producen el ruido eléctrico del generador de chispa se ignoren y solamente se tome en cuenta cuando haya una tensión de 0V en el pin de entrada del microcontrolador, una tensión de 0V es algo muy poco probable que ocurra con el ruido eléctrico, por esta razón en el código de programación se decidió dejar como “DLL == HIGH” para la ausencia de llama y “DLL == LOW” cuando hay presencia de llama.

Regresando al código, cuando se cumple la condición “DLL == HIGH && CI == 11” es debido a que se ha llagado a los 10 segundos y el soplete no ha encendido por lo que se apaga todos los relés (GC = LOW, EV1 = LOW, EV2 = LOW), se imprime en la pantalla ”Error” y se pone en valor falso la variable “Q2” que dará por terminado el ciclo de la función, por otro lado hay otra condición mas que es “DLL == LOW”, esta es en el caso que si se logro encender el soplete, por lo que manda a apagar el generador de chispa y mantiene encendido el piloto (GC = LOW, EV1 = HIGH), luego espera 1.5 segundos para encender la electro-válvula del soplete (EV2 = HIGH), esta espera es para permitir que la llama del piloto se estabilice debido a que el pase grande de gas del soplete podría apagar el piloto al liberar gran cantidad de gas con poco oxigeno, una ves encendido se asigna el valor de falso a la variable “Q2” y se espera 1 segundo a que estabilice la llama del soplete para luego terminar el ciclo de ignición.

2.3 Etapa de presentación de resultados

En esta sección se explica el proceso de implementación de los circuitos adicionales diseñados para el prototipo del sistema de control, y se muestra los resultados obtenidos en las pruebas realizadas para la verificar el funcionamiento del sistema.

2.3.1 Implementación de los circuitos

Las placas correspondientes a cada circuito fueron realizadas por separado para facilitar su testeo y revisión ante errores para luego ser interconectadas entre si y con el microcontrolador, el diseño de las placas PCB[8] se realizaron en Proteus 8.7 y construidas por el método de planchado, acá algunas imágenes de cada placa.

8 Revisar diseños PCB en sección 5 Anexos.

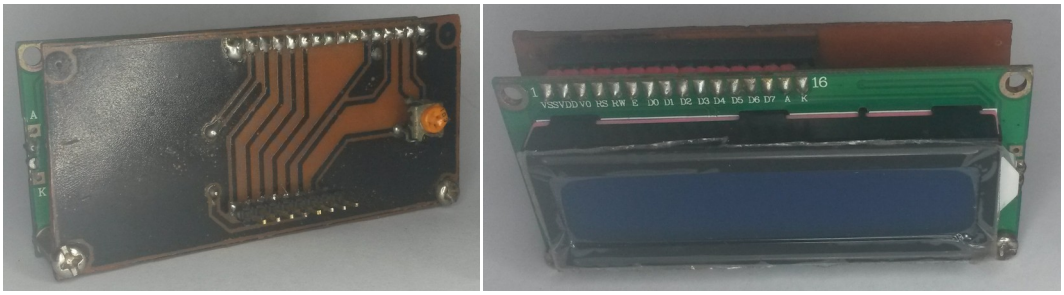


Fig.65 Placa de la base para la LCD 16X2, se muestra como quedara instalada en la LCD.

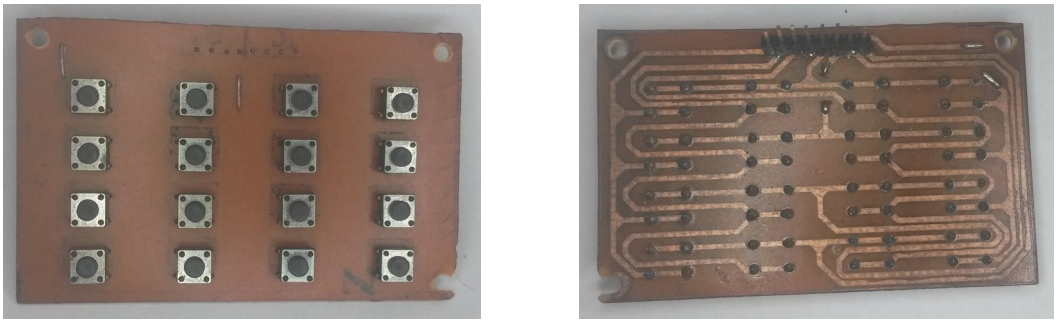


Fig.66 Placa correspondiente al teclado 4X4.

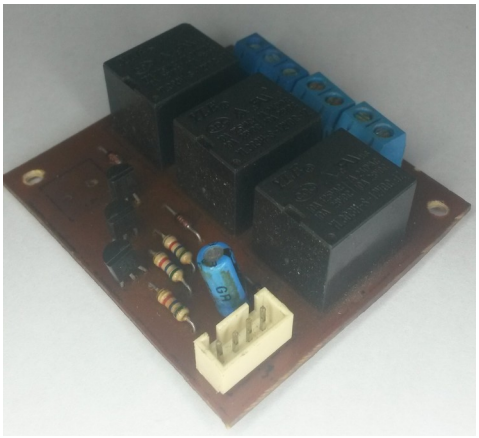


Fig.67 Placa con los relés de salida.

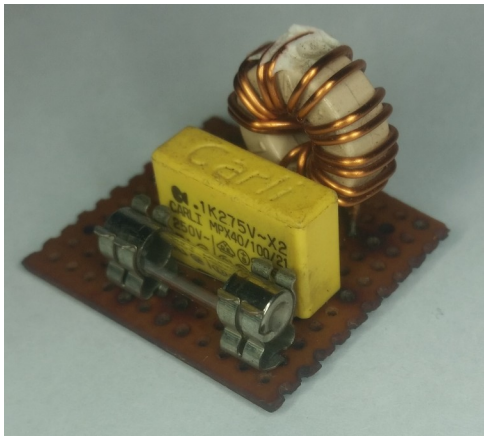


Fig.68 Placa correspondiente al filtro EMI.

Fig.69 Placa correspondiente al detector de llama.

Las imágenes anteriores corresponder a las placas que fueron necesarias construir, pero no hubo la necesidad de realizar placa alguna para el microcontrolador ya que se utilizo la placa de desarrollo Arduino Mega 2560, también con el sensor de temperatura que se comercializan listo para conectar y por ultimo el filtro EMI se construyo utilizando placas pre-perforadas ya que el circuito es muy sencillo y no amerita realizar un diseño PCB.

2.3.2 Ensamble y estructura del prototipo

Para construir la estructura del prototipo y asegurar las placas de los circuitos se utilizo placas de PVC de 2.5 mm con la idea de apilarlas en forma de sandwich para ser aseguradas con varillas roscadas y tuercas, estas laminas serán de 10X11 cm, siendo necesario 4 laminas para poder contener todos los circuitos del prototipo. En la parte frontal del prototipo estará ubicado la pantalla LCD 16X2 con el teclado 4X4.

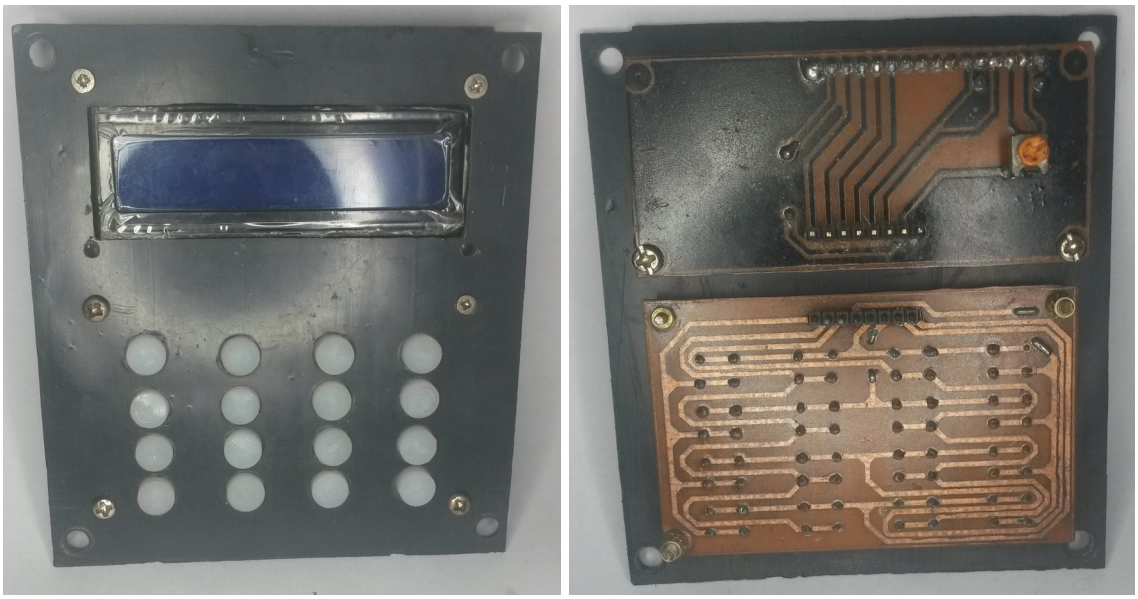


Fig.70 Cara frontal del prototipo, muestra como quedara asegurada la LCD y el teclado 4X4.

Como se puede ver en la imagen anterior así quedara la presentación del prototipo, para instalar la LCD 16X2 simplemente se realizo un agujero rectangular para asomar la pantalla y se atornillo a la lamina PVC, se puede apreciar donde quedara situado el trimpot utilizado para fijar el contraste, este sera configurado solo una ves, lo mismo hecho en la LCD se realizara con el teclado 4X4 con la diferencia en que se realizaron taladros de 8 mm exactamente en el centro de donde se situá cada botón, en estos taladros se introducen cilindros de plástico que poseen una pestaña en uno de sus extremos que le impiden salirse por al frente, por ultimo el teclado se atornilla sobre estos cilindros para que queden confinados, estos funcionaran como extensiones de los botones que al empujarlos presionan los botones.

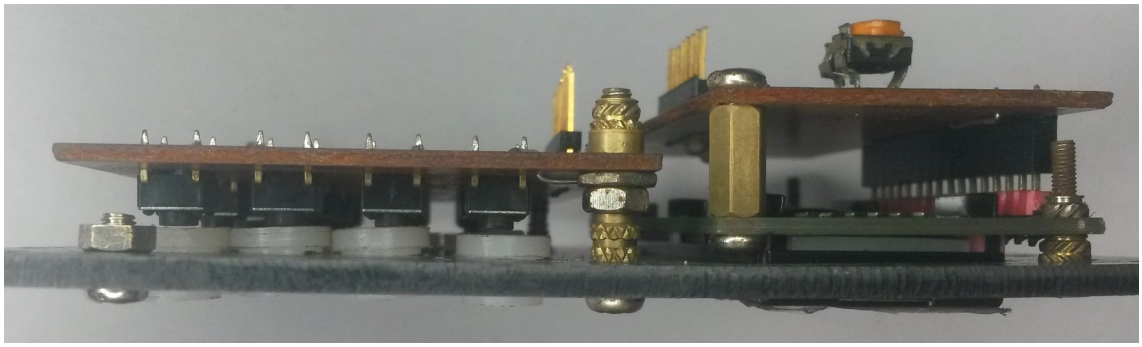


Fig.71 Cara frontal del prototipo vista desde el costado, muestra como queda asegurado el teclado 4X4 con los cilindros plásticos

De la misma forma se aseguran el resto de los circuitos del prototipo sobre mas laminas de PVC, estas como se dijo se aseguraran con las otras laminas entre si por medio de pernos pasantes, estas servirán como capas que forman un sandwich, ya asegurados se realiza la interconexión de las placas electrónicas entre si mediante cables que pasaran por agujeros que se realizaran en las laminas PVC, en la siguiente imagen se muestra como queda asegurado la placa de desarrollo Arduino Mega 2560 y como se le realizaron canales a los lados de la placa que permitirán la conexión mediante cables, también se puede apreciar los orificios realizados en cada esquina de la lamina PVC donde pasaran las varillas rocadas, uniendo la varilla a la lamina al apretar tuercas por ambos lados.

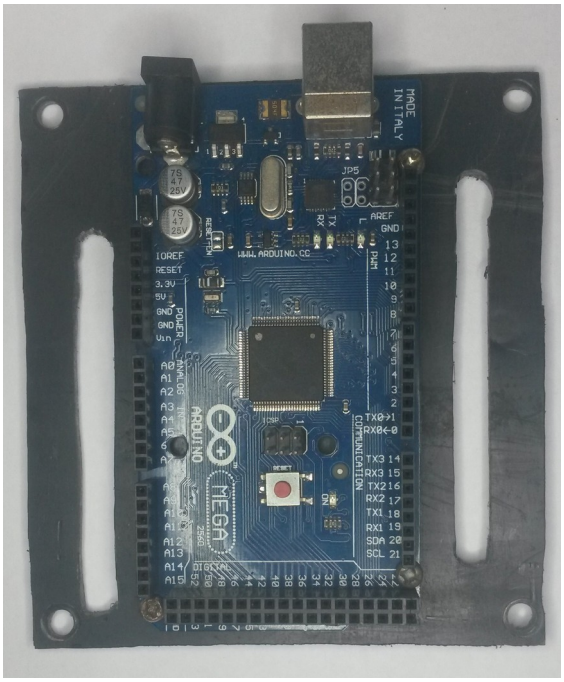


Fig.72 Muestra como queda asegurado el Arduino Mega, se puede ver los canales donde pasaran las conexiones.

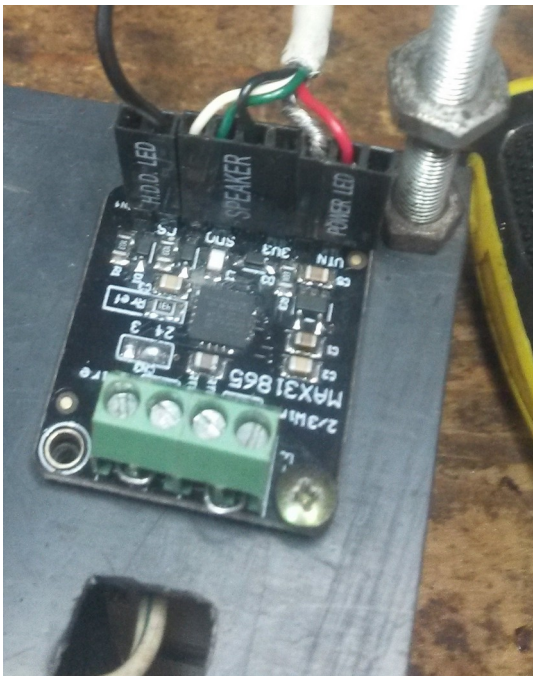


Fig.73 Modulo MAX6675.

Ya aseguradas todas las placas solo basta hacer pasar cada capa por las varillas a medida que se realizan las conexiones y se aprietan las tuercas.

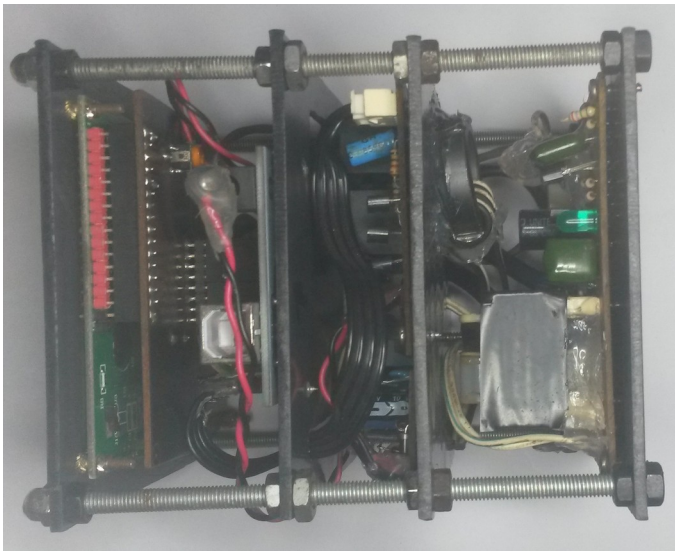


Fig.74 Vista desde arriba del prototipo, se puede ver las varillas roscadas con sus tuercas.

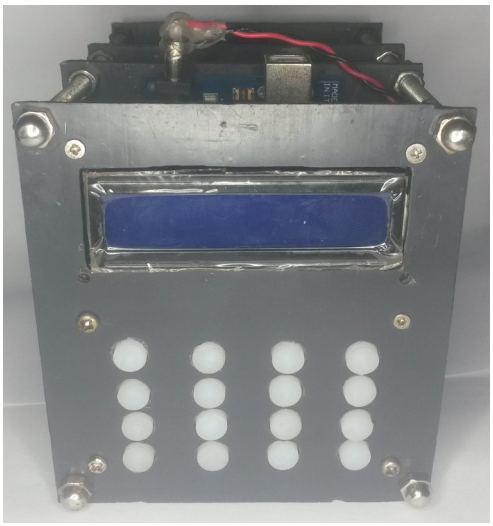


Fig.75 Vista frontal final del prototipo.

Fig.76 Parte trasera del prototipo con sus bornes de conexión.

Acá se puede ver como se aseguran las laminas PVC y como queda la parte trasera y frontal del prototipo, los bornes fueron asegurados de la misma forma que el LCD mediante un agujero rectangular y tornillos, en estos bornes se realizaran las conexiones con el resto del circuito del horno.

2.3.3 Montaje en el horno a gas

Una ves terminado el prototipo es cuestión de instalarlo en el horno a gas para proceder a realizar pruebas de funcionamiento y configuraciones finas del PID si lo amerita, como el prototipo pretende ser un modulo destinado al control de horno este se debe instalar como un dispositivo mas en el cableado del horno utilizando las normas que se requieren en la instalación de circuitos eléctricos industriales, a continuación se presenta un circuito eléctrico para la instalación de este prototipo como modulo, este es el circuito que se recomienda.

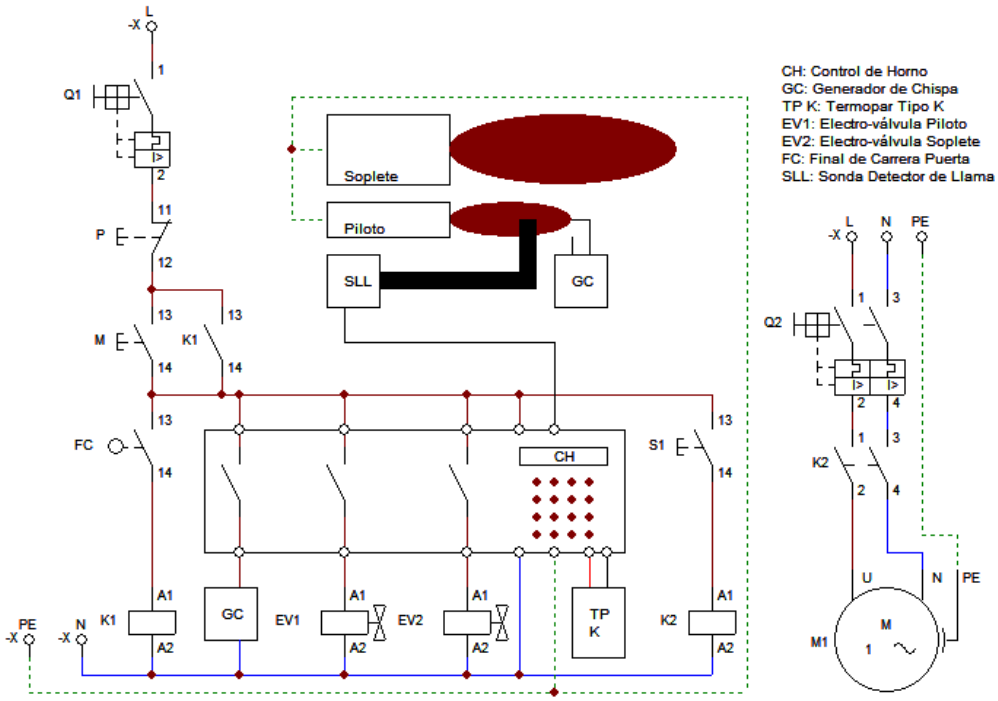


Fig.77 Circuito de control sugerido para la instalación del prototipo.

Como se puede apreciar el modulo control de horno es implementado con un circuito a contactor que se conoce como “paro y marcha”, en este tipo de circuitos se aconseja utilizar en sistemas industriales con la finalidad de evitar que el sistema se inicie solo si por alguna razón el suministro eléctrico se corta, obligando a siempre presionar marcha para poder poner en funcionamiento el circuito, también se puede apreciar la conexión del motor AC monofasico, este es gobernado principalmente por el interruptor “S1”, el contacto tipo final de carrera “FC” se encargara de apagar todo el sistema en caso de que se abra la puerta en pleno funcionamiento. La implementación del circuito anterior se realizo en un horno a gas con las siguientes características de la tabla.

Tabla 5
Características del horno a gas.

Dato	Descripción
Modelo	Garland The Master
N. Bandejas	5
Tamaño de bandejas	21” X 27”
Área interior del horno	24”A X 29”L X 27”P
Posee Piloto	Si
N. Electro-válvula del soplete	1
Posee Turbina	Si

Fig.78 Horno a gas Garland The Master, fue despojado de su control original.

Las siguientes imágenes corresponden a la instalación realizada en el horno a gas donde se realizaran las pruebas de funcionamiento. La mayoría de los dispositivos que son controlados por el prototipo de control de horno ya se encuentran en el horno, esto debido a que nos basamos en el tipo de horno que comúnmente se utiliza en al industria, el control se implemento en un horno que fue dado de baja por carecer del circuito de control completo y como se pretendía construir un prototipo de control de horno que pueda realizar todas las tareas que realizaba el circuito de control original, este es totalmente compatible por lo que lo único a realizar es las conexiones con el controlador.

Fig.79 Soplete del horno a gas, a la izquierda se puede apreciar la ubicación del electrodo sensor y la salida del piloto.

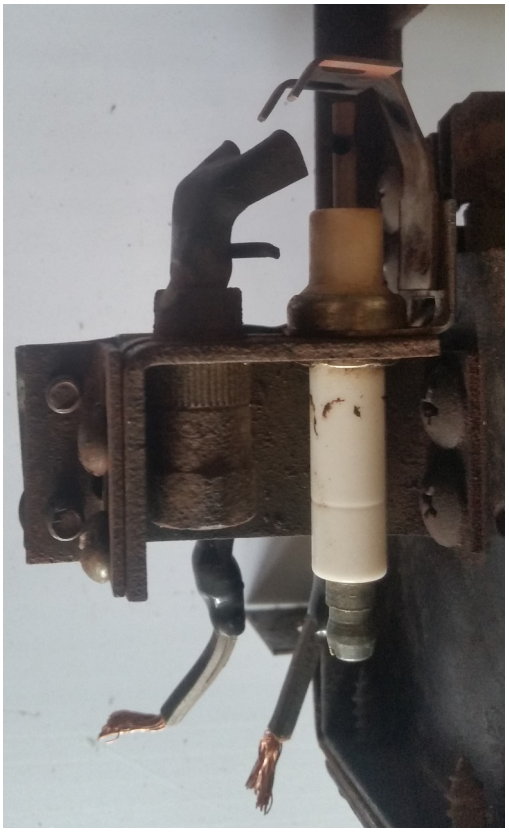
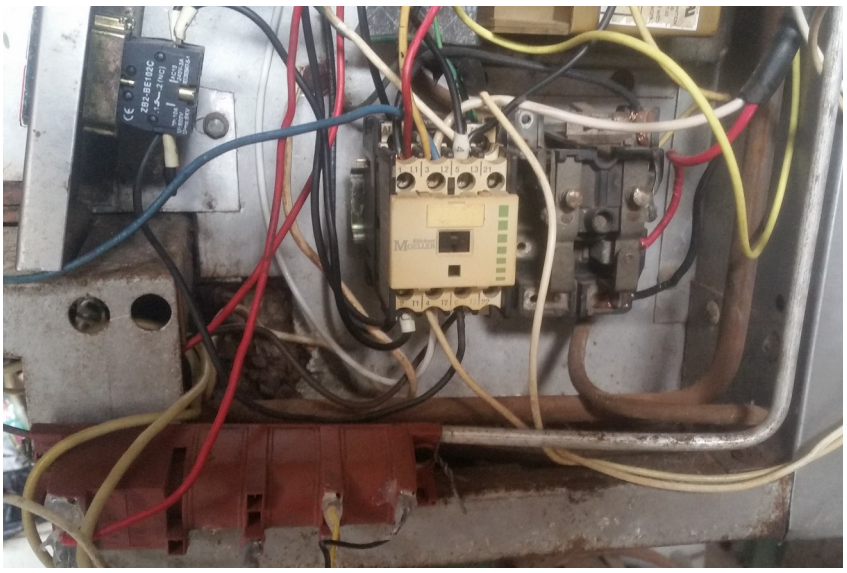


Fig.80 Acercamiento del electrodo sensor y la boquilla del piloto.

Fig.81 Electro-válvula del soplete y el piloto en un solo dispositivo.

En la imágenes anteriores se puede apreciar la ubicación del piloto y el electrodo sensor en el soplete del horno a gas, se aprecia que el electrodo sensor verifica la presencia de llama del piloto y no directamente en el soplete, también se aprecia en las imágenes anteriores que en este modelo de horno las dos electro-válvulas del piloto y soplete esta fusionada en un solo dispositivo, esto varia dependiendo del tipo de horno y la disponibilidad de repuestos pero siempre se tendrá acceso a la alimentación de sus bobinas.



Dispositivo rojo esquina inferior izquierda: generador de chispa.

Dispositivo negro esquina superior izquierda: Pulsadores.

Dispositivos blanco y negro centro: contactor y relé de fuerza.

Fig.82 Ubicación del contactor, generador de chispa y pulsador utilizado en el circuito del horno.

En la imagen anterior se puede apreciar los demás dispositivos citados en el diagrama sugerido para la instalación, se puede ver en el centro los contactores, el contactor blanco se utiliza para alimentar todo el circuito ademas de realizar el “paro y marcha” que permite el encendido del horno con solo pulsar una vez en el botón correspondiente a marcha, el relé negro a su costado se encarga de activar el motor AC para el uso de la turbina este es gobernado pro el contacto de puertas y la alimentación de todo el circuito, el dispositivo rojo es el generador de chispa, este generalmente su apariencia y forma varia dependiendo del modelo pero generalmente realizan la misma función, al ser alimentados generan chispas de alto voltaje en su salida. Y por ultimo el dispositivo en negro de la esquina superior izquierda corresponde a los pulsadores usados en el paro y marcha.

2.3.4 Resultados obtenidos

Una ves terminado el prototipo se pone a prueba la sintonización del PID realizada anteriormente, esto se verifica graficando la temperatura del horno con el PID en funcionamiento, el setpoint seleccionado para la prueba es de 200Cº ya que es una temperatura que se utiliza comúnmente para el horneado de productos como pizza, la siguiente imagen muestra el resultado de graficar la respuesta de la temperatura con el PID en funcionamiento.

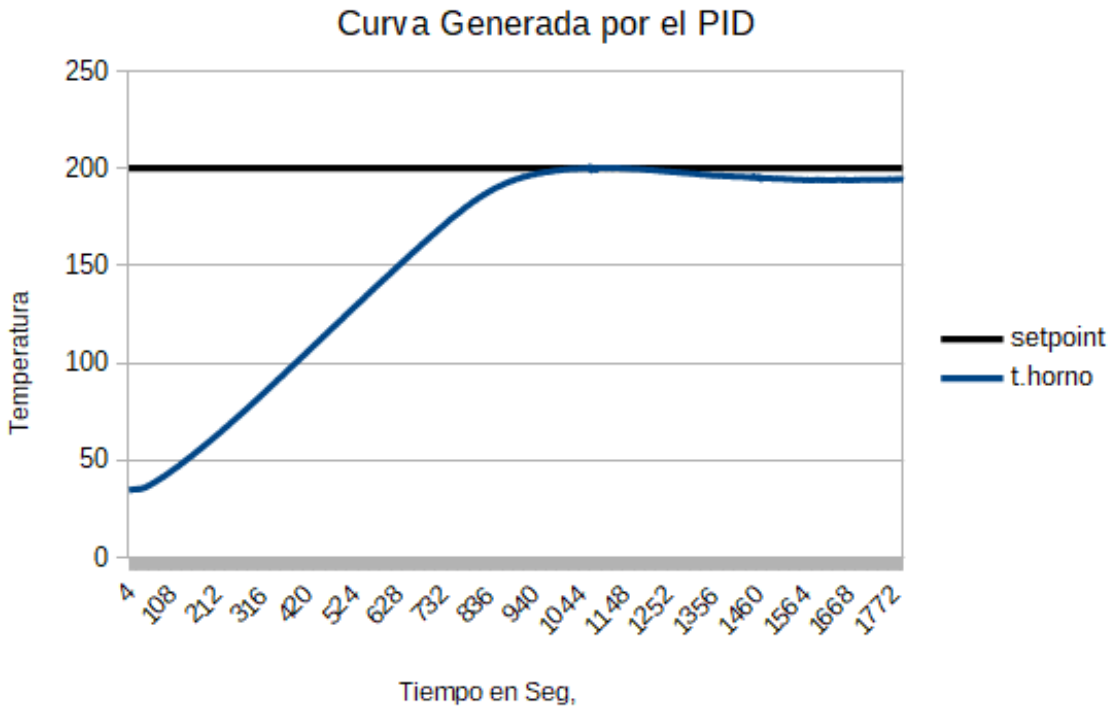


Fig.83 Comportamiento de la temperatura utilizando el PID.

Como se puede apreciar el sobre-impulso del controlador PID es casi inexistente deteniéndose el incremento de la temperatura casi justamente en los 200Cº, esto podría atribuirse a que el sistema es muy lento por lo que la histéresis de la temperatura no alcanza para sobrepasar el setpoint, pero tiene un pequeño inconveniente, y es el que la temperatura se mantiene por debajo del setpoint a unos 5Cº en promedio, acá se puede realizar ajustes finos en el controlador PID para intentar aumentar un poco el sobre impulso y dejar la temperatura justo es los 200Cº, recordemos que las constantes obtenidas del PID fueron las siguientes.

$$EC.16 \quad (1)k_P=7.15 \quad (2)k_i=0.04 \quad (3)k_d=328.9$$

Ahora el inconveniente es saber cual constante cambiar para obtener el ajuste fino, se puede deducir si se sabe cual es el papel de cada constante y como es el comportamiento de la planta a controlar, el horno es un sistema de respuesta lenta, si se le aplica toda la potencia del soplete la temperatura tardara en subir, pero a la ves tiene el problema que al cortar la potencia del soplete esta no produce que la temperatura deje de subir, esta puede seguir avanzando a pesar de haber apagado el soplete hasta bajar tiempo después, las constantes obtenidas de la sintonización del PID toman en cuenta este comportamiento del horno debido a que las calculamos en base a su curva característica, por lo que solo se debe modificar la constante que acelere la respuesta del control, esto porque queremos aumentar un poco el sobre-impulso.

Analizando las constantes tenemos que, la constante proporcional “kp = 7” no es muy alta pero tampoco baja, como es proporcional su aporte sera una multiplicación de la cantidad de error que tengamos por lo que dará un valor alto, esto hará que la salida se pongan al máximo cuando el setpoint esta muy lejos, algo que necesita el horno, la constante integral “ki = 0” produce que la salida aumente a medida que pasa el tiempo y el error no disminulla, es necesario en el horno que sea un valor muy bajo debido a que es un sistema lento y transcurre mucho tiempo, si esta constante fuese mas alta se produciría una acumulación del error que podría crear sobre-impulsos en la salida, y por ultimo la constante derivativa “kd = 328” esta producirá una reducción de la salida a medida que nos acercamos al setpoint, en el horno es necesaria debido a la histéresis de la temperatura, por esta histéresis necesitaremos que la reducción de la potencia del horno sea grande y previa a llegar al setpoint para poder detener el avance de la temperatura antes que lo sobrepase.

Teniendo lo anterior en cuenta se puede decidir que constante podemos modificar, en este caso puede ser la constate integral, la idea es aumentar la suma de errores para que se produzca un sobre-impulso controlado que permita mantener la temperatura exacta una ves se estabilice, estos valores propuestos son los siguientes.

$$EC.28 \quad (1)k_P=7 \quad (2)k_i=2 \quad (3)k_d=328$$

Aplicando los valores anteriores el resultado obtenido se puede ver en la siguiente imagen.

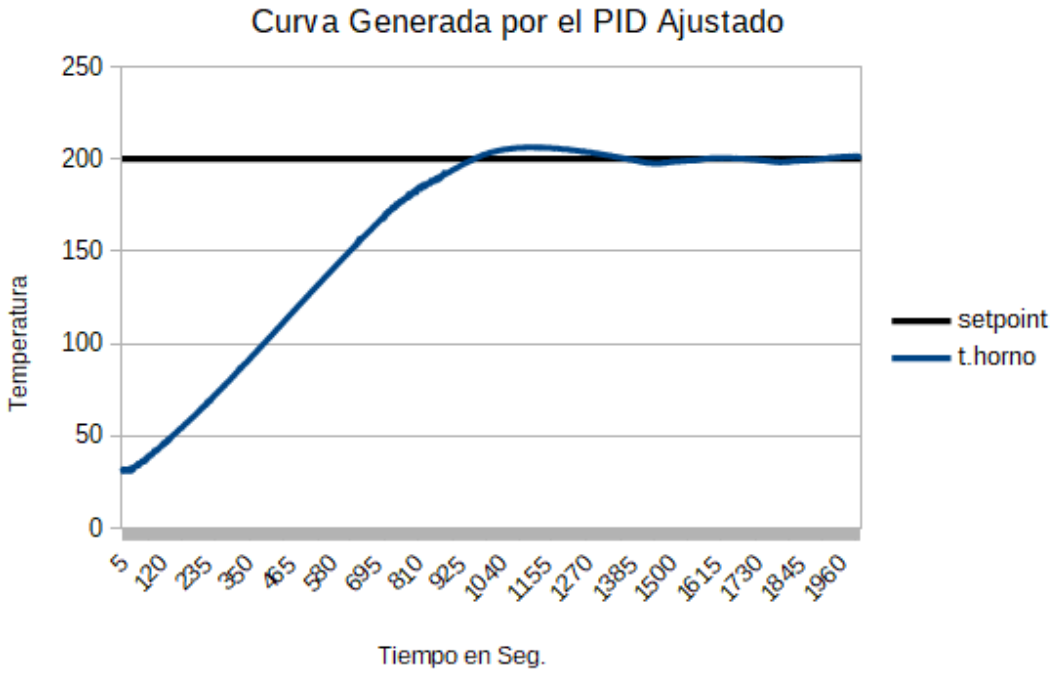


Fig.84 Comportamiento de la temperatura utilizando el PID con el ajuste fino.

Ya con el ajuste fino se puede apreciar que el sobre-impulso ha aumentado como era la intención, pero en consecuencia la temperatura se ha mantenido sobre los 200Cº una ves ha terminado el sobre impulso, dicho sobre-impulso solo fue hasta los 206Cº que representa un 3% con respecto al setpoint por lo que es aceptable, luego la temperatura se estanca entre los 198Cº a los 200Cº, este rango representa un 1% de variación en la temperatura o tolerancia, no es lo suficiente para considerarse malo, se puede realizar un buen horneado con este rango de variación de la temperatura.

Prueba de horneado

Para verificar mejor el funcionamiento del prototipo en el control del horno a gas se decide realizar ademas un horneado de algún producto de pan, idealmente el controlador debe poder hornear cualquier producto que no necesite realizar algún procedimiento extra a mitad del proceso de horneado, algunos ejemplos de productos que se esperan poder hornear con el prototipo son pan, biscochos, pizza, galletas entre otros, un ejemplo de producto que no se podría hornear con el prototipo podría ser el pan baguette debido a que necesita un proceso de horneado con vapor de agua.

Para realizar la prueba del prototipo se decidió hornear pizza, este producto se puede hornear a una temperatura de entre 150Cº a 200Cº con un tiempo de entre 5 a 10 minutos, habitualmente en los proceso de producción de pan los hornos que se utilizaran en la jornada de trabajo se deben pre-calentar antes de hornear cualquier producto, este pre-calentado se realiza debido a que el

producto se debe ingresar al horno con la temperatura de horneado ya alcanzada, esto debido a que el aumento graduado del la temperatura podría variar la calidad del producto, en conclusión se debe realizar un pre-calentamiento del horno a la temperatura de horneado, este se configura como si se tratara de hornear algo, se decide que el pre-calentamiento sea de 200Cº por 10 minutos.

Fig.85 Pre-calentamiento del horno a gas utilizando el prototipo.

El horno logro realizar el pre-calentamiento de forma adecuada alcanzando los 200Cº solo necesitando los 10 minutos. Inmediatamente se procede a realizar el horneado, se realizo el horneado de 8 pizzas de 12 pulgadas de diámetro a 200Cº y 8 minutos de horneado, ambos el pre-calentamiento y el horneado se realizan utilizando el motor AC y su turbina para hacer circular el aire en el interior del horno así asegurando un horneado parejo del producto, a continuación se puede apreciar las imágenes del proceso de horneado utilizando el prototipo.

Fig.86 Horneado en el horno a gas utilizando el prototipo.

En la siguiente imagen se puede apreciar el funcionamiento del soplete, como se pretendía hacer, el soplete se acciono gradualmente mediante ciclos de calentado manteniendo el piloto encendido para permitir su encendido instantáneo cuando se necesite calor, el PID realiza cortes a medida que se

acerca al setpoint de la temperatura permitiendo una temperatura de horneado estable.

Fig.87 Soplete del horno a gas en funcionamiento, se puede ver la tubería de gas del piloto y el soplete.

Fig.88 Resultado del horneado.

Como resultado las pizzas fueron horneadas exitosamente y el controlador no tubo un desvió anormal en la temperatura que afectara al producto, la temperatura estuvo manteniéndose estable una ves hecho el pre-calentamiento.

2.3.5 Costos del prototipo

Los costos del prototipo se muestran en las siguientes tablas, cabe señalar que algunos artículos se adquirieron por envío ya que no se encuentran fácilmente en el mercado local por lo que llevan un monto agregado de envió, las tablas muestran el costo por cada parte del proyecto para luego realizar una suma del prototipo total.

Circuitos interfaz de usuario

Tabla 6
Costos del circuito para la interfaz de usuario.

Articulo	Cantidad	Costo	Costo Total
LCD 16X2	1	$5	$5
Resistor 470	1	$0.15	$0.15
Trimpot 10k	1	$0.5	$0.5
Pulsadores	16	$0.05	$0.8
Costo Total			$6.45

Circuito salidas relés

Tabla 7
Costos del circuito para las tres salidas con relé.

Articulo	Cantidad	Costo	Costo Total
Relés 12V DC	3	$1.5	$4.5
Diodo 1N4148	3	$0.10	$0.30
Transistor C9014	3	$0.15	$0.45
Resistor 4k7	3	$0.15	$0.45
Terminal Block de 2	3	$0.23	$0.69
Condensador electrolítico 100uF	1	$0.15	$0.45
Costo Total			$6.84

Circuito detector de llama

Tabla 8
Costos del circuito del detector de llama.

Articulo	Cantidad	Costo	Costo Total
Transformador 5VA	1	$20	$20
Diodo 1N4001	4	$0.09	$0.36
Condensador electrolítico 1000uF	1	$0.5	$0.5
Condensador poliéster 100nF	4	$0.5	$2
Condensador poliéster 47nF	1	$0.5	$0.5

Resistor 1k	2	$0.15	$0.30
Resistor 10k	2	$0.15	$0.30
Resistor 3k9	1	$0.15	$0.15
Resistor 330	1	$0.15	$0.15
Resistor 1M5	2	$0.15	$0.30
Resistor 2M7	1	$0.15	$0.15
LED Verde	1	$0.15	$0.15
Transistor BC557	2	$0.15	$0.30
Transistor BC548	1	$0.15	$0.15
Opto-acoplador	1	$0.75	$0.75
Costo Total			$26.06

Otros

Tabla 9
Costos de los materiales extras utilizados.

Articulo	Cantidad	Costo	Costo Total
Arduino Mega 2560	1	$20.95	$20.95
Pines de conexión	1	$0.45	$0.45
PCB Virgen	1	$3	$3
Cargador de Móvil	1	$5	$5
MAX6675	1	$7.95	$7.95
Cables	1 Pack	$1	$1
Lamina PVC (2.5mm)	440 cm^2	$0.01	$4.4
Varilla roscada 3/16	52 cm	$0.12	$6.24
Tuercas	32	$0.03	$0.96
Bornera 12 bornes	1	$10	$10
Tornillos + tuercas (2mm)	19	$0.2	$3.8
Costo Total			$63.75

El costo total del prototipo se puede ver en la siguiente tabla que suma los costos de cada parte del proyecto incluyendo la mano de obra.

Tabla 10
Suma de todos los costos del prototipo.

Circuito	Costo Total
Interfaz de usuario	$6.45
Circuito salidas relé	$6.84
Circuito detector de llama	$26.06
Otros	$63.75
Mano de obra	$100
Costo Total Prototipo	$203.1

Visto el precio final del prototipo, no hay mucha diferencia con los ofrecido en el mercado, pero esto podría cambiar si es que se decide crear un producto final tomando en cuenta las recomendaciones mencionadas en las sección de recomendaciones, ademas hay que tomar en cuenta que el costo de mano de obra es debido al trabajo de diseño y construcción por parte del autor que aspira a ser ingeniero, por lo que en un producto final los costos de de inversión en diseño se pueden dividir entre las unidades vendidas y así reducir costos para volverlo mas competitivo.

Conclusiones

Luego de concluir el trabajo monográfico se logro obtener un prototipo de control de horno que realiza las funciones que se deseaban y se plantearon desde un inicio, un control que permita encender un horno a gas, contar su tiempo de horneado y su temperatura.

- Se logro construir y utilizar un circuito detector de llama por ionización basado en electrónica analógica que permite enviar un estado de en bajo cuando hay presencia de llama.
- Fue posible su construcción al implementar todos los circuitos y conexiones necesarias al microcontrolador tales como, las salidas relé, el sensor de temperatura, el detector de llama, uso de la pantalla y teclado y su alimentación de todo el prototipo desde una toma 120V AC.
- Se logro implementar un controlador PID para el control de temperatura del horno, su implementación fue posible al traducir su salida a una salida PWM de muy baja frecuencia con una sola electro-válvula, también fue posible sintonizar este controlador con el método de Ziegler-Nichols que se propuso, este método fue el de la curva de respuesta o en lazo abierto ya que es el método adecuado para este tipo de planta que es un horno.
- No hubo inconvenientes en la programación del microcontrolador, este pudo mostrar los datos en pantalla como también ingresarlos por medio de un teclado para ser almacenados en memoria, también pudo realizar la secuencia de encendido del horno, monitorear la presencia de la llama para tomar acciones cuando esta falle, como re-intentar el encendido y cancelar después de tres intentos, a la ves pudo calcular el PID y traducir su salida para poder apagar la electro-válvula en base al PID manteniendo el piloto encendido en todo momento, todo esto mientras realiza una cuenta regresiva del tiempo de horneado.
- Se realizaron pruebas y se pudo determinar que el control realiza su trabajo, esto al ponerlo a prueba horneando un producto (pizza) y concluyendo su horneado sin inconvenientes relacionados al control.

Recomendaciones

El prototipo cumple los objetivos propuestos al inicio del proyecto, dicho prototipo cumple los requerimientos básicos del control de un horno a gas, en algunos casos sera lo único necesario, esto lo hace que sea de carácter genérico por lo que no se especializa en un tipo de producto, pero a medida que se realizo el prototipo se pudieron hacer notar algunas posibles mejoras que se deben tomar en cuenta si se desea convertir al prototipo en un producto terminado y listo de comercializar.

- Unificar la mayoría de los circuito en una sola placa PCB única si es posible o bien una placa de potencia y otra de control, también dependerá del tipo de carcasa a utilizar, para realizar esta mejora se debe tomar en cuenta el volver mas inmune al microcontrolador ante el ruido.
- Utilizar otras tecnologías de componentes mas económicas: para intentar reducir mas los costes se puede implementar una pantalla de 7 segmentos para evitar usar una LCD 16X2, esto incluiría re-diseñar el menú para poder configurar los parámetros sin necesidad de visualizar una interfaz completa, se podría re-diseñar el detector de llama para evitar tener que usar un transformador ya que este puede ser algo costoso y voluminoso, esto implicaría evitar tener que usar una tensión alta AC (80V AC) en el detector de llama o intentar usar una fuente conmutada con un valor de tensión un poco mas bajo, también seria opción intentar que dicha fuente conmutada destinada a remplazar al transformador alimente al microcontrolador, pero esto implicaría resolver el problema de eliminar la AC que esta solapada a la DC de la fuente del detector de llama. Como es lógico el siguiente paso hacer si se desea mejorar este prototipo es utilizar directamente un microcontrolador que cumpla solo lo necesario para remplazar a la placa de desarrollo que se utilizo Ardino Mega 2560 así bajando el costo del prototipo.
- Crear la posibilidad de que el prototipo puede utilizar sondas RTD PT100 o PT1000 como también las sondas termopar tipo K o tipo J para hacer lo mas versátil al tipo de sonda disponible, esto se podría realizar conmutando entre dos circuitos lectores de la sonda de temperatura, pudiéndose configurar desde el menú o bien realizar alguna investigación sobre si existe algún integrado que permita esta característica.
- Otra mejora seria la posibilidad de poder accionar un ventilador en la zona de combustión, esta característica se puede utilizar para despejar de gas acumulado de intentos anteriores de ignición y evitar explosiones o llamaradas que sobresalgan, generalmente se aplica a sopletes que poseen un flujo de gas mayor al de un horno mediano.
- Diseñar una carcasa para el prototipo como producto final, esta puede ser construida en una impresora 3D como prototipo de encapsulado del proyecto, esta para poder acomodar los componentes en posiciones finales si se desea comercializar.

Bibliografía

Norma ISA S5.1 (2009). Instrumentation Symbols and Identification. ISA.

BASO Gas Products (2018). 24VAC Direct Spark Gas Ignition Control.

J. Park y S. Mackay (2003). Data Acquisition for Instrumentation and Control Systems. Oxford: Newnes.

Antonio Creus Solé (2010). Instrumentación Industrial 8va edición. México: MARCOMBO, S.A.

Ilber Adonayt Ruge Ruge. Método Básico para Implementar un Controlador Digital PID. Universidad de Cundinamarca.

(2017). Métodos de Sintonización de Controladores PID. Universidad Nacional de Tucuman.

Brett Beauregard (2018). Arduino PID Library. Recuperado de https://playground.arduino.cc/Code/PIDLibrary/

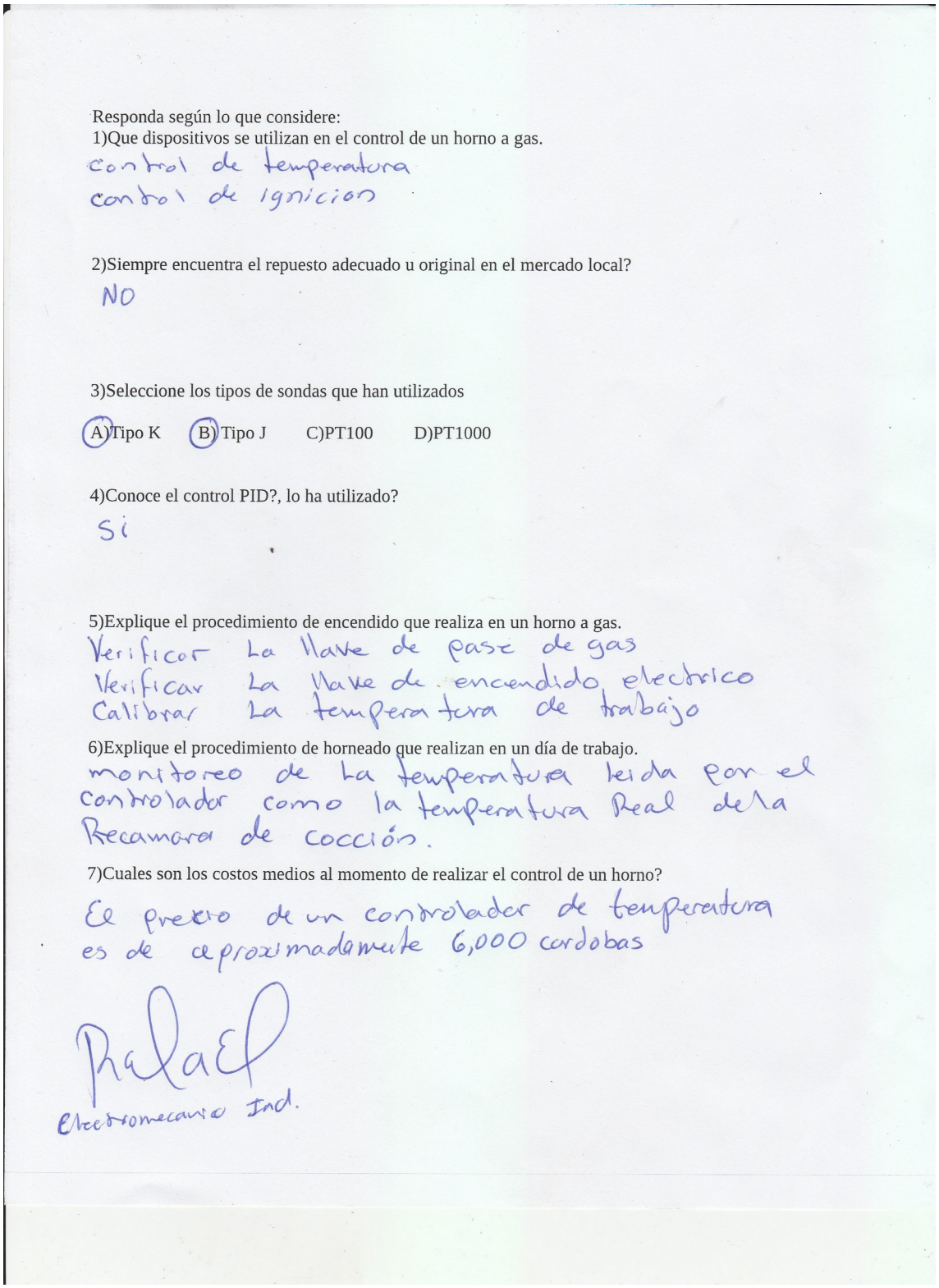
Construya su Video Rockola (2020). Construcción de un Transformador Casero. Colombia. Recuperado de http://www.videorockola.com/proyectos-electronicos/fuentes/construccion-de-un-transformador-casero-2/

BUSHELL (2018). Foros de Electrónica. Recuperado de https://www.forosdeelectronica.com/threads/construir-detector-de-flama.16320/

Robert L. Boylestad, Louis Nashelsky (2009). Electronica: Teoría de Circuitos y Dispositivos Electrónicos. México: Pearson Education.

Anexos

1 Pregunta realizadas en la entrevista a técnicos dedicados al mantenimiento de hornos a gas



Responda según lo que considere:

1)Que dispositivos se utilizan en el control de un horno a gas.

control de temperatura
control de ignicion

2)Siempre encuentra el repuesto adecuado u original en el mercado local?

NO

3)Seleccione los tipos de sondas que han utilizados

(A)Tipo K (B) Tipo J C)PT100 D)PT1000

4)Conoce el control PID?, lo ha utilizado?

Si

5)Explique el procedimiento de encendido que realiza en un horno a gas.

Verificar La llave de pase de gas
Verificar La llave de encendido electrico
Calibrar La temperatura de trabajo

6)Explique el procedimiento de horneado que realizan en un día de trabajo.

monitoreo de la temperatura leida por el controlador como la temperatura Real de la Recamara de cocción.

7)Cuales son los costos medios al momento de realizar el control de un horno?

El precio de un controlador de temperatura es de aproximadamente 6,000 cordobas

Rafael
Electromecanico Ind.

2 Asignación de pines Arduino Mega 2560

# Pin	Nombre Arduino	Nombre en Circuito	Tipo de Pin	Descripción
0	D0	Fila	Entrada	Fila 1 teclado matricial.
1	D1	Fila	Entrada	Fila 2 teclado matricial.
2	D2	Fila	Entrada	Fila 3 teclado matricial.
3	D3	Fila	Entrada	Fila 4 teclado matricial.
4	D4	Columna	Entrada	Columna 1 teclado matricial.
5	D5	Columna	Entrada	Columna 2 teclado matricial.
6	D6	Columna	Entrada	Columna 3 teclado matricial.
7	D7	Columna	Entrada	Columna 4 teclado matricial.
11	D11	EV2	Salida	Salida a relé EV2.
12	D12	EV1	Salida	Salida a relé EV1.
13	D13	GC	Salida	Salida a relé GC.
22	D22	E	Salida	Salida datos LCD Enable.
23	D23	RS	Salida	Salida datos LCD selector de registro.
24	D24	D4	Salida	Salida de datos LCD D4.
25	D25	D5	Salida	Salida de datos LCD D5.
26	D26	D6	Salida	Salida de datos LCD D6.
27	D27	D7	Salida	Salida de datos LCD D7.
31	D31	SS	Salida	Salida de datos SPI reinicio esclavo.
50	D50	MISO	Salida	Salida de datos serial SPI.
51	D51	MOSI	Entrada	Entrada de datos serial SPI.
52	D52	SCK	Salida	Salida reloj SPI.
53	D53	DLL	Entrada	Entrada sensor de llama.

3 Código completo implementado en el prototipo

```
#include <EEPROM.h>
#include <LiquidCrystal.h>
LiquidCrystal lcd(23, 22, 24, 25, 26, 27);
//Parametros:    (RS,  E, D4, D5, D6, D7)

#include <Keypad.h>
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
//define the cymbols on the buttons of the keypads
char hexaKeys[ROWS][COLS] = {
  {'*','7','4','1'},
  {'0','8','5','2'},
  {'#','9','6','3'},
  {'D','C','B','A'}
};
byte rowPins[ROWS] = {3, 2, 1, 0}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {7, 6, 5, 4}; //connect to the column pinouts of the keypad
//initialize an instance of class NewKeypad
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);

#include <PID_v1.h>
double Setpoint, Input, Output;
double Kp = 7, Ki = 2, Kd = 328;
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);

#include <max6675.h>
//                  (CLK, CS, DO)
MAX6675 thermocouple(52, 31, 50);

//////////////Salidas y Entradas////////////////////////////////////////////////////
int GC = 13;
int EV1 = 12;
int EV2 = 11;
int DLL = 53;
//////////////////Externos//////////////////////////////////////////////////////////
float Temperature = 0;
char keypressed;
//////////////////internos//////////////////////////////////////////////////////////
int SEG = 0, MIN = 0, Valor = 0;
int SizeTime = 1000;
unsigned long StartTime;
bool Q;

void setup(){
//////////Externos//////////////////////////////////////////////////////////////////
  pinMode(GC, OUTPUT);
  pinMode(EV1, OUTPUT);
  pinMode(EV2, OUTPUT);
  pinMode(DLL, INPUT);
  digitalWrite(GC, LOW);
  digitalWrite(EV1, LOW);
  digitalWrite(EV2, LOW);
//////////Presentacion//////////////////////////////////////////////////////////////
  lcd.begin(16,2);
  lcd.print("Control De Horno");
  delay(1000);
  lcd.clear();
  EEPROM.get(1, Valor);
  Setpoint = Valor;//Define la temperatura seleccionada como nuevo Setpoint del PID.
  lcd.print(Valor);//Imprime el valor de temperatura seleccionada.
  lcd.print("C");
  lcd.print((char)223);
  lcd.setCursor(0,1);
  Temperature = thermocouple.readCelsius();
  lcd.print(Temperature);//Imprime la temperatura actual.
  lcd.print("C");
```

```
  lcd.print((char)223);
  EEPROM.get(3, Valor);
  lcd.setCursor(9,1);
  lcd.print(Valor);
  lcd.print(" min.");
//////////Internos/////////////////////////////////////////////////////////////
  myPID.SetOutputLimits(0, 100);
  myPID.SetSampleTime(1000);
  myPID.SetMode(AUTOMATIC);
  StartTime = millis();
}

void loop(){
  if (millis() - StartTime > SizeTime){
    StartTime += SizeTime;
    Temperature = thermocouple.readCelsius();
    lcd.setCursor(0,1);
    lcd.print(Temperature);//Imprime la temperatura actual.
    lcd.print("C");
    lcd.print((char)223);
  }
  keypressed = customKeypad.getKey();//Toma el valor del keypad.
  if(keypressed){//Si hay un cambio en el keypad.
    if(keypressed == 'D'){//Equivalente a enter, se habre el menu.
      MENU();
    }
    if (keypressed == '#'){//Equivalente a inicio, se inicia el PID y la Cuenta
regresiva.
      INICIO();
    }
    digitalWrite(GC, LOW);
    digitalWrite(EV1, LOW);
    digitalWrite(EV2, LOW);
    delay(1000);
    lcd.clear();
    EEPROM.get(1, Valor);
    lcd.print(Valor);//Imprime el valor de temperatura seleccionada.
    lcd.print("C");
    lcd.print((char)223);
    lcd.setCursor(0,1);
    lcd.print(Temperature);//Imprime la temperatura actual.
    lcd.print("C");
    lcd.print((char)223);
    EEPROM.get(3, Valor);
    lcd.setCursor(9,1);
    lcd.print(Valor);
    lcd.print(" min.");
  }
}

void MENU(){
  int Posi = 1;
  Q = true;
  Valor = 0;
  lcd.clear();
  lcd.print("1.Set.Temp.    <");
  lcd.setCursor(0,1);
  lcd.print("2.Set.Time.");
  while(Q == true){
    keypressed = customKeypad.getKey();//Toma el valor del keypad.
    if(keypressed){
      if(keypressed == 'A'){
        lcd.setCursor(15,1);
        lcd.print(" ");
        lcd.setCursor(15,0);
        lcd.print("<");
        Posi = 1;
      }
```

```
      if(keypressed == 'B'){
        lcd.setCursor(15,0);
        lcd.print(" ");
        lcd.setCursor(15,1);
        lcd.print("<");
        Posi = 3;//A la posición para 3 por motivo de dar espacio entre los datos.
      }
      if(keypressed == 'D' && Posi == 1){
        lcd.clear();
        lcd.print("Temp.: ");
        EEPROM.get(Posi, Valor);
        lcd.print(Valor);
        lcd.print("C");
        lcd.print((char)223);
        GetValue(Posi);
        Q = false;
      }
      if(keypressed == 'D' && Posi == 3){
        lcd.clear();
        lcd.print("Time: ");
        EEPROM.get(Posi, Valor);
        lcd.print(Valor);
        lcd.print(" min.");
        GetValue(Posi);
        Q = false;
      }
    }
  }
}

void GetValue(int Posicion){//Funcion que toma el valor del keypad.
  char NumArreglo[3];
  int Cuenta = 0;
  Q = true;
  Valor = 0;
  for(int i = 0; i <= 3; i++){
        NumArreglo[i] = '0';
  }
  while(Q == true){
    keypressed = customKeypad.getKey();
    if(keypressed){
      if(keypressed == '#' || keypressed == 'A' || keypressed == 'B'){
        keypressed = '*';
      }
      if(keypressed != '*' && keypressed != 'C' && keypressed != 'D' && Cuenta <= 2)
{
        for(int i = 3; i >= 0; i--){
          NumArreglo[i] = NumArreglo[i - 1];
        }
        NumArreglo[0] = keypressed;
        lcd.setCursor(Cuenta,1);
        lcd.print(keypressed);
        Cuenta += 1;
      }
      if(keypressed == 'C'){//Equivalente a presionar "backspace".
        Cuenta = Cuenta - 1;
        lcd.setCursor(Cuenta,1);
        lcd.print(" ");
        for(int i = 1; i <= 3; i++){
          NumArreglo[i - 1] = NumArreglo[i];
        }
      }
      if (keypressed == 'D'){//Equivalente a presionar "enter".
        //Convertimos el array char en un numero int.
        Valor = NumArreglo[0] - 48;//Unidades
        Valor += (NumArreglo[1] - 48) * 10;//Decenas
        Valor += (NumArreglo[2] - 48) * 100;//Centenas
        if(Valor != 0){
```

```
          EEPROM.put(Posicion, Valor);
        }
        lcd.setCursor(14,1);
        lcd.print("OK");
        Q = false;
      }
    }
  }
}

void INICIO(){
  int CT2 = 0, WindowPercent = 0, OutPercent = 0, CPWM = 0;
  SEG = 0;
  MIN = 0;
  Q = true;
  bool P = false;
  unsigned long WindowStartTime;
  lcd.clear();
  EEPROM.get(1, Setpoint);//Lee el valor de Temperatura en memoria.
  //Setpoint = Valor;
  lcd.print(Valor);//Imprime el valor de temperatura seleccionada.
  lcd.print("C");
  lcd.print((char)223);
  EEPROM.get(3, MIN);//Lee el valor de Tiempo en memoria.
  lcd.setCursor(11,1);
  lcd.print(MIN);//Imprime el valor de tiempo seleccionado.
  lcd.print(":00");//Imprime con 0 segundos.
  Input = thermocouple.readCelsius();
  StartTime = millis();//Toma el tiempo de inicio de la tarea.
  while(Q == true){
////////////////////////////////////////////////////////////////////////////
//////////////////////Mietras halla llama.//////////////////////////////////////
    while(digitalRead(DLL) == LOW && Q == true){//Mietras halla llama.
////////////// Cada Segundo ////////////////////////////////////////////////////
      if (millis() - StartTime > SizeTime){// Cada Segundo
        StartTime += SizeTime;
        Input = thermocouple.readCelsius();
        lcd.setCursor(0,0);
        lcd.print(int(Setpoint));
        lcd.print("C");
        lcd.print((char)223);
        lcd.setCursor(0,1);
        lcd.print(Input);
        lcd.print("C");
        lcd.print((char)223);
        lcd.setCursor(11,0);
        lcd.print("     ");
        lcd.setCursor(13,0);
        lcd.print("ON");
        lcd.setCursor(11,1);
        lcd.print("     ");
        lcd.setCursor(11,1);
        lcd.print(MIN);
        lcd.print(":");
        if(SEG < 10){
          lcd.print("0");
        }
        lcd.print(SEG);
        if(SEG == 0){
         MIN--;
          SEG = 60;
        }
        SEG--;
////////////////////// PID y Salida Relay ///////////////////////////////////////
        myPID.Compute();
        OutPercent = int(Output);
        WindowPercent = int(CPWM / 0.2);
        if (OutPercent >= WindowPercent && digitalRead(DLL) == LOW && OutPercent !=
```

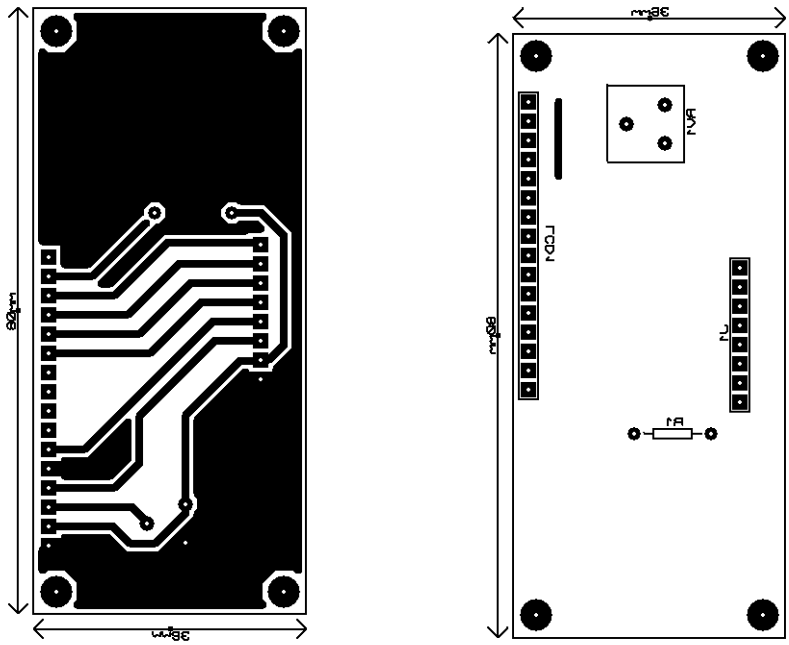
```
0 && Input < Setpoint){
          digitalWrite(EV2, HIGH);
        }
        else {
          digitalWrite(EV2, LOW);
        }
        CPWM++;
        if(CPWM == 21){
          CPWM = 0;
        }
      }
////////////////////////////////////////////////////////////////////////////
////////////////// Siempre ////////////////////////////////////////////////
////////////////////// Si hay cambios desde el teclado ////////////////////////
      do{
        keypressed = customKeypad.getKey();
        if(keypressed){
          if(keypressed == '#'){
            digitalWrite(GC, LOW);
            digitalWrite(EV1, LOW);
            digitalWrite(EV2, LOW);
            P = !P;
            lcd.setCursor(11,0);
            lcd.print("     ");
            lcd.setCursor(11,0);
            lcd.print("PAUSED");
            StartTime = millis();
          }
          if(keypressed == '*'){
            Q = false;
            lcd.setCursor(11,0);
            lcd.print("     ");
            lcd.setCursor(11,0);
            lcd.print("STOP");
          }
        }
      }while(P == true && Q == true);
////////////////////// Al terminar la cuenta ////////////////////////////////////
      if(MIN == 0 && SEG == 0){
        Q = false;
        lcd.setCursor(11,0);
        lcd.print("     ");
        lcd.setCursor(11,0);
        lcd.print("END");
      }
    }
////////////////////////////////////////////////////////////////////////////
//////////////////////Mietras no halla llama.////////////////////////////////////
    while(digitalRead(DLL) == HIGH && Q == true){
      digitalWrite(GC, LOW);
      digitalWrite(EV1, LOW);
      digitalWrite(EV2, LOW);
      IGN();
      CT2++;
      if(CT2 == 3){
        Q = false;
        lcd.setCursor(11,0);
        lcd.print("     ");
        lcd.setCursor(11,0);
        lcd.print("STOP");
      }
    }
    CT2 = 0;
  }
}

void IGN(){
  int CI = 0;
```

```
  bool Q2 = true;
  lcd.clear();
  lcd.print("IGN.");
  StartTime = millis();
  digitalWrite(GC, HIGH);
  digitalWrite(EV1, HIGH);
  while(Q2 == true){
    if (millis() - StartTime > SizeTime){
      StartTime += SizeTime;
      CI++;
      lcd.setCursor(9,0);
      lcd.print(CI);
      lcd.print(" Seg.");
    }
    if(digitalRead(DLL) == HIGH && CI == 11){
      digitalWrite(GC, LOW);
      digitalWrite(EV1, LOW);
      digitalWrite(EV2, LOW);
      lcd.clear();
      lcd.setCursor(11,0);
      lcd.print("Error");
      Q2 = false;
      delay(1000);
    }
    if(digitalRead(DLL) == LOW){
      delay(100);
      if(digitalRead(DLL) == LOW){
        digitalWrite(GC, LOW);
        digitalWrite(EV1, HIGH);
        delay(1500);
        digitalWrite(EV2, HIGH);
        lcd.clear();
        lcd.setCursor(13,0);
        lcd.print("ON");
        Q2 = false;
        delay(1500);
      }
    }
  }
}
```

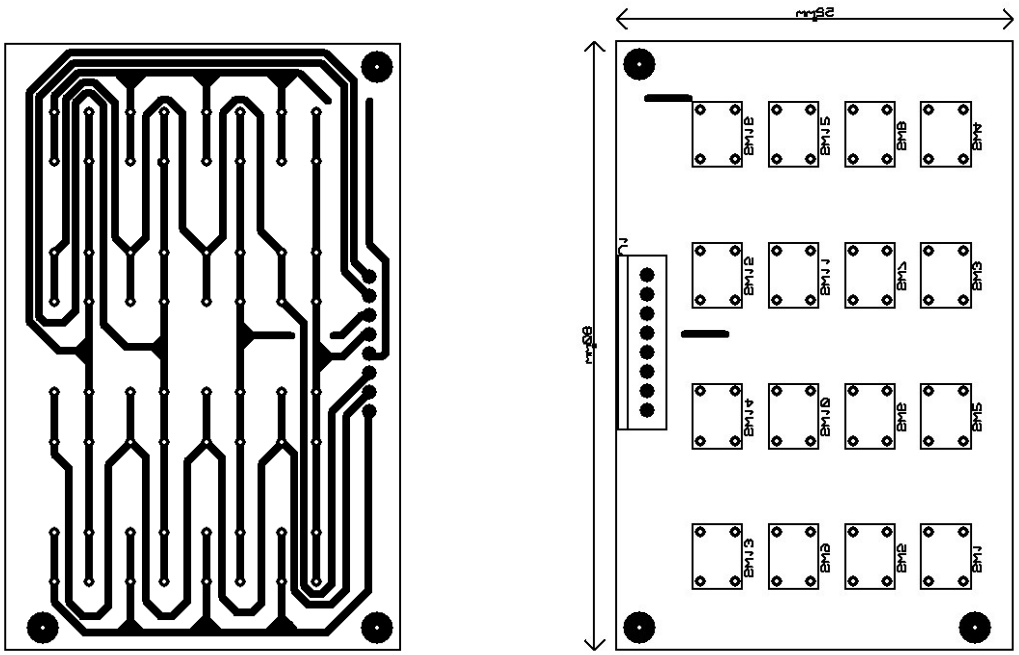
4 Script en Python 3.8.3 para el almacenamiento de los datos desde el serial

```
#Importa pyplot para realizar la gráfica.
from matplotlib import pyplot as plt
#Importa animation que permite actualizar la gráfica en intervalos concretos.
from matplotlib import animation
#Permite cambiar el estilo de nuestra gráfica.
from matplotlib import style
import serial #Importa librería para trabajar con el puerto serie.
import io #importando modulo io
import os #importando modulo os

#Cambia el estilo de nuestra gráfica.
style.use('fivethirtyeight')
#Creamos un objeto para almacenar la gráfica.
fig = plt.figure()
#Añadimos una "subgráfica" a nuestra ventana.
ax = fig.add_subplot(1,1,1)
#Definimos el titulo de la figura
fig.suptitle('Curva de respuesta - Horno a Gas', fontsize=14, fontweight='bold')

#Abrimos puerto Serie
#sustituir 'COM8' por el puerto que use el Arduino en tu PC.
ser = serial.Serial('COM8', 9600)
#ser.readline()

Temperatura = []
DatosTT = open('DatosTT.txt','w')
DatosTT = open('DatosTT.txt','a')

def plotea(i):#def es para definir una funcion creada por el usuario.
    datoString = str(ser.readline())
    print(datoString)#imprime la lista recien creada "datoString".
    #.split(",") genera una lista de los datos separados por "," provenientes del
puerto serial.
    ListaSerial = datoString.split(",")

    if i % 2 == 0:
        f = float(ListaSerial[1])
        Temperatura.append(f)
        #Agrega a la lista Temperatura el valor en la posicion "1" de la lista
"datoString".

    DatosTT.write(ListaSerial[0][2:])
    DatosTT.write(',')
    DatosTT.write(ListaSerial[1])
    DatosTT.write('\n')
    ax.clear() #Limpiamos la gráfica para volver a pintar.
    ax.plot(Temperatura) # Plotea los datos
    plt.ylabel('Temperatura')
    plt.xlabel('Tiempo en Seg.')

#Creamos animación para que se ejecute la función plotea con un intervalo de 100ms.
ani = animation.FuncAnimation(fig, plotea, interval = 100)
plt.show() #Muestra la gráfica.

DatosTT.close()
ser.close()
```

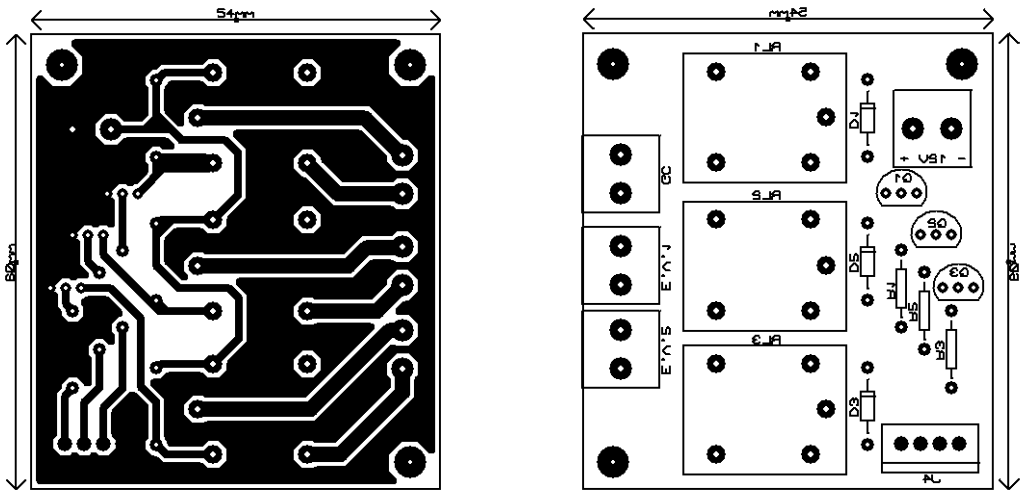
5 PCB de los circuitos implementados en el prototipo



PCB del circuito para la base del LCD 16X2.



PCB del circuito para el teclado matricial 4X4.



PCB del circuito para las salidas con relé.