

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA**

**FACULTAD DE INGENIERÍAS
CARRERA DE INGENIERÍA ELECTRÓNICA**

Tesis previa a la obtención del
título de Ingeniero Electrónico

**“DISEÑO E IMPLEMENTACIÓN DEL CONTROL DE
TEMPERATURA Y VELOCIDAD PARA EL REACTOR QUÍMICO
DE UN SISTEMA EXPERIMENTAL USADO EN LA
ELABORACIÓN DEL 5-HIDROXIMETILFURFURAL A PARTIR
DE MELAZA DE CAÑA PARA LA OBTENCIÓN DE
BIOCOMBUSTIBLE”**

AUTORES:

Julio Eduardo Cajamarca Villa

Juan Andrés Valverde Jara

DIRECTOR DE TESIS:

Msc. Ing. Walter Orozco.

CUENCA – ECUADOR

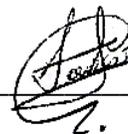
2012

Msc. Ing. Walter Orozco.

CERTIFICA

Haber dirigido y revisado el proyecto de tesis titulado: “DISEÑO E IMPLEMENTACIÓN DEL CONTROL DE TEMPERATURA Y VELOCIDAD PARA EL REACTOR QUÍMICO DE UN SISTEMA EXPERIMENTAL USADO EN LA ELABORACIÓN DEL 5-HIDROXIMETILFURFURAL A PARTIR DE MELAZA DE CAÑA PARA LA OBTENCIÓN DE BIOCOMBUSTIBLE”, realizado por los señores: Julio Eduardo Cajamarca Villa y Juan Andrés Valverde Jara. Por cumplir con todos los requisitos, autorizo su presentación.

Cuenca, 9 de febrero del 2009



Msc. Ing. Walter Orozco.

DECLARATORIA DE RESPONSABILIDAD

Nosotros: Julio Eduardo Cajamarca Villa y Juan Andrés Valverde Jara, autores del presente trabajo de tesis: **“DISEÑO E IMPLEMENTACIÓN DEL CONTROL DE TEMPERATURA Y VELOCIDAD PARA EL REACTOR QUÍMICO DE UN SISTEMA EXPERIMENTAL USADO EN LA ELABORACIÓN DEL 5-HIDROXIMETILFURFURAL A PARTIR DE MELAZA DE CAÑA PARA LA OBTENCIÓN DE BIOCOMBUSTIBLE”**, declaramos que:

Los conceptos desarrollados, análisis realizados y las conclusiones del presente trabajo son de exclusiva responsabilidad de los autores, y dando a conocer que el Capítulo I del presente fue realizado por los señores William Mejía y José Serrano.

Cuenca a 9 de Febrero de 2012.

Julio E. Cajamarca V.

Juan A. Valverde J.

ÍNDICE

1	OBTENCIÓN DEL 5-HIDROXIMETILFURFURAL.....	1
1.1	5-hidroximetilfurfural y sus características.	1
1.2	Usos potenciales del hidroximetilfurfural.....	1
1.3	Elementos que intervienen en la experimentación para la obtención del 5-hidroximetilfurfural.....	2
1.3.1	La fructosa	2
1.3.2	Catálisis en química orgánica	3
1.4	Reacción de deshidratación de la fructosa	6
1.5	Fases presentes en la reacción de deshidratación	7
1.6	Solventes empleados.....	9
1.6.1	Consideraciones para la selección del solvente	9
1.6.2	El agua	10
1.6.3	Características del 1-butanol (<i>n</i> -BUTANOL).....	11
1.7	Efecto de las sales inorgánicas.....	12
1.8	Posibles inconvenientes	14
2	CARACTERÍSTICAS Y MODELOS DE SISTEMAS DE CONTROL	15
2.1	Sistema de control analógico (en tiempo continuo).....	15
2.2	Sistemas de control digital (en tiempo discreto).....	16
2.2.1	Sistemas de control de datos muestreados	16
2.2.2	Sistemas de control digital	16
2.3	Sensores y Actuadores	17
2.3.1	Sensores o captadores	17
2.3.2	Actuadores	30
2.4	Estructuras y configuraciones de controladores.	32
2.4.1	Sistemas de control en lazo abierto.....	32
2.4.2	Sistemas de control en lazo cerrado.....	33
2.4.3	Configuraciones del controlador.....	36

2.5	Metodologías de modelación de controladores.....	37
2.5.1	Controlador PID.....	38
3	DISEÑO DEL SISTEMA DE CONTROL.....	42
3.1	Modelación del sistema del reactor químico.	42
3.1.1	Reactor discontinuo o tipo Bach de fase líquida.....	43
3.1.2	Motor de corriente continua controlado por inducido y con imán permanente.	47
3.2	Obtención de datos de información del sistema.	50
3.2.1	Adquisición de datos de la niquelina.	50
3.2.2	Adquisición de datos del motor de corriente continua con imán permanente.	66
3.3	Diseño del controlador para el sistema de temperatura del reactor.	69
3.4	Diseño de los controladores para el sistema de velocidad de agitación del reactor.....	75
3.5	Simulación del sistema.	76
3.5.1	Simulación del sistema de temperatura del reactor discontinuo agitado.....	76
3.5.2	Simulación del sistema de agitación del reactor discontinuo agitado.....	78
3.6	Diseño de la interfaz humana máquina HMI.	80
3.6.1	Software, interfaz humana máquina HMI.....	80
3.6.2	Firmware, interfaz humana máquina HMI.....	85
3.6.3	Hardware.....	93
4	IMPLEMENTACIÓN Y PRUEBAS.....	101
4.1	Pruebas experimentales del comportamiento del sistema compensado.....	103
4.1.1	Pruebas del controlador de velocidad.	104
4.1.2	Pruebas para el ajuste del controlador de temperatura.....	104
4.1.3	Pruebas para la obtención del 5-Hidroximetilfulfural	111
4.2	Ajustes de controladores.	130
4.3	Análisis de resultados experimentales.	131
4.4	Implementación del sistema de control (física).	132
4.4.1	Pruebas iniciales.....	132
4.4.2	Construcción de la estructura mecánica.....	134
4.4.3	Tarjeta electrónica.....	136
4.4.4	Ensamble del Equipo.	137

ANEXO A.....	145
ANEXO B	152
ANEXO C	177
ANEXO D.....	180
ANEXO E	188
ANEXO F	189
ANEXO G.....	190
ANEXO H.....	191
ANEXO I	193
ANEXO J	195
BIBLIOGRAFÍA	201

DEDICATORIA

Dedico este trabajo, a una persona que siempre ha sido mi inspiración mi admiración pero a su vez me ha hecho mucha falta y ha cambiado toda mi vida, este trabajo es dedicado para ti Julia Vanesa, toda mi vida estarás en mi corazón.

Julio Eduardo Cajamarca Villa

A Dios por ser mi padre, mi confidente y regalarme cada maravilloso día para ser feliz. A ti Mami Mary por tu constancia, fuerza y dedicación, gracias por creer en tus hijos, por ser mi heroína y la mamá más mala del mundo☺, ¡tus sacrificios no serán en vano! A ti Papi Flaco por tu calidez, humildad, tu amor hacia los otros y por tu ejemplo al enseñarme el valor del perdón, por creer en mí cuando yo ya no lo hacía, gracias. A René, Ana, Mateo y Alex, por acompañarme, recuerden que el mundo no gira a nuestro alrededor y que el tiempo solamente es un suspiro ¡aprovechémoslo! A Mami Lola y Macario, gracias por estar ahí.

Juan Andrés Valverde Jara

AGRADECIMIENTOS

Agradezco este logro a Dios que me ha dado fuerzas para seguir y seguir, a mis padres que segundo a segundo se han preocupado por todo aquello que me pasa y por lo que siento, que me han dado fortaleza a cada instante, a mi familia que siempre me han apoyado y me alientan a seguir hacia adelante, y me faltarían hojas para escribir a todos los demás que han sido un apoyo o tal vez un tropiezo, que por todos ellos soy lo que soy y estoy aquí para seguir luchando.

Julio Eduardo Cajamarca Villa

A la Universidad Politécnica Salesiana mediante sus catedráticos. A nuestro director de tesis Walter por su apoyo y su ayuda. A toda mi familia (tíos, primos, etc.) por preguntarme cada día ¿cuándo te gradúas? A la familia de Julio por acogerme en su hogar, gracias. William y José por creer en nuestro talento y en nuestro país, a Julio, Silvana, Juan, Franco, Jorge, Pablo, Tania, Manuel, Sofi, Adrián, José, Xavier, Edison, Karla, Pato, Pao, Nina, César, Ricardo, Andrés, Katy, Oswaldo, Franklin, Verit©, Gonzalo, Martín, Carlos, David, Andre, Jorge, Marco, Joha, Luis, Cris, Gaby, Geova, Calin, Diana, Dany, Esteban, Santi, Aracely, Iván, Felipe, Isma, Marilyn, Paul, Rosa, Mauricio, Omar, etc. a sus familias y demás ya que por medio de sus discusiones, historias, preguntas y consejos puedo seguir creciendo y aprendiendo.

Juan Andrés Valverde Jara

CAPÍTULO I

1 OBTENCIÓN DEL 5-HIDROXIMETILFURFURAL

1.1 5-hidroximetilfurfural y sus características.

El 5-Hidroximetilfurfural es un compuesto orgánico derivado de la deshidratación de ciertos azúcares. Es altamente soluble¹ en agua, de aspecto amarillento caramelo. La molécula está conformada por un anillo furánico² (Figura 1.1), que contiene simultáneamente dos grupos funcionales, a saber un aldehído³ y un grupo que caracteriza a los alcoholes. Su fórmula condensada es C₆H₆O₃.

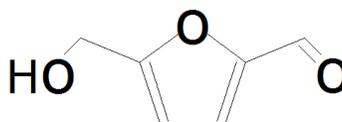


Figura 1.1 Molécula del 5-hidroximetilfurfural.

Sus propiedades físico-químicas se detallan a continuación:

Tabla 1.1 Propiedades físico-químicas del 5-hidroxumetilfurfural

PROPIEDAD FÍSICO-QUÍMICA	VALORES
Masa molar	126.11g/mol
Temperatura de ebullición	114-116°C (1mm Hg)
Punto de fusión	28-34°C (descomposición)
Densidad	1,243g/cm ³ a 25°C

1.2 Usos potenciales del hidroximetilfurfural

Mediante un proceso de hidrogenólisis, sobre un catalizador de Cu-Ru/C, y entre temperaturas que oscilan entre 220°C y 240°C el 5-Hidroximetilfurfural puede convertirse en el 2,5-Dimetilfurano, compuesto que tiene un potencial de convertirse en el biocombustible del futuro. La energía contenida en este biocombustible está alrededor de 31,5MJ/L, similar a la energía contenida en la gasolina comercial que es de 35MJ/L y

¹ **Soluble.**- Que se puede disolver.

² **Furano.**- Compuesto orgánico heterocíclico. Es un líquido claro, incoloro, altamente inflamable y muy volátil, con un punto de ebullición cercano al de la temperatura ambiente.

³ **Aldehídos.**- Son compuestos orgánicos

un 40% superior a la energía contenida en el etanol que bordea los 23MJ/L. Otro de los usos del 5-Hidroximetilfurfural consiste en la producción del ácido 2,5-furandicarboxílico, el cual se ha propuesto como un sustituto del ácido tereftálico en la producción de poliésteres.

1.3 Elementos que intervienen en la experimentación para la obtención del 5-hidroximetilfurfural.

1.3.1 La fructosa

La fructosa⁴, o levulosa, es una forma de azúcar encontrada en las frutas y en la miel. Es un monosacárido⁵ con la misma fórmula empírica que la glucosa pero con diferente estructura. Es una cetohexosa (6 átomos de carbono). Su poder energético es de 4 kilocalorías por cada gramo. Su fórmula química es C₆H₁₂O₆. Las principales propiedades físico-químicas (Tabla 1.2) se detallan a continuación:

Tabla 1.2 Propiedades físico-químicas de la fructosa.

PROPIEDAD FÍSICO-QUÍMICA	VALORES
Masa molar	180,15g/mol
Temperatura de ignición	360°C
Solubilidad en agua	790g/L a 20°C
Punto de fusión	100-110°C (descomposición)
Densidad	1650g/L a 20°C
Valor de PH	5-6(100g/L, H ₂ O,20°C)

La fructosa (Figura 1.2) será el punto de partida para llevar a cabo la reacción de obtención del 5-Hidroximetilfurfural, disolviéndola con fines de experimentación en una fase que además de esta contendrá cloruro de sodio y un ácido fuerte como el ácido clorhídrico.

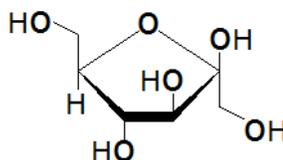


Figura 1.2 Molécula de fructosa.

⁴ **Fructuosa.**- Es una forma de azúcar encontrada en las frutas y en la miel.

⁵ **Monosacáridos.**- O azúcares simples son los glúcidos más sencillos, que no se descomponen para dar otros compuestos, conteniendo de tres a seis átomos de carbono.

1.3.2 Catálisis en química orgánica

“Un catalizador es una sustancia que aumenta (o disminuye) la rapidez de una reacción química sin consumirse ni modificarse en la reacción. La rapidez de una reacción química depende de la barrera de energía que debe salvarse en el proceso de convertir reactivos en productos.

La altura de esta barrera está representada por la energía libre de activación, por lo tanto un catalizador aumenta la rapidez de una reacción química al proporcionar una ruta para la reacción con menor energía libre de activación.

Se han estudiado las formas en que un catalizador⁶ puede disminuir la energía libre de activación, y se detallan a continuación:

- a. La reacción catalizada y no catalizada pueden tener mecanismos diferentes, pero similares y el catalizador proporciona una forma de convertir el reactivo (Figura 1.3) en una especie menos estable.

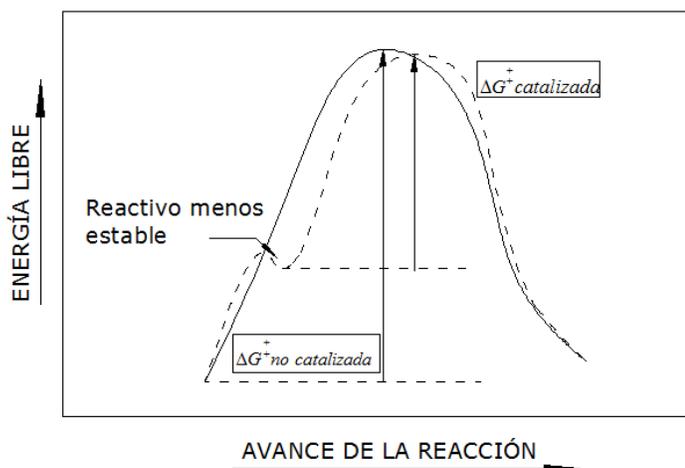


Figura 1.3 El catalizador convierte al reactivo en una especie menos estable.

- b. Las reacciones catalizada y no catalizada pueden tener mecanismos diferentes (Figura 1.4), pero similares, y el catalizador proporciona una forma de hacer más estable el estado de transición.

⁶ **Catalizador.-** Es una sustancia que está presente en una reacción química en contacto físico con los reactivos, y acelera, induce o propicia dicha reacción sin actuar en la misma.

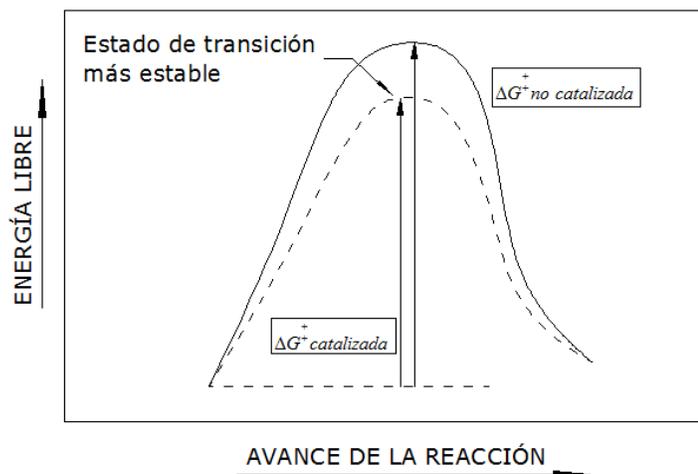


Figura 1.4 El catalizador estabiliza el estado de transición.

- c. El catalizador puede cambiar por completo el mecanismo de la reacción y proporcionar una ruta alterna (Figura 1.5), con menor energía libre de activación que la reacción no catalizada.⁷

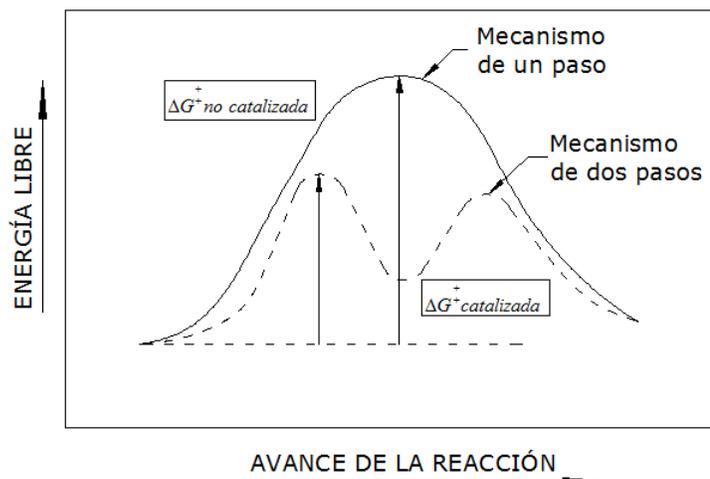


Figura 1.5 La reacción catalizada se realiza por una ruta energéticamente más favorable.

Un catalizador se agrega a la reacción en cantidades catalíticas pequeñas mucho menores que la cantidad de sustancia del reactivo, en forma típica **del 1 al 10% de la cantidad de moles del reactivo**.

Hay que notar también que el valor de la constante de reacción es independiente de cualquier fenómeno catalítico, es decir, el catalizador no altera la cantidad de

⁷ Yurkanis P. Química Orgánica. Quinta edición 2008. Person Education. México. Pág. 1063-1065.

producto que se forma cuando la reacción llega al equilibrio, sino que únicamente cambia la rapidez con la que se forma el producto.

Un catalizador puede proporcionar distintas alternativas para proporcionar una ruta más favorable de reacción:

- I. Puede aumentar la susceptibilidad de un electrófilo⁸ frente a un ataque nucleófilo.
- II. Puede aumentar la reactividad de un nucleófilo⁹.
- III. Puede aumentar la propiedad saliente de un grupo y convertirlo en una base más débil.
- IV. Puede aumentar la estabilidad de un estado de transición.

1.3.2.1 Catálisis ácida

Se propone en esta investigación, un mecanismo de deshidratación de la fructosa catalizada por un ácido, para la formación del 5-hidroximetilfurfural. Un catalizador ácido aumenta la rapidez de la reacción al ceder un protón a un reactivo. Hay dos tipos de catálisis ácida: una catálisis ácida específica y una catálisis ácida general.

En la *catálisis ácida específica* (Figura 1.6) el protón se transfiere al reactivo antes que transcurra el paso lento en la reacción, mientras que en la *catálisis ácida general* (Figura 1.7) el protón se transfiere al reactivo durante el paso lento de la reacción.

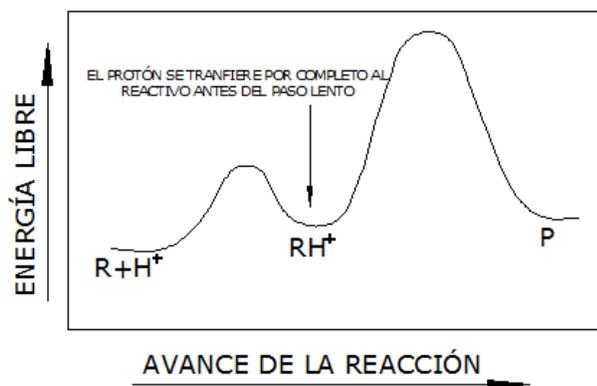


Figura 1.6 Catálisis ácida específica.

⁸ **Electrófilo.**- Reactivo químico atraído hacia zonas ricas en electrones que participa en una reacción química aceptando un par de electrones formando un enlace con un nucleófilo.

⁹ **Nucleófilo.**- Es una especie que reacciona cediendo un par de electrones libres a otra especie (el electrófilo), combinándose y enlazándose covalentemente con ella.

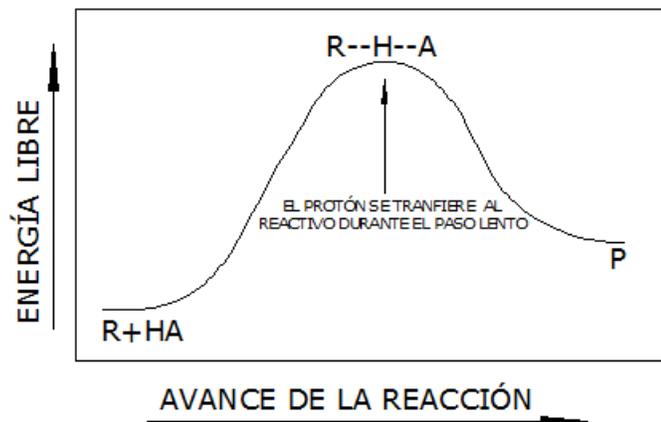


Figura 1.7 Catálisis ácida general.

Tanto la catálisis ácida específica como la general aumentan la rapidez de la reacción en la misma forma donando un protón para facilitar ya sea la formación o la ruptura de enlaces. Los dos tipos de catálisis ácidas solo difieren en el grado con el que se transfiere el protón en el estado de transición del paso lento de la reacción.

Un catalizador ácido específico debe ser un ácido suficientemente fuerte como para protonar¹⁰ totalmente al reactivo antes que comience el paso lento, mientras que un catalizador ácido general puede ser un ácido más débil porque solo transfiere en forma parcial un protón en el estado de transición del paso lento.

1.4 Reacción de deshidratación de la fructosa

La reacción de deshidratación de la fructosa, se llevará a cabo en medio acuoso en presencia de un ácido fuerte para formar el 5-Hidroximetilfurfural. Sin embargo, según lo expuesto anteriormente es necesario definir tentativamente un mecanismo de reacción que explique el fenómeno de deshidratación en presencia de un ácido fuerte. El primer paso de esta reacción (Figura 1.8) es la interacción de un protón donado por el ácido clorhídrico a la fructosa.

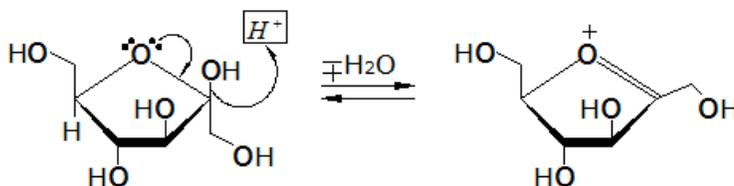


Figura 1.8 Acción del protón sobre la molécula de la fructosa.

¹⁰ **Protonar.**- Es decir mezclar una solución ácida.

Después de la formación de este compuesto intermedio ciclo furánico, se pueden obtener productos secundarios que no son de interés en esta reacción. El punto verdaderamente interesante se presenta en la acción de un grupo nucleofílico o básico (Figura 1.9) debido a la participación de compuestos que contienen un grupo halógeno como pueden ser cloruro de sodio, cloruro cromoico o cromoso, después de cuya acción se forma un último intermediario antes de la producción del 5-Hidroximetilfurfural.

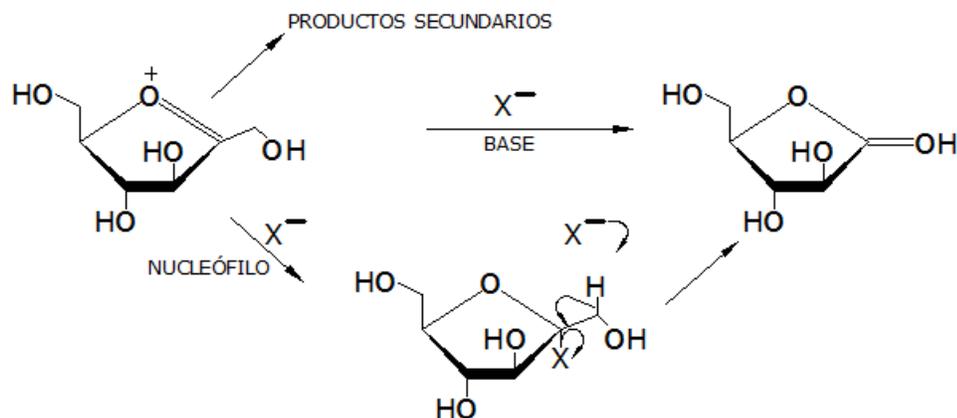


Figura 1.9 Acción de un nucleófilo y una base debido a la presencia de compuestos halogenados.

Finalmente, el último intermediario se transforma en el 5-Hidroximetilfurfural (Figura 1.10), completándose de esa manera la totalidad de la reacción de deshidratación de la fructosa.

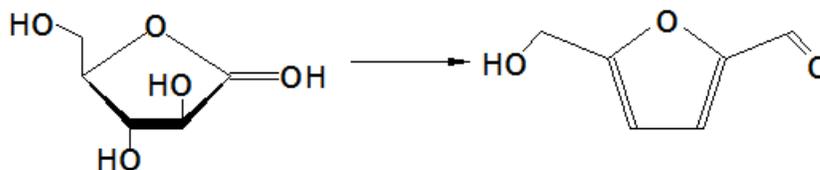


Figura 1.10 La reacción de deshidratación de la fructosa ha sido completada.

1.5 Fases presentes en la reacción de deshidratación

Es posible obtener el 5-hidroximetilfurfural en un reactor bifásico (Figura 1.11), usando para ello solventes de bajo punto de ebullición, de modo que se elimine la necesidad de separaciones costosas. La fase acuosa reactiva contendrá el catalizador ácido (solución de ácido clorhídrico de molaridad conocida) y la fructosa, mientras que la fase orgánica parcialmente miscible en agua contendrá 1-butanol, un solvente orgánico, cuya función será la de extraer el 5-Hidroximetilfurfural que se va produciendo, sin intervenir en la reacción.

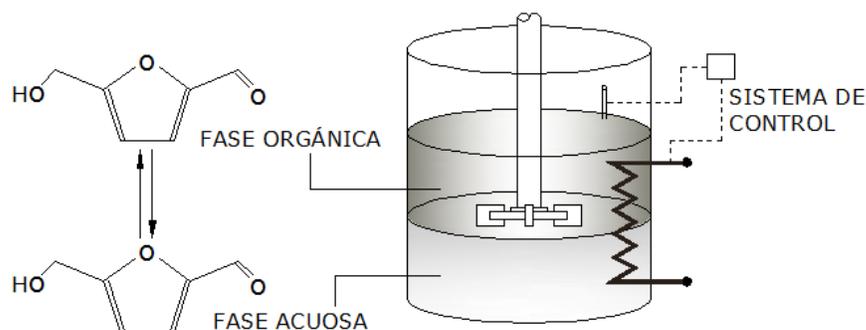


Figura 1.11 Reactor bifásico conteniendo la fase orgánica y la fase acuosa.

Para poder determinar el grado de extracción de las fases, utilizaremos un coeficiente K_i , que determinará el reparto del producto en las fases, a saber:

$$K_i = \frac{[HMF]_{FASE\ ORGÁNICA}}{[HMF]_{FASE\ ACUOSA}} \quad \text{Ecuación 1.1}$$

El rendimiento vendrá descrito por la relación:

$$Y_G = \frac{N_{HMF\ PRODUCIDOS}}{N_{FRUCTOSA\ REACCIONADOS}} \quad \text{Ecuación 1.2}$$

Esta ecuación en realidad nos permite calcular el rendimiento global, y se expresa en función del número de moles y no de flujo molar debido a que los experimentos se realizarán en un reactor discontinuo con agitación constante. Otro parámetro importante que mide la especificidad de la reacción hacia el producto que queremos obtener se define como selectividad y su cálculo se realiza como sigue:

$$S_G = \frac{Y_G}{Y_{SUB}} = \frac{Y_G}{1 - Y_G} \quad \text{Ecuación 1.3}$$

Es decir, la selectividad se expresa como el cociente entre el rendimiento de producto deseado y el rendimiento de producto no deseado.

Se ha demostrado que el rendimiento está directamente relacionado con el coeficiente de reparto K_i (Dumesic et al., 2007). Al incrementarse K_i el rendimiento también lo hace, es decir, mientras más eficiente sea el solvente en la extracción del 5-

Hidroximetilfurfural se elevará este coeficiente de reparto y por lo tanto se incrementará el rendimiento, o equivalentemente la eficacia de la remoción de 5-Hidroximetilfurfural de la fase acuosa previene la formación de reacciones secundarias.

1.6 Solventes empleados

1.6.1 Consideraciones para la selección del solvente

La eficiente producción de 5-Hidroximetilfurfural requiere la minimización de reacciones secundarias no deseadas que envuelven al reactante, a las reacciones intermedias y al producto. Múltiples sistemas de reacción han sido reportados para la producción de 5-Hidroximetilfurfural usando diferentes solventes.

La deshidratación de la fructosa usando ácidos minerales es generalmente un proceso no selectivo (rendimientos <20%) debido a la degradación del 5-Hidroximetilfurfural por reacciones de rehidratación formando ácido levulínico y ácido fórmico. Contrariamente, los solventes apróticos con alto punto de ebullición como el dimetil-sulfóxido suprimen las reacciones secundarias no deseadas y generan altos rendimientos de 5-Hidroximetilfurfural mayores al 90%. Sin embargo, la separación del 5-Hidroximetilfurfural de inclusive trazas del disolvente es difícil y requiere una elevada cantidad de energía para su separación.

Los sistemas bifásicos en el cual un solvente orgánico inmisible en agua es adicionado para extraer continuamente el 5-Hidroximetilfurfural de la fase acuosa ofrece una importante ventaja en que el producto es separado del reactante y del medio de reacción, de este modo se protege contra las reacciones de degradación. Sin embargo, la mayoría de los solventes de extracción usados muestran un pobre reparto del 5-Hidroximetilfurfural hacia la fase orgánica. Recientemente, se ha desarrollado una estrategia catalítica de producción del 5-Hidroximetilfurfural con gran rendimiento a partir de soluciones concentradas de fructosa en un reactor bifásico por un incremento del reparto del 5-Hidroximetilfurfural en la fase orgánica.

La fase acuosa reactiva en el reactor bifásico contiene un catalizador ácido y un azúcar y la fase de extracción contiene un solvente orgánico parcialmente miscible (butanol, hexanol, etc.).

Algunos factores necesitan ser considerados en la apropiada selección de un solvente para extraer el 5-Hidroximetilfurfural en un sistema que contenga una sal inorgánica. En primer lugar, el proceso de producción del 5-Hidroximetilfurfural en su conjunto ha sido evaluado en función de los pasos que se llevarán a cabo en el procesamiento para el cual el producto final va dirigido. Por ejemplo, para la síntesis de

éteres a partir del 5-Hidroximetilfurfural (derivados de la reacción de eterificación del 5-Hidroximetilfurfural con alcoholes) para su aplicación como aditivos de combustibles, el uso de alcoholes primarios como solventes es ideal cuando se considera el procesamiento posterior del 5-Hidroximetilfurfural. Similarmente, en términos de producción de alcanos de cadena larga para su uso como biocombustibles (por condensación aldólica) y extracción de 5-Hidroximetilfurfural con cetonas convertiría al solvente en un reactante en el subsecuente paso de condensación aldólica.

Para procesos que comprenden hidrogenólisis seguida por destilación (producción del compuesto químico intermedio denominado 2,5-Dimetilfurano) es adecuado el uso de un solvente saturado con un bajo punto de ebullición como el Tetrahidrofurano, para evitar la degradación del solvente y minimizar la energía gastada durante la destilación. Adicionalmente los solventes que pueden ser obtenidos directamente desde biomasa¹¹ son sostenibles y son de fácil integración a un proceso de biorefinería¹².

Actualmente, los principales solvente derivados de la biomasa son el etanol, isopropanol, n-butanol y acetona, por lo tanto nuevas técnicas están siendo desarrolladas continuamente para sintetizar nuevos bio-solventes.

Finalmente, otra consideración para la selección del solvente incluye la estabilidad química, lo que quiere decir que el solvente sea inerte respecto a los componentes del sistema.

1.6.2 El agua

El agua es el más antiguo de los disolventes. Desde un punto de vista medioambiental el agua es un excelente disolvente, ya que no es tóxico ni inflamable. Las particulares propiedades del agua, en especial su elevado orden interno impuesto por la asociación de sus moléculas mediante enlaces de hidrógeno son los directos responsables de sus ventajas y limitaciones como solvente y como medio de reacción. A causa de las interacciones hidrofóbicas, debidas a dicho orden, algunas reacciones orgánicas son aceleradas en agua, llegando incluso a modificarse su selectividad. El mayor inconveniente para usar el agua como medio de reacción es la baja solubilidad en la misma de muchos compuestos orgánicos. Para solventar esta limitación se usan a menudo medios orgánicos acuosos. En muchas ocasiones se usan estos sistemas lo que

¹¹ **Biomasa.**- Materia orgánica originada en un proceso biológico, espontáneo o provocado, utilizable como fuente de energía.

¹² **Biorefinería.**- Estructura que integra procesos de conversión de biomasa y equipamiento para producir combustibles, energía y productos químicos a partir de la biomasa.

mejora notablemente la separación de productos al final de la reacción. En estos casos es frecuente el empleo de *catalizadores de transferencia de fase* para mejorar la cinética del proceso. Un catalizador de transferencia de fase actúa transfiriendo un reactivo a través de la interface.

Una estrategia alternativa es el uso de agentes tenso activos, para que ayuden a solubilizar tanto reactantes como catalizadores.

1.6.3 Características del 1-butanol (*n*-BUTANOL)

El 1-Butanol es un alcohol primario con cuatro carbonos en su estructura, cuya fórmula molecular es $C_4H_{10}O$. Sus isómeros¹³ incluyen al isobutanol, 2-butanol y el *terc*-butanol. Se obtiene como un subproducto de la fermentación de azúcares y de otros carbohidratos, lo que abre la posibilidad de su obtención como un disolvente procedente de biomasa como se mencionó anteriormente. El *n*-butanol es un disolvente polar prótico. Esto significa que este alcohol contiene un hidrógeno que está unido a un átomo electronegativo. La gran diferencia de electronegatividades entre los átomos de oxígeno y de hidrógeno combinado con el pequeño tamaño del átomo de hidrógeno, hacen que pueda formar fácilmente enlaces de hidrógeno con los solutos¹⁴. Es volátil, de apariencia clara y tiene un fuerte olor a alcohol. Es miscible con muchos solventes orgánicos e incompatible con oxidantes fuertes. Respecto a su solubilidad en agua comparada con otros alcoholes, se puede observar que solo es soluble en un 8,3%, tal como lo indica la tabla 1.3:

Tabla 1.3 Solubilidad de algunos alcoholes en el agua.

ALCOHOL	SOLUBILIDAD EN AGUA
Metanol	En todas proporciones
Etanol	En todas proporciones
Butanol	8,3 %
Octanol	Prácticamente insoluble

¹³ **Isómeros.-** Propiedad de ciertos compuestos químicos que con igual fórmula química, es decir, iguales proporciones relativas de los átomos que conforman su molécula, presentan estructuras moleculares distintas y, por ello, diferentes propiedades.

¹⁴ **Soluto.-** Componente de una disolución que se disuelve en otro. El soluto se considera siempre como el componente minoritario de la mezcla.

Sus propiedades físico-químicas vienen dadas a continuación:

Tabla 1.4 Propiedades físico-químicas del 1-butanol.

PROPIEDAD FÍSICO-QUÍMICA	VALORES
Masa molar	74,12g/mol
Temperatura ebullición	116-118°C
Punto de fusión	-90°C
Densidad	810 g/L a 25°C
Solubilidad en agua	63,2g/L
Momento dipolar	1,52D

1.7 Efecto de las sales inorgánicas

La adición de cloruro de sodio en la fase acuosa mejora el reparto del 5-Hidroximetilfurfural hacia la fase extracción por sus propiedades y conduce a incrementar el rendimiento del 5-Hidroximetilfurfural sin el uso de solventes de alto punto de ebullición.

Es importante notar que el incremento relativo del coeficiente de reparto K_i depende de la naturaleza del solvente. Por ejemplo, el sistema conformado por 1-butanol y cloruro de sodio muestra (Figura 1.12) un incremento en el coeficiente de reparto del doble (de 1,6 a 3,2) con respecto al sistema con el mismo disolvente pero sin sal. También la 2-Butanona presenta un incremento en el coeficiente de reparto del triple con respecto al sistema sin sal inorgánica (de 1.8 a 5.4).

*Figura 1.12 Efecto del NaCl en el coeficiente de reparto K_i en sistemas bifásicos. Las barras oscuras corresponden a sistemas con presencia de NaCl mientras que las plomas representan al sistema sin NaCl.
Fuente: Roman-Leshov Y., Dumesic J.A. (2009). Top Catal.*

Este efecto es atribuido a que el NaCl induce un efecto de “salting out¹⁵” en el solvente de extracción, mediante el cual los iones en solución alteran las fuerzas intermoleculares entre los líquidos en equilibrio, resultando en un aumento de la inmiscibilidad sobre el sistema y por lo tanto, el decrecimiento de la solubilidad mutua de las fases orgánica e inorgánica mejora la extracción del 5-Hidroximetilfurfural de la fase acuosa.

El efecto de “salting out” depende de la naturaleza de las interacciones iónicas entre todos los componentes del sistema, y estos dependen de parámetros como concentración, temperatura y presión.

Es importante destacar que el efecto de “salting out” tiene otros dos atributos relevantes. Primero podemos aprovechar para utilizar solventes que en ausencia de sales inorgánicas son completamente miscibles en el agua, citando algunos como el 1-propanol, 2-propanol entre otros.

También se puede aprovechar este efecto en sistemas parcialmente miscibles. Para ello se define la Temperatura Superior Crítica de Solubilidad como la temperatura a la cual una mezcla de líquidos con solubilidad parcial se vuelve completamente miscible. La presencia de sales inorgánicas incrementa esta temperatura crítica de manera que

¹⁵ **Salting out.**- Es una técnica de extracción de proteínas que consiste en concentrar la fase líquida tanto como sea requerido para que las proteínas se precipiten, lo que se produce es que al solubilizar la sal, esta se rodea de agua, limitando las moléculas que solubilizan a las proteínas, logrando que al final, no haya moléculas disponibles de agua precipitando las proteínas.

permite a los sistemas bifásicos parcialmente miscibles permanecer inmiscibles inclusive a altas temperaturas.

1.8 Posibles inconvenientes

Se ha afirmado anteriormente que los catalizadores ácidos en diversos procesos de obtención del 5-Hidroximetilfurfural provocan reacciones secundarias aumentando significativamente el costo del producto. En el agua en condiciones de acidez el 5-Hidroximetilfurfural se descompone en ácido levulínico y en ácido fórmico por rehidratación (Figura 1.13). El ácido levulínico es particularmente difícil de separar del producto de nuestro interés. Al minimizar la exposición del 5-Hidroximetilfurfural a soluciones acuosas ácidas a temperaturas elevadas, la pérdida del rendimiento de producto en ácido levulínico se mantiene muy baja. Es por ello que se debe aumentar la eficiencia en la extracción del producto desde la fase acuosa, usando cloruro de sodio.

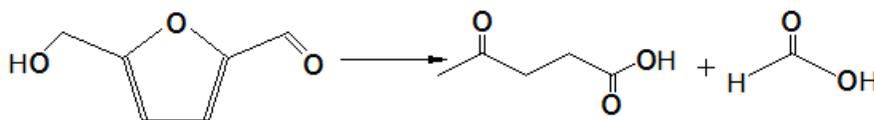


Figura 1.13 Formación de ácido levulínico y ácido fórmico al exponer al HMF a soluciones acuosas ácidas a temperaturas elevadas.

CAPÍTULO II

2 CARACTERÍSTICAS Y MODELOS DE SISTEMAS DE CONTROL

Los sistemas de control automático son ampliamente usados en automatización, sistemas industriales, robots, cadenas de montaje en serie, etc. para mejorar la calidad, cantidad y el costo del producto; mejorar la calidad de vida del ser humano ya sea con una lavadora, la domótica¹⁶ en el hogar, un automóvil asistido por computador, los sistemas de navegación satelital; en la innovación, los avances tecnológicos y científicos, etc.

Un sistema automático de control es un conjunto de componentes y elementos físicos conectados o relacionados entre sí, para que regulen y dirijan por si mismos su comportamiento, sin intervención de agentes externos (incluido el ser humano) y corrijan los posibles errores que se presenten, para ello manipulan las variables de control del proceso (entrada) de tal manera que se dominan las variables de salida para alcanzar los valores determinados.

2.1 Sistema de control analógico (en tiempo continuo).

Un sistema de control analógico en tiempo continuo trabaja con señales que son funciones de la variable continua tiempo t (Figura 2.1).

Si de una señal $x(t)$ en tiempo continuo se puede tomar cualquier valor en un intervalo (\mathbf{a}, \mathbf{b}) , donde \mathbf{a} puede ser $-\infty$ y \mathbf{b} $+\infty$, entonces la señal en tiempo continuo es llamada señal analógica. (Hsu, 1995, pág. 2)

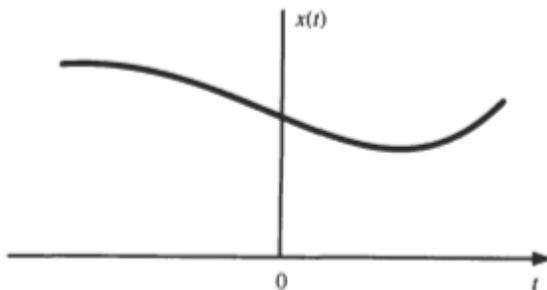


Figura 2.1 Representación gráfica de una señal en tiempo continuo. (Hsu, 1995)

Cabe recalcar que existen sistemas de control CA que ocupan señales moduladas, por otro lado los sistemas de control cd carecen de dicha modulación, a estos sistemas se los puede modelar mediante ecuaciones diferenciales.

¹⁶ **Domótica.**- Conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda.

2.2 Sistemas de control digital (en tiempo discreto).

Kuo sostiene una diferencia entre los sistemas de control digital y los sistemas de datos muestreados (Sistemas de control automático, 1996, pág. 18).

2.2.1 Sistemas de control de datos muestreados

Un sistema de control de datos muestreados (Figura 2.2) tiene una visión más general del sistema ya que utiliza señales que solamente están definidas en instantes de tiempo discreto y se encuentran en forma de pulsos de datos, por lo que la información que recibe solo se encuentra en instantes específicos de tiempo.



Figura 2.2 Diagrama de bloques de un sistema de control de datos muestreados. (Golnaragui & Kuo, 2010)

Las variables de estos sistemas solo pueden cambiar en valores discretos de tiempo (Figura 2.3), es decir en instantes denominados nT o t_n ($n=0, 1, 2, \dots$), donde estos equivalen a los tiempos en los que se puede realizar una medición física, se pueden describir en ecuaciones en diferencias¹⁷ después de su discretización.

Cuando de una señal en tiempo discreto $x[n]$ se puede tomar solo un número finito de valores, entonces la llamamos señal digital.

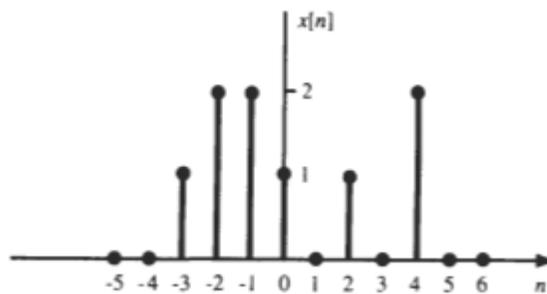


Figura 2.3 Representación gráfica de una señal en tiempo discreto. (Hsu, 1995)

2.2.2 Sistemas de control digital

El sistema de control digital usa señales que forman parte de un código como puede ser BCD o binario y que para poder utilizarlas en un controlador necesita de un computador o control digital.

¹⁷ Son similares a las ecuaciones diferenciales, sustituyendo las funciones por sucesiones.

La mayoría de los procesos que se desean controlar son analógicos, por lo que es absolutamente necesario digitalizarlos con convertidores A/D (analógico-digital), muestreando sus señales en tiempo discreto y codificándolas numéricamente (comúnmente en código binario).

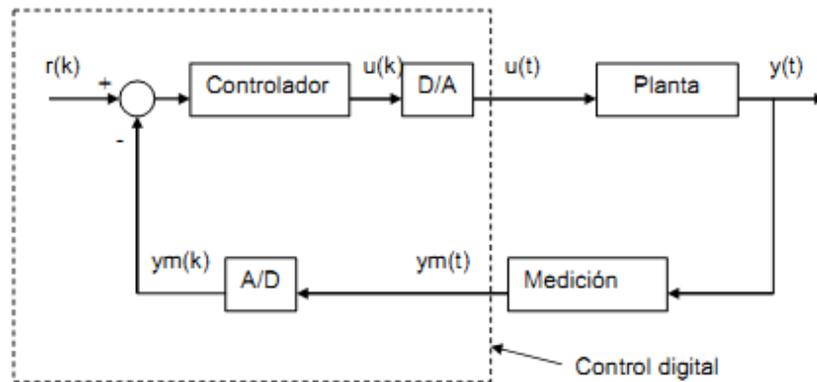


Figura 2.4 Diagrama de bloques de un sistema de control digital. (Zambrano Pérez, 2009)

Este sistema muestra que la señal de control de la planta luego de ser acondicionada es digitalizada, para que el control digital calcule el error que existe y por medio del controlador generar la señal de control, para que esta pueda actuar en la planta debe necesariamente ser convertida por un convertidor D/A (digital-analógico).

2.3 Sensores y Actuadores

En nuestro medio es necesario conocer los acontecimientos que suceden a nuestro alrededor, poder medir de alguna manera un fenómeno y modificarlo para obtener un resultado deseado, esto se puede dar dentro de la industria o en otras instancias, para poder efectuar el proceso de *medir-decidir-actuar*, para lo cual se necesita instrumentos que nos permitan llevar a cabo la medición (sensores) y manipulación (actuadores) del proceso como tal.

2.3.1 Sensores o captadores

Son aquellos que permiten representar una manifestación de cualidades o fenómenos físicos transformándolos a otra magnitud (habitualmente eléctrica) y tener un equivalente a la medición realizada para poder manipularla y cuantificarla, estos suelen ser contruidos por componentes pasivos (resistencias variables, PTC, NTC, LDR, etc., y componentes que varían su magnitud en función de alguna variable), y componentes activos.

Estos dispositivos se clasifican según su arquitectura interna, teniendo una variedad muy extensa: según el diseño de los fabricantes, las aplicaciones para lo que fueron destinados, los materiales de construcción, etc., a continuación presentamos una

síntesis de la clasificación de los sensores para posteriormente centrarnos en dos tipos, los de temperatura y velocidad:

Síntesis de Clasificación de Sensores

- Sensores de Fuerza.
- Sensores de Torque.
- Sensores de Presión.
- Sensores de Flujo-Caudal.
- Sensores de Nivel.
- Sensores de Desplazamiento Lineal y Angular, Posición.
- Sensores de Velocidad, Aceleración y Vibraciones.
- Sensores de Sonido.
- Sensores de Luz.
- Sensores Magnéticos.
- Sensores de Proximidad.
- Sensores de Humedad.
- Sensores Químicos.
- Sensores de Deformación.
- Sensores de Temperatura

2.3.1.1 Sensores o transductores¹⁸ de temperatura

Es aquel dispositivo que nos permite medir la temperatura de un entorno y nos da un equivalente en una señal analógica o digital. Dependiendo de aplicación que se vaya a dar se los catalogan de diferentes maneras, por su precisión, área de uso y rangos en los que van a ser manejados, entre otros parámetros.

2.3.1.2 Tipos de sensores de temperatura

- 1 Termopares.-** Es un sensor formado por la unión de dos metales de características diferentes que producen un voltaje inducido por la diferencia de temperatura debido a efectos termoeléctricos.
- 2 Resistivos.-** Se basan en la variación de la resistencia por la temperatura, con coeficiente positivo PTC y de coeficiente negativo las NTC llamados termistores. En este grupo también se encuentran los PT100.
- 3 Semiconductores.-** Su característica está dada por la variación de la conducción de la unión p-n al estar polarizados directamente.

¹⁸ **Transductor.-** Transforma un tipo de variable física de entrada en uno diferentes de salida

2.3.1.3 Termopares

Son muy robustos y económicos a más de tener una buena estabilidad, se encuentran constituidos por dos hilos de diferente material siendo estos puros o de aleaciones homogéneas que se encuentran unidos en un extremo, esta unión toma el nombre de unión de medición.

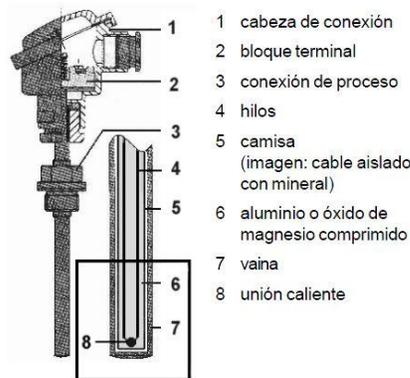


Figura 2.5 Esquema con las partes de una termo-resistencia. (Iñi, 2010)

La base de funcionamiento del termopar se basa en el efecto Seebeck (Thomas Seebeck, 1770-1831), este consiste en la generación de corriente por dos metales heterogéneos (unión caliente y unión fría) que se encuentran a diferentes temperaturas.

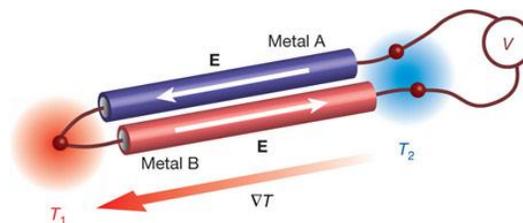


Figura 2.6 Efecto Seebeck. (Taringa)

En los termopares no se puede hacer una medición de manera directa, debido a que si se conecta en los terminales un voltímetro para medir la tensión de Seebeck¹⁹, esta unión genera otro punto de medida variando así la medida original del termopar, puesto que forman corrientes que atraviesan el circuito, es por ello que para poder realizar una medición se tiene que forzar que el sistema vaya a un valor de temperatura conocida como se muestra en la Figura 2.7.

¹⁹ Tensión de Seebeck.- Hace referencia al efecto Seebeck.

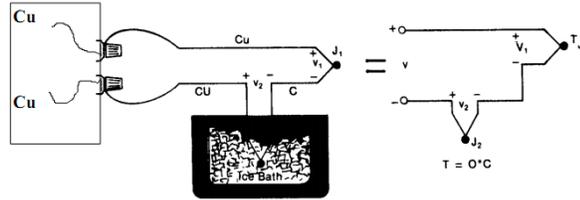


Figura 2.7 Unión de referencia con baño termostático. (Muñoz, Universitat de Valencia, 2005)

De esta manera conseguimos que la medida que nos da el voltímetro ahora sea:

$$V = V_1 - V_2 \cong \alpha(t_{j1} - t_{j2})$$

Ecuación 2.1

Donde t_{j1} y t_{j2} son las temperaturas de las uniones en grados kelvin, si queremos convertir esto a grados centígrados tenemos:

$$V = V_1 - V_2 \cong \alpha[(T_{j1} + 273,15) - (T_{j2} + 273,15)] = \alpha(T_{j1} - T_{ref})$$

Ecuación 2.2

De esta manera vemos que para realizar la medición hay que tener un voltaje de referencia. Dado que este no es el tema específico de estudio, la explicación completa se obviará y se concluye que en la ecuación 2.2 se muestra que el valor del voltaje es la temperatura que se está midiendo menos la temperatura de referencia (suele ser un sensor de temperatura ambiente).

2.3.1.4 Resistivo

2.3.1.4.1 RTD

Es un dispositivo basado en la dependencia de la temperatura que tienen los materiales para variar sus características resistivas, con un rango de medida que oscila entre los -200°C a 800°C . Pueden ser construidas de platino por las prestaciones que tiene, permitiendo medidas más exactas y estables a más de una mejor linealidad en un rango de 0°C a 500°C , a diferencia de las que son construidas con níquel o aleaciones de níquel, que no son tan exactas ni lineales pero son más económicas.

Para representar la variación de temperatura se toma en cuenta el coeficiente de temperatura del material, que determina la variación resistiva por cada grado de temperatura que varía, esto se encuentra dado por la siguiente ecuación:

$$R_t = R_0 \cdot (1 + \alpha \cdot t)$$

Ecuación 2.3

Dónde:

R_t = Resistencia en ohms a $t^{\circ}\text{C}$
 R_0 = Resistencia en ohms a 0°C

α = Coeficiente de temperatura de la resistencia cuyo valor entre 0°C y 100°C es de 0.003850 $\Omega \left(\frac{1}{\Omega}\right) \left(\frac{1}{^\circ\text{C}}\right)$ en la escala practica de temperaturas internacional (IPTS-68).
 t = Temperatura actual.

Las RTD's se comportan como transductores con la ayuda de circuitos que transforman la temperatura en señales de voltaje, teniendo diferentes métodos de conexión los mismos que se detallan a continuación:

- **Método de 2 hilos.-** Es uno de los más sencillos pero a su vez es el menos confiable de todos, puesto que se suma directamente la resistencia de los conductores que van conectados al sensor (Figura 2.8), en este caso el valor resistivo del sensor $R(t)$ nos da la relación directa de la temperatura, pero al sumarse los valores de los conductores tendríamos $R(t) + R_{c1} + R_{c2}$

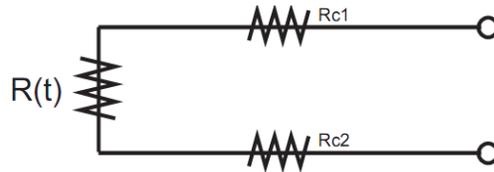


Figura 2.8 Método de 2 hilos. (Arian, 2008)

Esta variación se puede explicar de la siguiente manera; si tenemos una temperatura de 100°C el valor resistivo sería 138.5 Ω , si los cables tienen un valor resistivo de $R_{c1} = 1.4 \Omega$ y $R_{c2} = 1.6 \Omega$, tenemos que el valor total es de 141.5 Ω que equivale a un valor de temperatura de 108°C, existiendo un valor de 8°C de diferencia del valor original, para disminuir el valor resistivo del mismo se utilizan cables de un diámetro considerable, por ejemplo un cable 18 AWG que tiene un valor resistivo de 0.0193 Ω/m .

- **Método de 3 hilos.-** Es un método más efectivo que el anterior, consiste en un sistema basado en el puente de Wheatstone, la condición fundamental que se debe dar en este caso, es que la resistencia de los tres cables sea de igual valor (Figura 2.9), además se considera que el dispositivo lector del sensor debe estar adecuado para este tipo de medición.

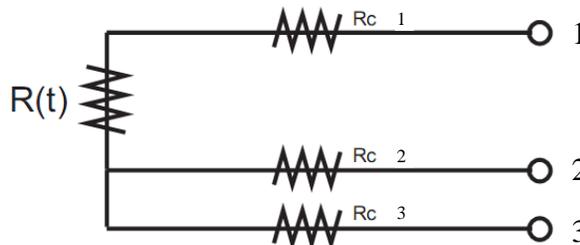


Figura 2.9 Método de 3 hilos. (Arian, 2008)

En este caso el sistema mide la resistencia que existe entre 1 y 2, luego se mide la resistencia entre 2 y 3, procediendo a continuación a restar los valores de las mismas obteniendo el valor de la $R(t)$, eliminando así el valor resistivo de los cables.

- **Método de 4 hilos.-** Este es más preciso que sus predecesores, pero también es mucho más costoso, por el tipo de instrumentos que se utiliza.

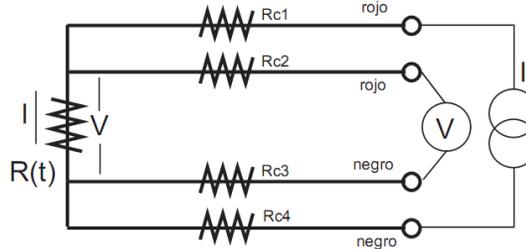


Figura 2.10 Método de 3 hilos. (Arian, 2008)

En este caso se hace circular una corriente por R_{c1} y R_{c4} la cual al pasar por $R(t)$ provocará una tensión que será medida por un voltímetro de alta impedancia en R_{c2} y R_{c3} , al no haber circulación de corriente por estos cables la tensión que se mida será directamente la de $R(t)$, al tener el voltaje y la corriente podemos mediante la ley de Ohm obtener el valor resistivo (Figura 2.10).

Una RDT-PT100 toma este nombre debido a su resistencia aproximada de 100Ω a 0°C , siendo de precio ligeramente elevado, pero es ideal para trabajar a temperaturas bajas que oscilan entre los -100°C a 200°C .

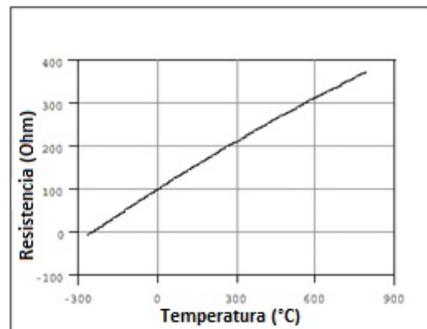


Figura 2.11 Comportamiento de un PT100. (Thermibel)

2.3.1.4.2 Los PTC y NTC

Son resistencias construidas con cristales de óxido metálico para que presenten un alto coeficiente de temperatura y variar así su valor resistivo. Las PTC's suelen estar fabricadas de Bario, Estroncio y Titanio, presentando un coeficiente de temperatura

positivo, mientras que las NTC pueden tener mezclas de óxidos como son Magnesio, Níquel, Cobre, Hierro, Cobalto y presentan un coeficiente de temperatura negativo.

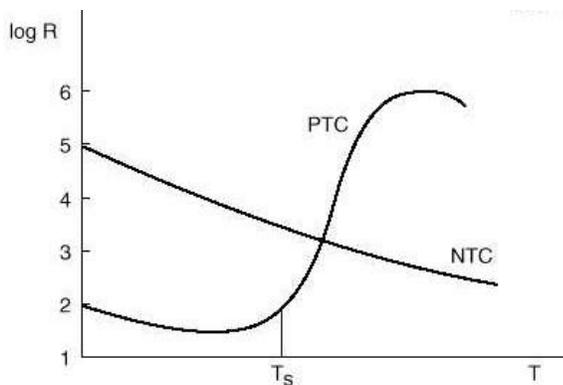


Figura 2.12 Curvas características resistencia/temperatura de PTC y NTC. (IFEnt)

Las PTC no son muy utilizadas en relación de las NTC, que tienen un valor típicamente entre 50Ω y $1M\Omega$ a 25°C pudiendo mejorar su respuesta al colocar una resistencia en paralelo que logra una respuesta más lineal de la NTC.

2.3.1.5 Semiconductores

La corriente de conducción de una unión p-n polarizada directamente responde de manera exponencial, a la tensión directa e inversa a la temperatura, esto se aprecia en la ecuación:

$$I = I_{SAT} \times e^{\left(\frac{q}{KT \times V}\right)}$$

Ecuación 2.4

Dónde:

- I_{SAT} = corriente de saturación
- q = carga del electrón
- k = constante de Boltmann
- T = temperatura
- V = tensión

Por ello se puede tomar como sensor de temperatura a un diodo semiconductor ordinario, siendo este no de precisión como los anteriores, pero nos puede dar unos muy buenos resultados a un muy bajo costo dependiendo de la aplicación, para utilizar un diodo como sensor se necesita de una buena calibración y la corriente de excitación debe ser lo más estable posible, utilizando una fuente de corriente constante o caso contrario una resistencia conectada a una fuente de voltaje estable. El coeficiente de temperatura que tiene un diodo está alrededor de $2,3 \text{ mV}/^\circ\text{C}$, teniendo una variación casi lineal.

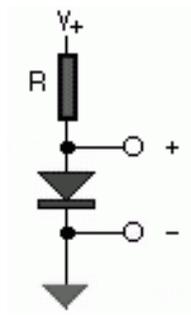


Figura 2.13 Diodo utilizado como sensor de temperatura. (Carletti, 2009)

2.3.1.6 Criterios de Selección del Sensor

El diseñador debe estar consciente del tipo de utilidad que va a dar al mismo, para así poder escoger el más idóneo para su aplicación, tomando en cuenta ciertos parámetros como son:

- Temperatura máxima.
- Rangos de temperatura que se necesitan medir.
- Exactitud.
- Velocidad de respuesta.
- Costos.
- Requerimientos de mantención.
- Fiabilidad.

Además de estos parámetros se da a conocer (Figura 2.14) las temperaturas a las cuales trabajan los diferentes tipos de sensores para tener una idea más amplia de su elección.

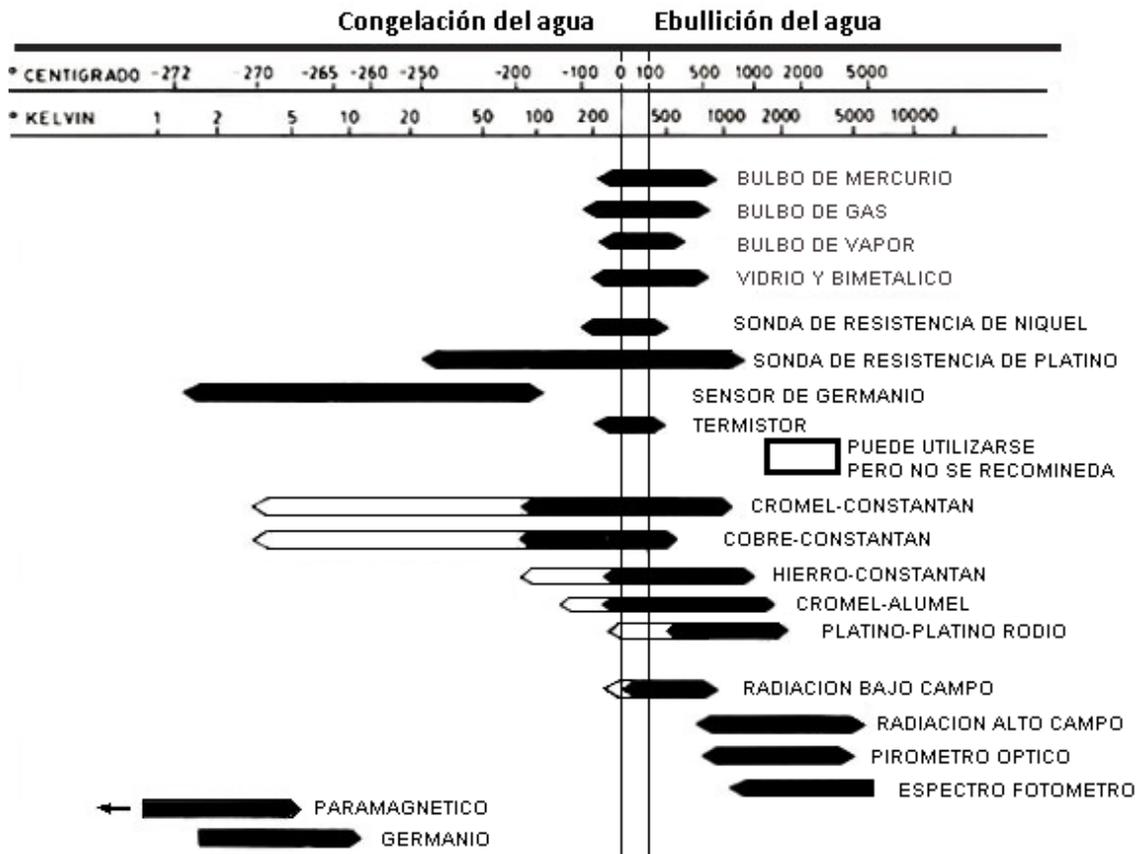


Figura 2.14 Sensores de temperatura. (LIMPI)

Consideramos el rango en que se manejará la medición, para escoger el sensor que cubra a este, y se verá la necesidad de manejar otros instrumentos alternos si se requieren. Luego de realizar la selección del sensor, se debe tener presente si el mismo cubre la exactitud que se necesita al realizar la medición, dependiendo de la aplicación que se le vaya a dar. En la industria, se tienen los de clase AA²⁰ que son de mejor exactitud, clase A y clase B, comercializados también con fracciones de tolerancia, como por ejemplo 1/3A.

Se debe de considerar también las condiciones en las que se va a realizar la medición, saber si el sensor se va a encontrar en superficies de contacto o sin contacto, medición en inmersión, medir en gas, líquido o sólido.

Estos parámetros son las más importantes que debemos tomar en cuenta para la correcta elección de un sensor en el momento de la construcción de un sistema y que cumpla con los requerimientos del mismo.

²⁰ Clase AA.- (1/3B) = $\pm(0.1+0.0017*t)^{\circ}\text{C}$ o $100.00 \pm 0.04\Omega$ a 0°C .

A continuación se expone (Tabla 2.1) un cuadro comparativo de ventajas y desventajas de algunos sensores.

Tabla 2.1 Ventajas y desventajas de los sensores de temperatura.

	TERMOPAR	RTD	TERMISTOR	SENSOR IC
VENTAJAS	-270°C a 1800°C	-260°C a 850°C	-80°C a 150°C	-55°C a 150°C
	Simple	Muy estable	Tiempo de	Buena linealidad
	Rudo (aplicaciones de alta vibración y golpes)	Amplio alcance de operación	respuesta rápida	Barato
	Económico	Buena exactitud	Medición de 2 hilos	
	Amplia variedad de formas físicas	Mejor linealidad que el termopar	Cambios grandes de resistencia vs. Temperatura	
	Amplio alcance de temperaturas	Excelente intercambiabilidad	Alta resistencia	
	Buena intercambiabilidad		Pequeños Baratos Buena estabilidad	
DESVENTAJAS	No lineal	Caro	No lineal	Temperatura <250°C
	Baja tensión	Sensible a vibraciones y golpes	Requiere fuente de corriente	Requiere alimentación
	Requiere compensación por unión de referencia	Tiempo de respuesta relativamente lento	Alcance de temperatura limitado	Tiempo de respuesta lento
	Baja sensibilidad	Relativamente lento	Frágil	Configuraciones limitadas
	Relativamente baja estabilidad	Requiere fuente de corriente		
		Cambios pequeños de resistencia		
		Medición de 4 hilos		

2.3.1.7 Sensores de Velocidad

Al igual que los anteriores, los sensores de velocidad nos permiten conocer la velocidad de desplazamiento de un objeto, teniendo que considerarse inicialmente que tipo de velocidad es la que se quiere conocer, en esta sección nos centraremos a la velocidad angular pues en la mayoría de los procesos industriales se requiere conocer las revoluciones de algún eje giratorio. En la Tabla 2.2 se muestra algunos tipos de velocidades y los instrumentos para medir las mismas.

Tabla 2.2 Instrumentos y unidades de medición de velocidad

TIPO DE VELOCIDAD	UNIDAD DE MEDICIÓN	TIPO DE INSTRUMENTO
Lineal	Km/h	Velocímetro
Angular	Radianes/segundos	Tacómetro
Frecuencia	Ciclos/segundos	Frecuencímetro

Para realizar las mediciones se tienen diferentes tipos de métodos tanto mecánicos, eléctricos y electrónicos. En la actualidad para realizar mediciones que sean superiores a las 100 RPM se suelen utilizar instrumentos electrónicos los que pueden ser:

- Tacómetros con y sin contacto.
- Encoders.

2.3.1.7.1 Tacómetros con y sin contacto

Son instrumentos que nos permiten tomar una medición con respecto a un eje en movimiento. Los tacómetros con contacto tienen el sensor directamente en el eje para que el instrumento nos indique la velocidad de giro del sistema como se muestra en la Figura 2.15.



Figura 2.15 Tacómetro con contacto. (EXTECH INSTRUMENTS)

A diferencia de los anteriores, los tacómetros sin contacto (Figura 2.16) como indica su nombre no necesitan unirse al eje, estos emiten una señal a través de un LED que luego será identificada con una foto celda, teniendo que ser esta señal direccionada al eje en movimiento, a una distancia entre los 50 a 2000 mm (dependiendo de las características del equipo que se esté utilizando), este eje tendrá una marca que permite un reflejo del LED a la foto celda para determinar de esta manera la cantidad de vueltas.

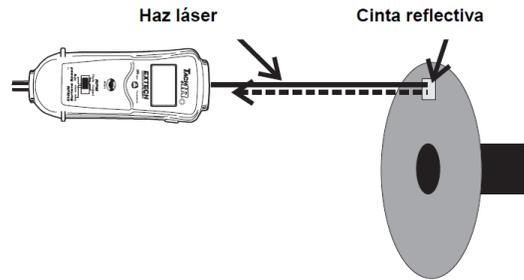


Figura 2.16 Tacómetro sin contacto. (EXTECH INSTRUMENTS)

2.3.1.7.2 Encoders

Los Encoders son sensores que generan señales digitales en respuesta al movimiento. Pueden ser para velocidad lineal o angular dependiendo en el sistema que se utilice, también permiten medir a más de la velocidad la posición. Existen dos tipos de tecnologías: la óptica y la magnética.

2.3.1.7.2.1 Encoders Ópticos

Estos sensores utilizan un vidrio que tiene grabado un patrón de líneas en el mismo, o un disco metálico o de plástico con ranuras para ser detectados por los sensores ópticos, existen dos tipos de sensores ópticos los incrementales y absolutos. Disponen de dos tipos de salidas: los incrementales (para medir velocidad) y absolutos (para medir posición).

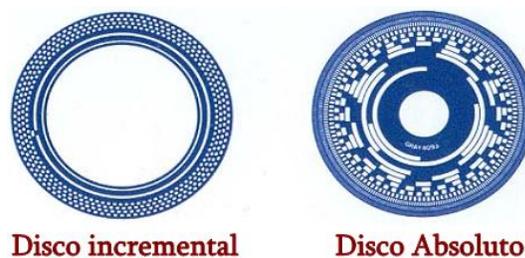


Figura 2.17 Tipos de encoders ópticos. (West instruments de México)

2.3.1.7.2.1.1 Encoders incrementales

Permiten que un sensor óptico detecte el número de segmentos que tiene el disco, y la posición cero del mismo la da una señal llamada índice, marcador o canal Z. En la Figura 2.18 se aprecia el funcionamiento de un encoder para aplicaciones donde no es importante el sentido de la dirección (unidireccional).

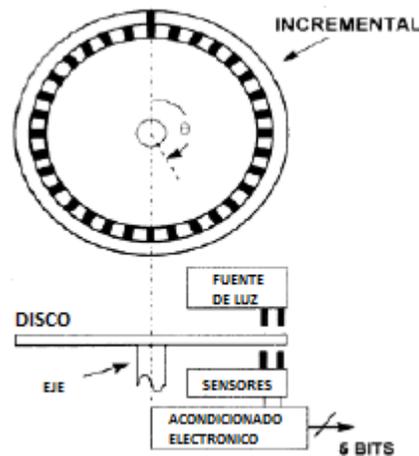


Figura 2.18 Vista lateral y superior del disco del encoder incremental. (Pérez, 2010)

La salida en cuadratura se usa para saber el sentido de la dirección (bidireccional). Este tiene dos canales desfasados 90 grados eléctricos uno del otro. Esto es importante en sistemas de control donde la posición se puede revertir y en donde ciertas vibraciones de la máquina, por ejemplo, podrían ocasionar lecturas erróneas (al usar encoders unidireccionales).

Para duplicar la resolución (x2) se rastrean los flancos de subida y bajada de un canal y para cuadruplicarla (x4) se cuentan los flancos de los dos canales.

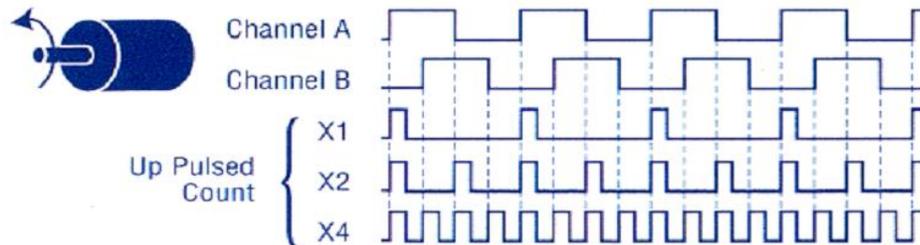


Figura 2.19 Encoder incremental en cuadratura. (West instruments de México)

2.3.1.7.2.1.2 Encoders absolutos

Estos nos permiten conocer la posición exacta de su ubicación en cualquier instante, ya que tienen una codificación Manchester en su disco.

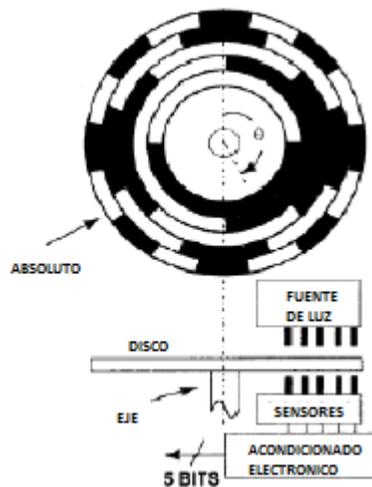


Figura 2.20 Disco del encoder absoluto. (Pérez, 2010)

Funcionan por la emisión y recepción de luz, usualmente el emisor es un diodo LED y el receptor un fotodiodo que detecta la reflexión de la luz sobre alguna superficie que se interponga en su camino.

2.3.1.7.2.2 Encoders Magnéticos

Son sensores muy resistentes al polvo, grasa, humedad, etc., que funcionan por medio de reluctancia²¹ variable, detectando campos magnéticos causados por el movimiento de un objeto ferromagnético²² cuando un diente de engrane se mueve a través del sensor rotatorio de reluctancia (magneto recolector), este envía una señal al sistema que indicará las rpm que da el eje.

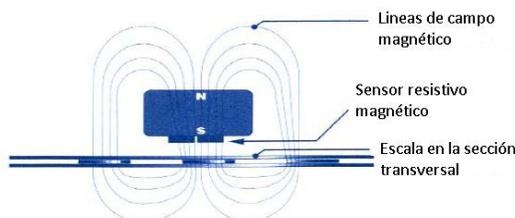


Figura 2.21 Encoder magnético. (West instruments de México)

2.3.2 Actuadores

Los actuadores son elementos que nos permiten alterar el resultado de un proceso, con la finalidad de obtener un resultado requerido, siendo capaces de generar una fuerza a partir de líquidos, gas o energía eléctrica, clasificándose de esta manera en:

²¹ **Reluctancia.**- Resistencia que ofrece un circuito al flujo magnético.

²² **Ferromagnético.**- Se aplica al mineral, como el hierro o el níquel, que tiene un gran poder magnético.

- Actuadores eléctricos.- Utilizados en aparatos electrónicos y mecánicos.
- Actuadores hidráulicos.- Utilizados donde el trabajo requiera de potencia.
- Actuadores neumáticos.- Utilizados en la industria para posicionamiento.

Tabla 2.3 Tipos de actuadores.

	Actuador Neumático	Actuador Eléctrico	Actuador Hidráulico
Fuerza Generadora de Movimiento	Presión de aire	Energía eléctrica	Presión hidráulica
Elemento Motriz	Embolo, Pistón o Veleta	Motor Eléctrico	Embolo, Pistón o Veleta
Transmisión de Fuerza o Torque	Eje o Cremallera	Reductor	Eje
Conversión mecánica	Yugo o Piñón	No hay	Yugo o Piñón

La elección de los actuadores dependerá del proceso en que se vaya a utilizar y del medio en que van a operar, a continuación se citara la clasificación de actuadores para tener una mayor referencia, considerándose que solo se dará una explicación trivial de estos debido a que no es el punto de análisis al cual nos centramos.

2.3.2.1 Actuadores eléctricos.

Los actuadores eléctricos son aquellos en los cuales intervienen una señal eléctrica para su funcionamiento y manipulación, habitualmente estos suelen ser motores o servo motores que pueden ser actuadores directos o que se encuentran dentro de algún sistema para su operación.

Debemos considerar los espacios y el lugar en donde va a trabajar el actuador para según esto ver las características del mismo, pudiendo ser: el tamaño, el torque que necesita el sistema para su funcionamiento, la presión de la línea a la que va a estar sometido el actuador, establecer un porcentaje de sobredimensionamiento que podría oscilar entre el 10% hasta el 50%, se debe tener presente la carrera angular o el giro que va a tener (90°,180°, multivuelta.), si es un sistema multivuelta se considera el número de vueltas para completar su función, se debe también considerar el material con el que se encuentra construido y el medio en el que se va a trabajar (corrosivos, no corrosivos). Tomando estas consideraciones se debería de hacer una correcta elección del actuador para el sistema requerido.

2.3.2.2 Actuadores hidráulicos.

Los actuadores necesitan de una señal para su funcionamiento, esta señal sería un fluido a presión (regularmente aceite), que es inyectada por el extremo del actuador el cual genera una fuerza de expansión en el recipiente, generando un movimiento del actuador.

El torque que es generado por el actuador va a ser directamente proporcional con la presión hidráulica que ingresa al sistema, esta puede variar según la forma que tenga el actuador y a la posición en la que se encuentra. Estos suelen emplearse en sistemas de gran tamaño, que requieran velocidad y tengan una resistividad elevada.

2.3.2.3 Actuadores neumáticos.

El funcionamiento del actuador neumático se realiza mediante aire comprimido, generando de esta manera una expansión entre el espacio que tiene el émbolo²³ y la pared, produciendo de esta manera un movimiento del cuerpo, el torque que se genera va a ser directamente proporcional al torque que genera el aire comprimido, no obstante este tendrá dependencia del diseño del actuador y de su posición actual. Las consideraciones para la elección del actuador son similares a la de los actuadores eléctricos, pero a más de las indicadas, existe una consideración muy importante que se debe hacer en el momento de la elección, siendo esta la presión del aire que va a soportar el actuador, en este caso se verificará que la presión que será ejercida no sea mayor a la que soporta el actuador, caso contrario se debería cambiar de actuador o a su vez colocar un control de presión a la entrada.

2.4 Estructuras y configuraciones de controladores.

2.4.1 Sistemas de control en lazo abierto.

Benjamin C. Kuo divide en dos partes a los sistemas en lazo abierto: el controlador y el proceso controlado. La señal de referencia r llamada también entrada, comando o set point es la que se aplica al controlador. Este tiene una señal actuante u con la que controla el proceso, tal que la variable controlada y se comporte de acuerdo a normas ya establecidas (Kuo, Sistemas de control automático, 1996, pág. 9)

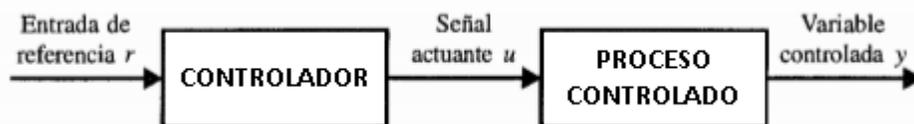


Figura 2.22 Elementos de un sistema de control en lazo abierto. (Kuo, Sistemas de control automático, 1996)

Ventajas:

²³ **Émbolo.**- Disco que se ajusta y mueve alternativamente en el interior de una bomba para comprimir un fluido o para recibir de él movimiento.

- Diseño simple.
- Número de componentes relativamente bajo.
- Económico.
- No tiene problemas de estabilidad.

Desventajas:

- La planta no es conocida exactamente por lo que no se puede lograr el inverso perfecto.
- No se puede llegar a controlar plantas inestables.
- No compensa perturbaciones externas.
- No logra satisfacer requerimientos de desempeño crítico.

El diagrama de bloques de un sistema de control en lazo abierto es:

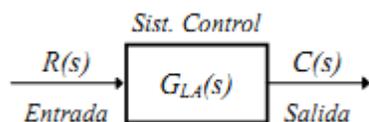


Figura 2.23 Diagrama de bloques equivalente de un sistema de control en lazo abierto. (Edicions UPC, 1999)

La función de transferencia directa del sistema de control en lazo abierto es:

$$G_{LA}(s) = \frac{C(s)}{R(s)} \quad \text{Ecuación 2.5}$$

Ya que un sistema se encuentra formado por el control, la planta, etc. entonces:

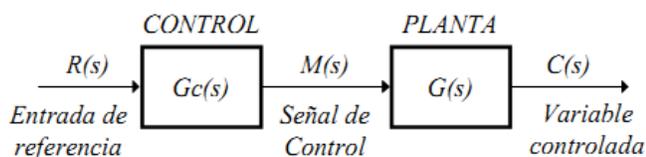


Figura 2.24 Diagrama de bloques de un sistema de control en lazo abierto. (Edicions UPC, 1999)

Obtenemos su función de transferencia:

$$G_{LA}(s) = \frac{C(s)}{R(s)} = \frac{M(s)}{R(s)} \cdot \frac{C(s)}{M(s)} = G_C(s) \cdot G(s) \quad \text{Ecuación 2.6}$$

2.4.2 Sistemas de control en lazo cerrado.

Con un sistema en lazo cerrado se obtiene un control más exacto sobre la variable controlada y, ya que siempre se toma en cuenta el error que existe entre la entrada r y la salida del proceso que se está controlando.

Farid Golnaraghi y Benjamin C. Kuo explican que en un sistema de control de velocidad en lazo cerrado la variable controlada ω_r tiene que mantenerse en el nivel preestablecido de la señal de referencia r que en este caso es ω . Si se aplica un par a la carga T_L , este corrige automáticamente su error gracias a su detección por medio de la diferencia entre el set point y la señal del transductor de velocidad (Automatic control systems, 2010, pág. 7).

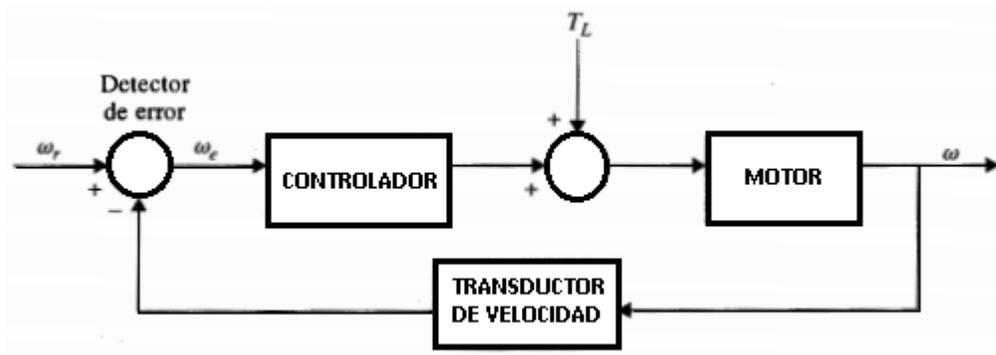


Figura 2.25 Elementos de un sistema de control en lazo cerrado de velocidad en ralentí. (Kuo, Sistemas de control automático, 1996)

Ventajas:

- Se puede controlar plantas inestables.
- Compensa perturbaciones externas.
- Se puede llegar a controlar los sistemas aunque existan errores en el modelado.

Desventajas:

- Requiere un alto número de componentes.
- Costosa.
- Se introducen ruidos al medir las variables del sistema.
- Puede hacerse inestable.

Es viable analizar la función de transferencia de los sistemas en realimentación negativa no unitaria ya que son los más usados, a más que si $H(s)$ es uno, entonces el sistema tendría realimentación negativa unitaria.

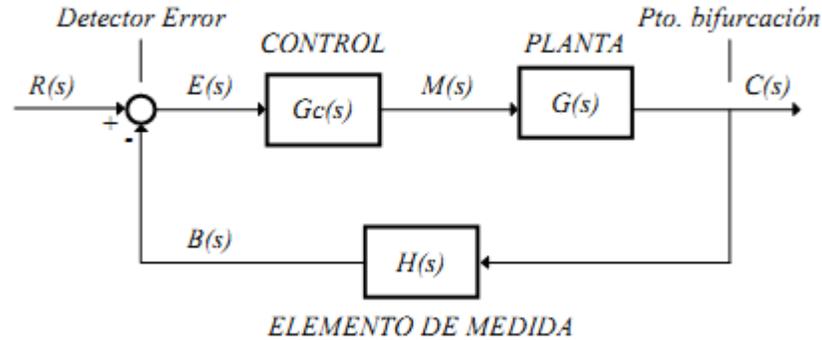


Figura 2.26 Diagrama de bloques de un sistema de control en lazo cerrado con realimentación negativa no unitaria. (Edicions UPC, 1999)

El cual tiene un sistema de ecuaciones en el dominio de Laplace:

$$\begin{cases} C(s) = G_C(s) \cdot G(s) \cdot E(s) \\ E(s) = R(s) - B(s) \\ B(s) = H(s) \cdot C(s) \end{cases} \quad \text{Ecuación 2.7}$$

Entonces:

$$C(s) = \frac{G_C(s) \cdot G(s)}{1 + G_C(s) \cdot G(s) \cdot H(s)} \cdot R(s) \quad \text{Ecuación 2.8}$$

Donde $C(s)$ es la transformada de Laplace de la señal de salida, $R(s)$ es la de la entrada, $E(s)$ la del error y $B(s)$ la de realimentación.

La función de transferencia directa relaciona la señal de error y la salida:

$$G_D(s) = \frac{C(s)}{E(s)} = G_C(s) \cdot G(s) \quad \text{Ecuación 2.9}$$

La función de transferencia en lazo abierto relaciona la señal de error y la realimentación, esta es el producto de todas las funciones de transferencia del lazo de control.

$$G_{LA}(s) = \frac{B(s)}{E(s)} = G_C(s) \cdot G(s) \cdot H(s) \quad \text{Ecuación 2.10}$$

La función de transferencia en lazo cerrado relaciona la señal de entrada y la salida.

$$G_{LC}(s) = \frac{C(s)}{R(s)} = \frac{G_C(s) \cdot G(s)}{1 + G_C(s) \cdot G(s) \cdot H(s)} = \frac{G_d(s)}{1 + G_{LA}(s)} \quad \text{Ecuación 2.11}$$

Cabe resaltar que si la función de transferencia directa es alta ($G_C(s) \cdot G(s) \gg 1$) y que tenga realimentación unitaria ($H(s) = 1$), entonces el sistema posee una robustez muy significativa frente a perturbaciones externas y variaciones de parámetros internos.

2.4.3 Configuraciones del controlador.

En el diseño de controladores nos basamos en una configuración fija, primero elegimos la configuración básica del controlador y en donde estará ubicado, lo que encontramos son las características de compensación del controlador que cumplan los requerimientos de diseño.

Un sistema que tenga un grado de libertad tiene un solo controlador, por tanto un sistema de dos grados de compensación tendrá dos controladores.

2.4.3.1 Compensación en cascada (serie)

El controlador está localizado en serie con el proceso a ser controlado.

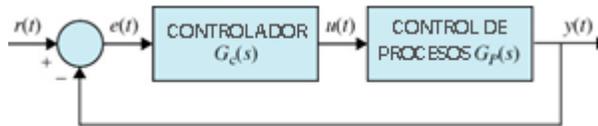


Figura 2.27 Sistema de control en cascada. (Kuo, Sistemas de control automático, 1996)

2.4.3.2 Compensación en realimentación (paralelo).

El controlador es localizado en el lazo menor de retroalimentación.

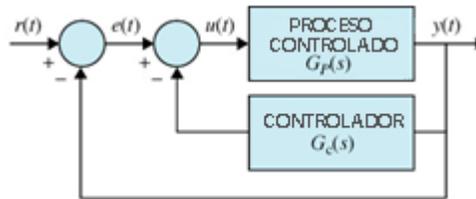


Figura 2.28 Sistema de control en realimentación (paralelo). (Kuo, Sistemas de control automático, 1996)

2.4.3.3 Compensación en serie-realimentada.

Un controlador está ubicado en serie con el proceso a ser controlador y el segundo controlador en paralelo con el mismo proceso.

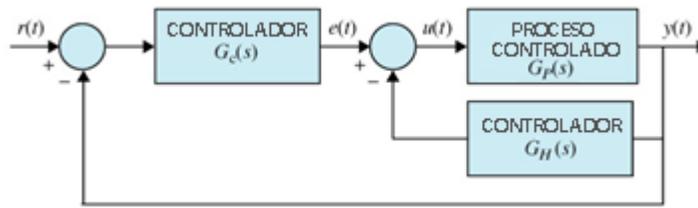


Figura 2.29 Sistema de control en serie y paralelo (dos grados de libertad). (Kuo, Sistemas de control automático, 1996)

2.4.3.4 Compensación directa con compensación serie.

Los dos controladores se encuentran en trayectoria directa en una configuración estilo serie - serie uno con respecto al otro

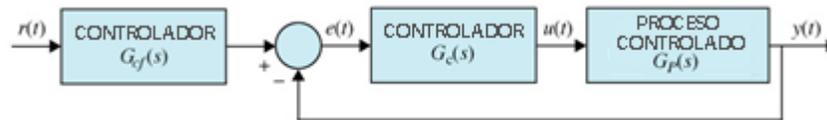


Figura 2.30 Compensación directa con compensación serie (dos grados de Libertad). (Kuo, Sistemas de control automático, 1996)

2.4.3.5 Compensación en pre-alimentación.

Los compensadores están en paralelo y localizados en serie al proceso, exclusivamente en la trayectoria directa del sistema.

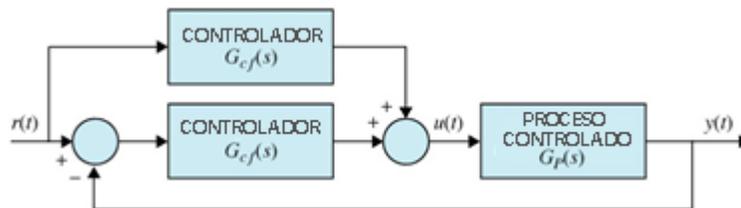


Figura 2.31 Compensación en pre alimentación (dos grados de Libertad). (Kuo, Sistemas de control automático, 1996)

La compensación directa con compensación serie y la compensación en pre alimentación tienen la característica de que el controlador $G_{cf}(s)$ no se encuentra en el lazo del sistema por lo que se pueden cancelar ceros y polos de la función de transferencia en lazo cerrado.

2.5 Metodologías de modelación de controladores.

El diseño de un sistema de control comienza por un correcto modelo de la planta, es decir, del proceso físico que se va a controlar. Para ello se necesita conocer la relación que existe entre la entrada y la salida del sistema mediante su función de transferencia,

para obtenerla podemos formularla de dos maneras: la primera es deducir su modelo matemático a partir de ecuaciones diferenciales que describan sus procesos físicos y la otra es tomar al sistema como una caja negra a la que la vamos a estudiar generando señales a su entrada y viendo que comportamiento tienen en su salida. Este comportamiento podemos modelarlo ya sea en el plano s para sistemas de control analógico o en el plano z para los sistemas de control digital. Obtenida la función de transferencia del sistema procedemos a diseñar un controlador adecuado que cumpla satisfactoriamente los requerimientos de diseño necesarios.

Para diseñar el controlador usamos técnicas convencionales de diseño como el lugar geométrico de las raíces, el método de respuesta en frecuencia, etc. Pudiéndolos usar en controladores digitales basados en modelos continuos de controladores P, PI, PID, en adelanto, en retraso, etc. y aplicándoles técnicas de discretización mediante la transformada z o en base a sus ecuaciones en diferencias. Otra forma es diseñar los controladores directamente en el dominio del tiempo discreto, para ello el proceso a controlar debe ser discretizado.

En la práctica es conveniente usar procedimientos sistematizados de sintonización de controladores, los primeros investigadores fueron Zieger y Nichols (1942), a partir de ellos se desarrollaron muchos métodos y aun es un tema que se encuentra en investigación. Para usarlos es necesario conocer la dinámica del sistema mediante su modelo y los parámetros del controlador utilizado.

En resumen para implementar un lazo de control debemos:

- Modelar el proceso a controlar.
- Establecer los criterios de diseño (tiempo de subida, tiempo de establecimiento, máximo sobresalto, error en estado estacionario e_{ss}).
- Seleccionar el método de sintonía del controlador.
- Calcular los parámetros del controlador.
- Simular el comportamiento del sistema.
- Ajustar el controlador.
- Comprobar el comportamiento del controlador en el proceso real.

2.5.1 Controlador PID

Aunque las nuevas, efectivas, y metodológicas teorías de diseño en el campo del control automático continúan desarrollándose, el control PID sigue siendo el controlador más ampliamente usado por la industria gracias a su gran tasa costo/beneficio, su uso relativamente sencillo y su gran capacidad de proveer una performance satisfactoria a muchas tareas de control.

2.5.1.1 Control de tres términos

Este tipo de controlador tiene las propiedades de tres tipos de acciones, podemos representarlas de tres formas diferentes. Primero, su representación simbólica está en la figura 2.32(a). Segundo, en la figura 2.32(b) se muestra los operadores en el dominio del tiempo y la figura 2.32(c) da la transformada de Laplace del controlador en el dominio de s.

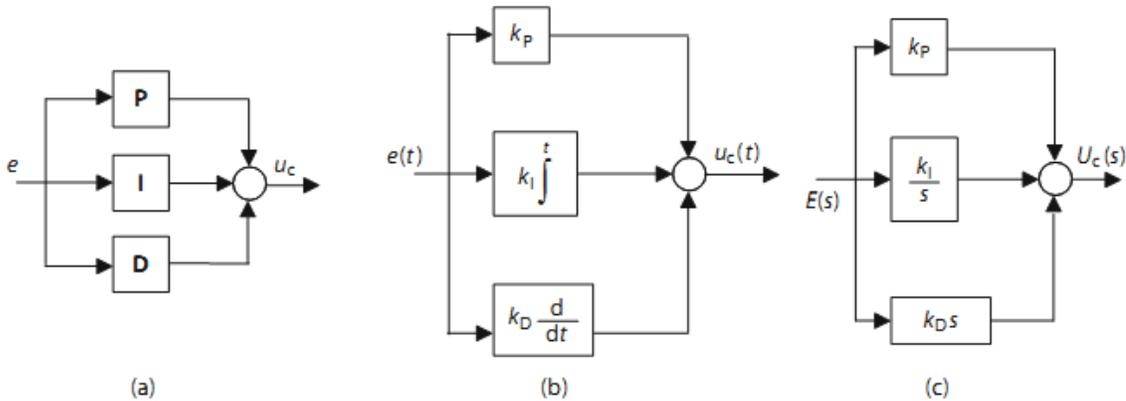


Figura 2.32 Representaciones de un controlador PID. (Johnson & Moradi, 2005)

2.5.1.1.1 Proporcional (P)

Ya que su salida es el error multiplicado por una ganancia k_p , <<el aumento de la ganancia del control proporcional permite reducir el error en estado estacionario>> (Edicions UPC, 1999, p. 130), aunque no logra eliminarlo. Modificando este parámetro podemos reducir el tiempo de establecimiento del sistema, pero hay que tener cuidado debido a que si el parámetro es muy bajo el sistema puede responder incorrectamente, y si es muy alto puede llegar a causar inestabilidad.

Dominio del tiempo $u_c(t) = k_p e(t)$ Ecuación 2.12

Dominio del Laplace $U_c(s) = k_p E(s)$ Ecuación 2.13

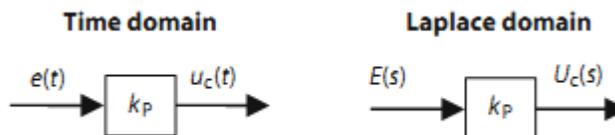


Figura 2.33 Diagramas de bloques del término proporcional. (Johnson & Moradi, 2005)

2.5.1.1.2 Integral (I)

Su salida es la integral del error, por lo tanto esta puede tener <<una señal de control diferente de cero aunque el error sea cero... El control integral permite obtener

un error estacionario nulo en un sistema de control>> (Edicions UPC, 1999, p. 135) ya que se introduce un polo en $s=0$ (a esta acción se la conoce como reseteo automático). Este valor en la salida del controlador depende de los valores pasados del error. Al ajustar esta ganancia podremos remover el offset de la referencia, pero si es muy alta puede causar oscilaciones, y si es muy baja la respuesta será demasiado lenta.

Dominio del tiempo $u_c(t) = k_I \int^t e(\tau) d\tau$ Ecuación 2.14

Dominio de Laplace $U_c(s) = \left[\frac{k_I}{s}\right] E(s)$ Ecuación 2.15

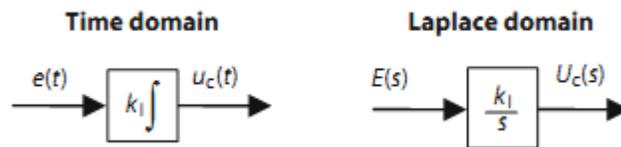


Figura 2.34 Diagramas de bloques del término integral. (Johnson & Moradi, 2005)

Al juntar las acciones proporcional e integral en el llamado control PI se logra resolver la respuesta oscilatoria de sistemas y el error de estado estacionario en un control proporcional puro.

2.5.1.1.3 Derivativo (D)

Su salida es la derivada del error, <<tiene como finalidad “anticipar hacia dónde va el proceso”>> (García Jaimes, 2010, p. 198), por lo que está al tanto de la dinámica que maneja el sistema y aumenta su sensibilidad. No puede usarse solo ya que no logra <<responder a una señal de error constante... debido a su acción anticipativa>> (Edicions UPC, 1999, p. 140). Al modificar esta ganancia podremos reducir la naturaleza oscilatoria de la respuesta en lazo cerrado, pero si es muy baja su respuesta es lenta y si es muy alta puede causar oscilaciones.

Dominio del tiempo $u_c(t) = k_D \frac{de}{dt}$ Ecuación 2.16

Dominio de Laplace $U_c(s) = [k_D s] E(s)$ Ecuación 2.17

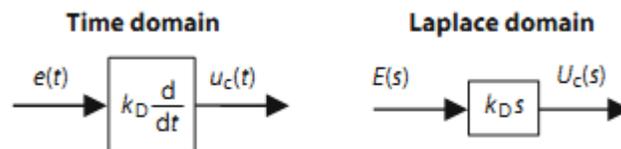


Figura 2.35 Diagramas de bloques del término derivativo. (Johnson & Moradi, 2005)

2.5.1.2 Control PID en base al computador

2.5.1.2.1 Aproximación discreta del control PID convencional

En base a la ecuación del control PID en el dominio de s se obtiene la discretización de la función de transferencia.

$$G_{PID}(s) = K_p + \frac{K_i}{s} + K_d \quad \text{Ecuación 2.18}$$

Para ello usamos el método TUSIN para la integral y el método rectangular hacia atrás para la derivada, que son los más utilizados:

$$G_{PID}(s) = K_p + \frac{K_i T_m (z + 1)}{2(z - 1)} + \frac{K_d (z - 1)}{T_m z} \quad \text{Ecuación 2.19}$$

$$G_{PID}(s) = \frac{\left(K_p + \frac{K_i T_m}{2} + \frac{K_d}{T_m}\right) z^2 + \left(-K_p + \frac{K_i T_m}{2} - \frac{2K_d}{T_m}\right) z + \frac{K_d}{T_m}}{z(z - 1)} \quad \text{Ecuación 2.20}$$

$$G_{PID}(s) = \frac{U(z)}{E(z)} = \frac{\left(K_p + \frac{K_i T_m}{2} + \frac{K_d}{T_m}\right) + \left(-K_p + \frac{K_i T_m}{2} - \frac{2K_d}{T_m}\right) z^{-1} + \frac{K_d}{T_m} z^{-2}}{(1 - z^{-1})} \quad \text{Ecuación 2.21}$$

Entonces la función de transferencia en forma de ecuación en diferencias para el control PID por computador es:

$$u[k] = u[k - 1] + \left(K_p + \frac{K_i T_m}{2} + \frac{K_d}{T_m}\right) e[k] + \left(-K_p + \frac{K_i T_m}{2} - \frac{2K_d}{T_m}\right) e[k - 1] + \frac{K_d}{T_m} e[k - 2] \quad \text{Ecuación 2.22}$$

CAPITULO III

3 DISEÑO DEL SISTEMA DE CONTROL

La planta del sistema es un reactor químico que está definido como: un dispositivo diseñado específicamente para que las reacciones químicas ocurran en condiciones totalmente controladas para obtener productos especificados (Caccavale, Iamarino, Pierri, & Tufano, 2011, pág. 10) <<La agitación del contenido es esencial, debido a que el flujo interior debe estar en constante circulación y así producir una mezcla uniforme>> (Mendiburu Díaz, 2005, pág. 23).

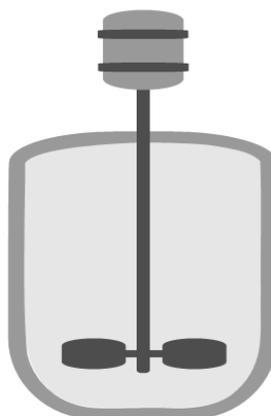


Figura 3.1 Reactor ideal tipo Batch (BR).

3.1 Modelación del sistema del reactor químico.

Al necesitar un ambiente controlado para los experimentos químicos, estos deben presentar algunos requerimientos físicos para dicha reacción y cumplir con ciertos requisitos de construcción que conlleven a un desempeño óptimo del sistema. Antes de continuar nos preguntamos: ¿Qué es una reacción química? ¿Qué hace un reactor químico? Ya que en el presente capítulo no abordaremos estos temas se hará una breve descripción para tener un conocimiento previo del tipo de sistema que se quiere controlar.

Según (Diaz) una **reacción química** es una operación unitaria que tiene por objeto distribuir de forma distinta los átomos de ciertas moléculas (compuestos reaccionantes o reactantes²⁴) para formar otras nuevas (productos).

²⁴**Reactante.**- Sustancia que participa en una reacción química produciendo otra u otras diferentes de las primitivas.

Un **reactor químico** es el lugar físico donde se producen las reacciones químicas. En su interior tiene un ambiente controlado y puede ser modificado según la conveniencia que el operador tenga para llevar a cabo el proceso requerido. Los reactores tienen diferentes modelos, formas, tamaños y se usan para diferentes tipos de reacciones (Levenzpiel, 2002).

El tipo de reactor que cumple los requerimientos de la reacción según el contexto y lo anteriormente planteado es de tipo discontinuo agitado en fase líquida.

3.1.1 Reactor discontinuo o tipo Batch de fase líquida.

Los reactores discontinuos agitados (Batch) son llamados así debido a que durante la operación, ninguna clase de material es agregado o retirado del mismo. Los reactivos son cargados previamente y el contenido del reactor es descargado al final (Mann, 2009, pág. 159). Considérese que las condiciones de temperatura y agitación en su interior deben ser las requeridas para que la reacción se lleve a cabo.

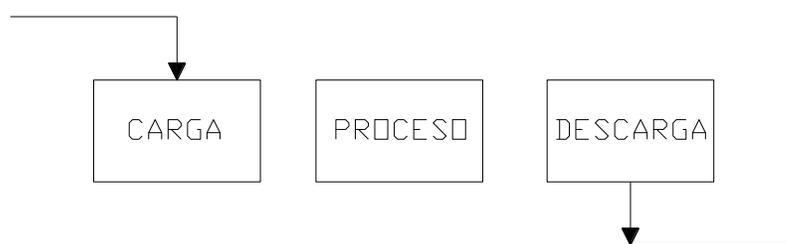


Figura 3.2 Proceso discontinuo

Para que el sistema tenga un desempeño óptimo, cuenta con ciertos requisitos de construcción:

Reactor: Es del tipo tanque agitado continuo. Tiene una tapa por la que entra y sale el material químico y puede ser sellada herméticamente para que el vapor que se genere no salga al exterior, asegurando que no exista ningún cambio en el volumen de la mezcla y protegiendo totalmente al operario de alguna fuga.

Agitador: Es del tipo magnético. Está dotado de un sistema de control que le permite mantener un movimiento uniforme y casi perfecto, asegurando así un excelente sellado del reactor (sin fugas mecánicas) y una mezcla limpia.

Camisa de Refrigeración: Es un sistema de enfriamiento basado en la circulación (intercambio de calor) de líquido refrigerante, cuenta con dos orificios, uno para la entrada del fluido y otro para la salida (en nuestro caso agua fría); un tanque de almacenamiento y una bomba de recirculación del líquido.

El reactor controla la temperatura (siendo este el objetivo principal), siempre manteniéndose dentro de un rango de operación especificado, esto se conseguirá variando la potencia de la niquelina al controlar el voltaje de alimentación. La camisa de refrigeración funcionará cuando el tiempo preestablecido de la reacción haya finalizado, haciendo que el material químico dentro del reactor recupere el estado líquido y que no se produzcan reacciones secundarias, ajenas al proceso mismo.

VARIABLES CENSADAS:

- Temperatura en el interior del reactor (por medio de un PT100).
- Velocidad del agitador (a través de un encoder digital).

VARIABLES MANIPULADAS:

- Potencia de la niquelina (por medio del voltaje de alimentación).
- Velocidad del agitador (a través del voltaje de alimentación del motor).

El modelo del reactor dependerá completamente del comportamiento de la niquelina, pues se tomara como despreciable la transferencia de calor que existe hacia el exterior considerándose que entre el calentamiento que tendrá la niquelina y el que se lleva a cabo en la reacción dependerá si es una reacción exotérmica²⁵ o endotérmica²⁶, este análisis se llevará a continuación para saber la temperatura que debe tener la niquelina para producir la reacción requerida.

Ya que se necesita conocer la energía que debe de transferir al sistema (en este caso la niquelina), se parte de la ecuación 3.1 (balance de la energía), para llegar a la temperatura necesaria que producirá la reacción.

$$\left[\begin{array}{c} \text{Acumulación} \\ \text{de} \\ \text{energía} \\ \text{dentro} \\ \text{del sistema} \end{array} \right] = \left[\begin{array}{c} \text{transferencia de} \\ \text{energía dentro de} \\ \text{la frontera} \\ \text{del sistema} \end{array} \right] - \left[\begin{array}{c} \text{transferencia de} \\ \text{energía fuera de} \\ \text{la frontera} \\ \text{del sistema} \end{array} \right] + \left[\begin{array}{c} \text{energía} \\ \text{generada} \\ \text{dentro} \\ \text{del sistema} \end{array} \right] - \left[\begin{array}{c} \text{energía} \\ \text{consumida} \\ \text{dentro} \\ \text{del sistema} \end{array} \right]$$

Ecuación 3.1

Para el diseño del equipo se considera las siguientes suposiciones que simplifican la ecuación planteada:

1. No existe transferencia de energía fuera del sistema
2. La energía consumida o generada dependerá de los compuestos químicos dentro del reactor.

²⁵ **Exotérmica.-** es cualquier reacción química que desprende calor.

²⁶ **Endotérmica.-** es cualquier reacción química que absorbe energía.

Quedando la ecuación:

$$\left[\begin{array}{l} \text{Acumulación de} \\ \text{energía dentro} \\ \text{del sistema} \end{array} \right] = \left[\begin{array}{l} \text{transferencia de} \\ \text{energía dentro de la} \\ \text{frontera del sistema} \end{array} \right] \pm \left[\begin{array}{l} \text{energía} \\ \text{del sistema} \end{array} \right] \quad \text{Ecuación 3.2}$$

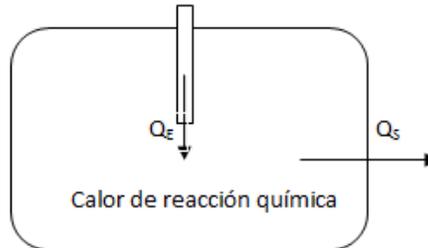


Figura 3.3 Diagrama del reactor químico discontinuo agitado

Aplicando estos parámetros en la ecuación del balance de energía:

$$\dot{Q}_E + \text{Calor liberado por reacción} - \dot{Q}_S = \frac{dE}{dt} \quad \text{Ecuación 3.3}$$

Donde tenemos:

\dot{Q}_E = El calor que aporta la niquelina.

\dot{Q}_S = El calor que sale del reactor (como se dijo será nulo).

Calor liberado por la reacción = Energía del sistema.

$\frac{dE}{dt}$ = Energía dentro del sistema con respecto al tiempo.

Ya que se trabaja en estado estable, el calor liberado por la reacción (energía del sistema) está dado por:

$$\text{Calor liberado por la reacción} = (-\Delta H_R)(-r_A)V \quad \text{Ecuación 3.4}$$

H_R = Entalpia de la reacción.

r_A = Velocidad de la reacción.

V = Volumen de la reacción.

Mientras que el calor de aporte \dot{Q}_E (niquelina) al sistema estará representado por:

- el coeficiente global de transferencia de calor entre la niquelina y el líquido.
- el área de transferencia de calor (área de la niquelina) y la variación de la temperatura entre la niquelina y el líquido.

Teniendo la ecuación:

$$\dot{Q}_E = U_{N-L} \cdot A \cdot (T_{niq} - T_{liq})$$

Ecuación 3.5

Dónde:

U_{N-L} = Coeficiente global de transferencia de calor.

A = Área de la niquelina.

T_{niq} = Temperatura de la niquelina.

T_{liq} = Temperatura del líquido.

La acumulación de energía con la variación del tiempo está dada por la cantidad de masa de los reactivos (M), el calor específico (Cp) de los mismos y la variación de temperatura con respecto al tiempo, con esto obtenemos:

$$\frac{dE}{dt} = M \cdot Cp \cdot \frac{dT}{dt}$$

Ecuación 3.6

Al sustituir en la ecuación 3.2 se obtiene:

$$(-\Delta H_R)(-r_A)V + U_{N-L} \cdot A \cdot (T_{niq} - T_{liq}) = M \cdot Cp \cdot \frac{dT}{dt}$$

Ecuación 3.7

Se considera que la reacción se llevara a cabo en el punto de estabilización de la temperatura, y el tiempo que se demora en llegar a este será el mínimo que nos permita las características del sistema.

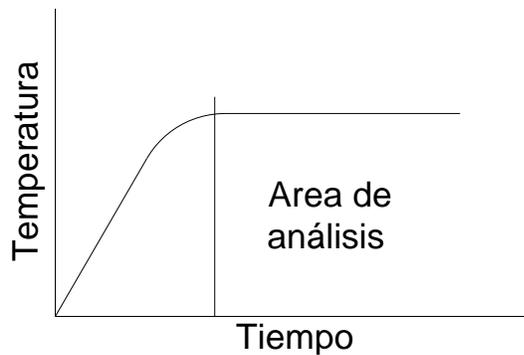


Figura 3.4 Temperatura de la reacción en el tiempo

Es así como se llega a:

$$(-\Delta H_R)(-r_A)V = U_{N-L} \cdot A \cdot (-T_{niq} + T_{liq})$$

Ecuación 3.8

Dada la ecuación 3.8, despejamos el parámetro que nos interesa (temperatura de la niquelina) debido que es el dispositivo a controlar:

$$T_{niq} = T_{liq} - \frac{(-\Delta H_R)(-r_A)V}{U_{N-L} \cdot A}$$

Ecuación 3.9

Cabe mencionar que los parámetros que se encuentran en la Ecuación 3.9 en algunos casos son dados por tablas y otros son proporcionados por un ingeniero químico, dependiendo de los reactantes y reactivos que vayan a ser colocados dentro del reactor químico.

3.1.2 Motor de corriente continua controlado por inducido y con imán permanente.

El motor de c.c. proporciona energía mecánica rotacional al agitador del reactor químico a partir de energía eléctrica en forma de corriente continua. La función de transferencia tendrá una aproximación lineal despreciando los efectos de segundo orden como la histéresis y la caída de tensión en las escobillas. El voltaje de entrada se aplicará al inducido del motor.

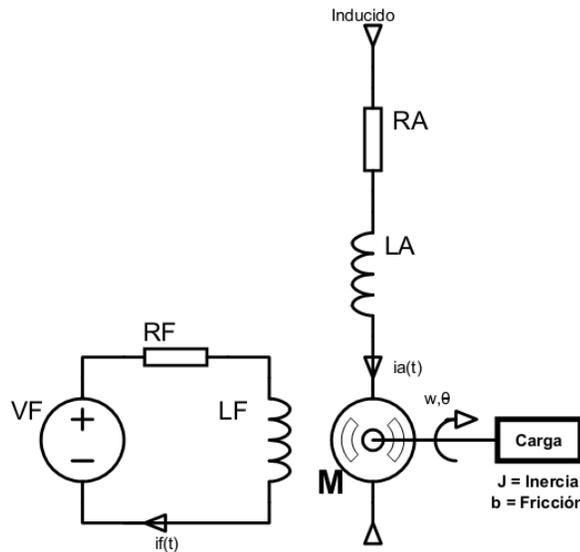


Figura 3.5 Representación del motor de corriente continua controlado por inducido y con excitación por imán permanente. (Dorf & Bishop, 2005)

En la demostración de Dorf y Bishop (Sistemas de control moderno, 2005, p. 58) la corriente de excitación i_f es proporcional al flujo ϕ (siempre que no haya saturación):

$$\phi = K_f i_f \tag{Ecuación 3.10}$$

El par del motor $T_m(t)$ está relacionado directamente con la corriente de inducido $i_a(t)$:

$$T_m(t) = K_l \phi i_a(t) = K_l K_f i_f(t) i_a(t) \tag{Ecuación 3.11}$$

La variable de control es la corriente de inducido $i_a(t)$. El campo en el estator i_f es constante (es establecido por un imán permanente):

$$T_m(t) = K_l \phi i_a(t) = (K_l K_f i_f) i_a(t) \quad \text{Ecuación 3.12}$$

$$T_m(t) = K_m i_a(t) \quad \text{Ecuación 3.13}$$

Con la transformada de Laplace tenemos:

$$T_m(s) = K_m I_a(s) \quad \text{Ecuación 3.14}$$

K_m = Es la permeabilidad del material magnético.

El par del motor $T_m(s)$ es igual al par proporcionado por la carga $T_L(s)$ más el par de perturbación $T_d(s)$:

$$T_m(s) = T_L(s) + T_d(s) \quad \text{Ecuación 3.15}$$

Dónde:

$$T_L(s) = Js^2\theta(s) + bs\theta(s) \quad \text{Ecuación 3.16}$$

La corriente del inducido I_a esta relacionada con el voltaje de entrada $V_a(s)$:

$$V_a(s) = (R_a + L_a s)I_a(s) + V_b(s) \quad \text{Ecuación 3.17}$$

Donde $V_b(s)$ es la fuerza contraelectromotriz (f.c.e.m.) que es proporcional a la velocidad angular del motor:

$$V_b(s) = K_b \omega(s) \quad \text{Ecuación 3.18}$$

Por lo tanto:

$$I_a(s) = \frac{V_a(s) - K_b \omega(s)}{(R_a + L_a s)} \quad \text{Ecuación 3.19}$$

Tenemos con $T_d(s) = 0$:

$$K_m I_a(s) = Js^2\theta(s) + bs\theta(s) \quad \text{Ecuación 3.20}$$

$$K_m \frac{V_a(s) - K_b \omega(s)}{(R_a + L_a s)} = Js^2\theta(s) + bs\theta(s) \quad \text{Ecuación 3.21}$$

$$\frac{V_a(s) - K_b \omega(s)}{\theta(s)} = \frac{(Js^2 + bs)(R_a + L_a s)}{K_m} \quad \text{Ecuación 3.22}$$

Siendo la velocidad angular es $\omega(s) = \theta(s)s$, despejando la posición angular $\theta(s) = \frac{\omega(s)}{s}$.

$$\frac{V_a(s) - K_b\omega(s)}{\frac{\omega(s)}{s}} = \frac{(Js^2 + bs)(R_a + L_as)}{K_m} \quad \text{Ecuación 3.23}$$

$$\frac{V_a(s)s}{\omega(s)} - \frac{K_b\omega(s)s}{\omega(s)} = \frac{(Js^2 + bs)(R_a + L_as)}{K_m} \quad \text{Ecuación 3.24}$$

$$\frac{V_a(s)s}{\omega(s)} - K_b s = \frac{(Js^2 + bs)(R_a + L_as)}{K_m} \quad \text{Ecuación 3.25}$$

$$\frac{V_a(s)s}{\omega(s)} = \frac{(Js^2 + bs)(R_a + L_as)}{K_m} + K_b s \quad \text{Ecuación 3.26}$$

$$\frac{V_a(s)s}{\omega(s)} = \frac{(Js^2 + bs)(R_a + L_as) + K_b K_m s}{K_m} \quad \text{Ecuación 3.27}$$

$$\frac{V_a(s)}{\omega(s)} = \frac{(Js + b)(R_a + L_as) + K_b K_m}{K_m} \quad \text{Ecuación 3.28}$$

En la condición de estado estacionario $K_b = K_m$ (Dorf & Bishop, 2005, pág. 61), obtenemos la función de transferencia del sistema:

$$G(s) = \frac{\omega(s)}{V_a(s)} = \frac{K_m}{(R_a + L_as)(Js + b) + K_m^2} \quad \text{Ecuación 3.29}$$

Para la mayoría de los motores la constante de tiempo del inducido $\tau_a = \frac{L_a}{R_a}$ es despreciable:

$$G(s) = \frac{\omega(s)}{V_a(s)} = \frac{K_m}{R_a(Js + b) + K_m^2} \quad \text{Ecuación 3.30}$$

$$G(s) = \frac{\omega(s)}{V_a(s)} = \frac{K_m}{R_a b + K_m^2} \frac{1}{(\tau_l s + 1)} \quad \text{Ecuación 3.31}$$

Donde $\tau_1 = \frac{R_a J}{R_a b + K_m^2}$ es la constante de tiempo equivalente.

3.2 Obtención de datos de información del sistema.

En la teoría de control, uno de los puntos más importantes es saber el comportamiento del sistema en su salida con respecto a una señal de ingreso, para este propósito existen diferentes tipos de señales de prueba que pueden ser aplicadas dependiendo de los datos que se necesite saber, las mismas que pueden ser:

- **Escalón.-** la característica de esta señal son sus cambios abruptos en la entrada al sistema, este comportamiento nos permite tener una idea del tiempo que necesita el sistema para que su señal se establezca, en otras palabras se puede saber en qué tiempo el sistema alcanza su estado estacionario.
- **Rampa.-** se pueden tener cambios lineales de la señal de entrada a una velocidad moderada.
- **Parábola.-** a diferencia de la señal de rampa implica cambios no lineales de la señal de entrada a velocidad moderada.

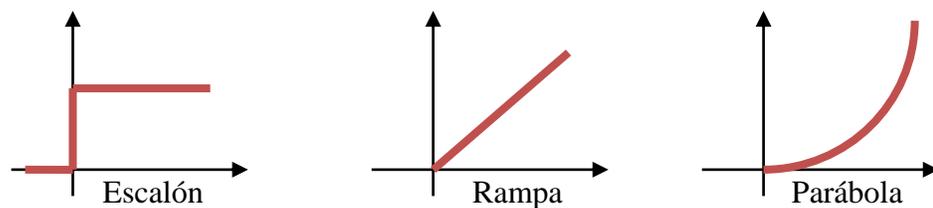


Figura 3.6 Tipos de señales de prueba

Al observar los diferentes tipos de señales de pruebas para los sistemas, se observa que para obtener la planta de nuestro sistema (la niquelina y el motor) la más idónea será la señal tipo escalón, por ello nos valemos de herramientas computacionales para este propósito, a más cabe acotar que la adquisición de datos será diferente en el caso de la niquelina como en el motor.

A continuación se comenzara con la explicación y obtención de datos de la planta de la niquelina, luego se procederá a explicar la obtención de la planta del motor.

3.2.1 Adquisición de datos de la niquelina.

Para la adquisición de datos de la niquelina, se cuenta con un sensor PT100 clase A de tres hilos, que será conectado a un controlador marca DELTA que nos permite tomar muestras cada 500ms a través de un puerto de comunicación 485 con protocolo MODBUS. Para la interface con el usuario se utilizó el programa LabVIEW 2011, en el cual se realizó una programación gráfica explicada a continuación:

Se comienza creando un **Project** en el programa, donde se colocara los **VI**²⁷ y/o servicios que necesite el programa. En la figura 3.7 se observa la creación de un nuevo **VI**:

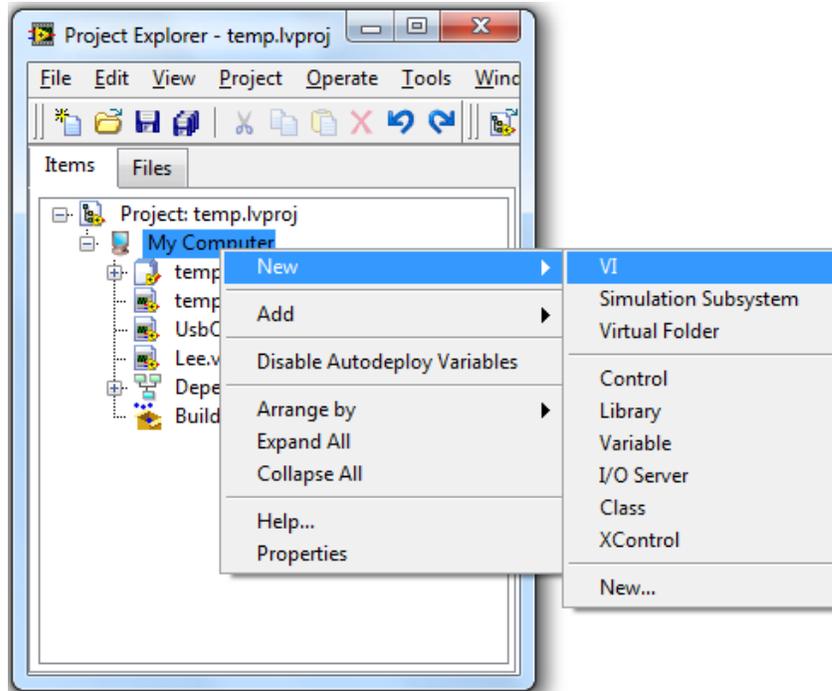


Figura 3.7 Creación del Proyecto en LabVIEW 2011

A continuación se crea los servicios de comunicación **I/O Service**, escogiendo la opción de MODBUS que será la requerida para la adquisición de datos (figura 3.8):

²⁷ **VI.**- Instrumentos Virtuales en programas desarrollados con LabVIEW.

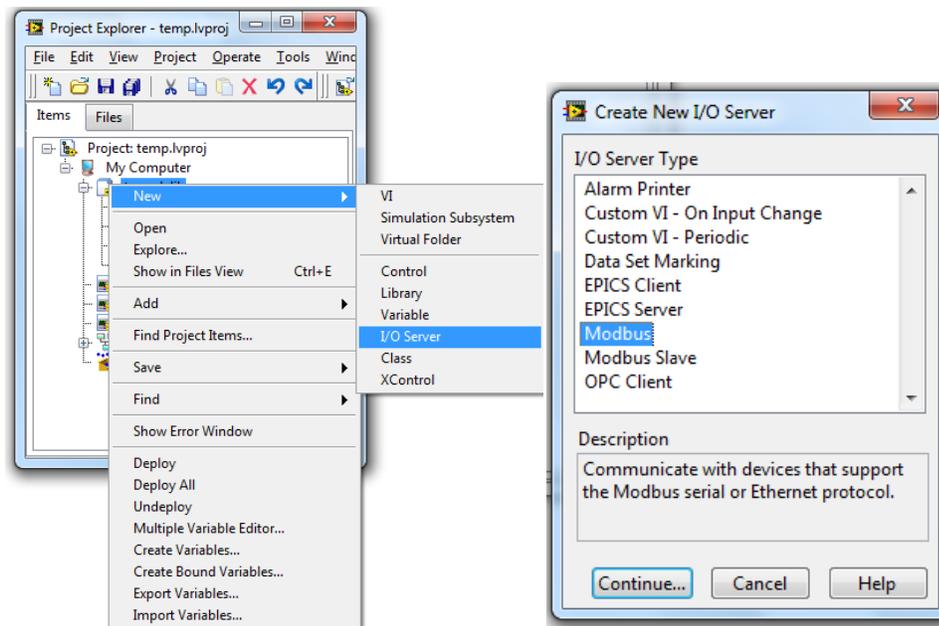


Figura 3.8 Creación y configuración del I/O Server

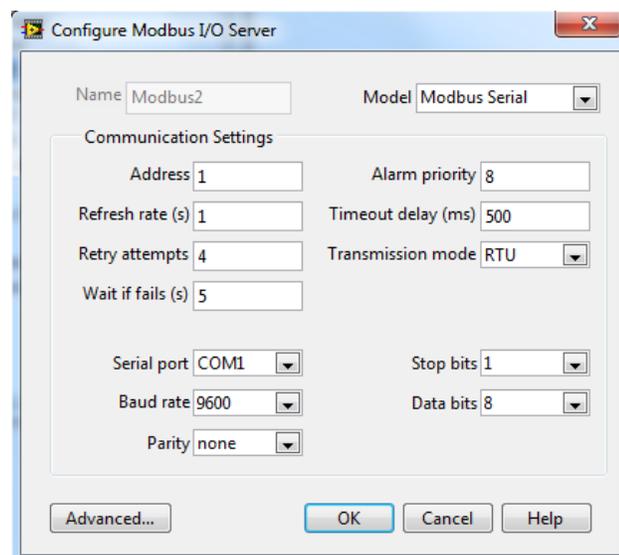


Figura 3.9 Configuración del I/O server Modbus.

Los parámetros de comunicación deben ser los mismos que se encuentran colocados en el controlador DELTA. Se escogen las variables que se necesitan (debido que cada una de ellas en el protocolo MODBUS tiene una dirección específica dada por el fabricante) por medio de **Bound Variables** con las direcciones 404097 y la 404098, que son las posiciones de memoria donde se encuentran los datos de temperatura medida y asignada respectivamente.

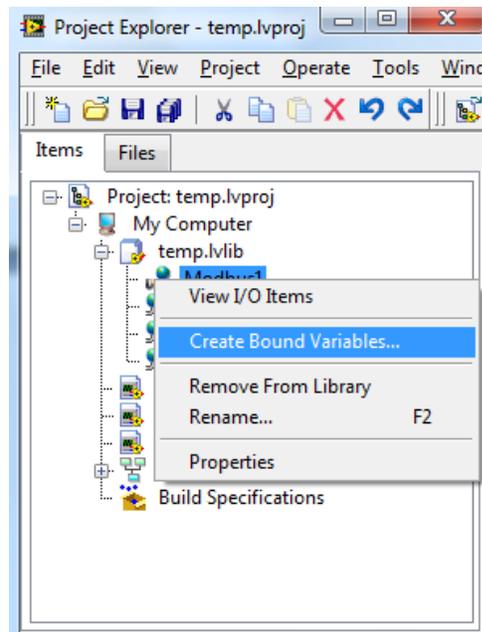


Figura 3.10 Creación de Bound Variables

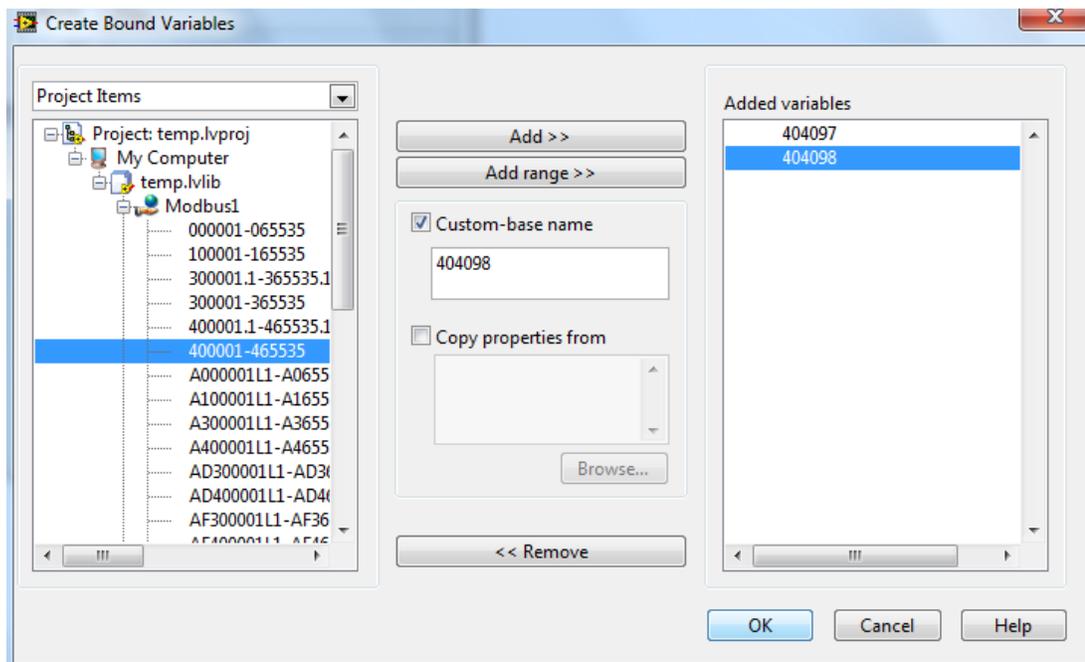


Figura 3.11 Selección de las variables a utilizar.

A estas variables se las utiliza para la programación dentro del VI. La interface gráfica tiene la opción de asignar el valor de la temperatura, el tiempo de adquisición, indicadores del valor de la temperatura adquirida y del tiempo total del proceso (minutos y segundos), el panel frontal se complementa con dos pantallas gráficas, la superior muestra como varía la temperatura en relación al tiempo en el momento de la

adquisición, mientras que la inferior muestra la respuesta total adquirida al finalizar el proceso.

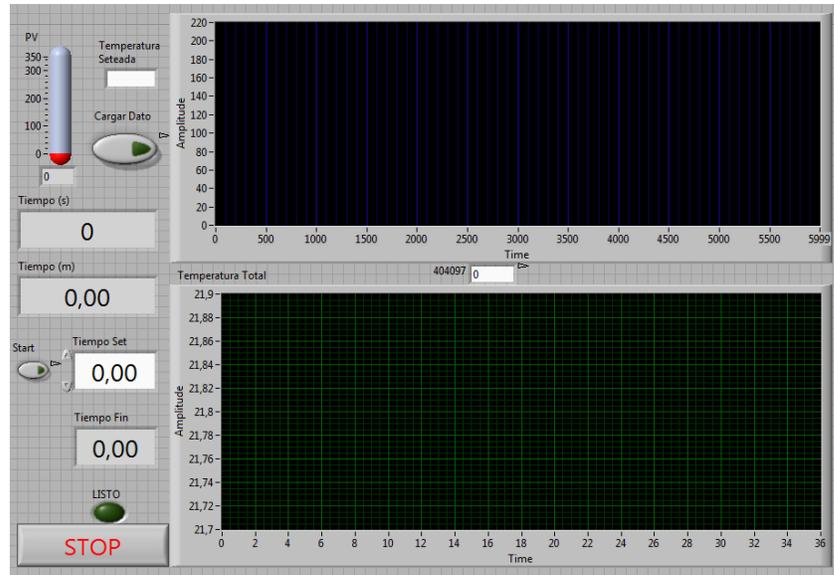


Figura 3.12 Pantalla de interfaz usuario.

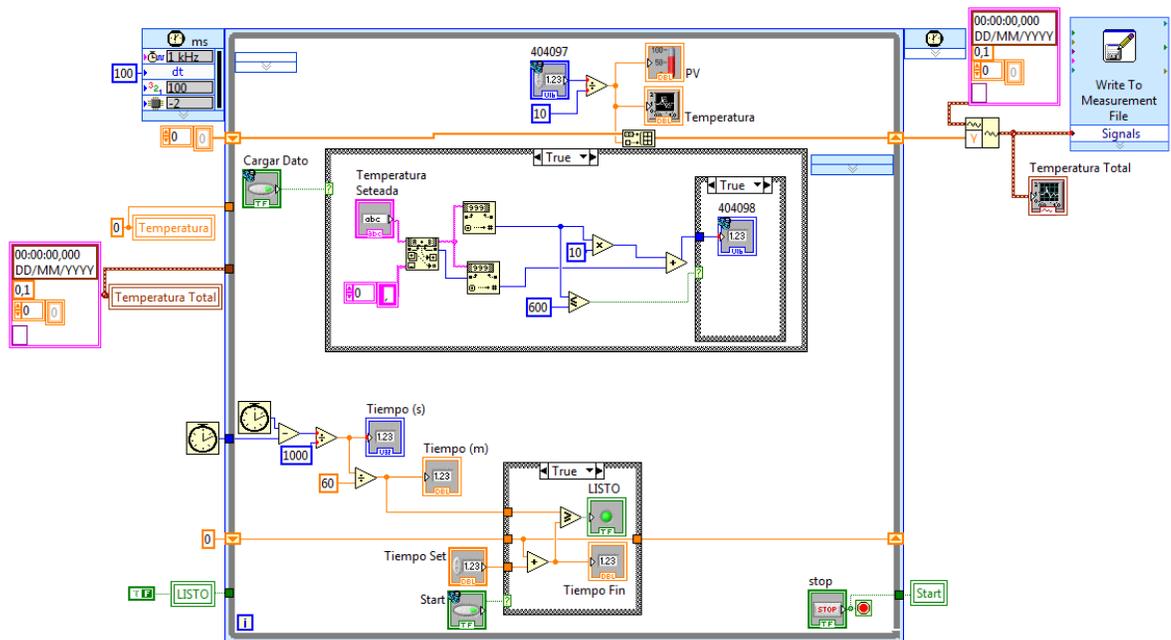


Figura 3.13 Diagrama de bloques en LabVIEW 2011

La niquelina tiene un voltaje de alimentación 40 Vac obtenido de un dimmer que provee un voltaje constante y estable durante todo el tiempo de adquisición y muestreo.

El circuito implementado para ese objetivo es el siguiente:

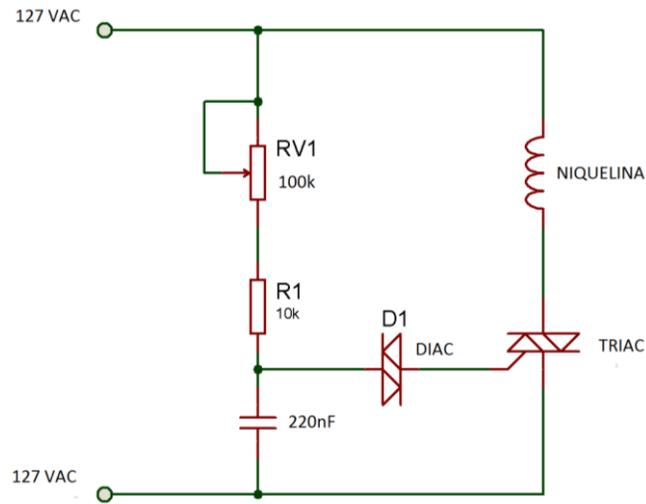


Figura 3.14 Dimmer calibración a 40 Vac para niquelina.

Durante el proceso de adquisición de datos, los valores de los componentes del circuito (figura 3.14) no se cambiaron para que la probabilidad de error disminuya, asegurando que el voltaje de alimentación de la niquelina se mantendrá en todas las muestras y que se pueda apreciar su comportamiento en diferentes instantes de tiempo.

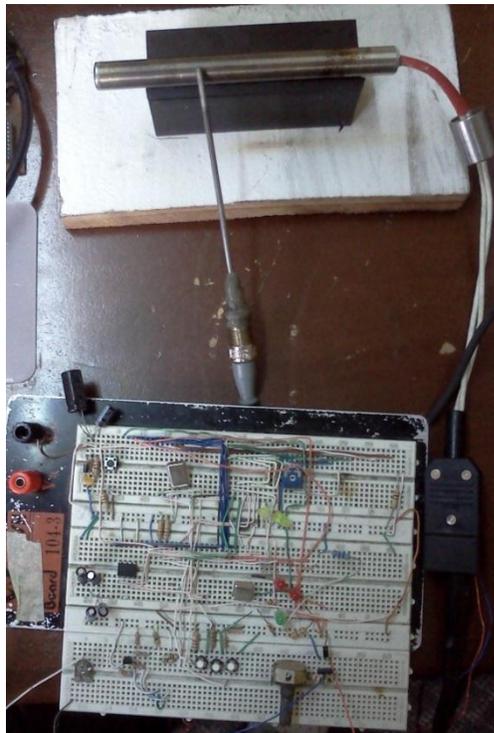


Figura 3.15 Adquisición de datos de la niquelina.

Pese a que las muestras adquiridas han sido varias, no se procedió a mostrar las gráficas obtenidas por el LABVIEW 2011, sino más bien se tomaron las tablas de datos y se mostraron los resultados gráficos generados en EXCEL.

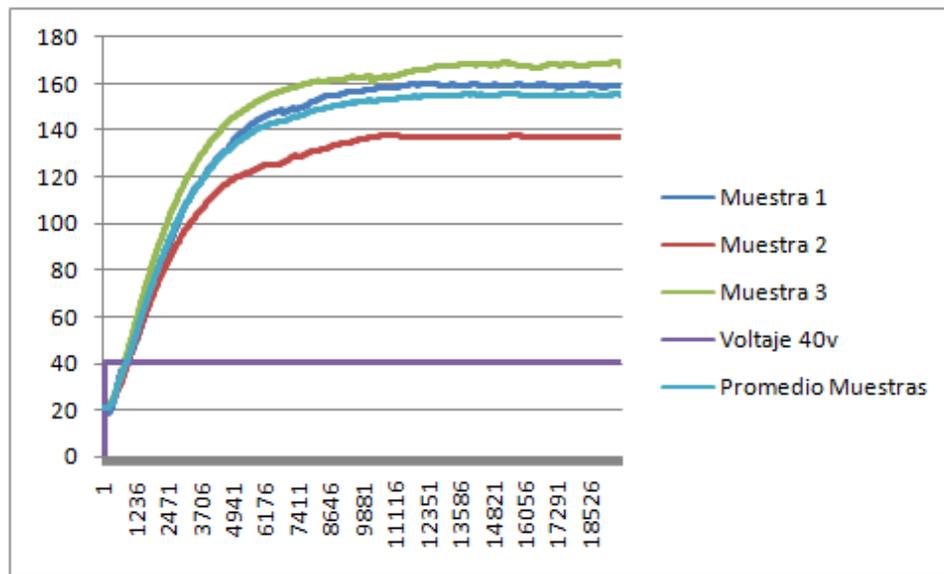


Figura 3.16 Muestras tomadas en la niquelina para 40v.

En la figura 3.16 se aprecia la variación de mediciones con una entrada del mismo voltaje pero en diferentes instantes de tiempo, es por ello que para la obtención de la planta del sistema se promedió todos los valores y a partir de este se realiza la obtención de la ecuación que representará el sistema, para esto utilizamos la herramienta matemática *IDENT* del programa **MATLAB R2011b**.

Antes de pasar a la obtención de la función de transferencia con el *IDENT*, se muestra una adquisición extra de datos realizada a lo largo de la niquelina para analizar su comportamiento, en la cual se colocaron 5 sensores LM 35 durante un escalón controlado, la adquisición de datos se realizó con **LabVIEW 2011** y los datos obtenidos se graficaron con **MATLAB R2011b** utilizando el siguiente código:

```

Editor - C:\Documents and Settings\Administrator\Desktop\tempe...
File Edit Text Go Cell Tools Debug Desktop Window Help
- 1.0 + ÷ 1.1 x
1 - clc
2 - clearall
3 - closeall
4
5 - load'datos_sensores.txt'
6 - plot(datos_sensores)
7 - gridon
8 - title('Comportamiento niquelina')
9 - xlabel('Muestras')
10 - ylabel('Temperatura °C')
11 - legend('1/14 niquelina',...
12 - '1/3 niquelina',...
13 - '1/2 niquelina',...
14 - '2/3 niquelina',...
15 - '2/14 niquelina')
16 - load'muestra40v_1_2_dimer2.txt'
17 - figure,plot(muestra40v_1_2_dimer2)
18 - gridon
19 - title('Niquelina p2 40v')
20 - xlabel('Muestras')
21 - ylabel('Temperatura °C')
22 - entrada_salida(:,1)=muestra40v_1_2_dimer2;
23 - entrada_salida(:,1)=40;
Ln 11 Col 28 OVR
    
```

Figura 3.17 Código de MATLAB

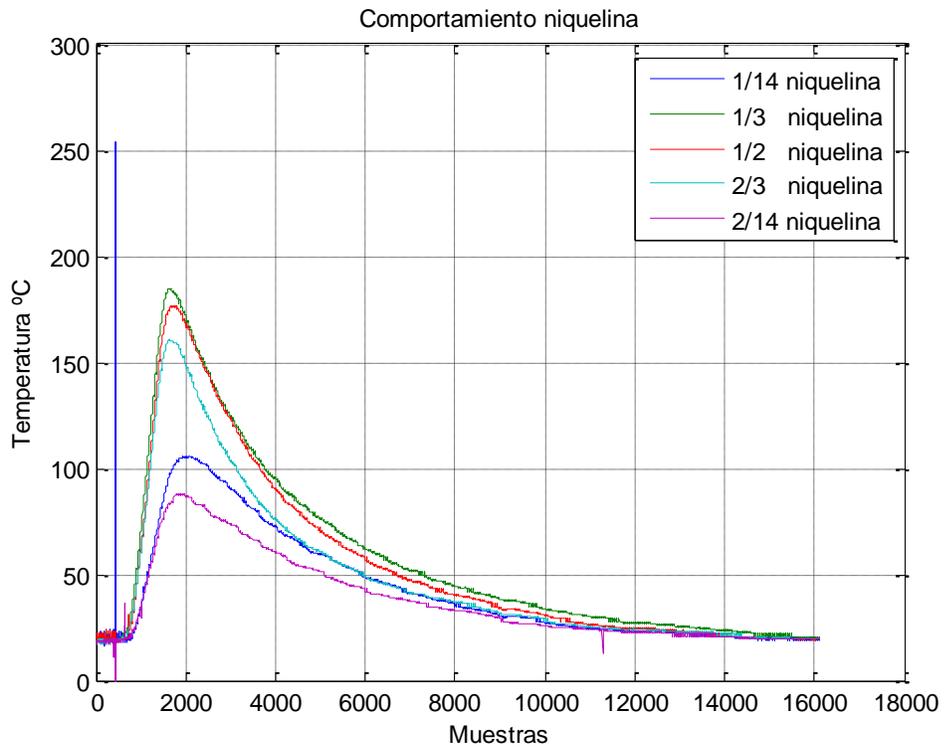


Figura 3.18 Comportamiento de la temperatura a lo largo de la niquelina

En la figura 3.18 se muestra que en la niquelina los puntos de mayor temperatura se generan en su parte central y conforme se va expandiendo la irradiación de calor a los extremos esta va perdiendo temperatura, por la capacidad que tiene el sensor LM-35 no se pudo realizar la adquisición a altas temperaturas por un periodo largo de tiempo, un hecho particular que no mostrado en esta grafica es que conforme avanza el tiempo, la irradiación hacia los extremos comienza a subir y con ello la diferencia de temperatura que existe a lo largo de la niquelina disminuye, las medidas a lo largo del tiempo se las realizaron con el sensor PT100 en el momento de la estabilización de las curvas, teniendo los siguientes valores:

Tabla 3.1 Valores de temperatura a lo largo de la niquelina



Temperatura en grados Centígrados					
←	114,5	128	152	156	190
←	140,8	159	169	160	193,5
←	151,8	159,2	168	174	195,9
←	127,8	144,7	148,8	160,9	165,7
←	80,1	96,2	94,7	95,6	96,1

Al observar la tabla 3.1 se concluye que no es mayormente significativa la variación de temperatura ya que nuestro punto de análisis será cuando el sistema se estabilice, a más de ello se observa que la temperatura varia significativamente mientras más se acerca a los cables y de ende esta parte no va a tener contacto con el compuesto químico, lo cual nos permite hacer despreciable esta consideración.

Una vez concluido con este breve análisis, procedemos ahora con la obtención de la planta con el **IDENT**, realizándolo de la siguiente manera:

Ingresamos en el **IDENT** de MATLABR2011b a través del CommandWindow

```
>>ident
```

Después de ingresar este comando se nos va a abrir la siguiente ventana:

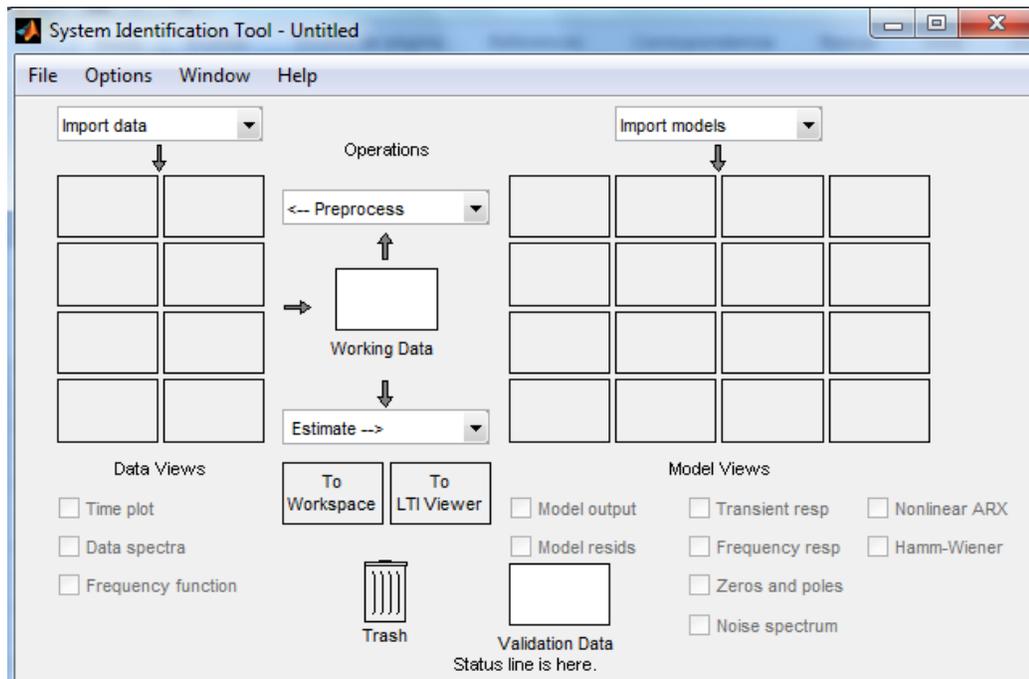


Figura 3.19 Ventana del System Identification Tool

Aquí seleccionamos el dato que vamos a cargar, en este caso lo realizamos en dominio del tiempo, pues es la manera de como adquirimos los datos a través del LABVIEW.

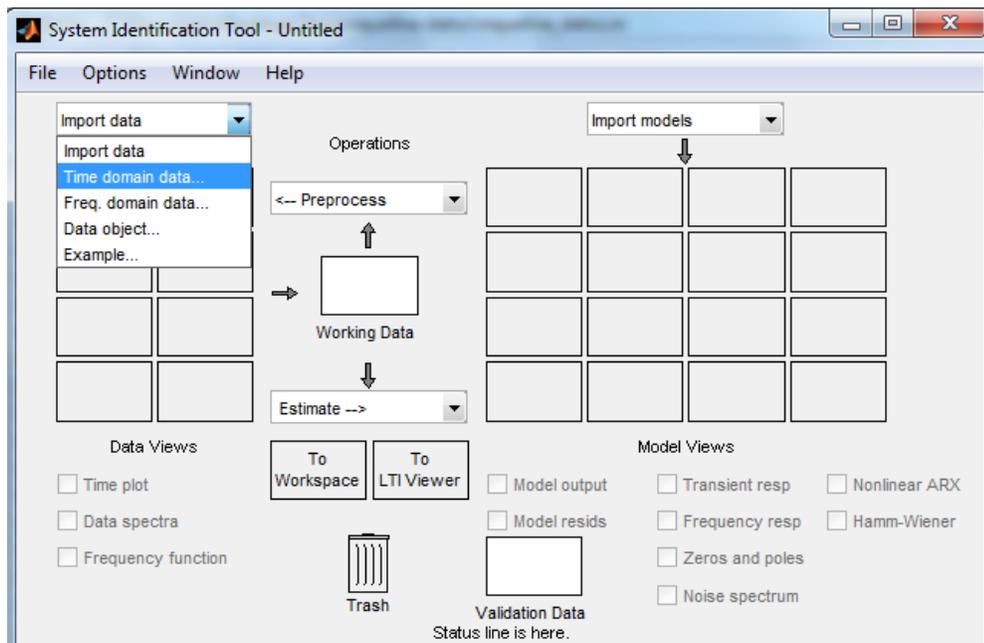


Figura 3.20 Carga de datos en el System Identification Tool

Se abre una ventana en la cual se ingresa el nombre de la matriz de datos que debe estar previamente cargada en el Workspace del MATLAB R2011b, en la ventana se muestran varias opciones, siendo la primera Data Format for Signals en la que se coloca Time-Domain Signals debido que todas las muestras están en dominio del tiempo, en el siguiente campo se pone la matriz de datos $entrada_salida(:,4)$ que es donde se encuentra el dato de ingreso del sistema (contiene un valor de 40 Vac) y $entrada_salida(:,5)$ donde se encuentra los valores de temperatura adquiridos con el LabVIEW, estas tablas se pueden apreciar en el respaldo digital adjunto.

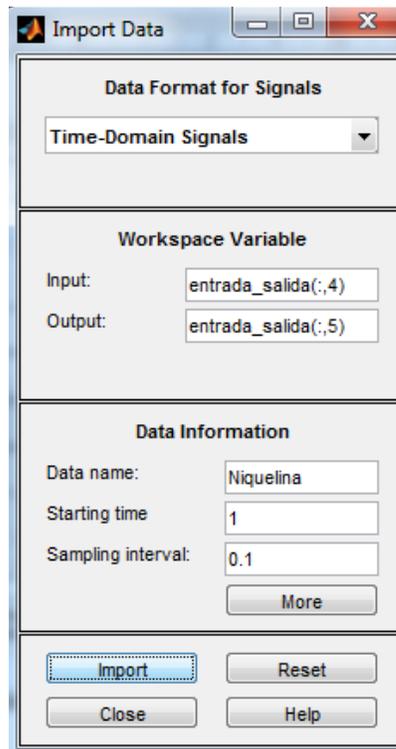


Figura 3.21 Selección de datos a procesar

Con esto se cargan los datos del sistema para su respectivo análisis, en la figura 3.22 se observa en el lado derecho, la gráfica superior en azul, es la resultante de los datos de temperatura, mientras que la inferior es la gráfica de la señal de entrada en este caso un escalón de 40 Vac.

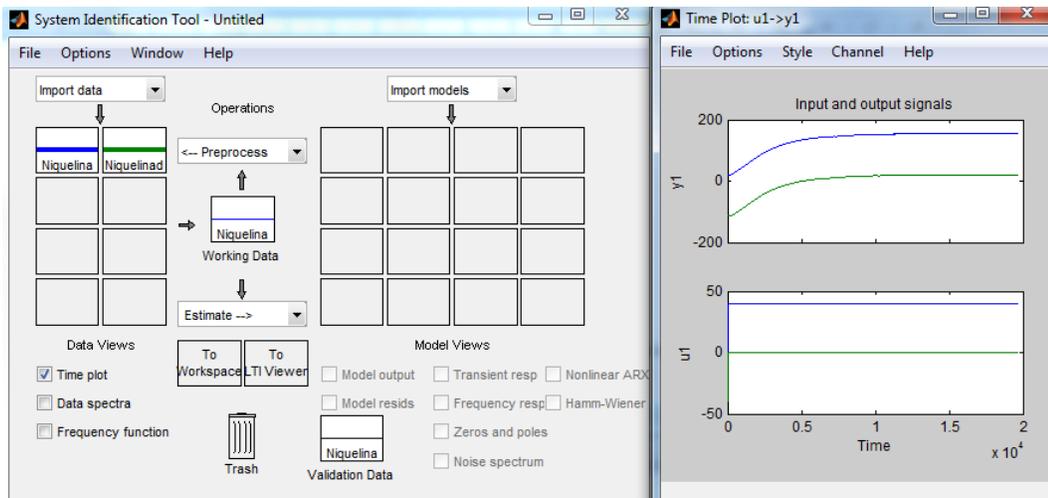


Figura 3.22 Datos que serán procesados

Para obtener la ecuación del sistema al cual vamos a hacer referencia, procedemos de la siguiente manera:

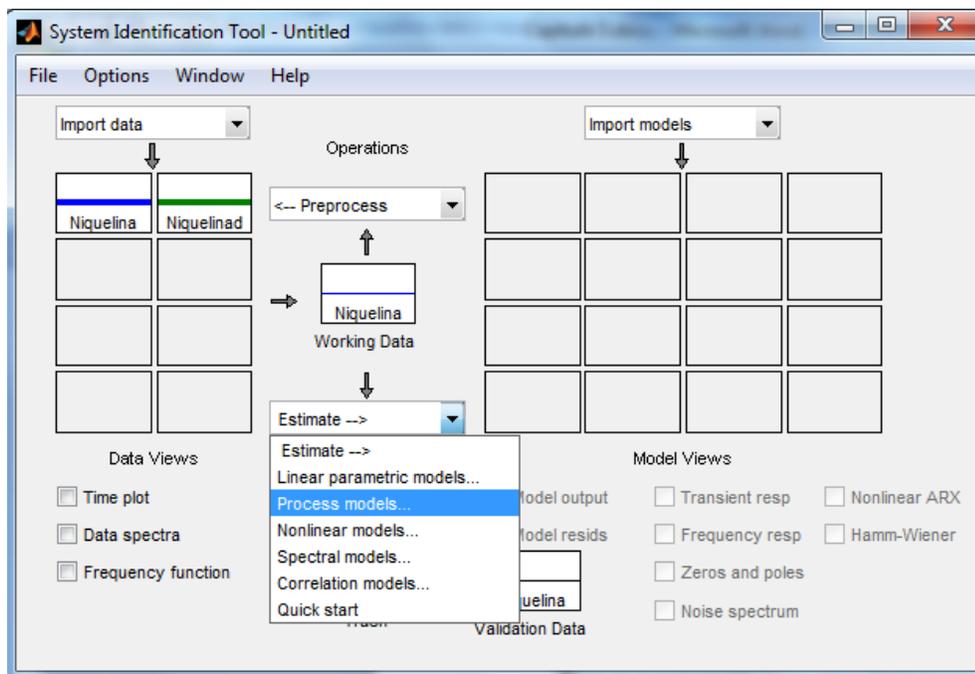


Figura 3.23 Selección del procesamiento a realizar

Se nos abre la siguiente ventana (figura 3.23) donde se podrá escoger el tipo de ecuación que queremos, tanto la cantidad de polos, el retardo, entre otras pues no compete al modelo que se necesita.

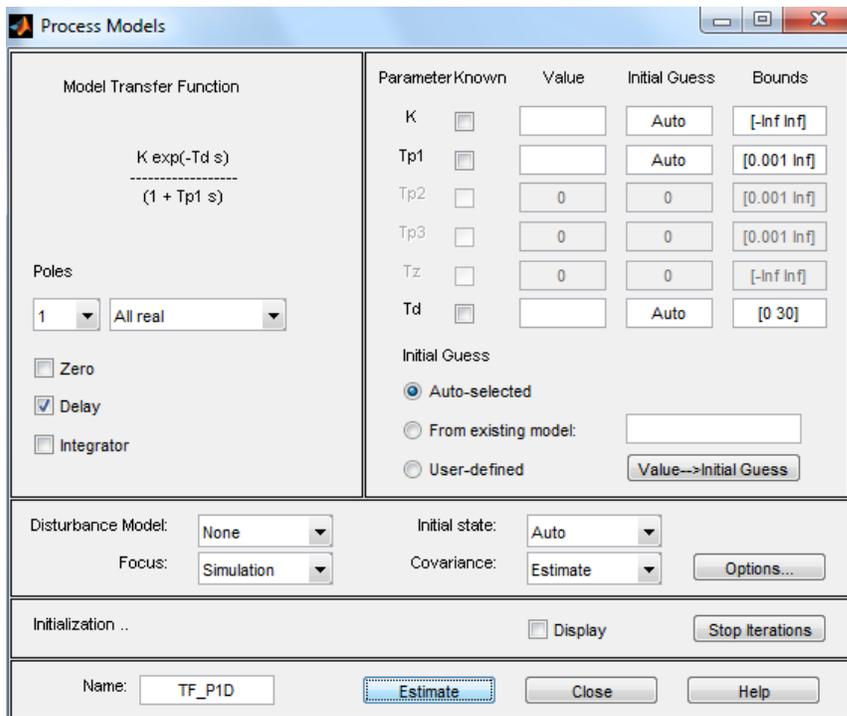


Figura 3.24 Selección de parámetros del procesamiento

Ingresamos los parámetros necesarios, comenzamos con un polo y sin retardo para obtener la función, y sucesivamente continuamos realizando un análisis con funciones de dos polos para escoger el más adecuado para nuestro análisis.

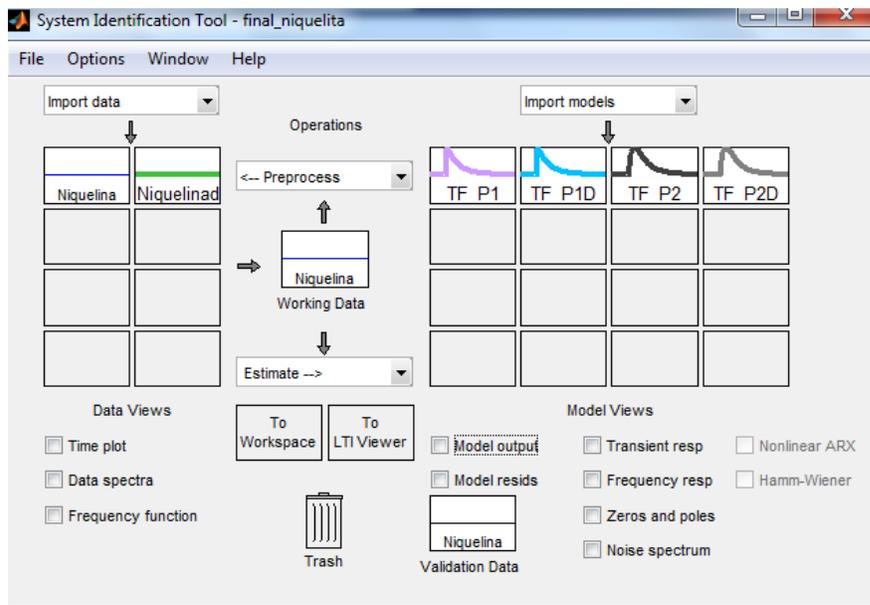


Figura 3.25 Datos obtenidos

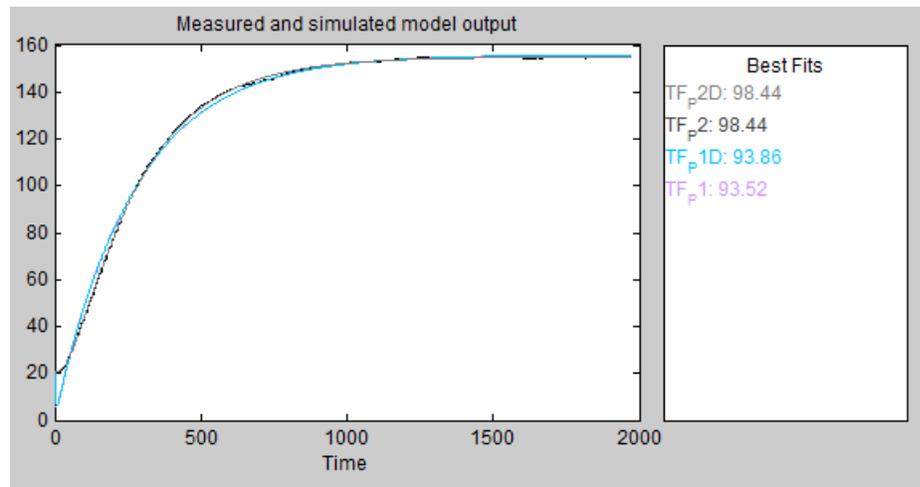


Figura 3.26 Respuestas del sistema al escalón

Para nuestro sistema podemos observar los siguientes datos obtenidos (Figura 3.26), los mismos que fueron analizados de la siguiente manera:

1. En color azul con un polo con retardo, con una aproximación de un 93.86%
2. En color verde con un polo sin retardo, con una aproximación de un 93.52%
3. En color rojo con dos polos sin retardo, con una aproximación de un 98.44%
4. En color turquesa con dos polo con retardo, con una aproximación de un 98.44%

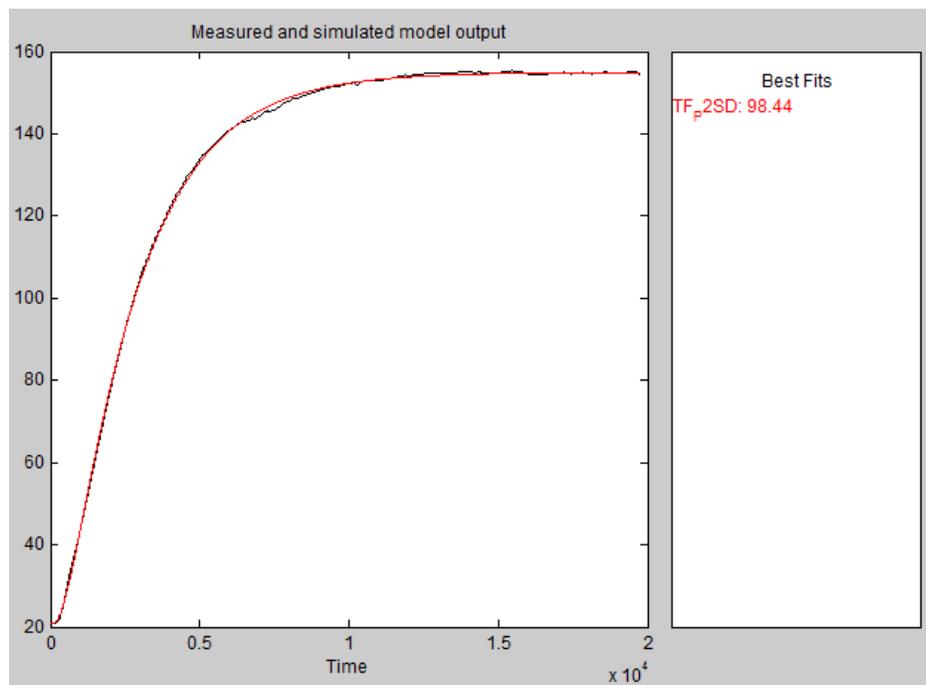


Figura 3.27 Selección del resultado a procesar

En este caso tomamos la mejor aproximación que se ha tenido, existiendo dos resultados que coinciden de igual manera (Figura 3.27), siendo la de dos polos con y sin retardo, tomaremos la que no tiene retardo por la factibilidad del desarrollo del controlador, obteniendo de esta manera los siguientes datos que son expuestos en el **comantwin** de MATLAB R2011b:

Process model with transfer function

$$G(s) = \frac{K_p}{(1+T_{p1}s)(1+T_{p2}s)}$$

with

$$\begin{aligned} K_p &= 3.8737 \pm 0.00034381 \\ T_{p1} &= 236.18 \pm 0.29965 \\ T_{p2} &= 55.45 \pm 0.42512 \end{aligned}$$

Estimated using PEM using Search Method = Auto from data set Niquelina

Loss function 0.291987 and FPE 0.292076

Created: 28-Nov-2011 12:45:31

Last modified: 28-Nov-2011 12:47:23

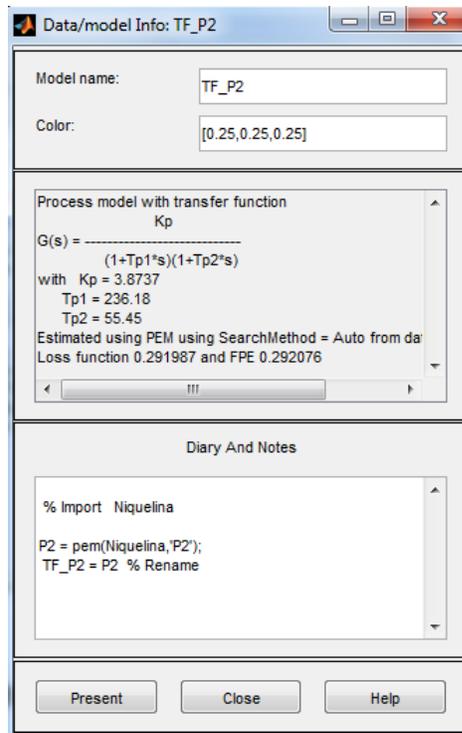


Figura 3.28 Parámetros obtenidos.

$$G(s) = \frac{3.8737}{(1 + 236.18 \cdot s)(1 + 55.45 \cdot s)} \quad \text{Ecuación 3.32}$$

$$G(s) = \frac{2.9843 \cdot 10^{-4}}{s^2 + 0.0224 \cdot s + 7.7046 \cdot 10^{-5}} \quad \text{Ecuación 3.33}$$

Para comprobar la función de transferencia obtenida, se modela el sistema en **Simulink** con el **System Identification Tool** y se ingresa un escalón en la entrada.

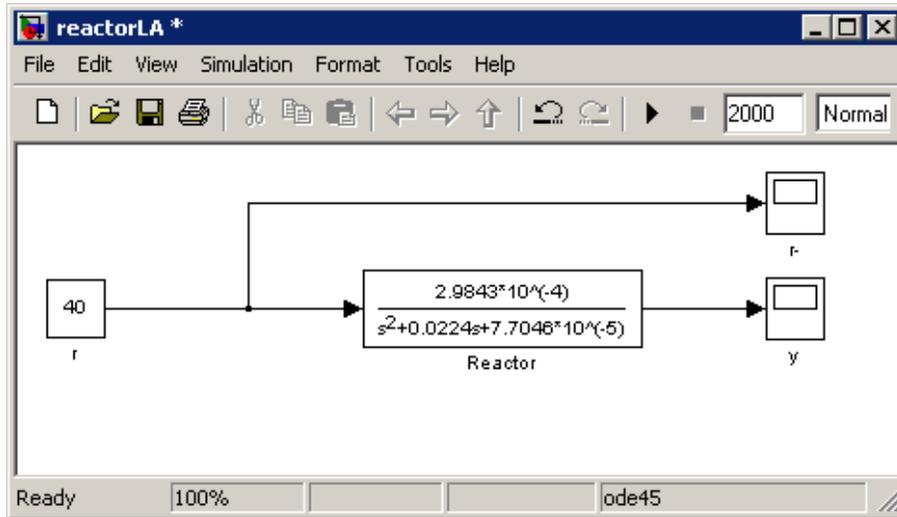


Figura 3.29 Diagramas en bloques de la respuesta

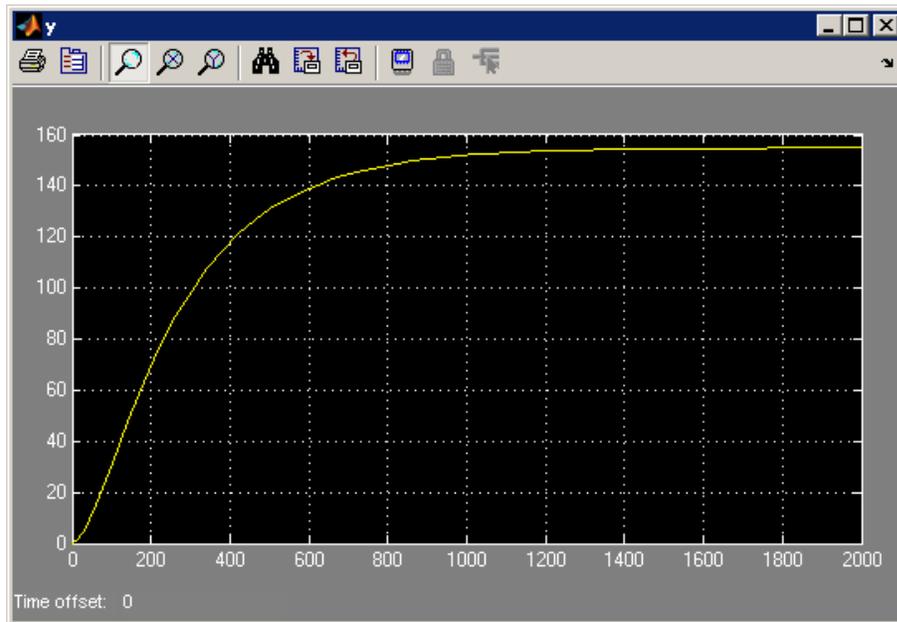


Figura 3.30 Respuesta a un escalón de 40 voltios

Como se observa la función de transferencia obtenida cumple con el modelo adquirido a través de LabVIEW.

3.2.2 Adquisición de datos del motor de corriente continua con imán permanente.

Debido que el fabricante no da a conocer todas las constantes necesarias del motor, para poder utilizar la función de transferencia la determinamos experimentalmente. Los datos serán adquiridos por **myDAQ** y analizados en **LabVIEW 2011** y pueden apreciar en el respaldo digital adjunto.



Figura 3.31 Tarjeta de adquisición MyDAQ de National Instruments

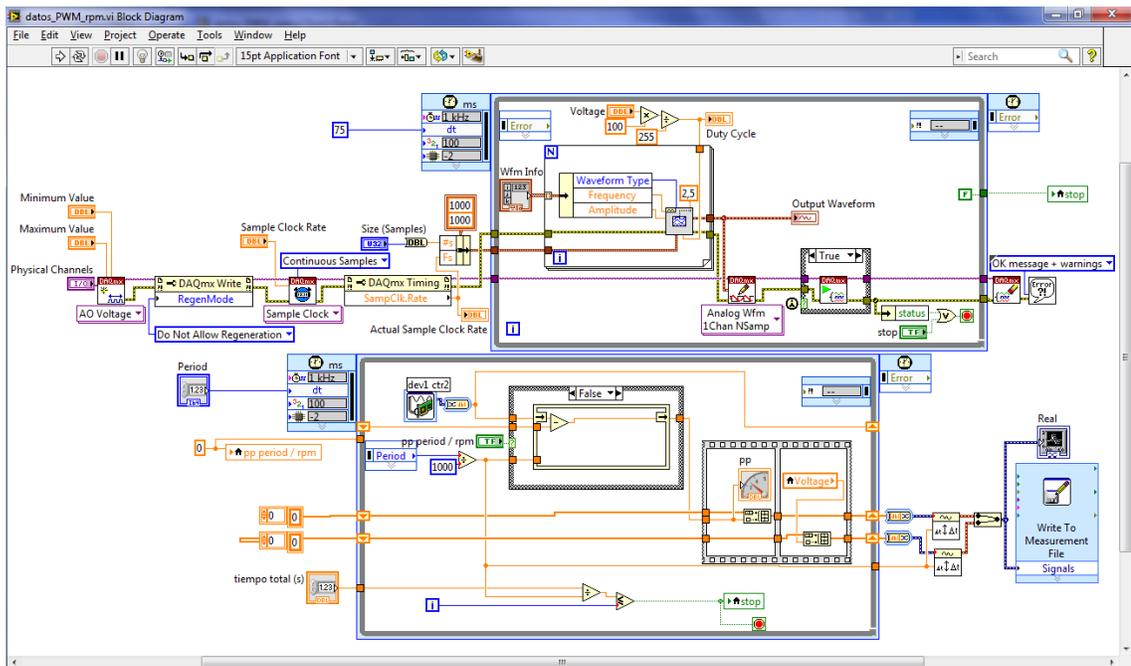


Figura 3.32 Diagrama de bloques en LabVIEW 2011 usado para generar el PWM y contar los pulsos del encoder a través de MyDAQ

El motor será controlado por PWM a una frecuencia de 10Hz que maneja un MOSFET. El máximo voltaje continuo que tendrá el motor será de 5 voltios.

Se realizó 10 pruebas de 20 segundos cada una y con un tiempo de muestreo de 100 milisegundos. Estas 10 pruebas se promediaron para eliminar errores en la medición.

La señal de excitación del MOSFET está representada por el valor en el registro PWM de 8 bits, es decir 255 para un ciclo de trabajo de 100% y 0 para uno de 0%:

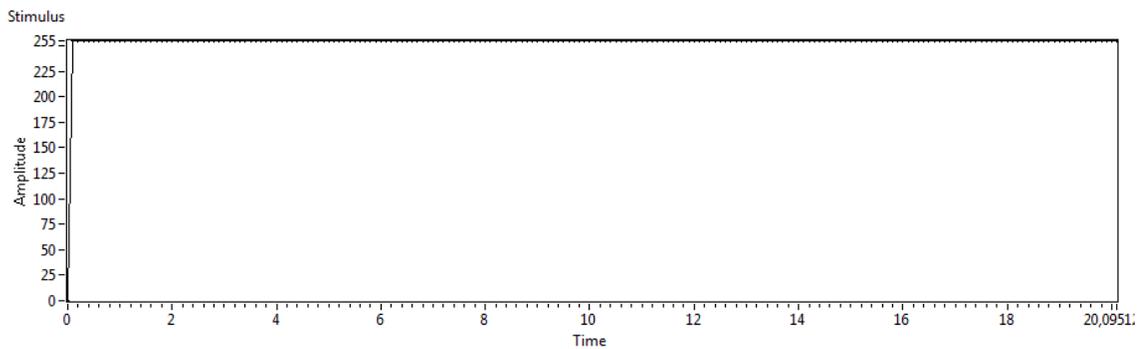


Figura 3.33 Señal de ingreso PWM con ciclo de trabajo al 100% (0xFF = 255)

El promedio de los datos adquiridos del encoder con un periodo de muestreo $T_s=0.1$ segundos es:

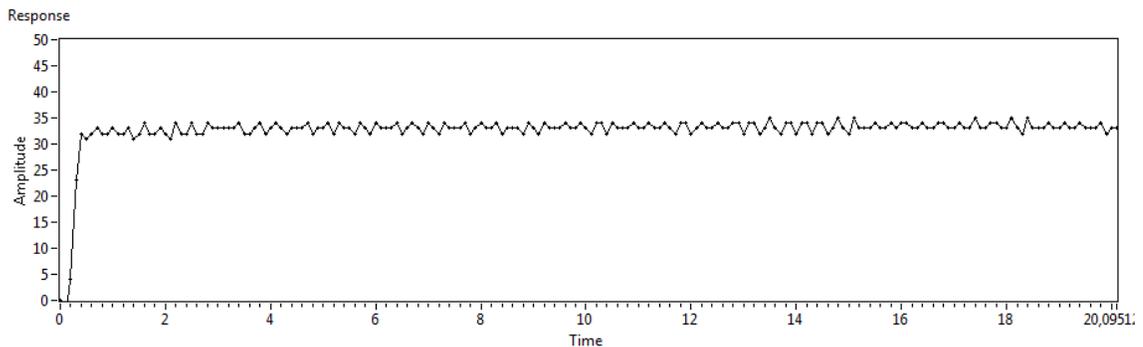


Figura 3.34 Señal de salida con un $T_s = 0.1$ segundos.

Usando el **Toolkit System Identification** y programando en **LabVIEW 2011**, analizamos los datos obtenidos y obtenemos la función de transferencia en función de Laplace y de z.

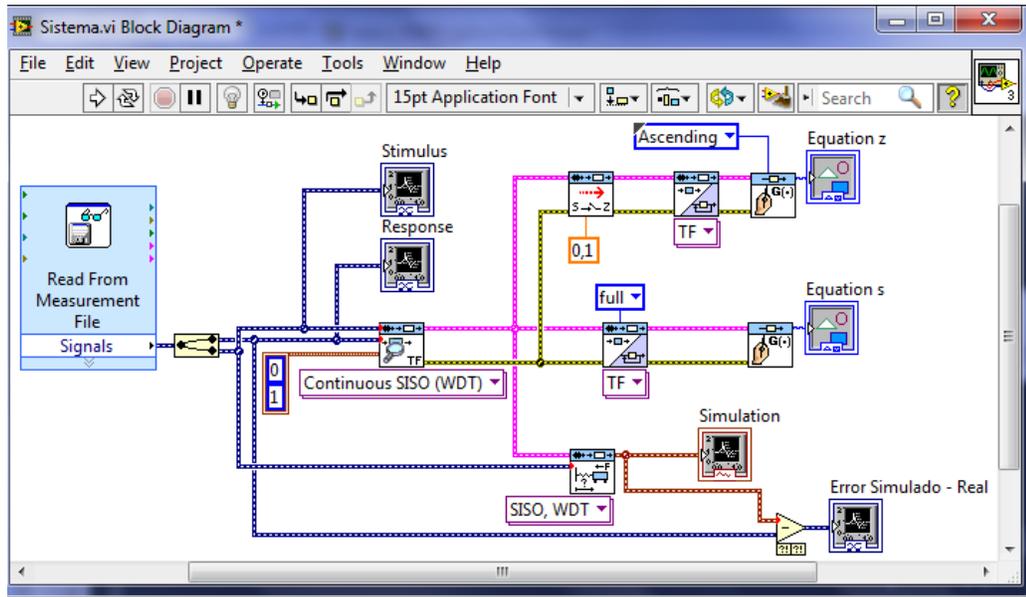


Figura 3.35 . Diagrama de bloques realizado en LabVIEW 2011 para sacar la función de trasferencia del motor.

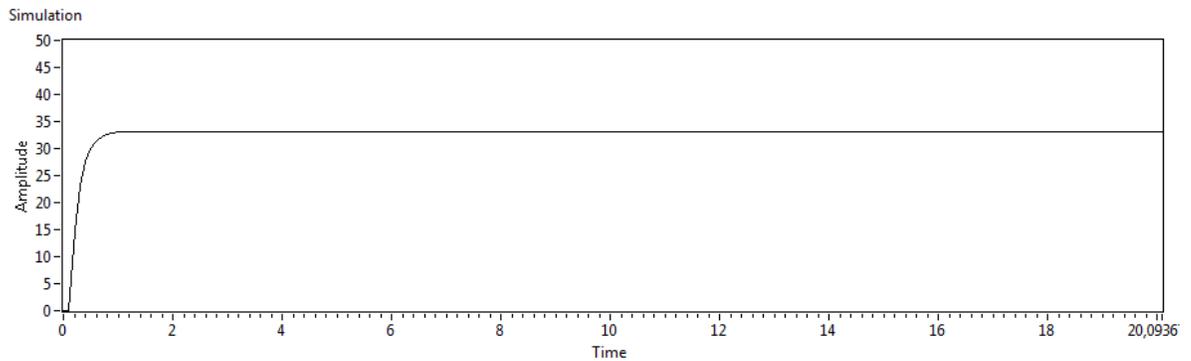


Figura 3.36 Señal de salida para la función de transferencia obtenida.

Por lo tanto las funciones de transferencia para el sistema de agitación son:

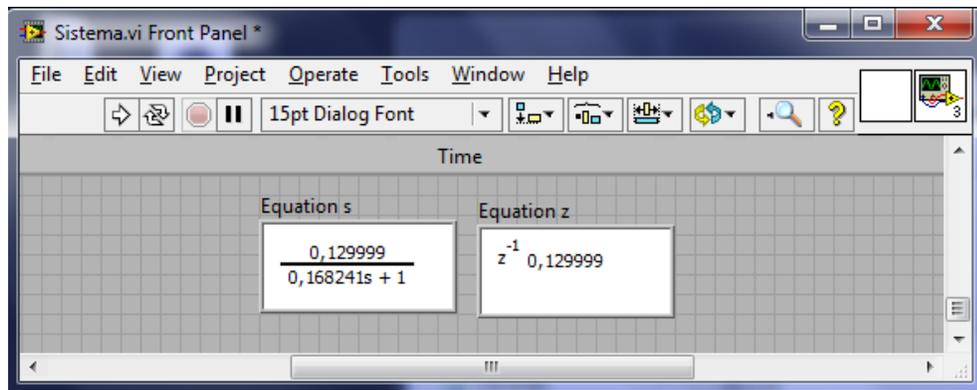


Figura 3.37 Funciones de transferencia para la agitación del reactor

3.3 Diseño del controlador para el sistema de temperatura del reactor.

La función de transferencia del sistema de temperatura del reactor químico es:

$$G(s) = \frac{3.8737}{(1 + 236.18 \cdot s)(1 + 55.45 \cdot s)} = \frac{2.9843 \cdot 10^{-4}}{s^2 + 0.0224 \cdot s + 7.7046 \cdot 10^{-5}} \quad \text{Ecuación 3.34}$$

Paso 1. Se determina el valor de la constante k para un error de estado estacionario (e_{ss}) igual a 0.002 con una entrada de tipo escalón (sistema de tipo 0).

$$K_p = \lim_{s \rightarrow 0} G(s) \quad \text{Ecuación 3.35}$$

$$K_p = \lim_{s \rightarrow 0} \frac{3.8737 * K}{(1 + 236.18 \cdot s)(1 + 55.45 \cdot s)} \quad \text{Ecuación 3.36}$$

$$K_p = 3.8737K \quad \text{Ecuación 3.37}$$

$$e_{ss} = \frac{1}{1 + K_p} \quad \text{Ecuación 3.38}$$

$$0.002 = \frac{1}{1 + 3.8737K} \quad \text{Ecuación 3.39}$$

$$0.002(1 + 3.8737K) = 1 \quad \text{Ecuación 3.40}$$

$$0.002 + 0.0077474K = 1 \quad \text{Ecuación 3.41}$$

$$K = \frac{1 - 0.002}{0.0077474} \quad \text{Ecuación 3.42}$$

$$K = 128.8174 \quad \text{Ecuación 3.43}$$

Al multiplicar el valor de k obtenido en G(s) tenemos que:

$$G(s) = \frac{2.9843 \cdot 10^{-4} * K}{s^2 + 0.0224 \cdot s + 7.7046 \cdot 10^{-5}} \quad \text{Ecuación 3.44}$$

$$G(s) = \frac{0.03844}{s^2 + 0.0224 \cdot s + 7.7046 \cdot 10^{-5}} \quad \text{Ecuación 3.45}$$

Paso 2. La función de transferencia en lazo cerrado de Gm(s) con H(s) = 1, ya que el controlador DELTA envía directamente el valor de temperatura del sensor.

$$M(s) = \frac{Gm(s)}{Gm(s) \cdot H(s) + 1} \tag{Ecuación 3.46}$$

$$M(s) = \frac{\frac{0.03844}{s^2 + 0.0224 \cdot s + 7.7046 \cdot 10^{-5}}}{\frac{0.03844}{s^2 + 0.0224 \cdot s + 7.7046 \cdot 10^{-5}} * 1 + 1} \tag{Ecuación 3.47}$$

$$M(s) = \frac{0.03844}{s^2 + 0.0224 \cdot s + 0.03851} \tag{Ecuación 3.48}$$

El código de MATLAB es:

```
>>gm = tf([0.03844],[1 0.0224 7.7046*10^-5])
>> m = feedback(gm,1)
>>step(m)
```

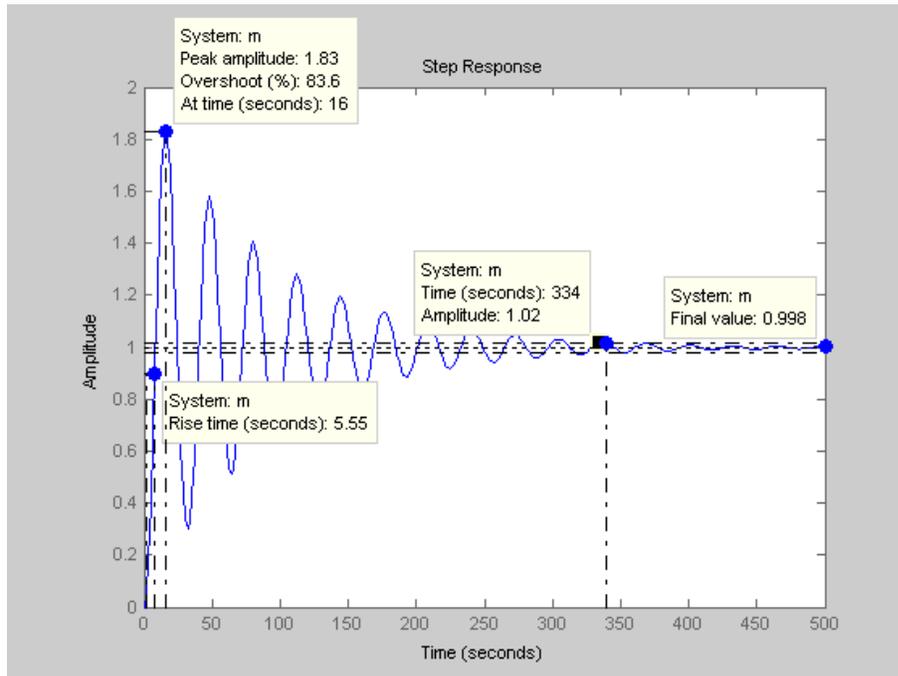


Figura 3.38 Respuesta al escalón unitario de M(s)

En la respuesta al escalón unitario se observa que el máximo sobrealto es de 83.6% lo que ocasionaría muchos problemas en el sistema y en la estabilidad de la reacción del compuesto químico.

Las especificaciones de diseño son las siguientes:

- Máximo sobrealto menor o igual al 0.5%
- Tiempo de subida menor a 20 seg.

- Tiempo de establecimiento menor a 30 seg.

La forma general de un controlador PID es:

$$G_c(s) = K_p + K_D s + \frac{K_I}{s} \quad \text{Ecuación 3.49}$$

$$G_c(s) = (1 + K_{D1}s) \left(K_{P2} + \frac{K_{I2}}{s} \right) \quad \text{Ecuación 3.50}$$

$$G_c(s) = K_{P2}(1 + K_{D1}s) \left(1 + \frac{K_{I2}}{K_{P2}} \right) \quad \text{Ecuación 3.51}$$

Dónde:

$$K_p = K_{P2} + K_{D1}K_{I2} \quad \text{Ecuación 3.52}$$

$$K_D = K_{D1}K_{P2} \quad \text{Ecuación 3.53}$$

$$K_I = K_{I2} \quad \text{Ecuación 3.54}$$

Paso 3: Para el diseño del controlador PD se multiplica la función de transferencia del sistema por $K_{P2}(1 + K_{D1}s)$, recordando que $K = K_{P2} = 128.8174$. La función de transferencia directa viene dada por:

$$G(s) = \frac{2.9843 \cdot 10^{-4}}{s^2 + 0.0224 \cdot s + 7.7046 \cdot 10^{-5}} \cdot K_{P2} \cdot (1 + K_{D1}s) \quad \text{Ecuación 3.55}$$

$$G(s) = \frac{0.03844}{s^2 + 0.0224 \cdot s + 7.7046 \cdot 10^{-5}} \cdot (1 + K_{D1}s) \quad \text{Ecuación 3.56}$$

La ecuación característica se obtiene a partir de la función de transferencia en lazo cerrado del sistema con $H(s) = 1$.

$$T(s) = \frac{G(s)}{1 + G(s)} \quad \text{Ecuación 3.57}$$

La ecuación característica es:

$$1 + G(s) = 0 \quad \text{Ecuación 3.58}$$

$$1 + \frac{0.03844(1 + K_{D1}s)}{s^2 + 0.0224 \cdot s + 7.7046 \cdot 10^{-5}} = 0 \quad \text{Ecuación 3.59}$$

$$s^2 + 0.0224 \cdot s + 7.7046 \cdot 10^{-5} + 0.03844(1 + K_{D1}s) = 0 \quad \text{Ecuación 3.60}$$

$$s^2 + 0.0224 \cdot s + 0.03844K_{D1}s + 7.7046 \cdot 10^{-5} + 0.03844 = 0 \quad \text{Ecuación 3.61}$$

$$s^2 + 0.0224 \cdot s + 0.03844K_{D1}s + 0.03851 = 0 \quad \text{Ecuación 3.62}$$

$$\frac{s^2 + 0.0224 \cdot s + 0.03844K_{D1}s + 0.03851}{s^2 + 0.0224 \cdot s + 0.03851} = 0 \quad \text{Ecuación 3.63}$$

$$1 + \frac{0.03844K_{D1}s}{s^2 + 0.0224 \cdot s + 0.03851} = 1 + G_{eq}(s) \quad \text{Ecuación 3.64}$$

$$G_{eq}(s) = \frac{0.03844K_{D1}s}{s^2 + 0.0224 \cdot s + 0.03851} \quad \text{Ecuación 3.65}$$

Como se observa el diseño se simplifica a un tipo P ya que solo se necesita encontrar la constante de la parte derivativa K_{D1} del problema.

El lugar geométrico de las raíces de $G_{eq}(s)$ es graficado en MATLAB 2011b. Obtenemos diferentes valores de la constante del compensador K_{D1} , las que vamos a analizar para alcanzar las mejores características en su diseño. Con el código de MATLAB:

```
>>Geq=tf([0.03844 0],[1 0.0224 0.03851])
>>rlocus(Geq); grid
```

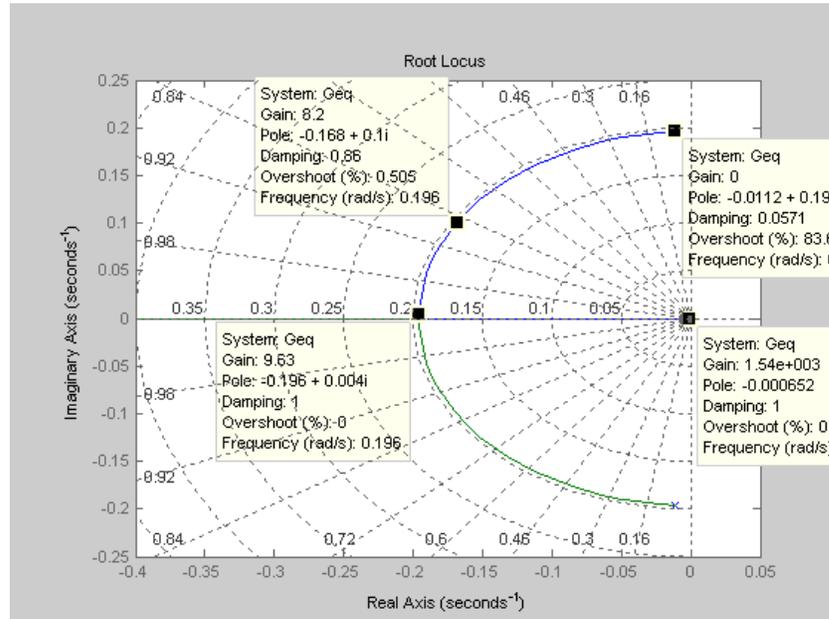


Figura 3.39 Lugar geométrico de las raíces

La gráfica del LGR muestra que la ganancia para un sobresalto de menor al 0.5% (Mp) debe ser mayor o igual a 8.2. Para un análisis profundo se obtiene una tabla para algunos valores de K_{D1} en la función de transferencia con el controlador incorporado.

$$G(s) = \frac{0.03844(1 + K_{D1}s)}{s^2 + 0.0224 \cdot s + 7.7046 \cdot 10^{-5}} \quad \text{Ecuación 3.66}$$

Tabla 3.2 Tabla para algunos valores de K_{D1} .

K_{D1}	Máximo sobresalto (Mp%)	Tiempo de subida seg.	Tiempo de establecimiento seg.
9.63	10.6	4.1	26.5
20	2.73	2.59	16.4
34	0.609	1.64	2.78
50	0.0307	1.14	2.01

Para una ganancia de 50 tenemos:

$$G(s) = \frac{0.03844(1 + 50s)}{s^2 + 0.0224s + 7.7046 \cdot 10^{-5}} \quad \text{Ecuación 3.67}$$

$$G(s) = \frac{1.922s + 0.03844}{s^2 + 0.0224s + 7.7046 \cdot 10^{-5}} \quad \text{Ecuación 3.68}$$

Graficando la respuesta al escalón del sistema en lazo cerrado en MATLAB:

```
>>Gc=tf([1.922 0.03844],[1 0.0224 7.7046*10^-5])
>> Mc=feedback(Gc,1)
>>step(Mc); grid
```

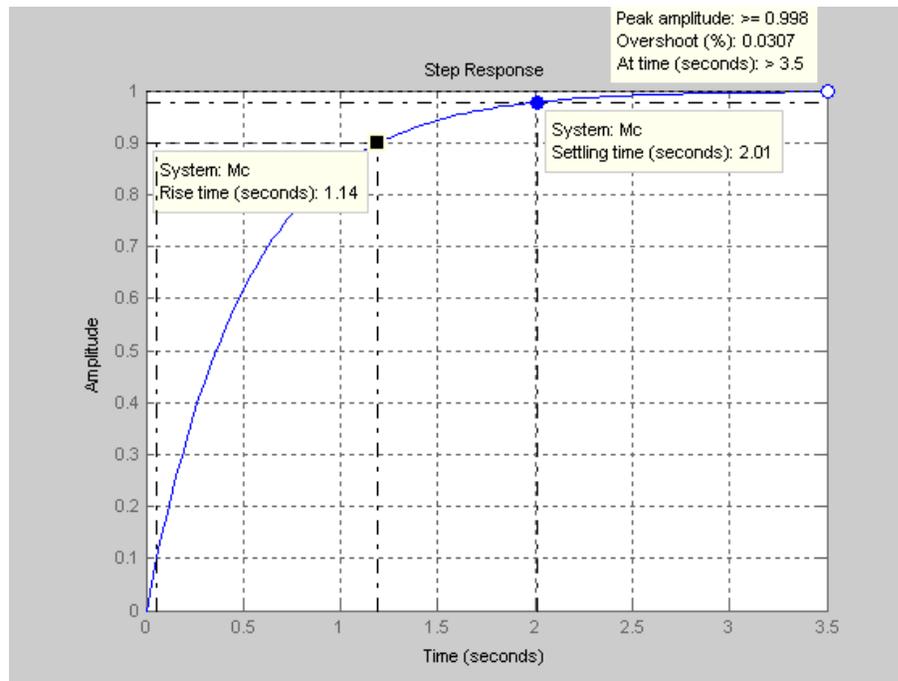


Figura 3.40 Respuesta en lazo cerrado del sistema de temperatura

Paso 4: Encontramos las constantes proporcional, derivativa e integral del control PID usando $K_{P2} = 128.8174$, $K_{D1} = 50$ y $K_{I2} = 0$ (ya que el control que satisface la estabilidad del sistema es PD).

La constante proporcional según la ecuación 3.52:

$$K_P = K_{P2} + K_{D1}K_{I2} \tag{Ecuación 3.69}$$

$$K_P = 128.8174 + 50 \cdot 0 \tag{Ecuación 3.70}$$

$$K_P = 128.8174 \tag{Ecuación 3.71}$$

Usando la ecuación 3.53 se obtiene la constante derivativa:

$$K_D = K_{D1}K_{P2} \tag{Ecuación 3.72}$$

$$K_D = 128.8174 * 50 \tag{Ecuación 3.73}$$

$$K_D = 6440.87 \tag{Ecuación 3.74}$$

Y la constante integral por medio de la ecuación 3.54

$$K_I = K_{I2} \tag{Ecuación 3.75}$$

$$K_I = 0$$

Ecuación 3.76

3.4 Diseño de los controladores para el sistema de velocidad de agitación del reactor.

La agitación del reactor no es un proceso crítico como el de la temperatura ya que sirve para que toda la reacción química tenga una mezcla homogénea y una temperatura uniforme, además que no estará expuesta a perturbaciones externas que intervengan en su estabilidad. La velocidad del motor depende directamente de la viscosidad del compuesto químico y esta, según la función de transferencia, depende del voltaje de alimentación aplicado a los bornes.

El sistema de control del motor está en lazo abierto, se comanda por el ciclo de trabajo porcentual de un modulador PWM²⁸ que trabajará a una frecuencia alta (15.6kHz) para no perder el torque del motor y para no utilizar frecuencias audibles.

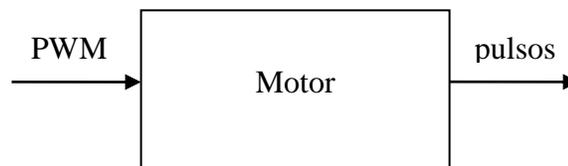


Figura 3.41 Sistema de agitación

El ciclo de trabajo (duty cycle) de una señal periódica es el tiempo en alto de su parte positiva relacionada con el periodo total de la onda:

$$D = \frac{\tau}{T} * 100 \quad [\%]$$

Ecuación 3.77

El ciclo de trabajo es **D**, τ es el tiempo en que la función es positiva (ancho del pulso) y el periodo total de la señal periódica es **T**.

Al trabajar con microcontroladores, asignamos una variable de 8 bits como registro PWM, tendríamos entonces que para un ciclo de trabajo de 0% la variable PWM tomaría el valor de 0 y para un $D = 100\%$ el valor de PWM sería de 255.

Aplicando estos conceptos, el valor de la variable PWM sería:

$$PWM = entero \left[\frac{D \cdot 255}{100} \right]$$

Ecuación 3.78

²⁸ Modulación por ancho de pulso.

El encoder tiene 48 pulsos por revolución (ppr) y cada $T_s=100$ milisegundos tenemos la lectura del número de pulsos, por lo que para poder medir en RPM tenemos que hacer la conversión primero a revoluciones por segundo (RPS):

$$RPS = \frac{(\#pulsos) \cdot Ts^{-1}}{ppr} \quad \text{Ecuación 3.79}$$

En RPM tenemos:

$$RPM = RPS * 60 \quad \text{Ecuación 3.80}$$

Por lo tanto:

$$RPM = \frac{(\#pulsos) \cdot Ts^{-1}}{ppr} * 60 \quad \text{Ecuación 3.81}$$

$$RPM = \frac{(\#pulsos) \cdot 0.1^{-1}}{48} * 60 \quad \text{Ecuación 3.82}$$

$$RPM = (\#pulsos) \cdot 12.5 \quad \text{Ecuación 3.83}$$

3.5 Simulación del sistema.

Es primordial simular el sistema para comprender como se va a comportar la temperatura del reactor y la mezcla que va a proveer el agitador, su importancia radica en prever los posibles problemas que podrían ocurrir al implementar el controlador en el sistema y si estos ocurren, corregirlos de una manera rápida, intuitiva y asistida gracias a las herramientas que hoy existen para este fin como es Simulink, un toolkit de MATLAB 2011.

3.5.1 Simulación del sistema de temperatura del reactor discontinuo agitado

En Simulink con los bloques de funciones se ingresa el sistema de control de temperatura junto con la planta (reactor químico). Como referencia se pone una temperatura de $r = 150$ °C para simular el sistema.

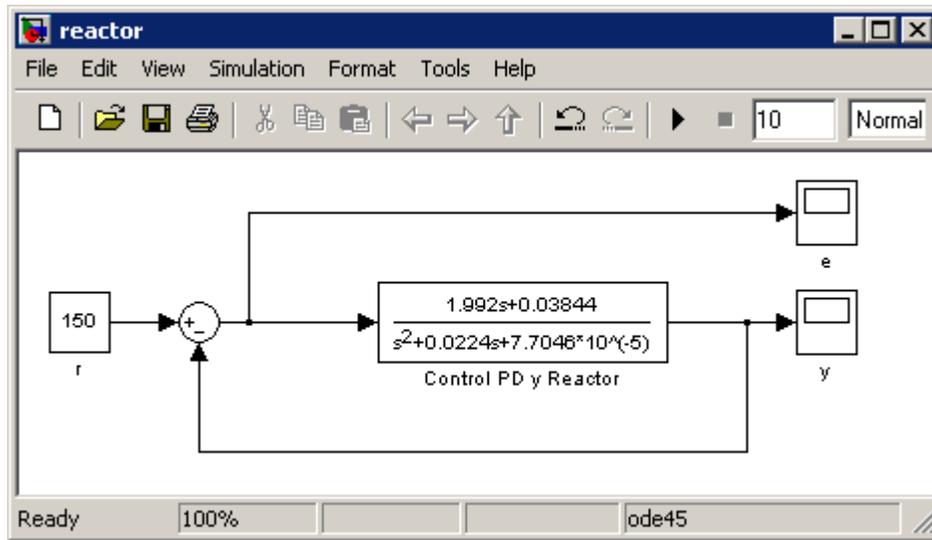


Figura 3.42 Sistema de control de temperatura del reactor químico

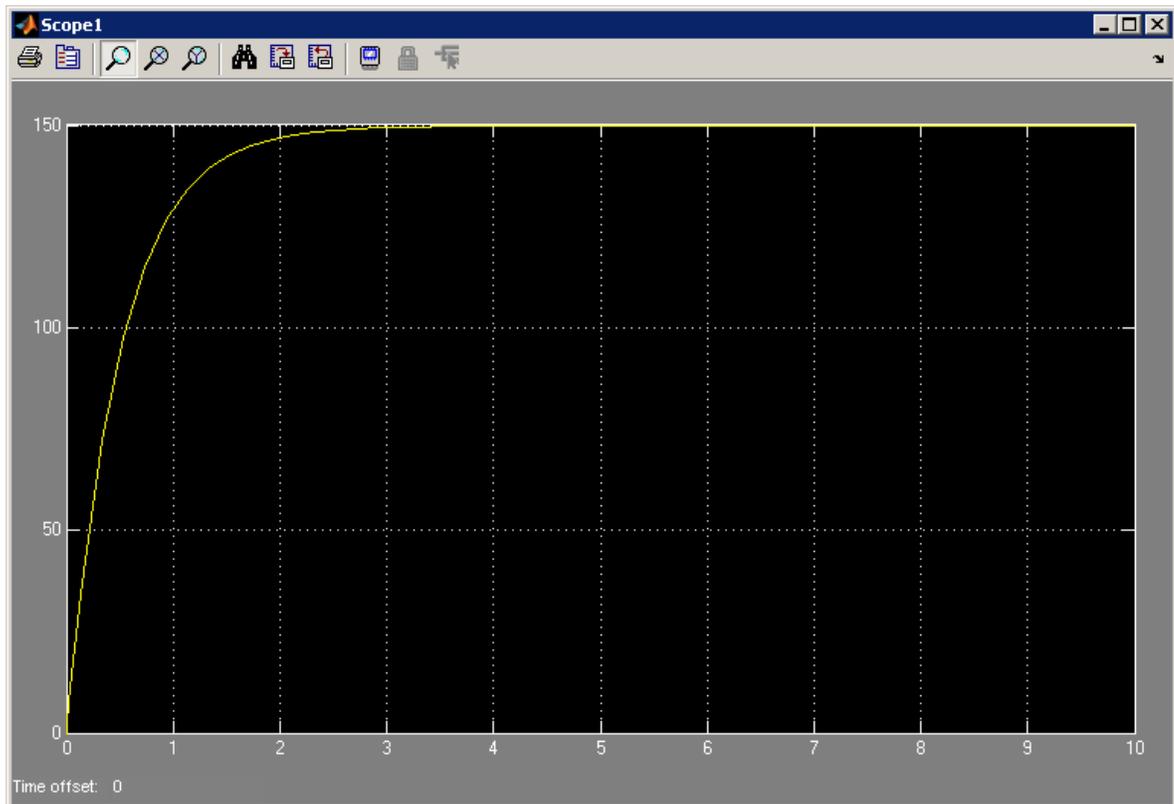


Figura 3.43 Salida del sistema de temperatura (°C)

La figura 3.43 muestra que el tiempo de establecimiento esta alrededor de los 3 segundos, pero en la vida real y al implementar el sistema vamos a toparnos con no linealidades que harán que el sistema no llegue a estabilizarse en un lapso tan rápido.

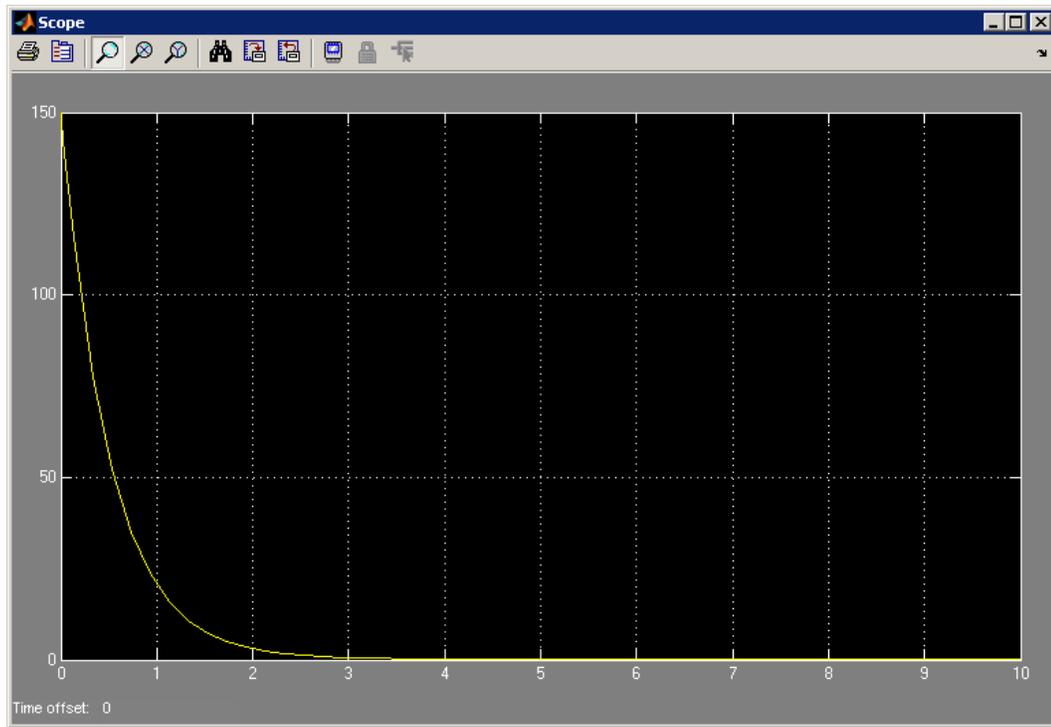


Figura 3.44 Señal de error del sistema de control de temperatura

Se observa en la figura 3.44 que el error del sistema disminuye conforme avanza el tiempo, hasta que este tiende a ser cero cuando llega al tiempo de establecimiento.

3.5.2 Simulación del sistema de agitación del reactor discontinuo agitado

Con los bloques de funciones, en Simulink se ingresa la función de transferencia del motor en lazo abierto, vamos a analizar al sistema con una entrada en su referencia, la que será el ciclo de trabajo de la modulación PWM representada en 8 bits (0 a 255).

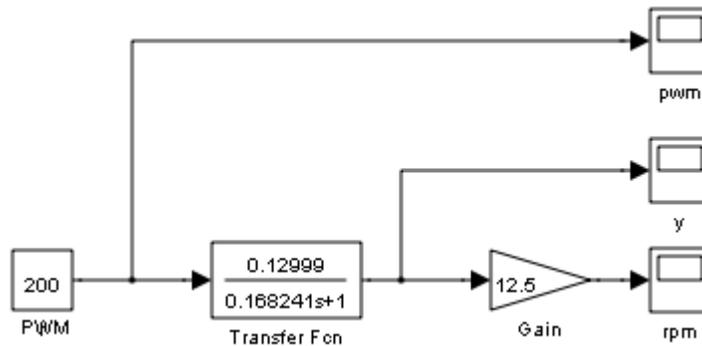


Figura 3.45 Sistema de control de agitación del reactor químico

En la de referencia r del sistema aplicamos una entrada de 200 para observar el comportamiento en la salida del sistema.

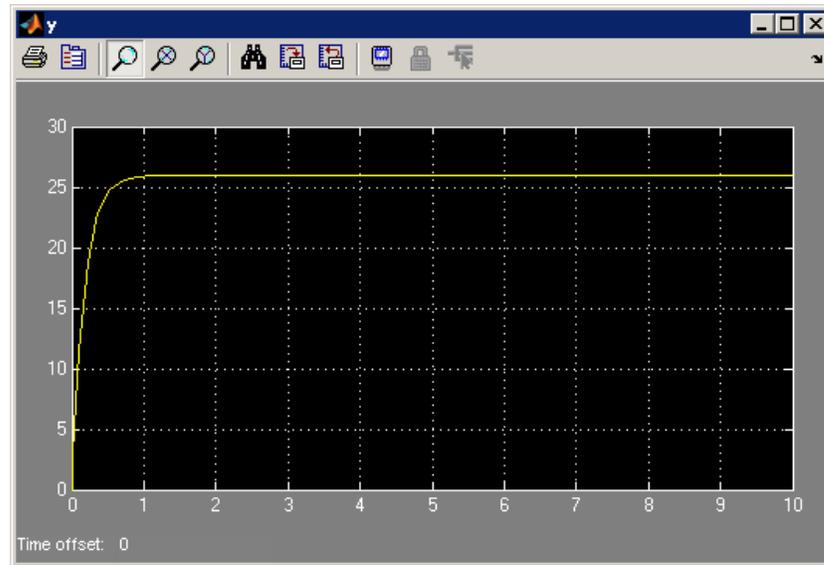


Figura 3.46 Salida del sistema de agitación en pulsos cada 100 milisegundos

Convirtiendo la salida del sistema a RPM tenemos:

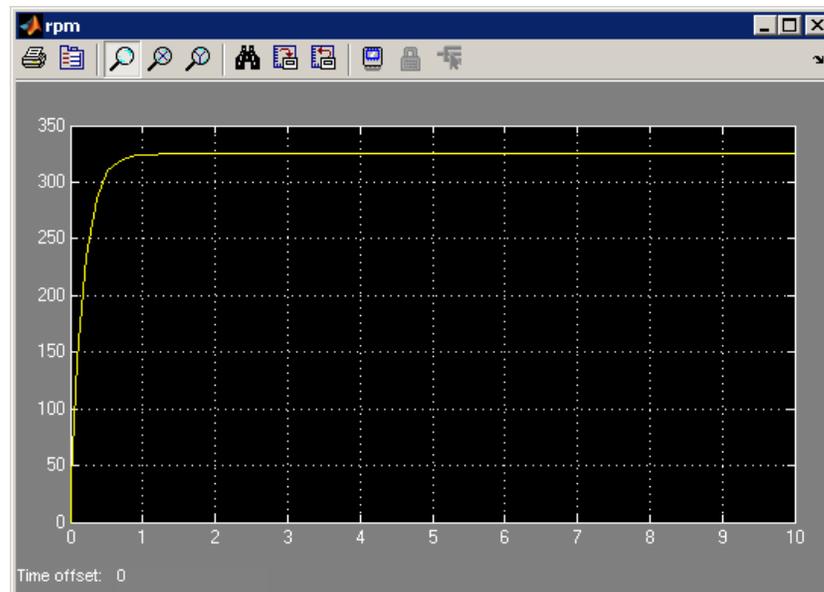


Figura 3.47 Salida del sistema de agitación en RPM

El tiempo de establecimiento en la figura 3.47 está alrededor de un segundo para estabilizarse, lo que es muy eficiente para el sistema de agitación (considérese que en la práctica este tiempo va a ser mucho mayor por las características de la niquelina).

3.6 Diseño de la interfaz humana máquina HMI.

El interfaz humano maquina consta de dos partes: del software desarrollado en LabVIEW2011 para PC y del firmware²⁹ desarrollado en microcontroladores PIC. En este apartado comenzaremos mencionando la elaboración del software y la explicación de cada parte del mismo, luego se continuara con el diseño del firmware. En el desarrollo del hardware se mostrara la realización, el diseño de la tarjeta electrónica y sus componentes.

3.6.1 Software, interfaz humana máquina HMI.

El software **HMI**³⁰ está desarrollado en LabVIEW2011 y usa el puerto serial (RS-232) para establecer una comunicación con el firmware del reactor químico, el toolkit VISA es el que nos provee de los VI necesarios para cumplir este objetivo. El programa tiene una configuración de máquina de estados. Al iniciar el programa el software busca el instrumento necesario para la comunicación, ni no se lo conecta, el programa no permite ninguna acción (excepto salir), mientras que si el instrumento es encontrado, el usuario puede ejecutar la acción que le convenga dependiendo de su necesidad, todas estas opciones serán descritas en este apartado. La programación (diagrama de bloques) está detallada en el ANEXO A. La interface con el usuario cuenta con varias opciones y una interfaz gráfica intuitiva que hacen que el manejo del software sea sencillo.

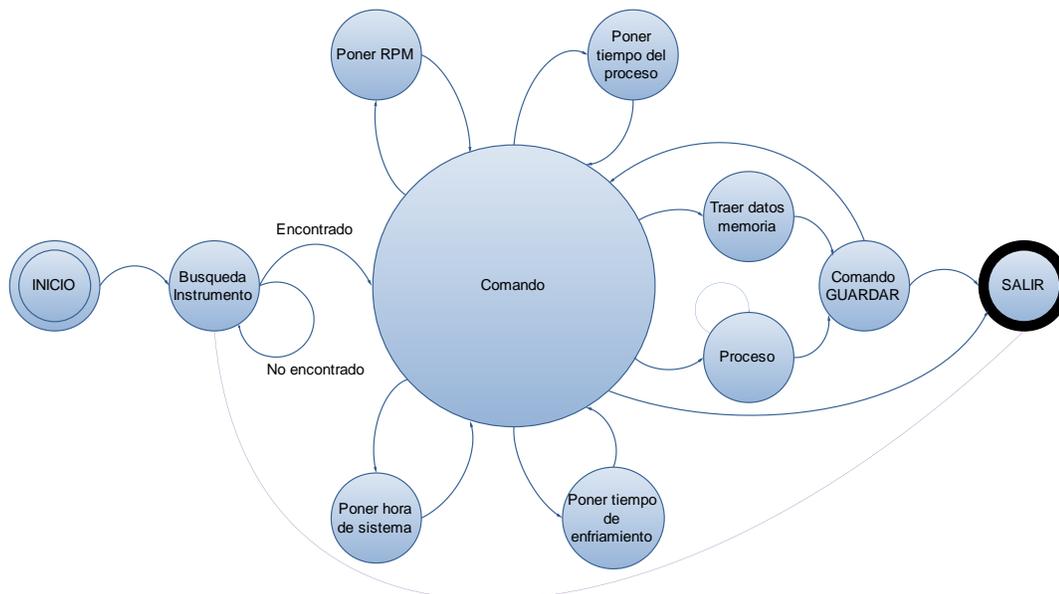


Figura 3.48 Diagrama de estados del software HMI

²⁹ **Firmware.**- según la IEEE es "La combinación de instrucciones de un dispositivo de hardware e instrucciones y datos de computadora que residen como software de solo lectura en ese dispositivo".

³⁰ **HMI.**- Human Machine Interface

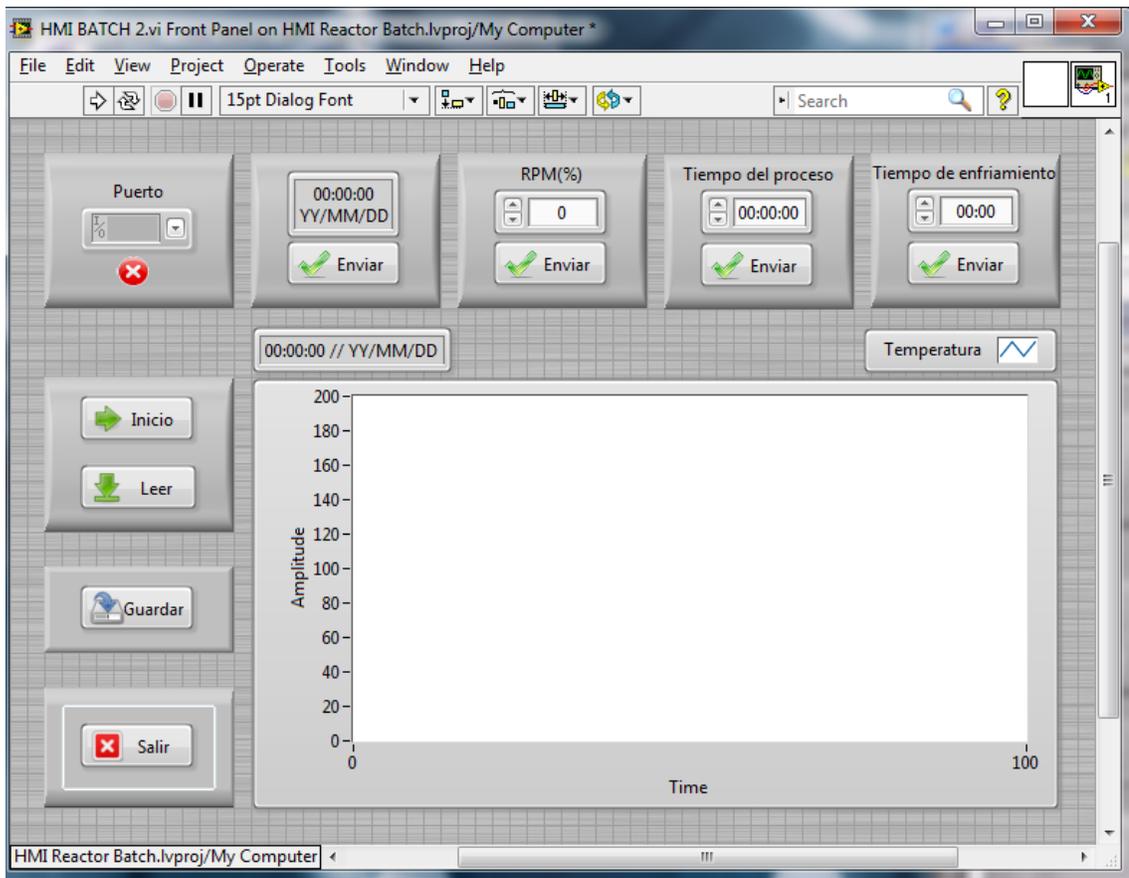


Figura 3.49 Panel frontal de la interface HMI del sistema

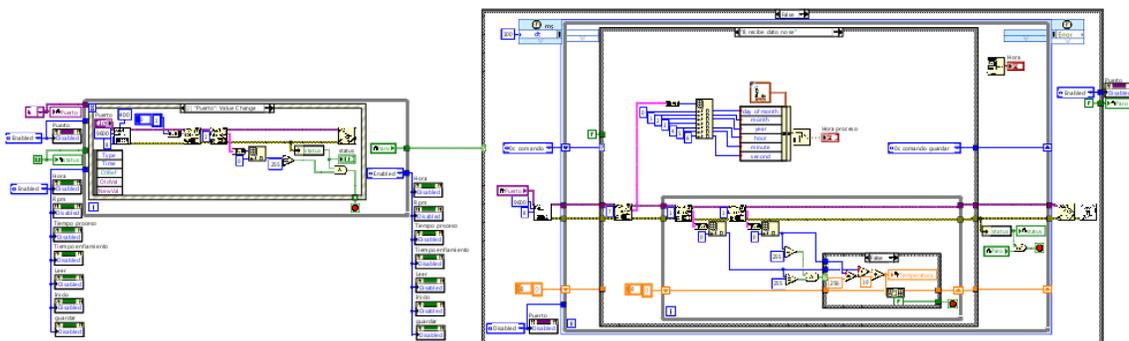


Figura 3.50 Diagrama de bloques de la interface HMI del sistema

3.6.1.1 Selección del puerto

En primer lugar existe la opción de seleccionar el puerto serial de la PC al cual el sistema está conectado.



Figura 3.51 Selección del puerto serial

Cuando se elige un puerto, la interface muestra un indicador con un visto bueno de conexión establecida, caso contrario, mientras ningún puerto esta seleccionado, o se produjo un error en la conexión, el indicador muestra una equis.

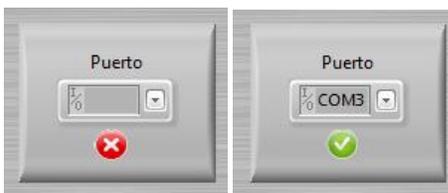


Figura 3.52 Indicador de conexión en sus dos estados.

3.6.1.2 Hora del sistema

Con esta opción el usuario puede igualar la hora del PC al sistema del reactor químico, para esto la interfaz de usuario muestra la hora actual y la fecha, y un botón que gestiona el envío de los datos necesarios.

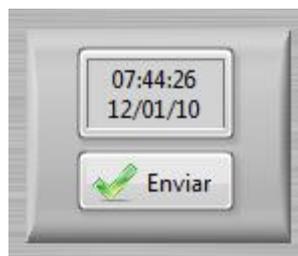


Figura 3.53 Gestión de la hora en el sistema

3.6.1.3 Revoluciones por minuto del sistema de agitación

Con el botón de “Enviar” se ingresa las revoluciones por minuto que el usuario requiera a través de la interface de usuario. Para ingresar las RPM en el control numérico podemos hacerlo por medio del teclado o usando los botones de incremento y decremento. Las RPM tienen rango entre 0% y 100% de la velocidad máxima.



Figura 3.54 Revoluciones por minuto

3.6.1.4 Tiempo del proceso

El tiempo total de proceso es gestionado por un control numérico en el que ponemos las horas, minutos y segundos que el sistema se mantendrá a cierta temperatura, para esto usamos los botones de incremento, decremento o el teclado. Al pulsar el botón “Enviar” los datos se envían al sistema electrónico del reactor químico.

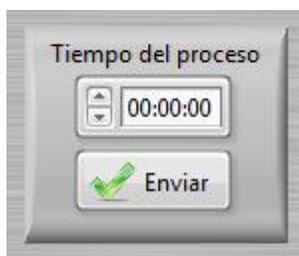


Figura 3.55 Tiempo del proceso

3.6.1.5 Tiempo de enfriamiento

Con el botón de “Enviar” el tiempo de enfriamiento que el usuario requiera se envía al sistema electrónico de control. Para ingresar este tiempo en el control numérico podemos hacerlo por medio del teclado o usando los botones de incremento y decremento.

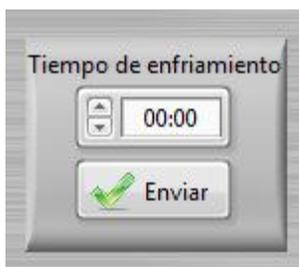


Figura 3.56 Tiempo de enfriamiento

3.6.1.6 Inicio del proceso y lectura de datos internos del sistema electrónico

Cuando todos los parámetros del sistema estén configurados, se puede empezar el proceso y la adquisición de datos usando el botón “Inicio”. En cambio, si el proceso fue realizado y guardado en memoria interna, los datos pueden ser visualizados a través del botón “Leer”.



Figura 3.57 Inicio del proceso y lectura de datos

3.6.1.7 Guardado de los datos adquiridos

Con el botón de “Guardar” podemos salvar los datos generados por el sistema o los que fueron guardados en la memoria interna del sistema electrónico.



Figura 3.58 Botón "guardar"

3.6.1.8 Gráfica de la forma de onda de temperatura

Este gráfico permite que el usuario observe la variación de la temperatura durante el tiempo que dura el proceso o cuando se leen los datos de temperatura guardados en la memoria interna del sistema electrónico. En la parte superior de la gráfica se observa un indicador numérico que muestra la fecha en la que el proceso se llevó a cabo.

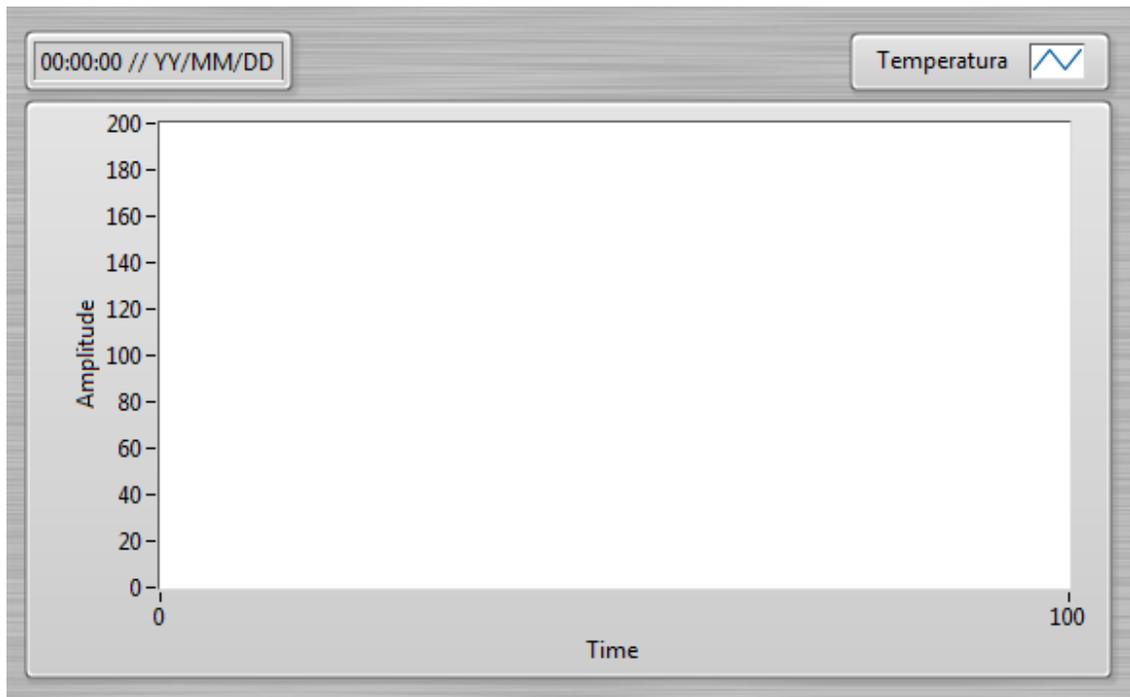


Figura 3.59 Gráfico de la forma de onda de la temperatura

3.6.1.9 Salir de la interfaz

Con el botón “Salir” el usuario puede terminar la sesión en el sistema de interface de usuario HMI para el control del reactor químico.



Figura 3.60 Salir del sistema

3.6.2 Firmware, interfaz humana máquina HMI.

Para que el software HMI del reactor se comunique con el equipo y realice las operaciones antes expuestas y siempre tomando en consideración las necesidades explicadas por la parte química, se elaboró el diseño electrónico del firmware de tal manera de poder asegurar una estabilidad y confiabilidad extrema.

3.6.2.1 Programa principal del sistema electrónico.

Se ha separado por bloques las diferentes partes del diseño del firmware para evitar un exceso de procesos dentro de un microcontrolador, pudiendo este llegar a bloquearse y desestabilizar todo el sistema, para ello se tomó la siguiente consideración:

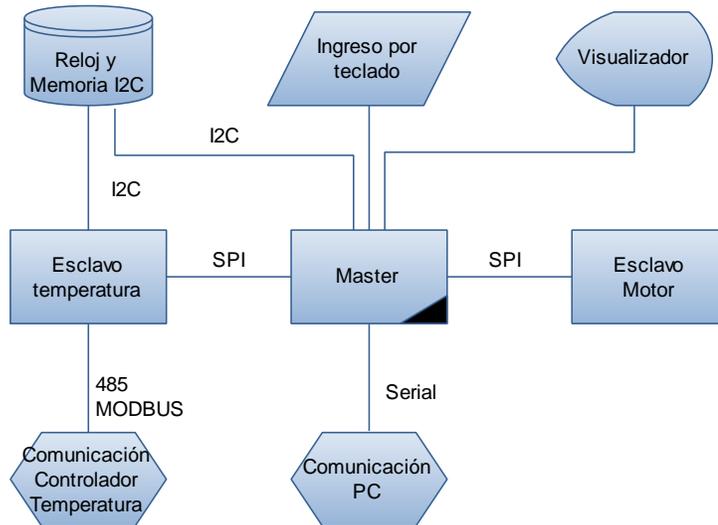


Figura 3.61 Diagrama de bloques del diseño electrónico

El bloque principal (Figura 3.62) está dado por el master que será implementado en un microcontrolador 18f452 que se comunicará con todos los dispositivos desde el programa principal del sistema, que maneja la comunicación con el computador (comunicación serial) al igual que los dispositivos de entrada (pulsantes), los de salida (visualizador LCD), los dispositivos esclavos a través de una comunicación SPI (comunicación con el controlador de temperatura y el controlador de velocidad), y por ultimo con la memoria interna I2C.

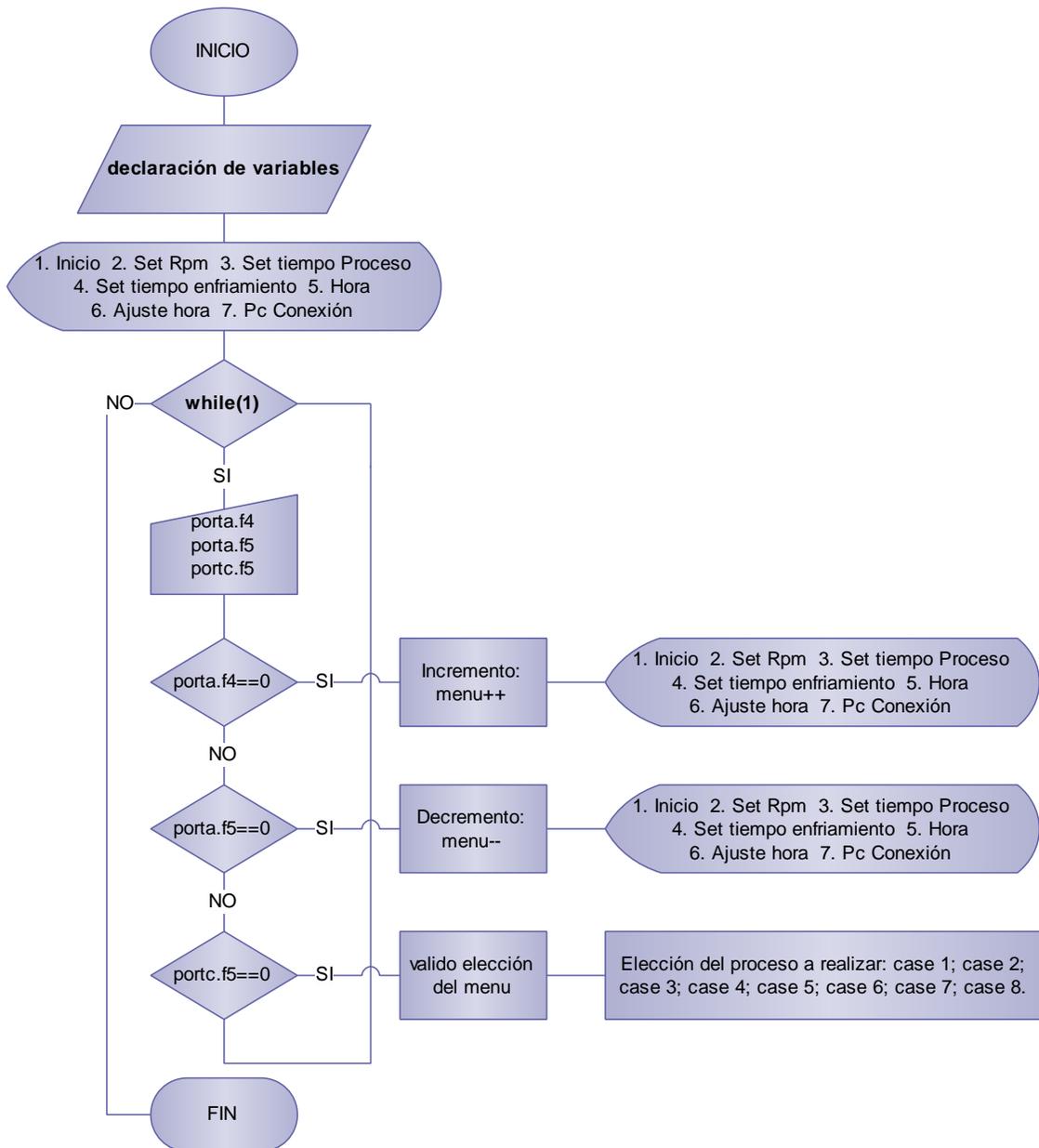


Figura 3.62 Diagrama de flujo del programa principal

Dentro de cada función que ejecuta el programa se encuentra una programación diferente, no se dará una explicación detallada de cada evento realizado en los “case”, pero se puede ver el código fuente realizado en el programa **mikroC PRO PIC 4.60** en el ANEXO B. Se opta por mostrar el resultado del mismo que fue simulado en el programa **PROTEUS**.

3.6.2.1.1 Funciones del reactor

Al iniciar el sistema nos pedirá que verifiquemos la temperatura a la cual va trabajar el reactor, esta temperatura debe de concordar con la que indica el controlador de temperatura, de no ser así se debería de variar la temperatura del controlador y ajustarla al valor deseado, una vez realizada esta verificación se dará en el botón “ENTER” para aceptar el valor de temperatura establecido.

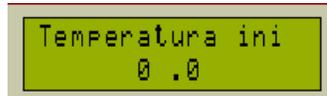


Figura 3.63 Visualización inicial del sistema

3.6.2.1.2 Inicio

La función “1. Inicio” le permite comenzar el proceso en el reactor, al aceptar esta opción da inicio con un indicador sonoro, se visualizara la fecha y hora en la que se está llevando a cabo el proceso, y esperara hasta que llegue a su valor te temperatura deseado, en este punto el indicador le mostrará el tiempo del proceso que usted ha seleccionado anteriormente y el tiempo que avance del mismo, una vez que el tiempo allá terminado el indicador le mostrará que su proceso ha sido terminado y que se iniciara el proceso de enfriamiento, el tiempo de enfriamiento se lo deberá de ajustar previamente si se requiere uno en específico, al terminar estos procesos el indicador sonoro también se activara dando por terminado.

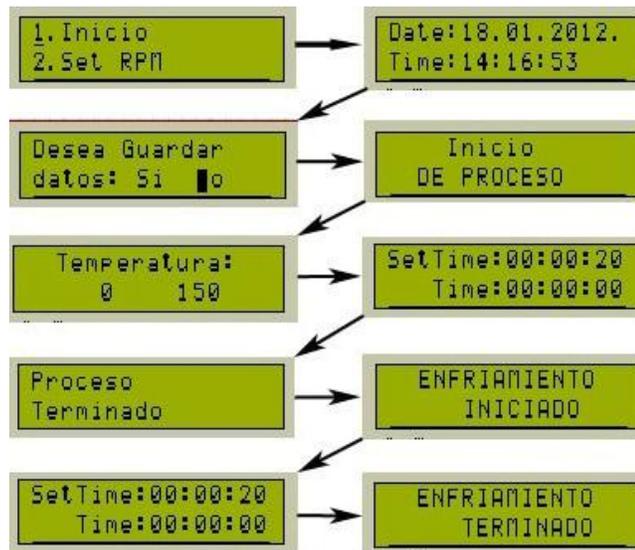


Figura 3.64 Visualización del menú inicio.

Si al validar la operación no se tiene previamente cargado el ajuste del tiempo del proceso se visualizará una pantalla de error y nos direccionará al ajuste del tiempo del proceso

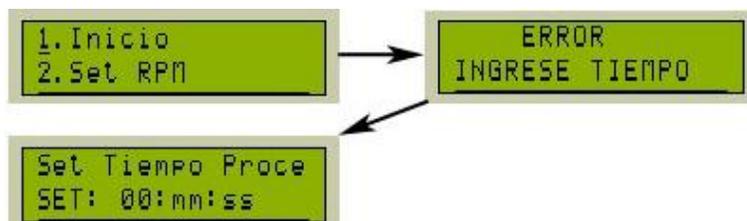


Figura 3.65 Visualización del menú inicio sin tiempo de proceso.

3.6.2.1.3 Ajustar velocidad

Para seleccionar la velocidad a la cual se requiere que funcione el reactor ingresamos en la opción “2. Set RPM”, con los controles SUBIR y BAJAR como lo dice su nombre podemos subir y bajar la velocidad del mismo hasta llegar a la velocidad deseada, en el indicador visual nos mostrara un porcentaje de la velocidad que se está ajustando y la velocidad del motor en RPM.

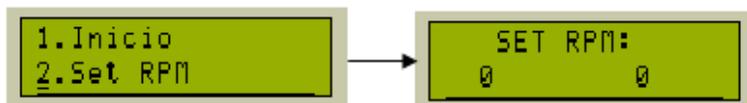


Figura 3.66 Visualización del menú set RPM.

3.6.2.1.4 Ajustar tiempo de proceso

El tiempo del proceso es el tiempo que durara el calentamiento del reactivo a una temperatura constante, en este caso como ejemplo si nuestro valor de temperatura es de 150°C, el tiempo de proceso será el tiempo que dura el sistema activo desde que la sustancia química alcanza esta temperatura, a partir de esto comenzara un conteo hasta que se cumple el tiempo establecido para finalizar el proceso e ingresar a la etapa de enfriamiento. El ajuste del mismo se lo realiza en “3. Set Tiempo Pro” en primera instancia se ajustara las horas validando esto se ajustaran los minutos y luego los segundos.



Figura 3.67 Visualización del menú Set tiempo de proceso.

3.6.2.1.5 Ajustar tiempo de enfriamiento

El tiempo de enfriamiento es el tiempo que entra a funcionar la bomba re-circulante de refrigerante para volver a la temperatura ambiente el compuesto químico, esta opción

es “4.Set Tiempo Enf”, en esta opción se ajustara primero los minutos y a continuación los segundos.



Figura 3.68 Visualización del menú Set tiempo de enfriamiento.

3.6.2.1.6 Hora

La opción “5.Hora” nos permite visualizar la fecha y hora con que se encuentra el reloj interno del reactor.



Figura 3.69 Visualización del menú hora.

3.6.2.1.7 Ajustar hora

La opción “6.Hora” nos permite ajustar los datos del reloj de tiempo real del sistema.



Figura 3.70 Visualización del menú Ajustar hora.

3.6.2.1.8 Conexión al PC

Esta función nos permite trabajar conjuntamente con el computador, recopilar información guardada en el reactor, o simplemente trabajar con todos los comandos anteriores por medio del computador.



Figura 3.71 Visualización del menú conexión PC.

3.6.2.2 Programa del controlador de velocidad.

Como se expuso en el punto 3.4 de esta sección, el control de velocidad que será implementado en un microcontrolador 16f648a es un PWM, (ver ANEXO C), a continuación se muestra el diagrama de flujo del control de velocidad.

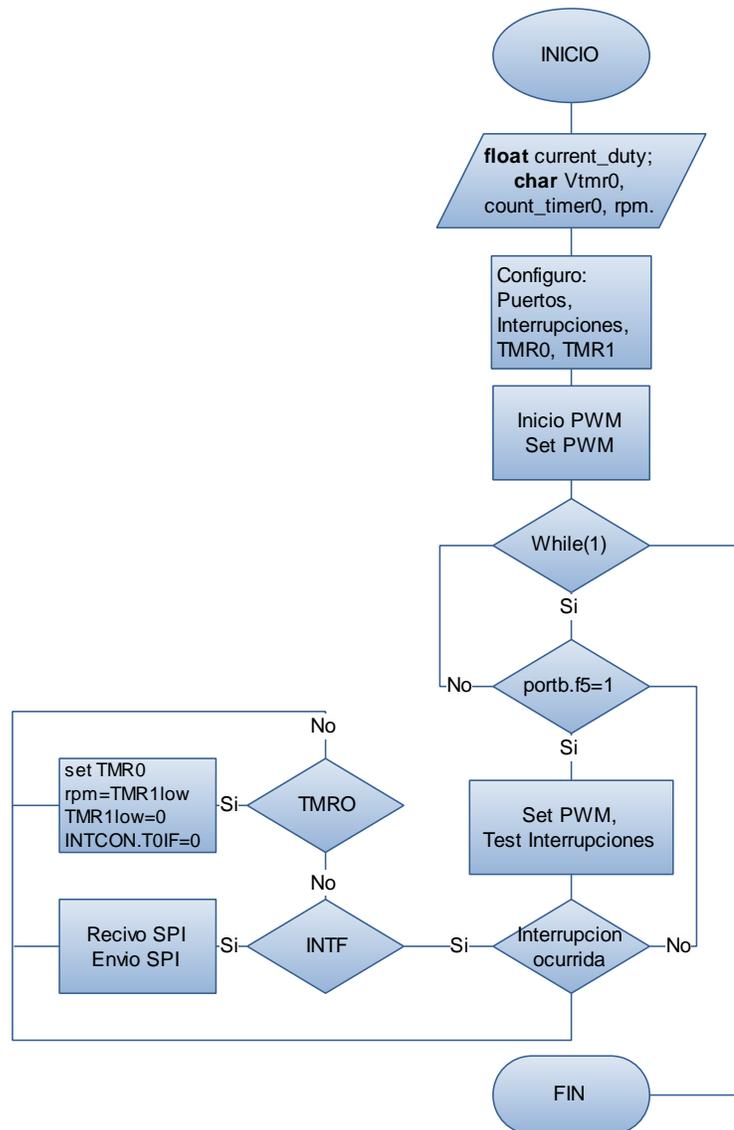


Figura 3.72 Diagrama de flujo del controlador de velocidad.

Al ser un PWM el que se utiliza, tenemos un control del motor por medio de frecuencia (como se expuso en 3.4) puede ser variada para lograr así el control, a más se implementara un tacómetro digital dentro del controlador que nos permita conocer la velocidad a la que gira nuestro sistema.

3.6.2.3 Programa del controlador de temperatura.

Al tener los datos calculados del nuestro controlador se procede a implementar el mismo, para nuestro caso se cuenta con un controlador de temperatura de la marca DELTA modelo DTB 4824, en el cual solo se ingresan las variables calculadas para el funcionamiento del control, por ello en este punto indicaremos los diagramas de flujo de

comunicación hacia dicho controlador y de esta manera poder obtener los datos para que luego puedan ser visualizados en el computador. Esta comunicación será implementada en un microcontrolador 16f876A, que se encargara de la comunicación entre el master y el controlador DELTA, (ver ANEXO D), la implementación de esta se basó en el siguiente diagrama de flujo:

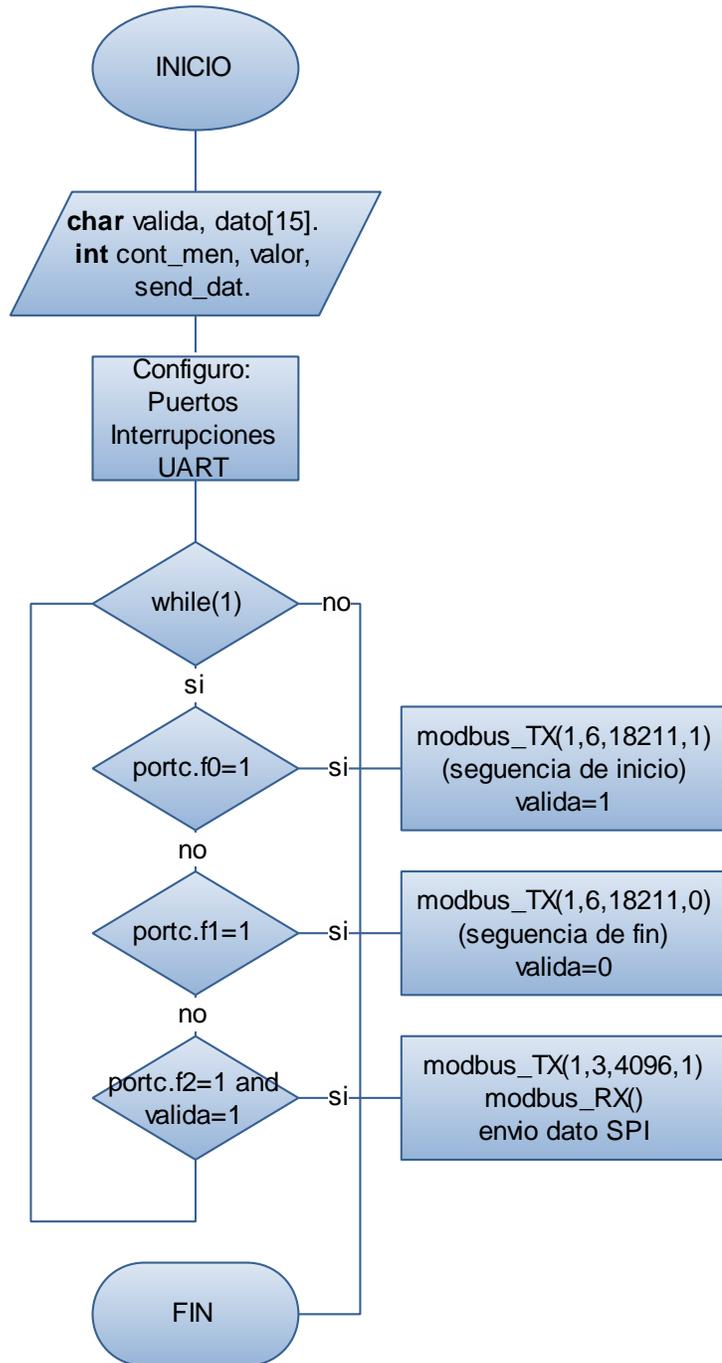


Figura 3.73 Diagrama de flujo de comunicación con el controlador

3.6.3 Hardware

3.6.3.1 Esquema del circuito electrónico.

A continuación se muestra el esquema del circuito electrónico que contiene los bloques que se indicaron en el punto 3.6.2.1, el esquema está realizado en PROTEL.

Master

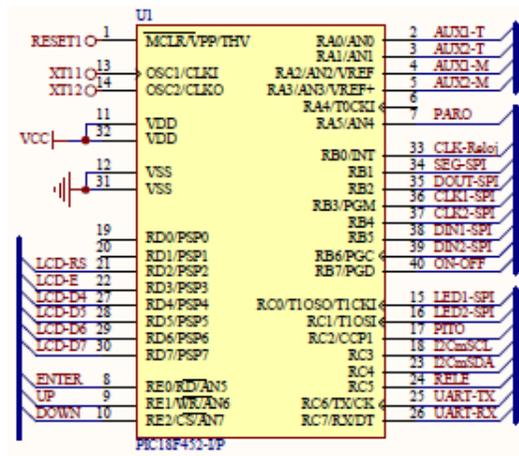


Figura 3.74 PIC-18f452 controlador master

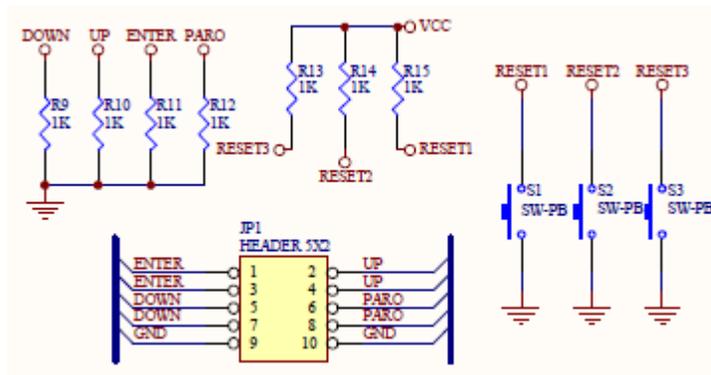


Figura 3.75 Acceso al master por pulsantes

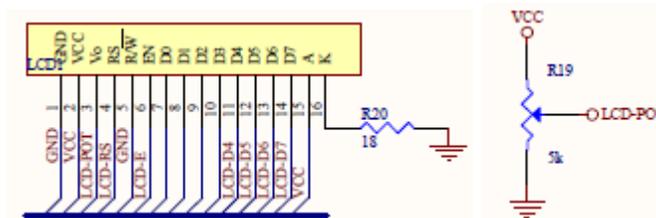


Figura 3.76 Conexión al visualizador

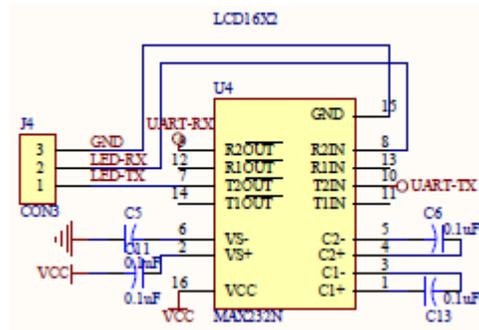


Figura 3.77 Comunicación serial al PC

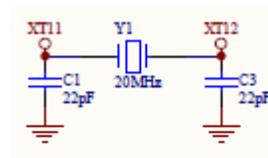


Figura 3.78 Oscilador del microcontrolador

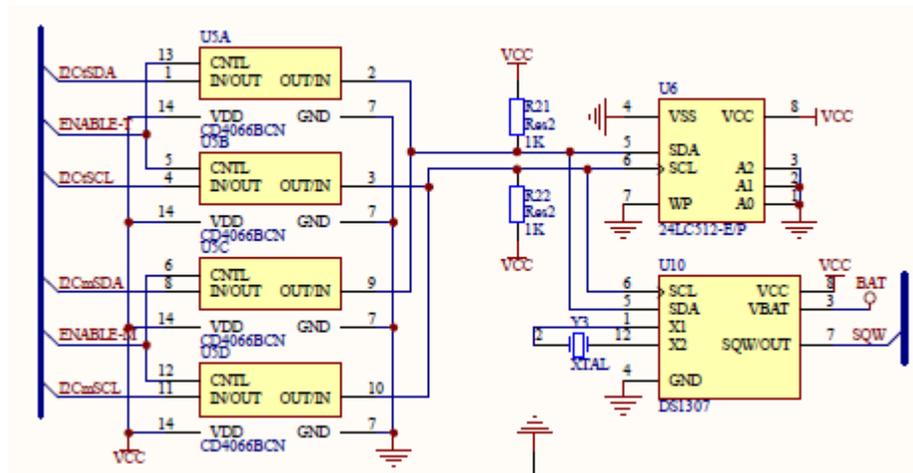


Figura 3.79 Comunicación I2C con memoria y reloj de tiempo real

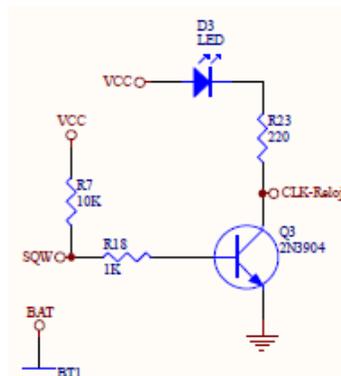


Figura 3.80 Esquema para conexión del pulso de reloj

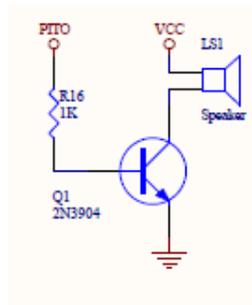


Figura 3.81 Indicador sonoro

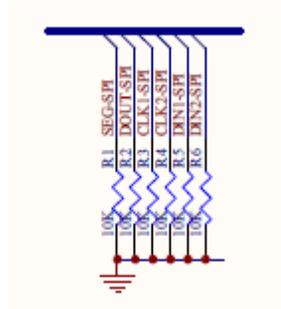


Figura 3.82 Resistencias para comunicación PSI

Esclavo Temperatura

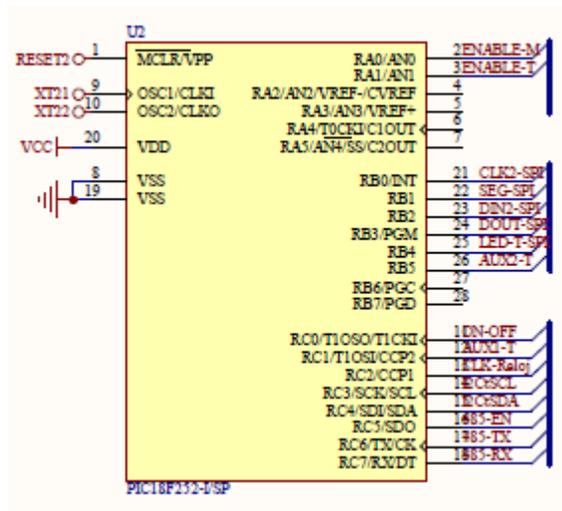


Figura 3.83 PIC-16f876A para comunicación con controlador delta

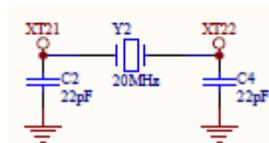


Figura 3.84 Oscilador microcontrolador

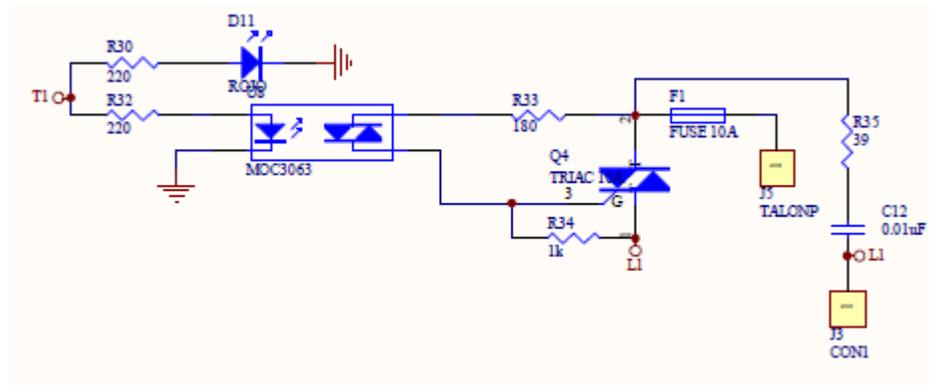


Figura 3.85 Accionador de la niquelina

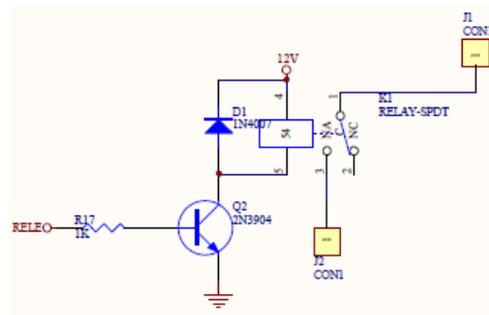


Figura 3.86 Accionador de la bomba refrigerante

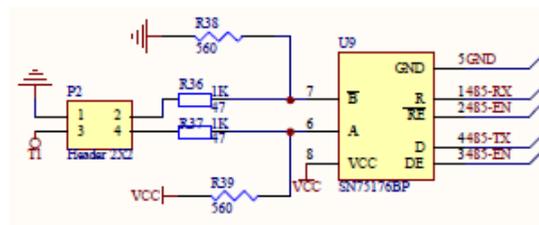


Figura 3.87 Comunicación 485 a comunicación serial

Esclavo Motor

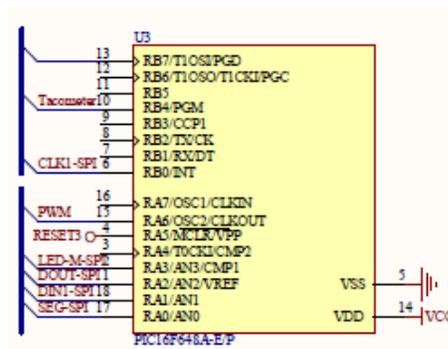


Figura 3.88 PIC-16f648A controlador de motor

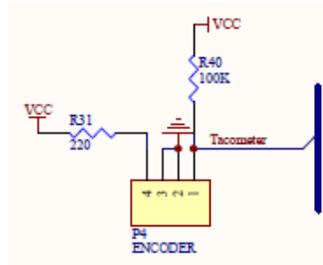


Figura 3.89 Conexión al encoder

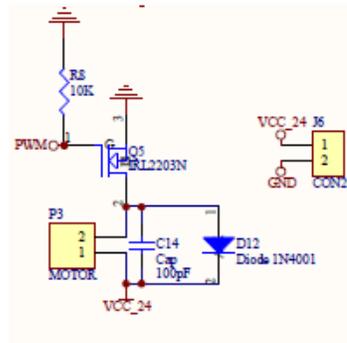


Figura 3.90 Conexión al motor

Indicadores luminosos

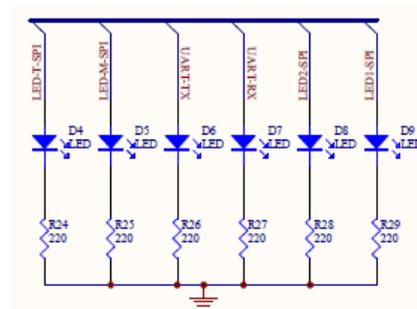


Figura 3.91 Luces LED de verificación

Estabilizador de la fuente de alimentación

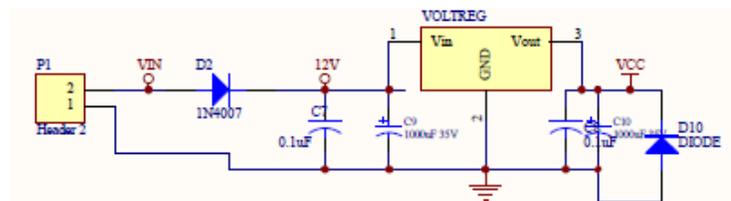


Figura 3.92 Alimentación de la tarjeta electrónica

3.6.3.2 Diseño PCB de la tarjeta electrónica.

Realizado el esquema del circuito a implementar, se procede a realizar el diseño del PCB, que es donde se soldaran todos los componentes electrónicos para el funcionamiento del control del reactor, quedando de la siguiente manera:

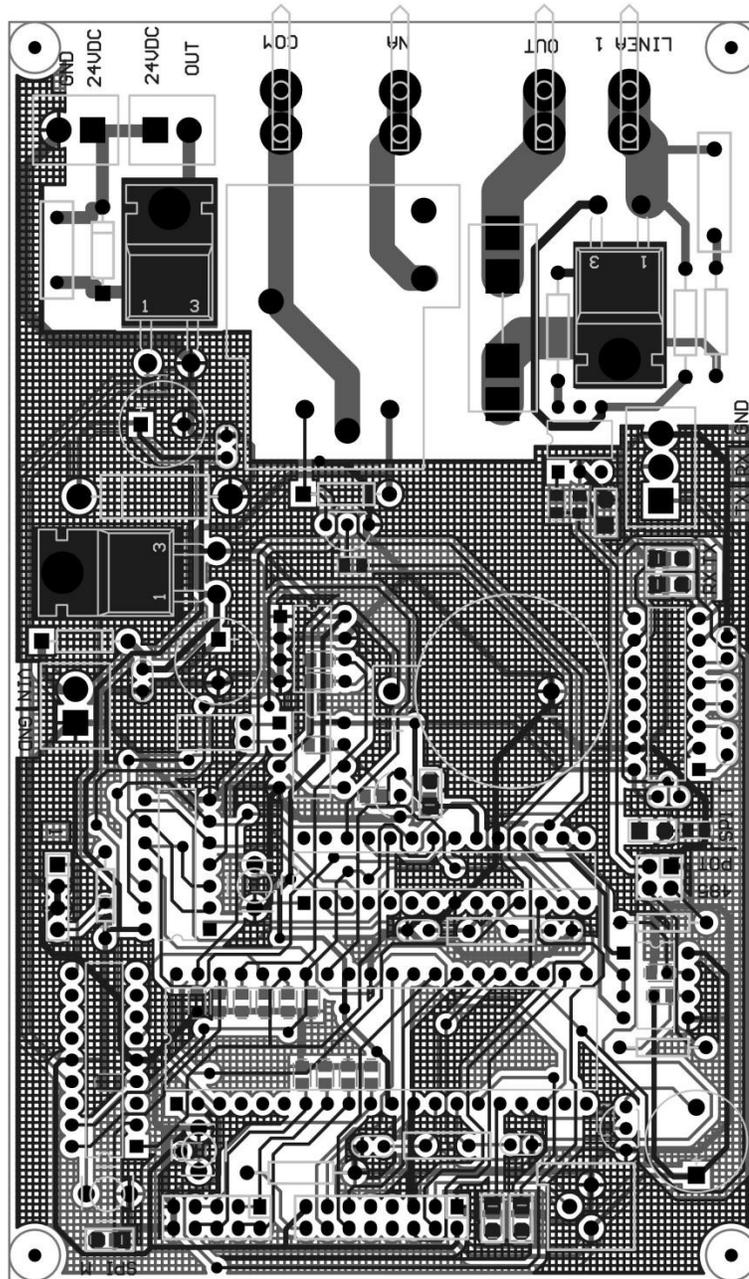


Figura 3.93 Diseño del PCB

Vistas 3D del PCB

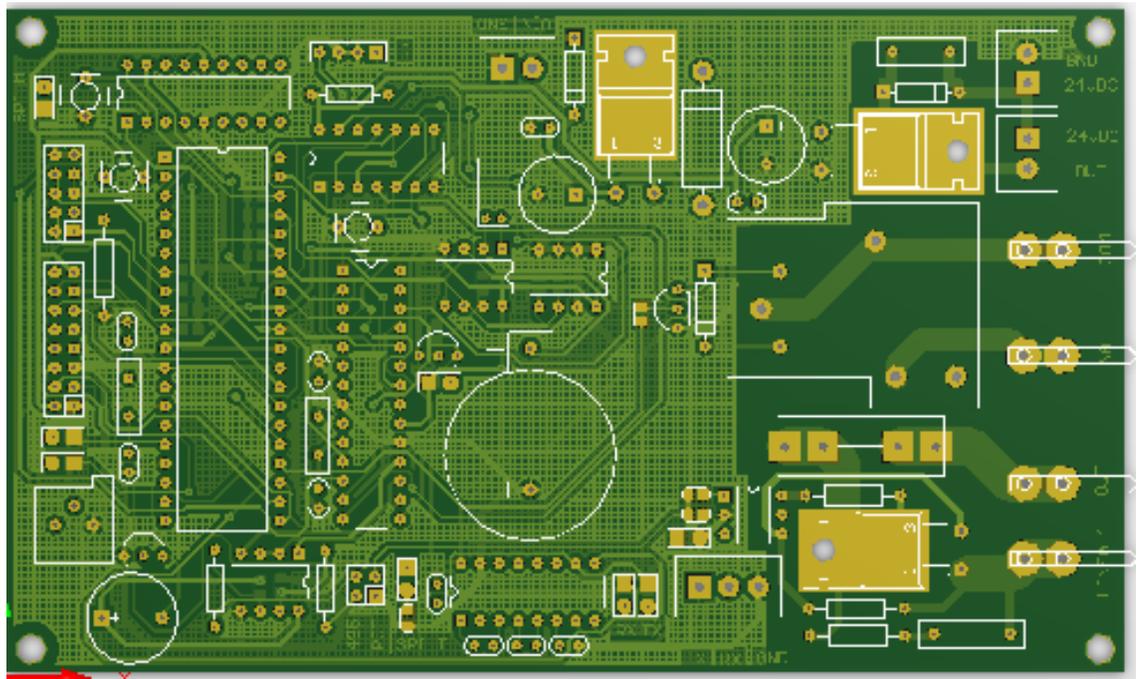


Figura 3.94 Vista de la tarjeta electrónica en 3D

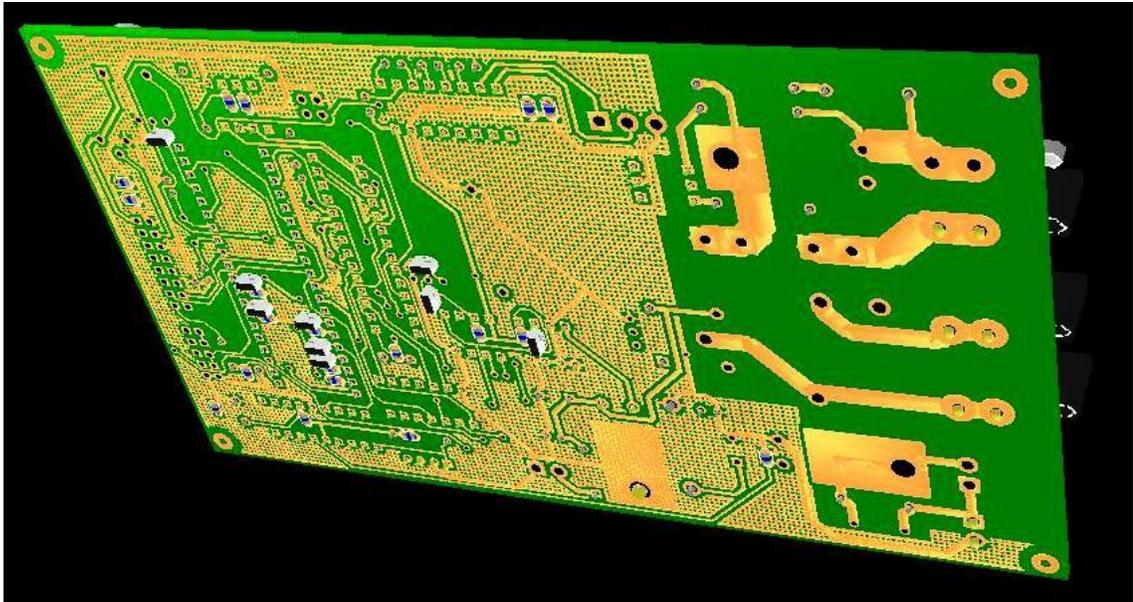


Figura 3.95 Vista Inferior de la tarjeta electrónica en 3D

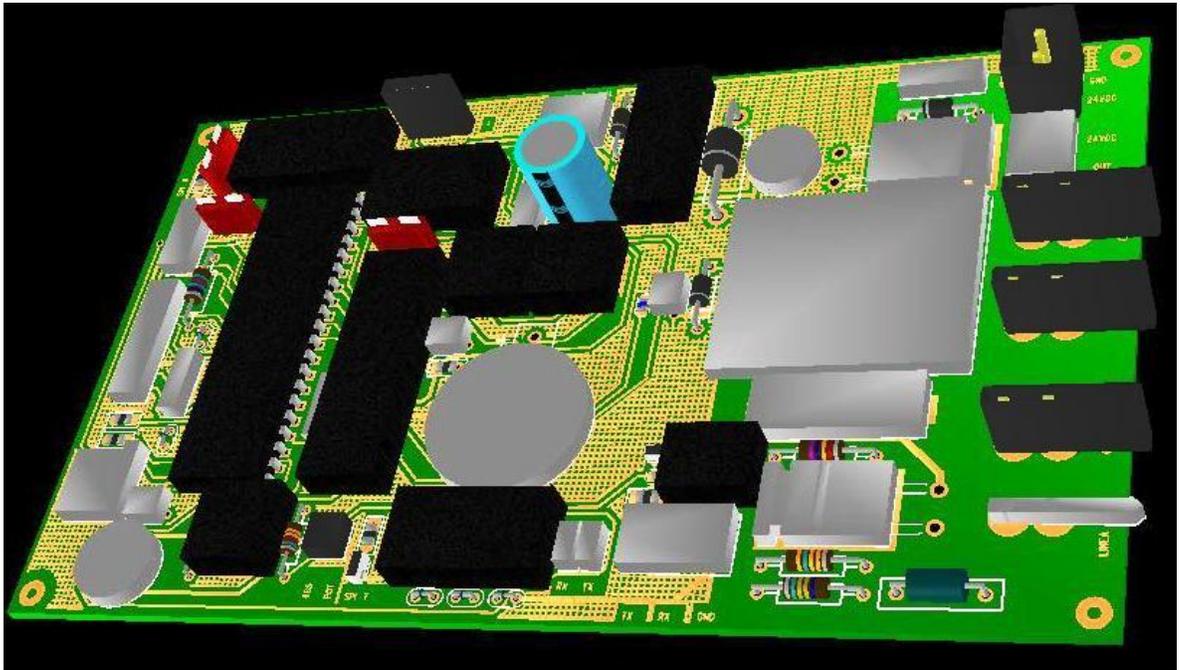


Figura 3.96 Vista superior de la tarjeta en 3D

CAPITULO IV

4 IMPLEMENTACIÓN Y PRUEBAS

Los controladores PID están presentes en el ambiente industrial, casi en exclusividad para procesos térmicos y químicos. Las constantes del control PID industrial de temperatura DELTA se encuentran en función del tiempo, para realizar esta transformación es necesario analizar algunos criterios para la implementación dentro de este dispositivo.

El control PID genera una señal de control $u(t)$ compuesta por tres términos aditivos: que son proporcionales a la señal de error $e(t)$, a su integral, y a su derivada. La ecuación del controlador PID serie (en términos de ganancia y en función del tiempo) es:

$$u(t) = K_p e(t) + K_I \int_0^t e(t) dt + K_D \frac{de(t)}{dx} \quad \text{Ecuación 4.1}$$

La ecuación 4.1 puede ser escrita en su forma ideal (parámetros de tiempo):

$$u(t) = K_p \left\{ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dx} \right\} \quad \text{Ecuación 4.2}$$

Dónde:

$$K_I = \frac{K_p}{T_i} \quad \text{Ecuación 4.3}$$

$$K_D = K_p \cdot T_d \quad \text{Ecuación 4.4}$$

La ganancia proporcional es K_p , T_i es el tiempo integral y T_d el tiempo derivativo. La función de transferencia en el dominio de Laplace es:

$$PID(s) = \frac{U(s)}{E(s)} = K_p \left\{ e(t) + \frac{1}{T_i s} + T_d s \right\} \quad \text{Ecuación 4.5}$$

El diagrama de bloques para el control PID ideal para la entrada de error $e(t)$ y la salida de control $u(t)$ es:

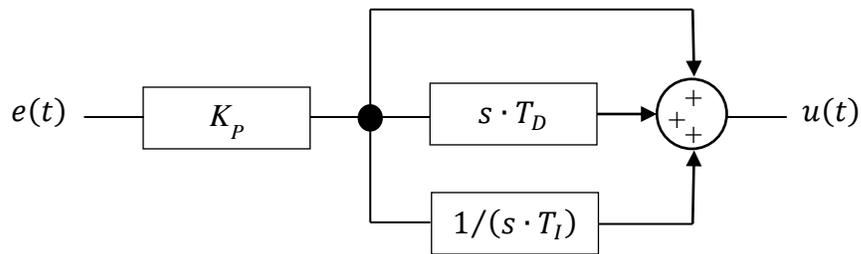


Figura 4.1 Diagrama de bloques del controlador PID ideal

El control PID puede comportarse como un control P, PI o PD al manipular las constantes de la siguiente manera:

- Un control tipo **P** cancelando tiempo derivativo $T_D = 0$ y el integral $T_I = +\infty$.
- Control **PI** anulando el tiempo derivativo $T_D = 0$.
- Para obtener el tipo **PD** se hace $T_I = +\infty$.

La constante proporcional en el ambiente de control industrial se representa por la conocida “Banda proporcional” que se refiere a un cambio porcentual en la variable controlada que produce un movimiento completo en el elemento final de control.

$$\%BP = \frac{100}{K_p} \quad \text{Ecuación 4.6}$$

En el capítulo 3 se encuentran las constantes calculadas del controlador PID:

$$K_p = 128.8174 \quad \text{Ecuación 4.7}$$

$$K_D = 6440.87 \quad \text{Ecuación 4.8}$$

Ya que el controlador calculado es del tipo PD tenemos:

$$K_I = 0 \quad \text{Ecuación 4.9}$$

Según la ecuación 4.4, la banda proporcional para el control de temperatura DELTA es:

$$\%BP = \frac{100}{K_p} \quad \text{Ecuación 4.10}$$

$$\%BP = \frac{100}{128.8174} \quad \text{Ecuación 4.11}$$

$$\%BP = 0.7762 \quad \text{Ecuación 4.12}$$

El tiempo integral se calcula por medio de la ecuación 4.4, con $K_I = 0$ (debido que el controlador es del tipo PD):

$$T_i = \frac{K_p}{K_I} \quad \text{Ecuación 4.13}$$

$$T_i = \frac{128.8174}{0} \quad \text{Ecuación 4.14}$$

Para eliminar la indeterminación, hallamos el límite cuando K_I tiende a cero:

$$T_i = \lim_{K_I \rightarrow 0} \frac{128.8174}{K_I} \quad \text{Ecuación 4.15}$$

$$T_i = +\infty \quad \text{Ecuación 4.16}$$

Al no permitirnos la configuración del controlador un valor “infinito”, lo que se hace es poner en la constante T_I el valor más alto disponible, en este caso $T_I = 9999$.

La constante de tiempo derivativo según la ecuación 4.5

$$T_d = \frac{K_D}{K_p} \quad \text{Ecuación 4.17}$$

$$T_d = \frac{6440.87}{128.8174} \quad \text{Ecuación 4.18}$$

$$T_d = 50 \quad \text{Ecuación 4.19}$$

4.1 Pruebas experimentales del comportamiento del sistema compensado.

Con los cálculos y simulaciones necesarias de los controladores realizados, el siguiente paso es realizar una serie de pruebas que nos permitan conocer los errores y aciertos que se dan en el control del sistema del reactor químico (control de temperatura y agitación del compuesto) para de esta manera, si algo falla, poder corregirlo, o a su vez realizar el ajuste necesario en los controladores tratado en el punto 4.2. A continuación se indicarán todas las pruebas realizadas del funcionamiento del reactor de una manera gráfica. Al principio se encuentran las pruebas realizadas en el primer modelo experimental (que se lo verá en el punto 4.4), el mismo que fue mejorando en la medida que surgieron errores, tomándolos en cuenta y siempre manteniendo las características necesarias que se plantearon, hasta llegar al modelo final.

4.1.1 Pruebas del controlador de velocidad.

El control de velocidad para la agitación de los compuestos está realizada por PWM (Capítulo 3), y la medición de velocidad a través de un encoder digital. Para comprobar la medición correcta de la velocidad del motor se contrasto con un tacómetro digital facilitado por los Laboratorios de la Universidad Politécnica Salesiana.



Figura 4.2 Contraste de velocidad con tacómetro digital.

4.1.2 Pruebas para el ajuste del controlador de temperatura.

A continuación se encuentran las pruebas experimentales realizadas en el reactor químico, las mismas que nos centraremos en la temperatura como se mencionó anteriormente es la parte más crítica. En cada una de ellas se adjunta la gráfica de temperatura (grados centígrados) medida en la reacción en función del tiempo (segundos), la temperatura de referencia y los parámetros PID que son:

- Banda proporcional “BP%”. En el control de temperatura delta se la encuentra como “P_O”.
- Constante de tiempo derivativo “T_d”.
- Constante de tiempo integral “T_i”.

Cada una de las pruebas servirá para el posterior ajuste del controlador. Se considerarán los mejores parámetros que tengan como resultado la mejor estabilidad en el control de la temperatura.

4.1.2.1 Prueba 1

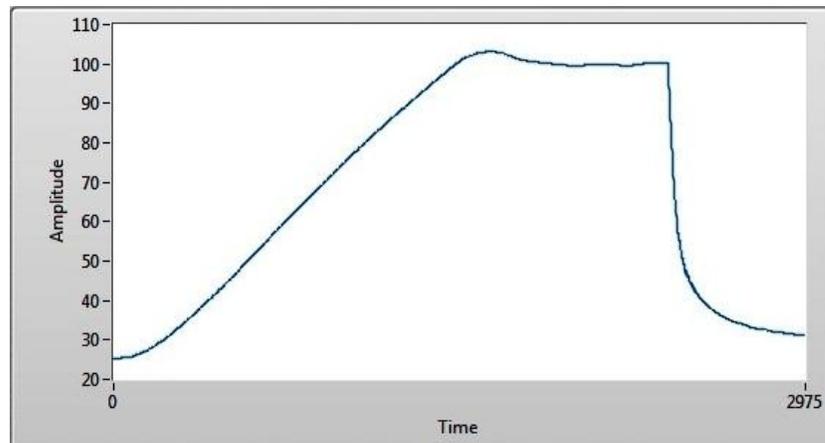


Figura 4.3 Prueba 1, ajustes del controlador.

- Temperatura de referencia = $100,0^{\circ}\text{C}$
- $P_O = 0,5$
- $T_i = 9999$
- $T_d = 50$

4.1.2.2 Prueba 2

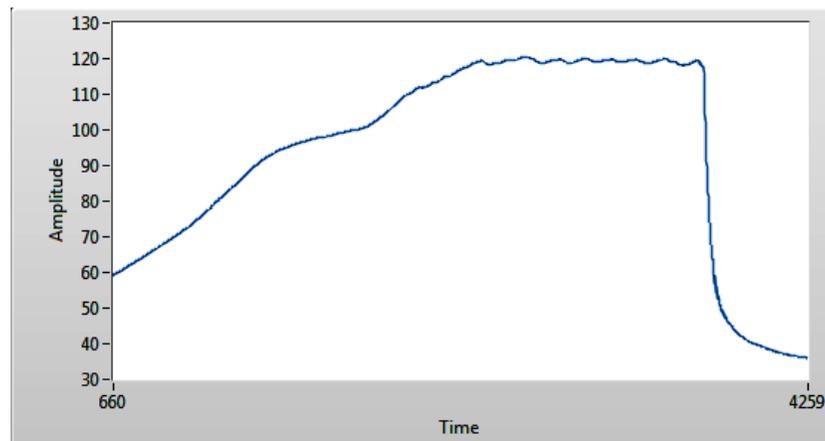


Figura 4.4 Prueba 2, ajustes del controlador.

- Temperatura de referencia = $120,0^{\circ}\text{C}$
- $P_O = 0,5$
- $T_i = 9999$
- $T_d = 50$

4.1.2.3 Prueba 3

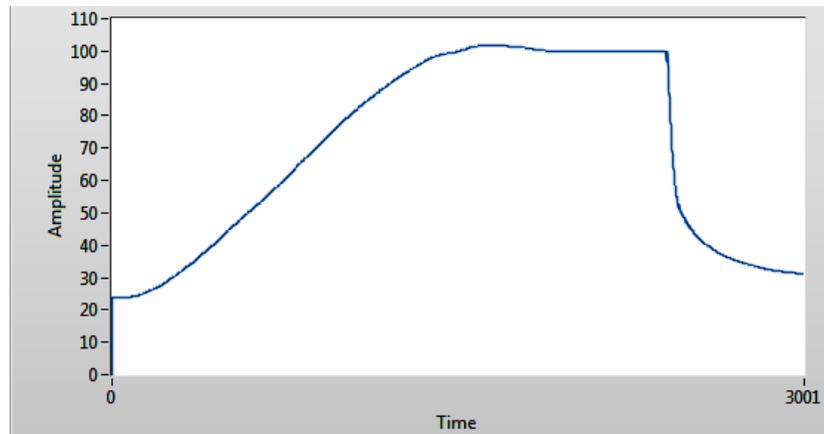


Figura 4.5 Prueba 3, ajustes del controlador.

- Temperatura de referencia = $100,0^{\circ}\text{C}$
- $P_O = 0,5$
- $T_i = 9999$
- $T_d = 50$

4.1.2.4 Prueba 4

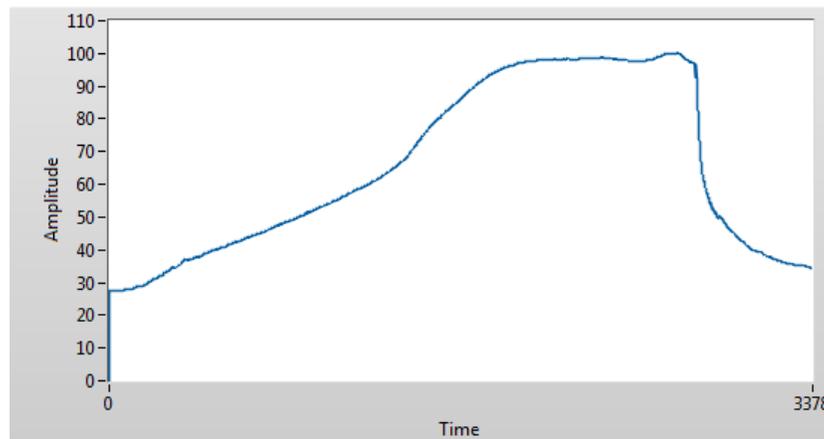


Figura 4.6 Prueba 4, ajustes del controlador.

- Temperatura de referencia = $100,0^{\circ}\text{C}$
- $P_O = 0,7$
- $T_i = 9999$
- $T_d = 50$

4.1.2.5 Prueba 5

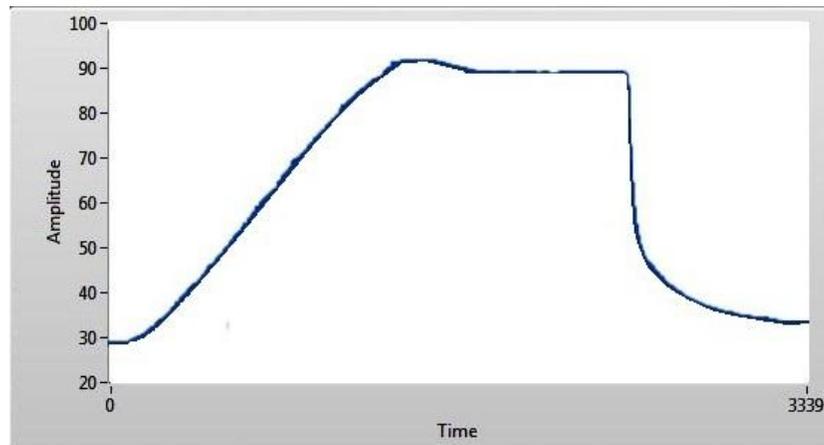


Figura 4.7 Prueba 5, ajustes del controlador.

- Temperatura de referencia = $90,0^{\circ}\text{C}$
- $P_O = 0,7$
- $T_i = 9999$
- $T_d = 50$

4.1.2.6 Prueba 6

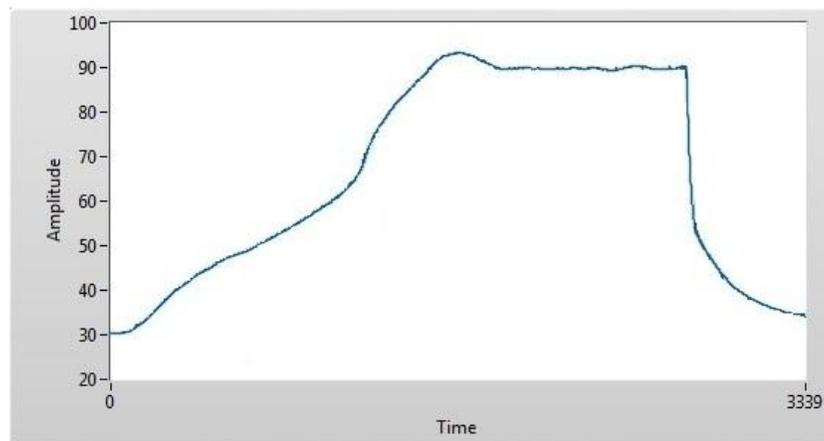
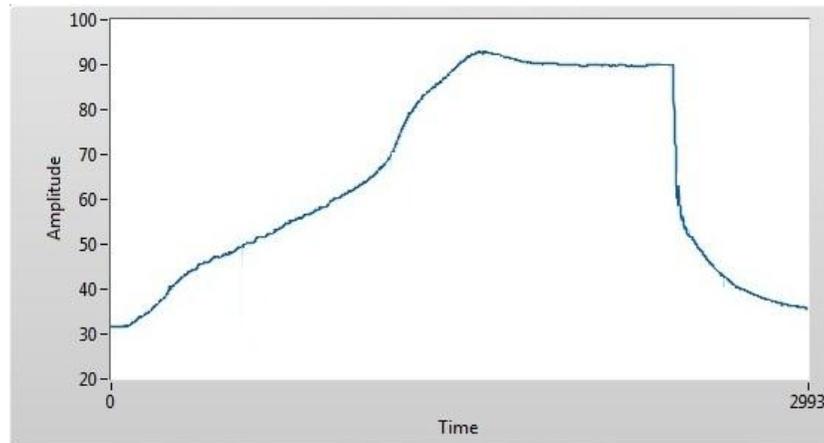
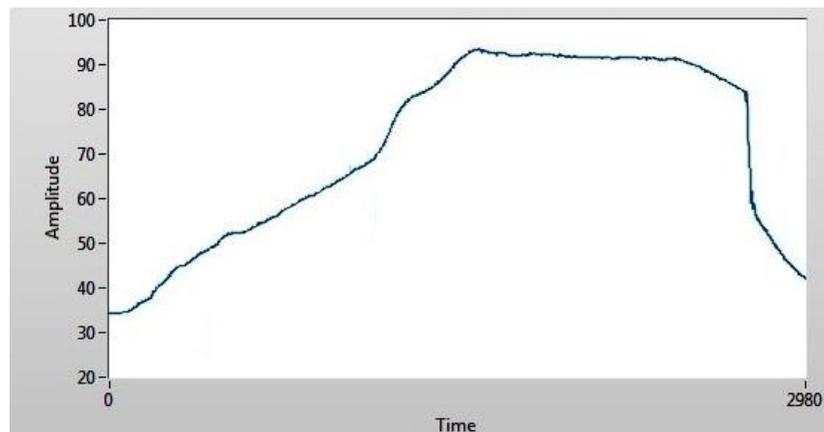


Figura 4.8 Prueba 6, ajustes del controlador.

- Temperatura de referencia = $90,0^{\circ}\text{C}$
- $P_O = 0,6$
- $T_i = 9999$
- $T_d = 40$

4.1.2.7 Prueba 7**Figura 4.9 Prueba 7, ajustes del controlador.**

- Temperatura de referencia = $90,0^{\circ}\text{C}$
- $P_O = 0,6$
- $T_i = 9999$
- $T_d = 70$

4.1.2.8 Prueba 8**Figura 4.10 Prueba 8, ajustes del controlador.**

- Temperatura de referencia = $90,0^{\circ}\text{C}$
- $P_O = 0,6$
- $T_i = 9999$
- $T_d = 150$

4.1.2.9 Prueba 9

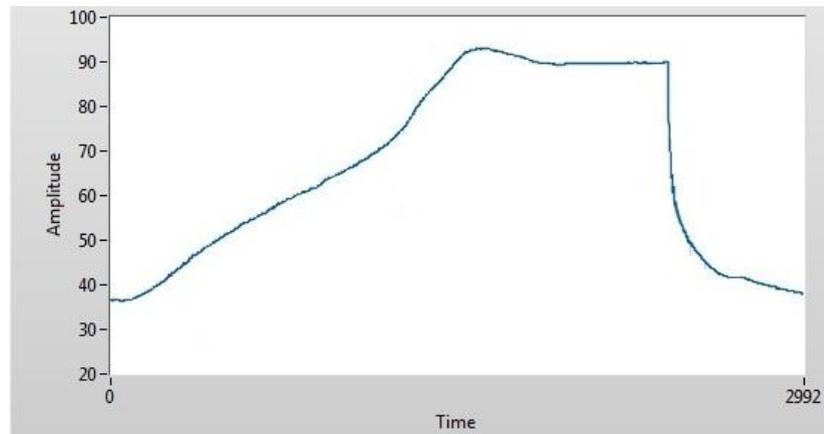


Figura 4.11 Prueba 9, ajustes del controlador.

- Temperatura de referencia = $90,0^{\circ}\text{C}$
- $P_O = 1,5$
- $T_i = 9999$
- $T_d = 50$

4.1.2.10 Prueba 10

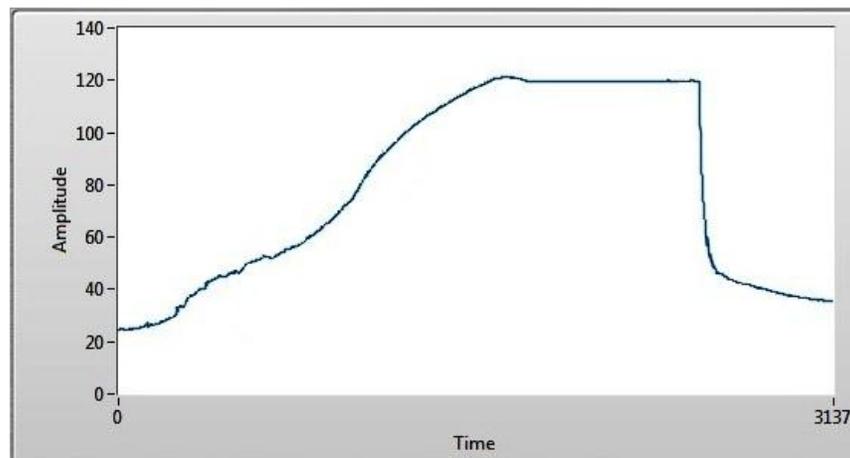
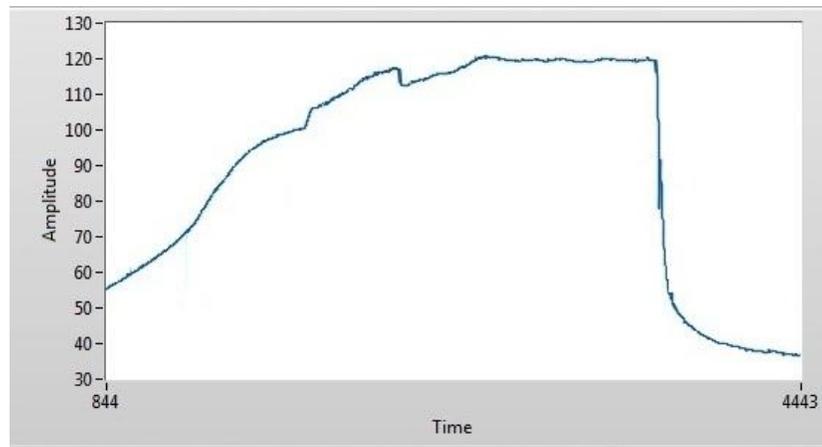
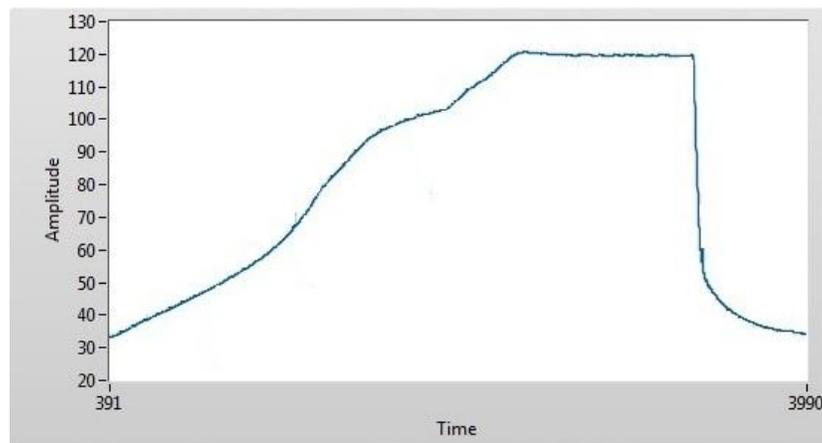


Figura 4.12 Prueba 10, ajustes del controlador.

- Temperatura de referencia = $120,0^{\circ}\text{C}$
- $P_O = 0,6$
- $T_i = 9999$
- $T_d = 50$

4.1.2.11 Prueba 11**Figura 4.13 Prueba 11, ajustes del controlador.**

- Temperatura de referencia = 120,0°C
- $P_O = 0,6$
- $T_i = 9999$
- $T_d = 50$

4.1.2.12 Prueba 12**Figura 4.14 Prueba 12, ajustes del controlador.**

- Temperatura de referencia = 120,0 °C
- $P_O = 0,4$
- $T_i = 9999$
- $T_d = 50$

4.1.3 Pruebas para la obtención del 5-Hidroxiacetilfulfural

Las pruebas se realizaron en el laboratorio del VLIR³¹ de la Universidad de Cuenca mediante un protocolo diseñado por el estudiante de Ingeniería Química William Andrés Mejía Galarza para la obtención del 5-Hidroxiacetilfulfural el cual restringe a cada uno de los experimentos a una temperatura y un tiempo en el estado estable diferente.

En cada prueba se adjunta la gráfica de temperatura (grados centígrados) medida en la reacción en función del tiempo (segundos), una descripción detallada de la evolución de cada experimentos, los problemas inesperados que surgieron y una síntesis de los resultados obtenidos.



Figura 4.15 Reactor químico experimental tipo batch.

³¹ Para mayor información visitar: <http://vlir-iuc.ucuenca.edu.ec/inicio.ks>



Figura 4.16 Fase acuosa reactiva

La base del compuesto de 18mL de fase acuosa reactiva estaba compuesta por:

- 175g de NaCl
- 25g de Fructosa
- 12,31g de HCl (o medir un volumen equivalente de 10,43 mL).

4.1.3.1 Prueba 1

Como se observa en la Figura 4.17 Prueba 1 en laboratorio, control de temperatura el control fue relativamente estable en el sistema, llegando a una temperatura de 110°C. El sistema de agitación no sufrió ningún tipo de anomalía y el sistema de enfriamiento fue correcto.

- Temperatura de referencia = 110.00 °C
- Temperatura máxima durante el estado estable = 110,40 °C
- Temperatura mínima durante el estado estable = 109,40 °C
- Variación de temperatura = -0,60 °C / +0,40 °C
- Tiempo de duración del experimento = 30 minutos 50 segundos 24 milisegundos
- Tiempo de inicio del estado estable = 15,80 minutos
- Tiempo de inicio del enfriamiento = 18,68 minutos
- Tiempo de duración del experimento en estado estable = 2,88 minutos

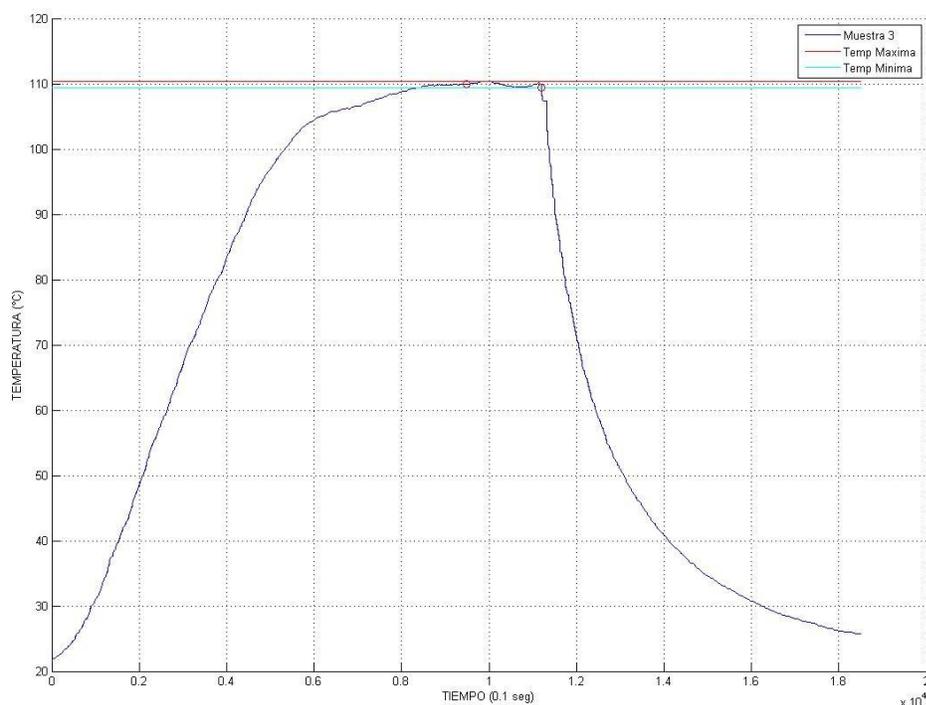


Figura 4.17 Prueba 1 en laboratorio, control de temperatura

4.1.3.2 Prueba 2

La Figura 4.18 muestra como la temperatura decae en el momento de alcanzar el valor de 110°C y no se mantiene estable debido a que surgieron problemas e imprevistos debido a fugas en el reactor químico, principalmente en la niquelina ya que el ácido incluido en la fase acuosa reactiva corroyó sus cables de alimentación. Se hicieron algunas pruebas antes de iniciar el experimento, pero este fallo.

- Temperatura de referencia = 110,00 °C
- Temperatura máxima durante el estado estable = 109,70 °C
- Temperatura mínima durante el estado estable = 109,10 °C
- Variación de temperatura = +0,70 °C / -0,10 °C
- Tiempo de duración del experimento = 29 minutos 39 segundos 6 milisegundos
- Tiempo de inicio del estado estable = 14,06 minutos
- Tiempo de inicio del enfriamiento = 15,11 minutos
- Tiempo de duración del experimento en estable = 1,04 minutos

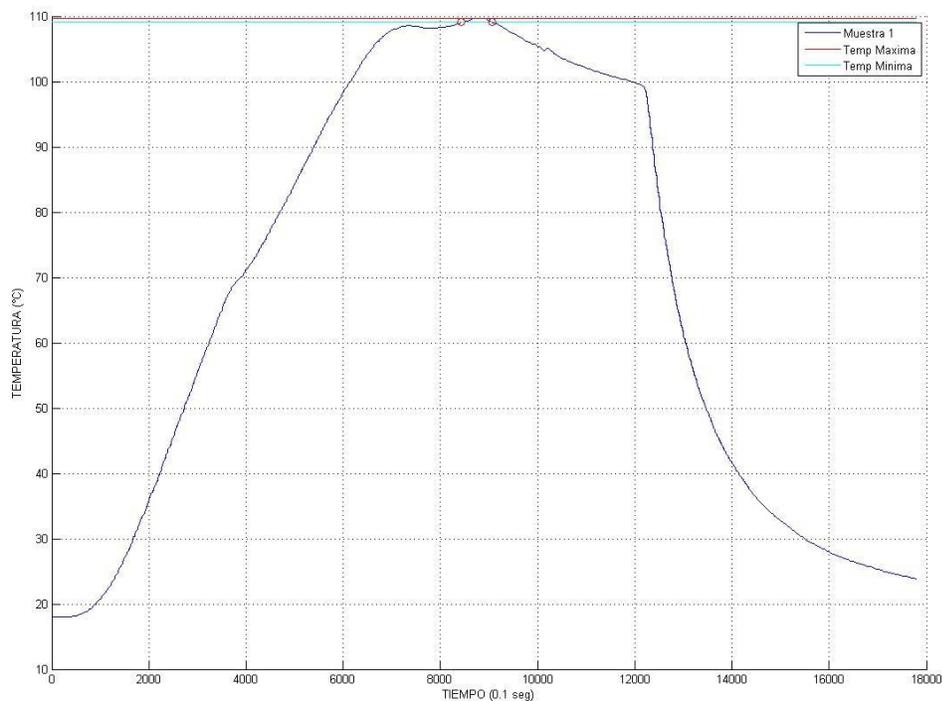


Figura 4.18 Prueba 2 en laboratorio, control de temperatura

4.1.3.3 Prueba 3

Se observa en la Figura 4.19 una gran inestabilidad del sistema en la parte del control de temperatura, esto se debe a un fallo en el control de velocidad de la agitación, por ello se decidió realizar un sistema de control separado por bloques (otro microcontrolador) para evitar estas anomalías en el sistema y que siempre esté operando de manera correcta.

- Temperatura de referencia = 110,00 °C
- Temperatura máxima durante el estado estable = 111,10 °C
- Temperatura mínima durante el estado estable = 109,10 °C
- Variación de temperatura = +1.10 °C / -0.90 °C
- Tiempo de duración del experimento = 39 minutos 57 segundos 36 milisegundos
- Tiempo de inicio del estado estable = 21.16 min
- Tiempo de inicio del enfriamiento = 25.13 min
- Tiempo de duración del experimento en estado estable= 3.97 min

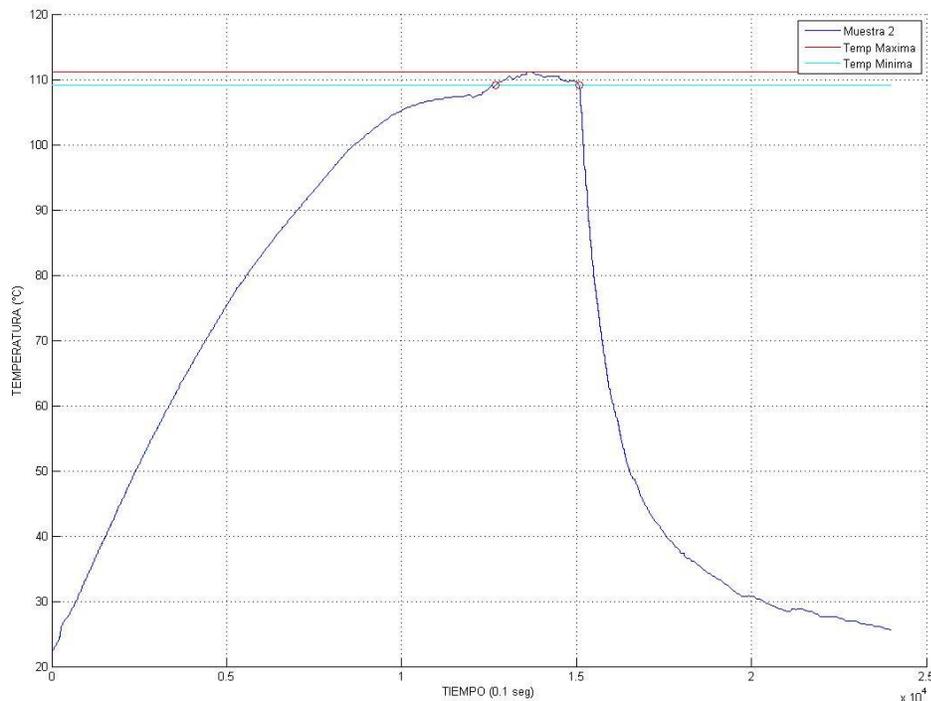


Figura 4.19 Prueba 3 en laboratorio, control de temperatura.

4.1.3.4 Prueba 4

En este caso se nota un gran error en el sistema causado por una falla en los conectores de la niquelina que no hacían el correcto contacto, estos fueron reemplazados para evitar problemas en posteriores pruebas. Como se vio en los ajustes, la niquelina como tal no presenta garantías completas de linealidad en su comportamiento, sino que es variable y modifica sus características haciendo que su comportamiento exacto sea desconocido.

- Temperatura de referencia= 110,00 °C
- Temperatura máxima durante el estado estable = 114,30 °C
- Temperatura mínima durante el estado estable = 110,00 °C
- Variación de temperatura = +4,3 °C / 0 °C
- Tiempo de duración del experimento = 55 minutos 27 segundos 48milisegundos
- Tiempo de inicio del estado estable = 42,00 minutos
- Tiempo de inicio del enfriamiento = 48,33 minutos
- Tiempo de duración del experimento en estado estable = 6,33 minutos

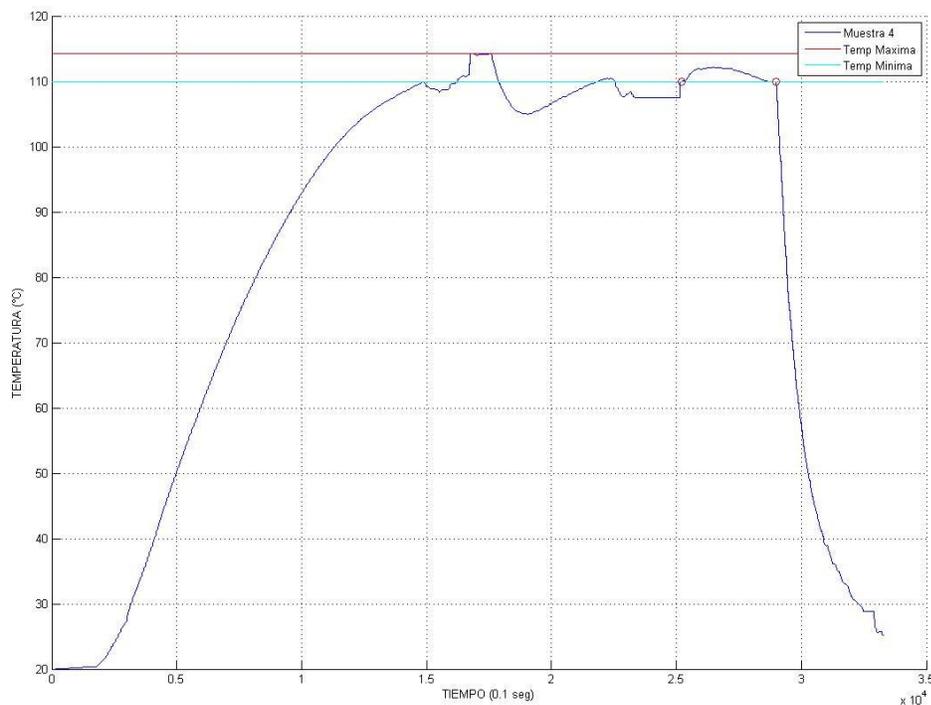


Figura 4.20 Prueba 4 en laboratorio, control de temperatura

4.1.3.5 Prueba 5

Se corrigieron todos los errores anteriormente generados en las pruebas previas, en este experimento se obtuvo un control adecuado tanto para el sistema de temperatura como el de agitación.

- Temperatura de referencia = 125,00 °C
- Temperatura máxima durante el estado estable = 126.40 °C
- Temperatura mínima durante el estado estable = 125.00 °C
- Variación de temperatura = +/- 0.70 °C
- Tiempo de duración del experimento = 44 minutos 47 segundos 54milisegundos
- Tiempo de inicio del estado estable = 31.82 minutos
- Tiempo de inicio del enfriamiento = 36.87 minutos
- Tiempo de duración del experimento en estado estable = 5.05 minutos

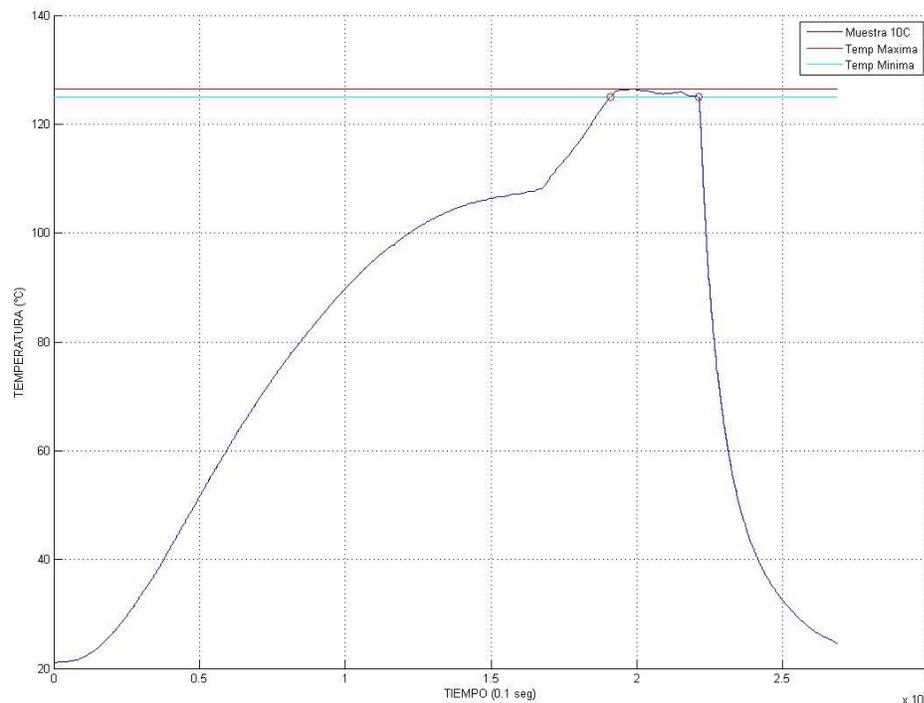


Figura 4.21 Prueba 5 en laboratorio, control de temperatura

4.1.3.6 Prueba 6

Sin ningún problema. Prueba exitosa.

- Temperatura de referencia = 140,00 °C
- Temperatura máxima durante el estado estable = 140,60 °C
- Temperatura mínima durante el estado estable = 139,80 °C
- Variación de temperatura = +0,60 °C / -0,40 °C
- Tiempo de duración del experimento = 51 minutos 9 segundos 54 milisegundos
- Tiempo de inicio del estado estable = 35,77 minutos
- Tiempo de inicio del enfriamiento = 41,97 minutos
- Tiempo de duración estable = 6,20 minutos

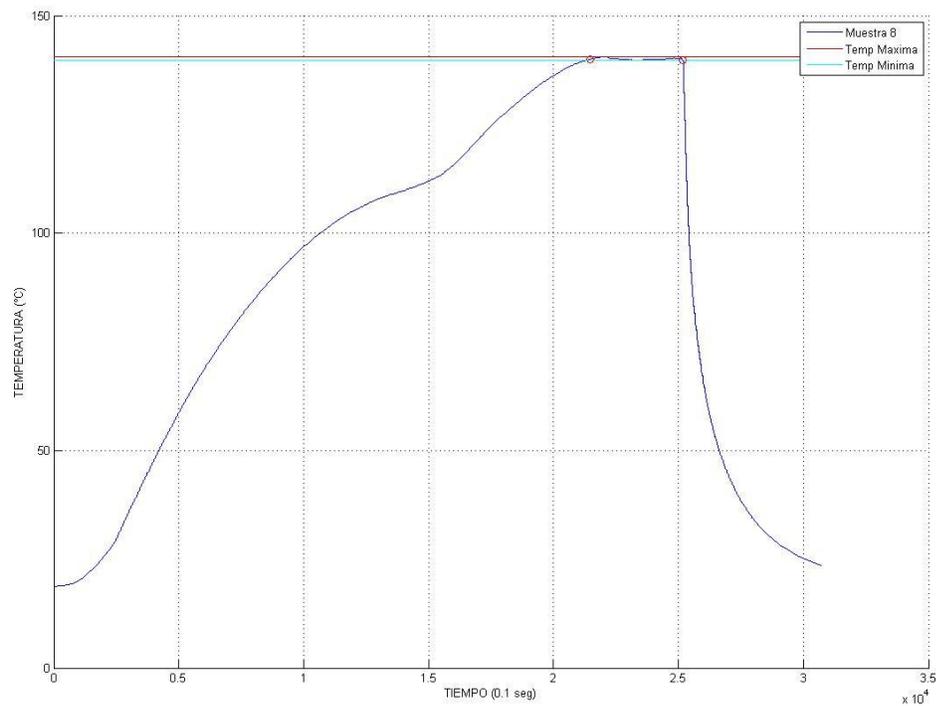


Figura 4.22 Prueba 6 en laboratorio, control de temperatura

4.1.3.7 Prueba 7

Sin ningún problema. Prueba exitosa.

- Temperatura de referencia = 140,00 °C
- Temperatura máxima durante el estado estable = 140,60 °C
- Temperatura mínima durante el estado estable = 139,60 °C
- Variación de temperatura = +0,60 °C / -0,40 °C
- Tiempo de duración del experimento = 47 minutos 45 segundos 30 milisegundos
- Tiempo de inicio del estado estable = 34,06 minutos
- Tiempo de inicio del enfriamiento = 38,44 minutos
- Tiempo de duración del experimento en estado estable = 4,38 minutos

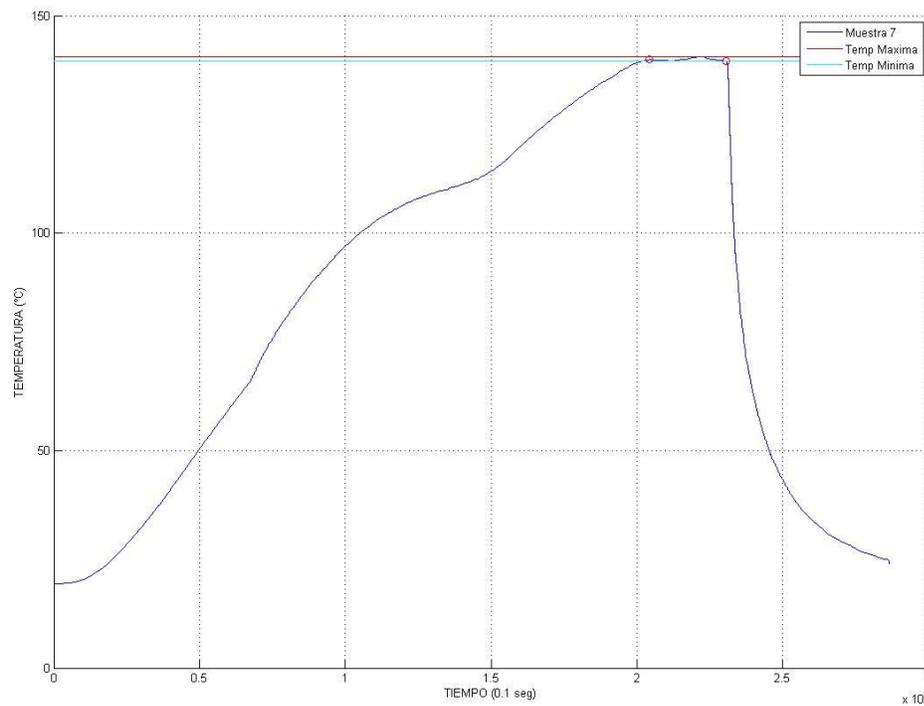


Figura 4.23 Prueba 7 en laboratorio, control de temperatura

4.1.3.8 Prueba 8

Sin ningún problema. Prueba exitosa.

- Temperatura de referencia = 140,00 °C
- Temperatura máxima durante el estado estable = 140,50 °C
- Temperatura mínima durante el estado estable = 139,70 °C
- Variación de temperatura = +0,50 °C / -0,30 °C
- Tiempo de duración del experimento = 51 minutos 33 segundos 18milisegundos
- Tiempo de inicio del estado estable = 34,40 minutos
- Tiempo de inicio del enfriamiento = 40,58 minutos
- Tiempo de duración del experimento en estado estable = 6,18 minutos

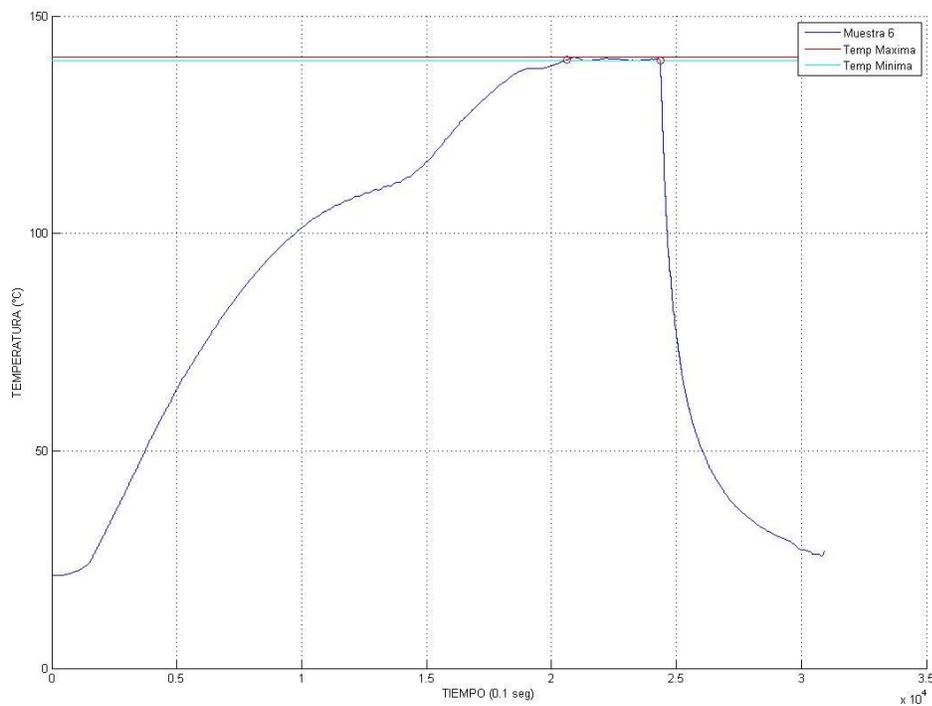


Figura 4.24 Prueba 8 en laboratorio, control temperatura

4.1.3.9 Prueba 9

Sin ningún problema. Prueba exitosa.

- Temperatura de referencia = 125,00 °C
- Temperatura máxima durante el estado estable = 125,90 °C
- Temperatura mínima durante el estado estable = 124,70 °C
- Variación de temperatura = +0,90 °C / -0,30 °C
- Tiempo de duración del experimento = 45 minutos 23 segundos 18 milisegundos
- Tiempo de inicio del estado estable = 26,46 minutos
- Tiempo de inicio del enfriamiento = 31,56 minutos
- Tiempo de duración del experimento en estado estable = 5,10 minutos

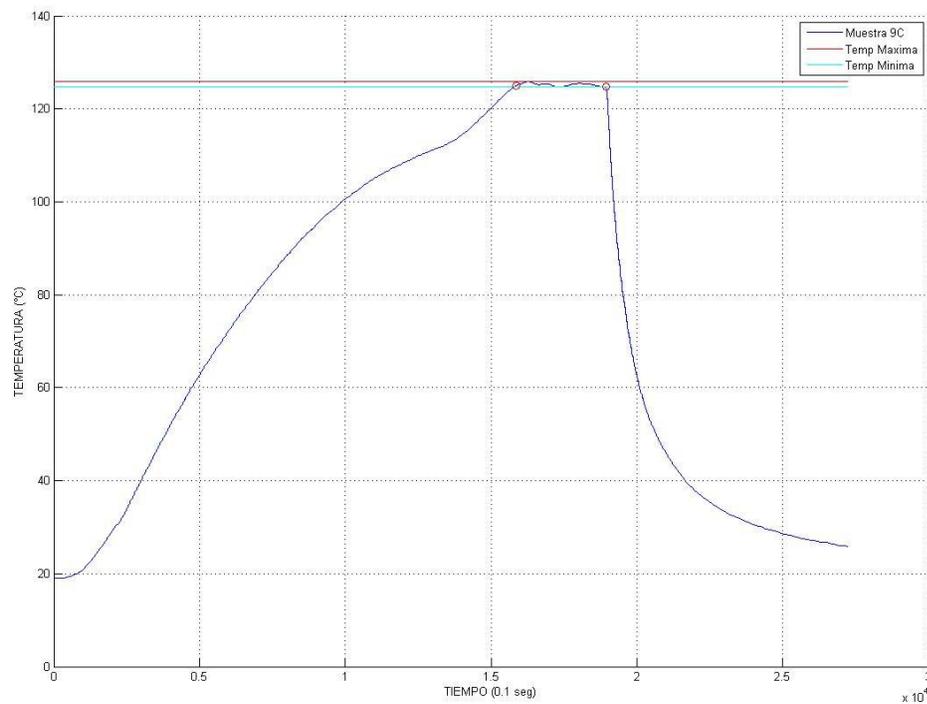


Figura 4.25 Prueba 9 en laboratorio, control de temperatura.

4.1.3.10 Prueba 10

Sin ningún problema. Prueba exitosa.

- Temperatura de referencia = 125,00 °C
- Temperatura máxima durante el estado estable = 125,30 °C
- Temperatura mínima durante el estado estable = 124,60 °C
- Variación de temperatura = +0,30 °C / -0,40 °C
- Tiempo de duración del experimento = 48 minutos 40 segundos 18 milisegundos
- Tiempo de inicio del estado estable = 31,77 minutos
- Tiempo de inicio del enfriamiento = 33,80 minutos
- Tiempo de duración del experimento en estado estable = 2,04 minutos

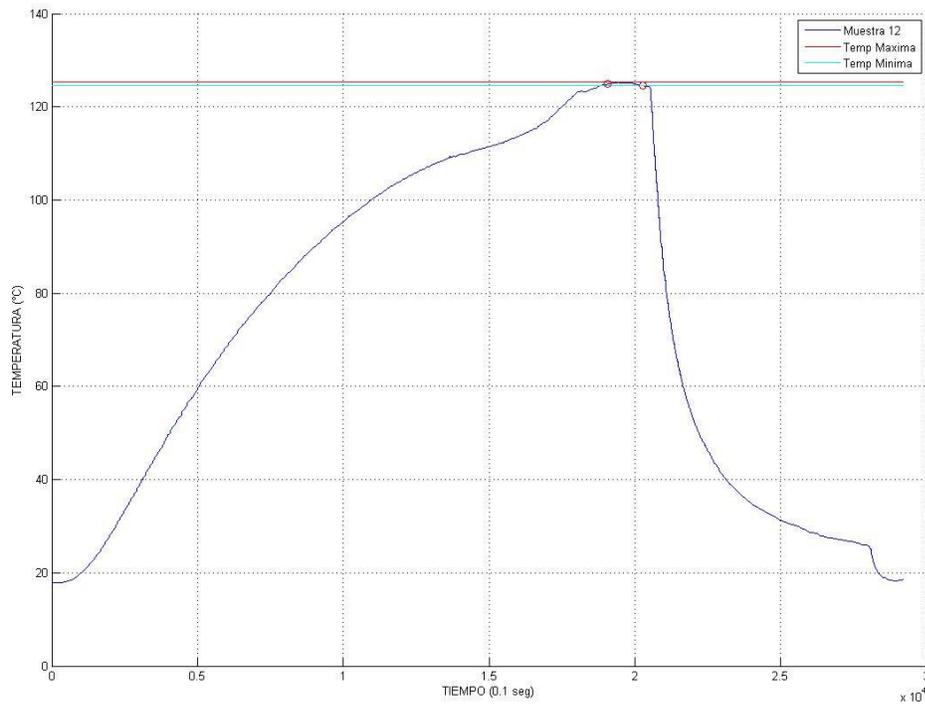


Figura 4.26 Prueba 10 en laboratorio, control de temperatura.

4.1.3.11 Prueba 11

Sin ningún problema. Prueba exitosa.

- Temperatura de referencia = 150,20 °C
- Temperatura máxima durante el estado estable = 150,60 °C
- Temperatura mínima durante el estado estable = 149,80 °C
- Variación de temperatura = +0,60 °C / -0,20 °C
- Tiempo de duración del experimento = 60 minutos 18 segundos 12 milisegundos
- Tiempo de inicio del estado estable = 42,75 minutos
- Tiempo de inicio del enfriamiento = 47,83 minutos
- Tiempo de duración del experimento en estado estable = 5,09 minutos

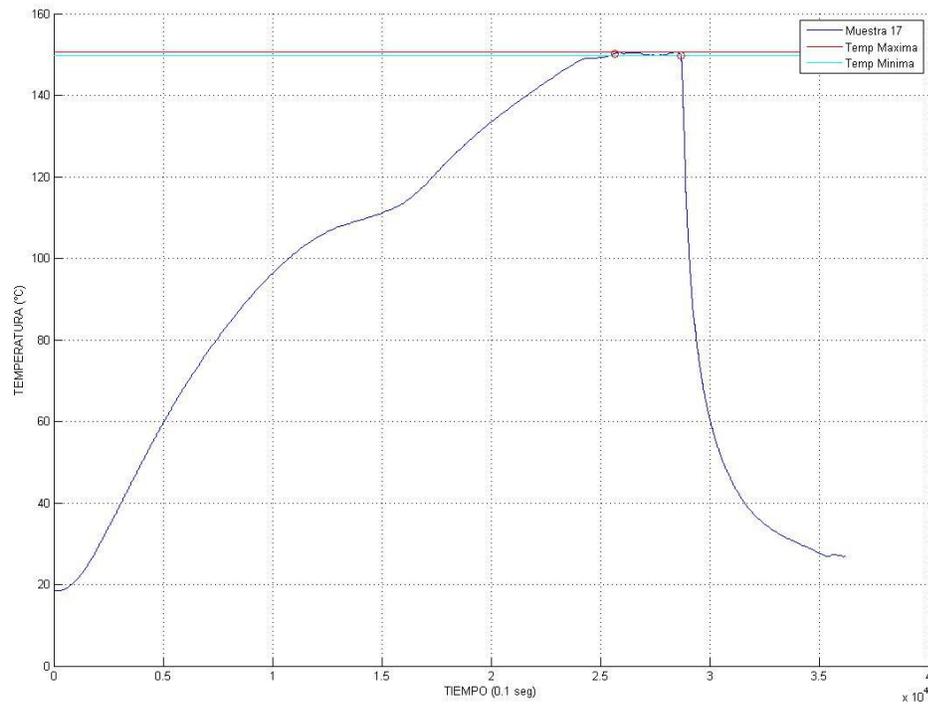


Figura 4.27 Prueba 11 en laboratorio, control de temperatura.

4.1.3.12 Prueba 12

Sin ningún problema. Prueba exitosa.

- Temperatura de referencia = 125,00 °C
- Temperatura máxima durante el estado estable = 125,20 °C
- Temperatura mínima durante el estado estable = 124,40 °C
- Variación de temperatura = +0,20 °C / -0,60 °C
- Tiempo de duración del experimento = 59 minutos 27 segundos 36 milisegundos
- Tiempo de inicio del estado estable = 39,62 minutos
- Tiempo de inicio del enfriamiento = 44,71 minutos
- Tiempo de duración del experimento en estado estable = 5,09 minutos

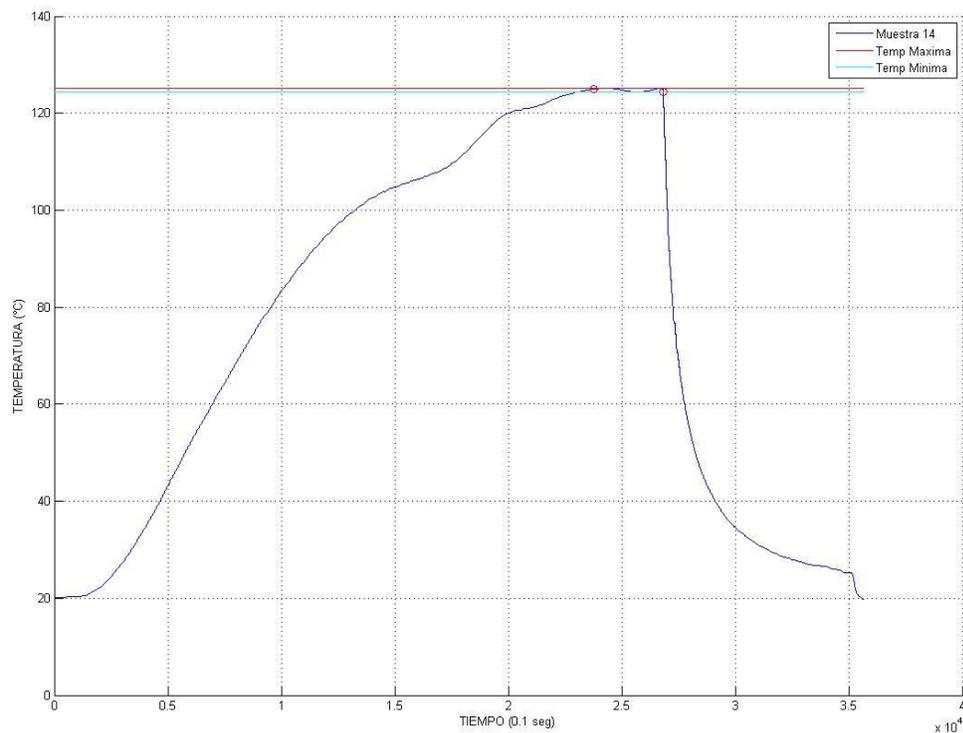


Figura 4.28 Prueba 12 en laboratorio, control de temperatura.

4.1.3.13 Prueba 13

Sin ningún problema. Prueba exitosa.

- Temperatura de referencia = 99,70 °C
- Temperatura máxima durante el estado estable = 100,10 °C
- Temperatura mínima durante el estado estable = 99,20 °C
- Variación de temperatura = +0,40 °C / -0,50 °C
- Tiempo de duración del experimento = 44 minutos 49 segundos 36 milisegundos
- Tiempo de inicio del estado estable = 23,91 minutos
- Tiempo de inicio del enfriamiento = 29,57 minutos
- Tiempo de duración del experimento en estado estable = 5,66 minutos

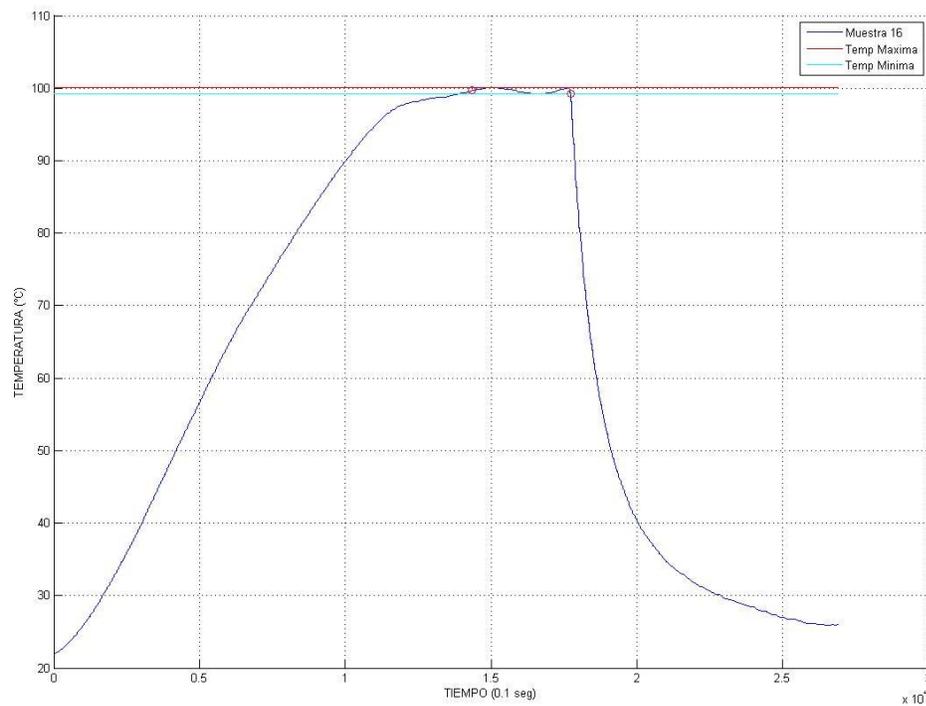


Figura 4.29 Prueba 13 en laboratorio, control de temperatura.

4.1.3.14 Prueba 14

Sin ningún problema. Prueba exitosa.

- Temperatura de referencia = 125,00 °C
- Temperatura máxima durante el estado estable = 125,70 °C
- Temperatura mínima durante el estado estable = 124,40 °C
- Variación de temperatura = +0,70 °C / -0,60 °C
- Tiempo de duración del experimento = 62 minutos 33 segundos 24 milisegundos
- Tiempo de inicio del estado estable = 31,41 minutos
- Tiempo de inicio del enfriamiento = 36,48 minutos
- Tiempo de duración del experimento en estado estable = 5,07 minutos

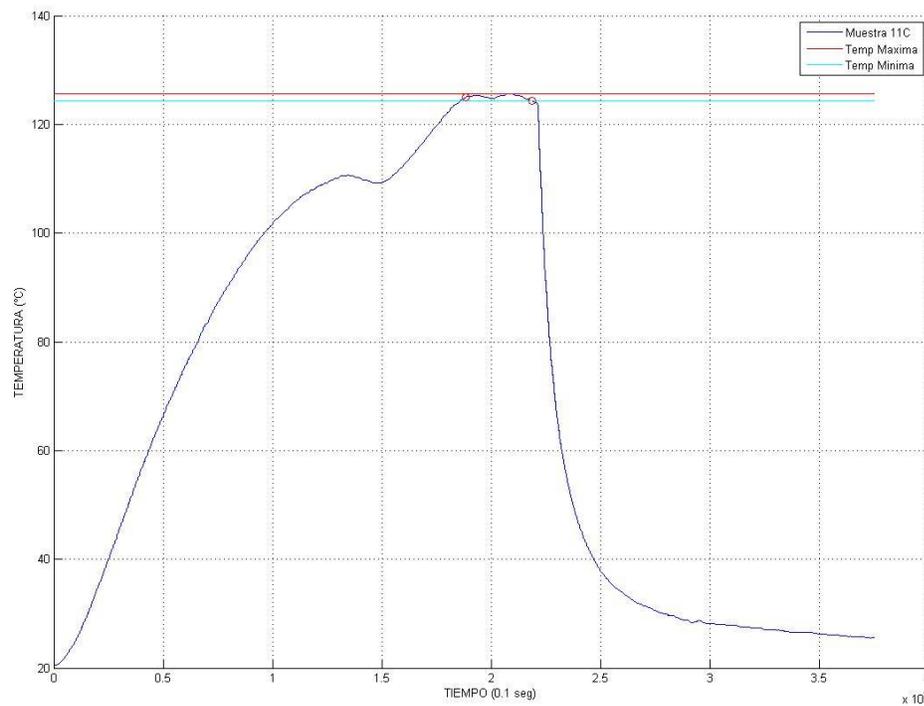


Figura 4.30 Prueba 14 en laboratorio, control de temperatura.

4.1.3.15 Prueba 15

Sin ningún problema. Prueba exitosa.

- Temperatura de referencia = 125,00 °C
- Temperatura máxima durante el estado estable = 125,40 °C
- Temperatura mínima durante el estado estable = 124,60 °C
- Variación de temperatura = +/- 0,40 °C
- Tiempo de duración del experimento = 56 minutos 6 segundos 42 milisegundos
- Tiempo de inicio del estado estable = 36,52 minutos
- Tiempo de inicio del enfriamiento = 41,60 minutos
- Tiempo de duración del experimento en estado estable = 5,07 minutos

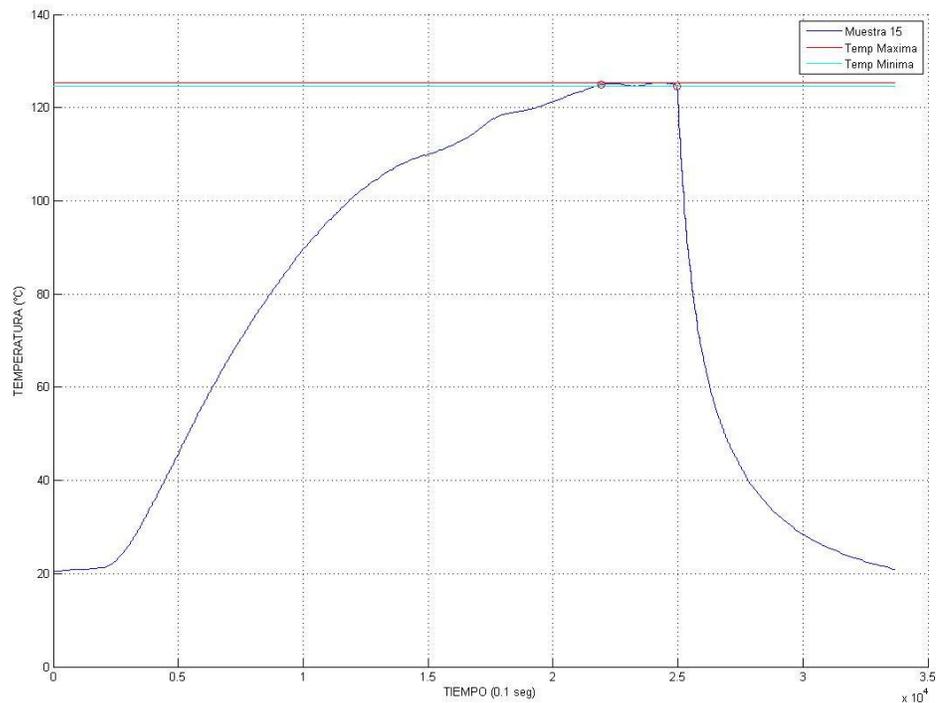


Figura 4.31 Prueba 15 en laboratorio, control de temperatura.

4.1.3.16 Prueba 16

Sin ningún problema. Prueba exitosa.

- Temperatura de referencia = 125,00 °C
- Temperatura máxima durante el estado estable = 125,70 °C
- Temperatura mínima durante el estado estable = 124,60 °C
- Variación de temperatura = +0,70 °C / -0,40 °C
- Tiempo de duración del experimento = 52 minutos 24 segundos 24 milisegundos
- Tiempo de inicio del estado estable = 31,45 minutos
- Tiempo de inicio del enfriamiento = 38,74 minutos
- Tiempo de duración del experimento en estado estable = 7,29 minutos

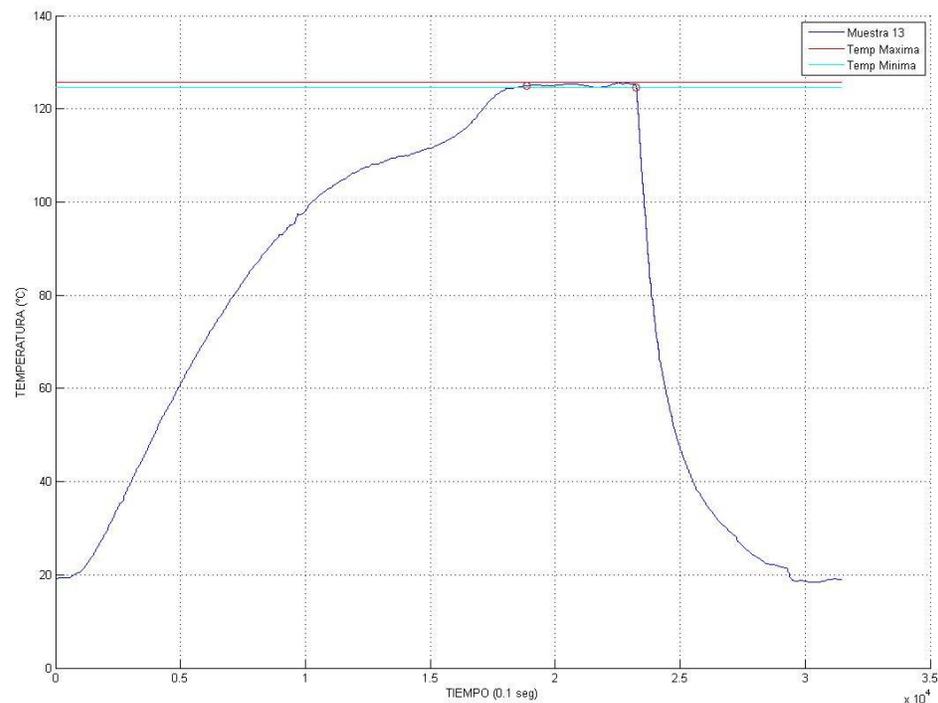


Figura 4.32 Prueba 16 en laboratorio, control de temperatura.

Cada uno de los resultados obtenidos al realizar el experimento en el reactor químico tipo BATCH fue etiquetado, almacenado y refrigerado en un lugar seguro (Figura 4.33), debido que el 5-Hidroximetilfulfural no puede estar expuesto a la luz porque altera sus características para su correcto análisis, a más de ello una exposición continua al mismo sin las protecciones debidas tienen un comportamiento como agente cancerígeno.



Figura 4.33 5-Hidroximetilfulfural resultante del experimento en el reactor batch.

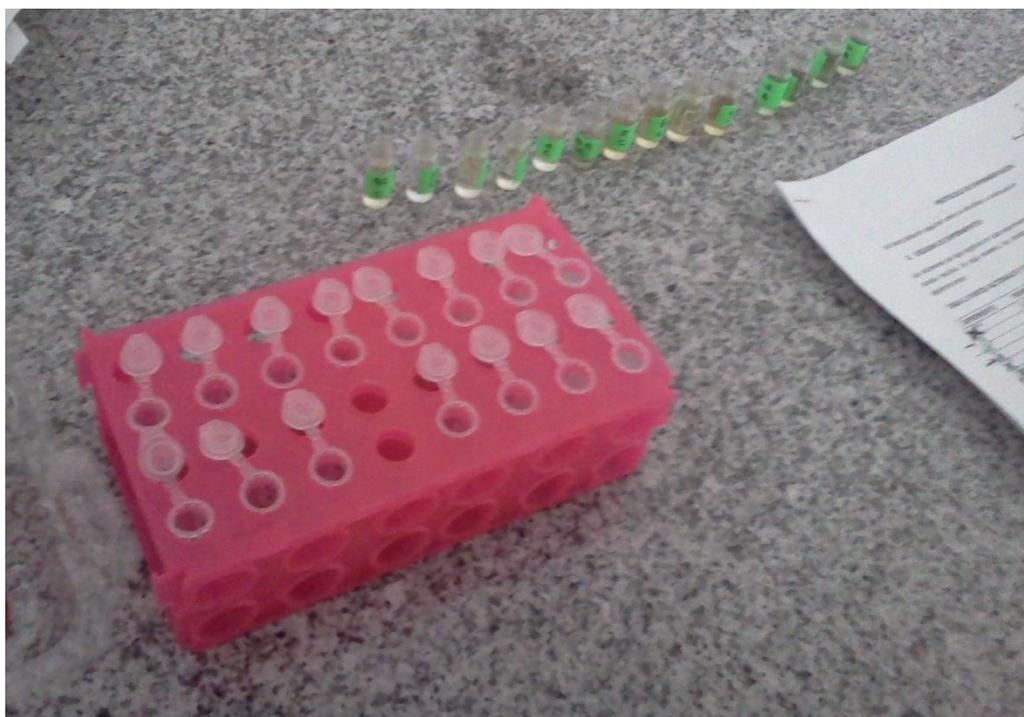


Figura 4.34 Muestras realizadas en el reactor batch

4.2 Ajustes de controladores.

Después de haber obtenido los valores de las constantes de una forma analítica, para implementarlos se han transformado en constantes de tiempo, la banda proporcional es $BP\% = 0.7$ (Ecuación 4.12), el tiempo integral (Ecuación 4.15) tiende a infinito ya que es un control PD por lo que se ingresara el valor más alto posible que es $T_i = 9999$ y el tiempo derivativo (Ecuación 4.18) con un valor de $T_d = 50$.

Las pruebas se realizaron cambiando el valor de las constantes, en diferentes tiempos de duración, a diferentes horas y a diferentes temperaturas para llegar al resultado deseado. Para cada valor de temperatura se desarrolló una tabla en la que se detalla el valor de las constantes y el máximo sobresalto obtenido en grados centígrados ($^{\circ}C$).

Tabla 4.1 Parámetros del controlador de temperatura a $90^{\circ}C$

	Prueba 5	Prueba 6	Prueba 7	Prueba 8	Prueba 9
Po	0,7	0,6	0,6	0,6	1,5
Ti	9999	9999	9999	9999	9999
Td	50	40	70	150	50
sobresalto	2,8	3,3	2,9	3,5	2,9

Tabla 4.2 Parámetros del controlador de temperatura a $100^{\circ}C$

	Prueba 1	Prueba 3	Prueba 4
Po	0,5	0,5	0,7
Ti	9999	9999	9999
Td	50	50	50
sobresalto	2.18	1,63	0,2

Tabla 4.3 Parámetros del controlador de temperatura a $120^{\circ}C$

	Prueba 2	Prueba 10	Prueba 11	Prueba 12
Po	0,5	0,6	0,4	0,4
Ti	9999	9999	9999	9999
Td	50	50	50	50
sobresalto	0,3	1,2	0,7	0,8

En la Tabla 4.1 se puede observar que los valores obtenidos en la Prueba 5 son los más idóneos para una temperatura baja con un sobresalto de 2.8, no obstante no es el mejor resultado (esto lo explicaremos en el punto 4.3), mientras en la Tabla 4.2 se muestra una mejora sustancial a los de la Tabla 4.1 en la Prueba 4, que tiene un

sobresalto de 0.2, mientras que en la Tabla 4.3 se muestra que los valores que generan un sobresalto del 0,3 son los mismos valores que fueron calculados anteriormente, con esta serie de pruebas se concluye que el valor idóneo para el controlador es el calculado previamente.

4.3 Análisis de resultados experimentales.

Las pruebas realizadas han sido diversas, siendo separadas en dos grupos: las pruebas de ajustes y pruebas de laboratorio.

Concretándonos a las primeras se observa que las pruebas que se realizaron a baja temperatura como se muestra en la Tabla 4.1, presentan un sobresalto que oscila entre los 3,5°C, mientras que las pruebas con temperatura más elevada como se indica en la Tabla 4.3, mejoran el sobresalto que se genera en el sistema, siendo este de 0,3°C, esto es debido que la niquelina como tal no tiene un comportamiento lineal y varía su comportamiento a diferentes temperaturas, y mientras más nos acercamos al punto de análisis de la F.T. el sobresalto del sistema va disminuyendo, es por ello que si se desea trabajar en un rango amplio de temperaturas y evitar el sobre salto, se debería de realizar nuevos cálculos para sacar las constantes que se acoplen con los nuevos valores.

En las pruebas se laboratorio se observa que la niquelina tiene un comportamiento diferente al sobrepasar un valor de temperatura, pues tiene un cierto retardo para luego continuar subiendo, por esta razón se observa que en las pruebas de laboratorio existe un breve momento en el cual la temperatura tiende a estabilizarse para después continuar. Para explicarlo de una mejor manera, en la Figura 4.35 se observa que para los diferentes valores de voltaje, la niquelina tiene un comportamiento diferente, pero en cada uno de ellos tiene un instante en donde varía su comportamiento; durante el tiempo de subida se nota que el aumento de la temperatura no es continuo, más bien tiene un comportamiento exponencial hasta que se detiene por un instante. Esto se lo puede observar muy notoriamente en la curva de color turquesa, esta variación es característica de la niquelina, por lo cual no se puede hacer nada para corregirla. Para corregirlo necesariamente se tendría que cambiar la niquelina por una que tenga mejores prestaciones como una mayor potencia para evitar este inconveniente.

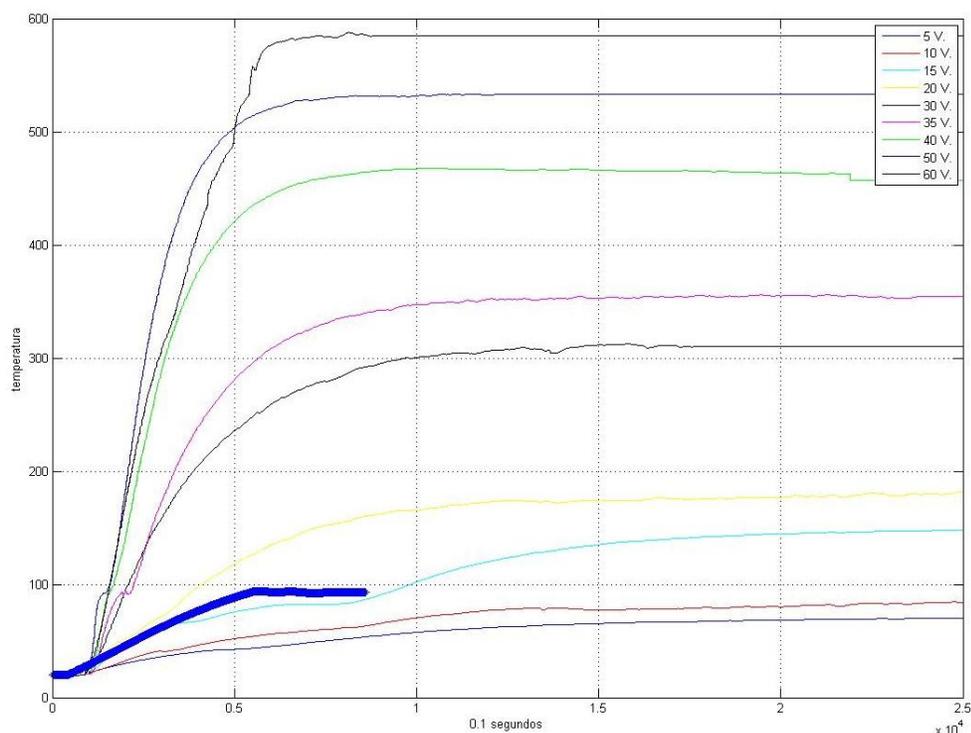


Figura 4.35 Comportamiento de la níquelina

4.4 Implementación del sistema de control (física).

En este apartado se trata el tema de la implementación física de los controladores de velocidad y de temperatura, el diseño de la placa electrónica que comanda a todo el sistema del reactor y el proceso de construcción de todo el sistema, que para su elaboración se tomó en cuenta todos los requerimientos necesarios por la parte química, se hicieron varias modificaciones en la parte mecánica para llegar a un sistema optimo y de acuerdo a las necesidades inicialmente planteadas.

En la parte electrónica interviene la fabricación de la tarjeta electrónica que contiene el sistema de control de la velocidad del motor, la comunicación entre el control de temperatura industrial delta, entre los microcontroladores y con el computador.

4.4.1 Pruebas iniciales

Para llegar a un sistema que cumpla con todos los requerimientos se tuvo que pasar por gran cantidad de pruebas ya que el diseño inicial no cumplía con todas las necesidades, para ello se realizaron varias modificaciones al diseño originalmente

planteado y teniendo como resultado un reactor que cumpla con las especificaciones necesarias. A continuación en la Figura 4.36 está el sistema inicial del reactor químico tipo batch.



Figura 4.36 Diseño inicial del reactor tipo batch

Este sistema inicial constaba de un agitador mecánico, a través de la tapa ingresaba un eje con movimiento circular que tenía una hélice en el extremo que va hacia la parte interna del reactor. Debido a la presión generada por la reacción de la fase acuosa reactiva, la alta temperatura que el sello mecánico tenía que soportar y las fugas generadas por la corrosión del ácido, fue modificado de tal manera que se obtuvo el sistema que se encuentra en la Figura 4.37, este tiene una agitación de tipo magnético (idea generada por el Ing. Pablo Arévalo, Universidad Politécnica Salesiana).



Figura 4.37 Diseño 2 del reactor batch

4.4.2 Construcción de la estructura mecánica.

Para la construcción física del reactor tipo Batch primero fue diseñado con el software CAD **SOLIDWORKS 2012**. Los planos realizados se encuentran en el ANEXO I.

El reactor químico ha sido construido en acero inoxidable de 3 mm que nos permite garantizar una correcta mantención de la presión dentro del sistema, pudiendo llegar hasta una presión de 140 psi según los cálculos realizados en el ANEXO J.

El reactor químico se encuentra conformado por la tapa del reactor y el envase, las cuales se detallan a continuación:

4.4.2.1 Tapa del reactor.

La tapa está realizada con 6 pernos de sujeción M4 X 12mm los que permiten su montaje y desmontaje. Cuenta además con dos orificios pasantes: uno para un manómetro de 100 PSI y el otro para el sensor de temperatura. El orificio del medio sirve como lugar de reposo para la niquelina, se encuentra soldado con un tubo de Ø15mm X 140mm.



Figura 4.38 Tapa del reactor

4.4.2.2 Envase del reactor.

El envase tiene 6 orificios pasantes M4 en donde se enroscan los pernos para poder realizar el sellado completo del reactor. Para procurar un enfriamiento uniforme cuenta con una chaqueta que permite la circulación del refrigerante sin que intervenga de ninguna manera en la parte química, para acceder a ella tiene dos Racor de 1/4" uno en su parte inferior (entrada de refrigerante) y el otro en su parte superior (salida de refrigerante).



Figura 4.39 Envase del reactor

4.4.2.3 Base del reactor.

La base del reactor está elaborada en acero inoxidable de 1,5 mm de espesor. Además de servir de soporte para el envase del reactor y del motor con el que se realiza la agitación, este permite la sujeción de todos los implementos electrónicos que se están

utilizando. Cuenta con cuatro orificios para los botones, dos para las mangueras del refrigerante (entrada y salida), uno para la alimentación del reactor, uno para la bomba re circulante, un orificio para el controlador y otro para el visualizador. En la parte superior tiene una entrada para el cable del sensor y otro para la toma de energía de la niquelina.



Figura 4.40 Base del reactor parte frontal



Figura 4.41 Base del reactor parte posterior

4.4.3 Tarjeta electrónica.

La tarjeta electrónica ha sido realizada según el diseño realizado en el capítulo 3, esta consta con algunos elementos de montaje superficial como resistencias y diodos LED. El resultado con todos los implementos electrónicos se ve a continuación:

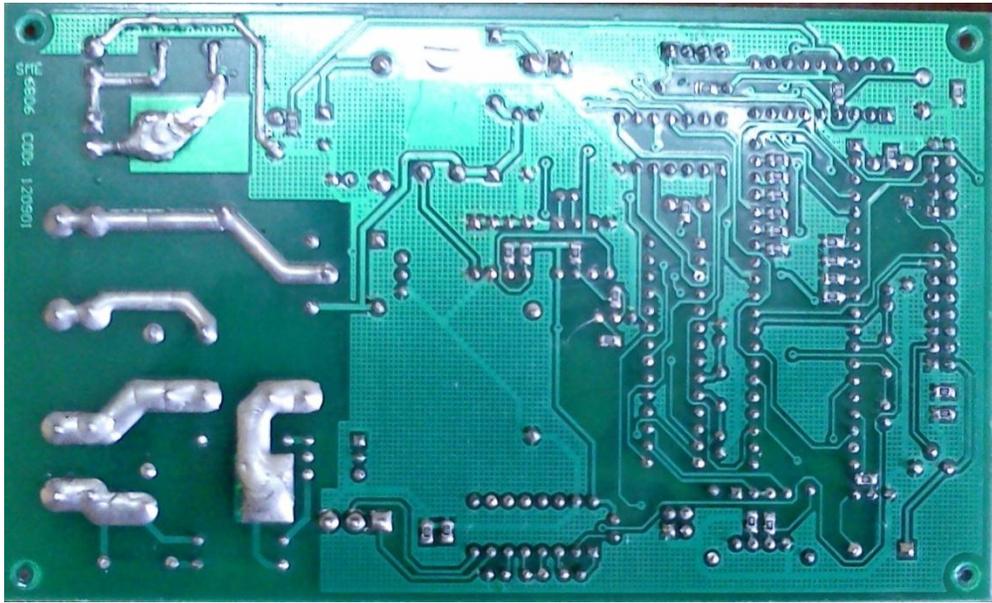


Figura 4.42 Vista inferior de la tarjeta electrónica

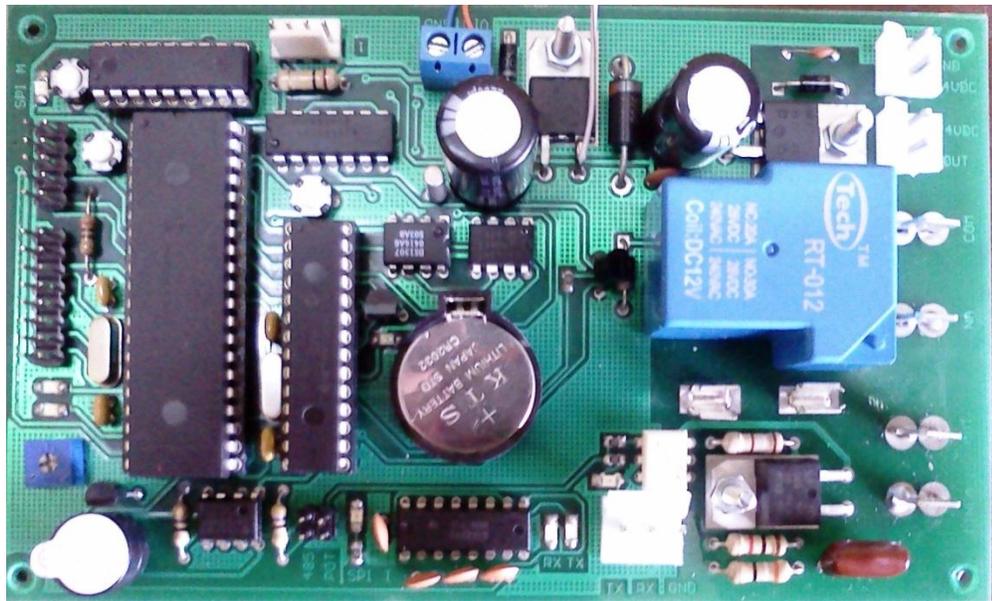


Figura 4.43 Vista superior de la tarjeta electrónica

4.4.4 Ensamble del Equipo.

Una vez realizado el proceso de cálculos, pruebas en el reactor, construcción de la estructura mecánica, diseño y fabricación del PCB (tarjeta electrónica) se puede proceder a ensamblar todo el sistema y cumplir con los objetivos iniciales. El equipo listo para su operación inmediata se lo puede ver en la siguiente figura:

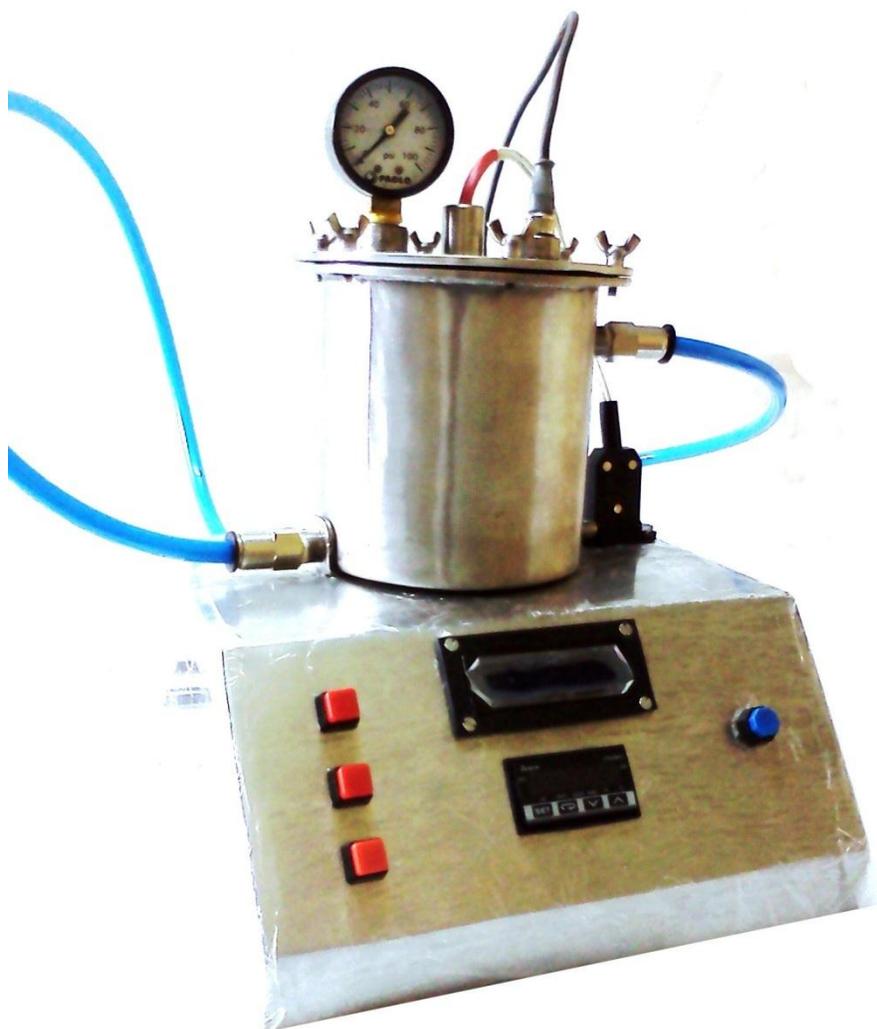


Figura 4.44 Reactor químico tipo BATCH

A continuación se muestran algunas de las imágenes antes de tener el producto final.



Figura 4.45 Pruebas del funcionamiento de la tarjeta

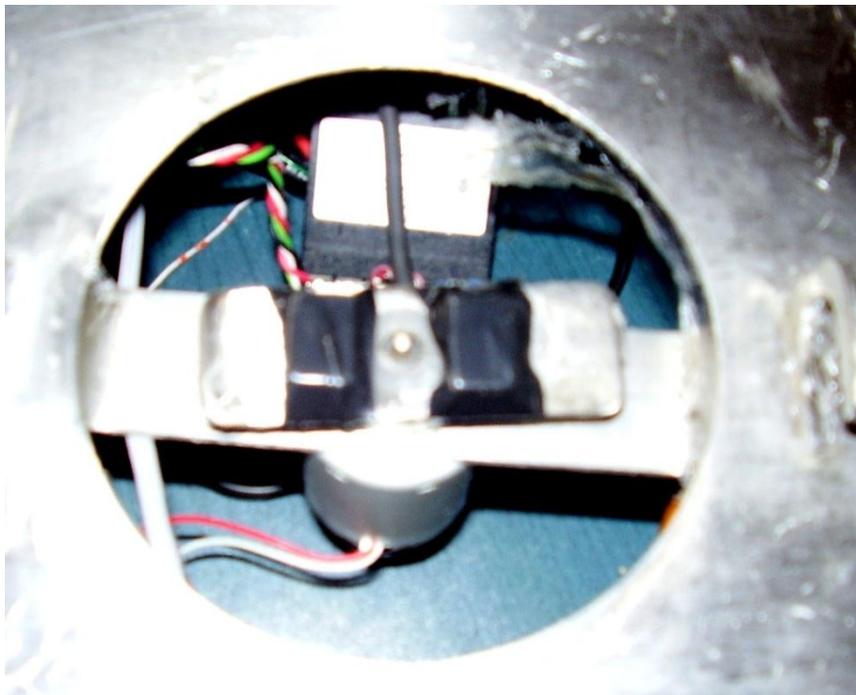


Figura 4.46 Colocación del motor y el agitador

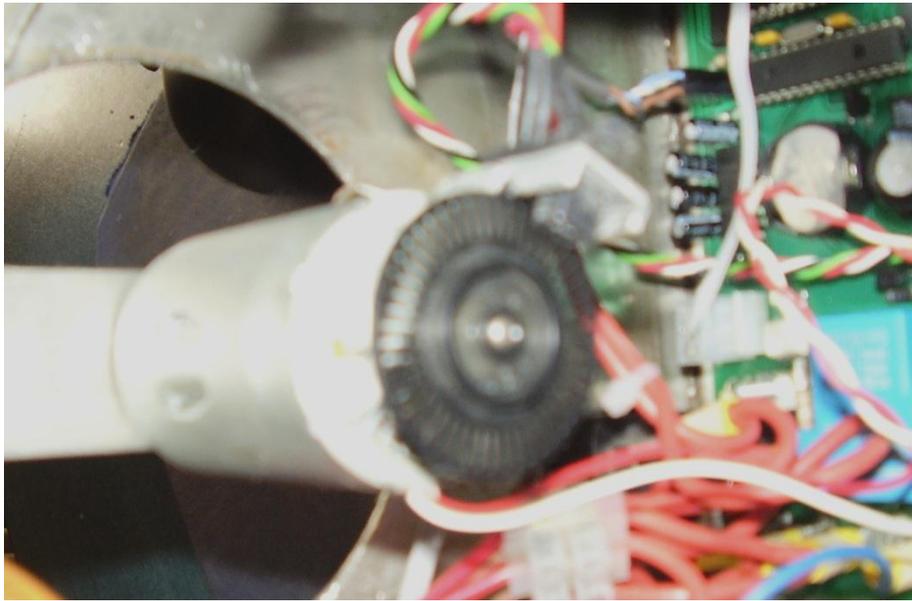


Figura 4.47 Colocación del encoder en el motor

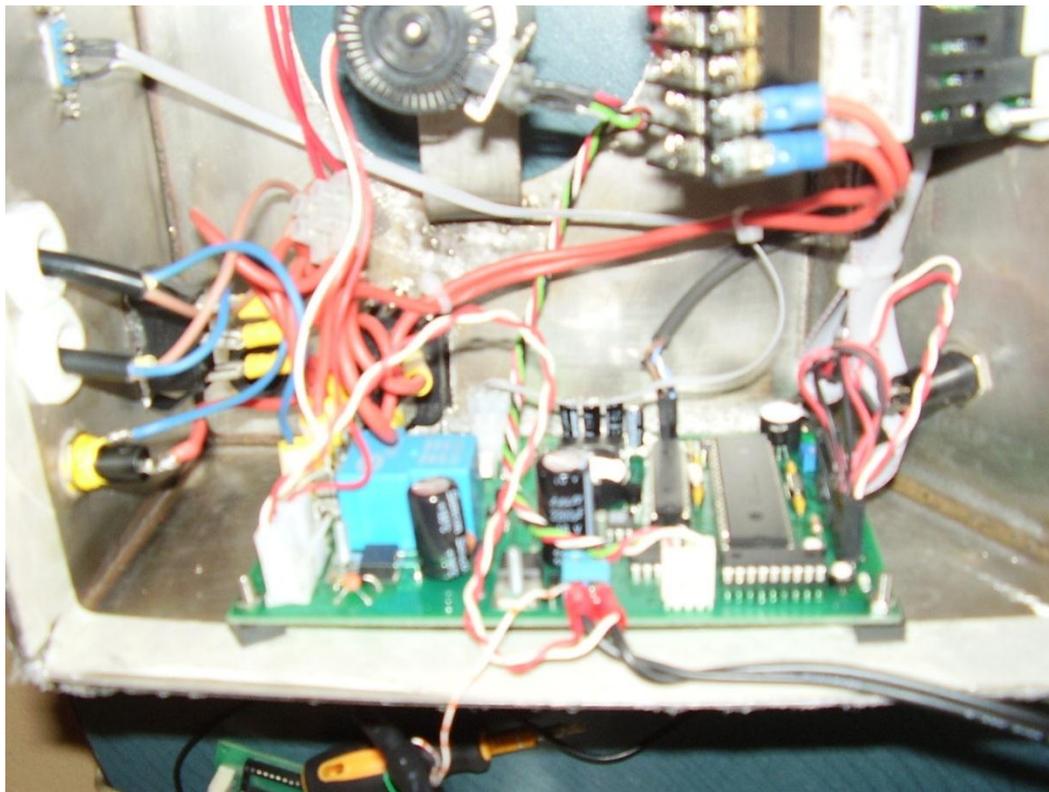


Figura 4.48 Cableado y sujeción de la tarjeta electrónica

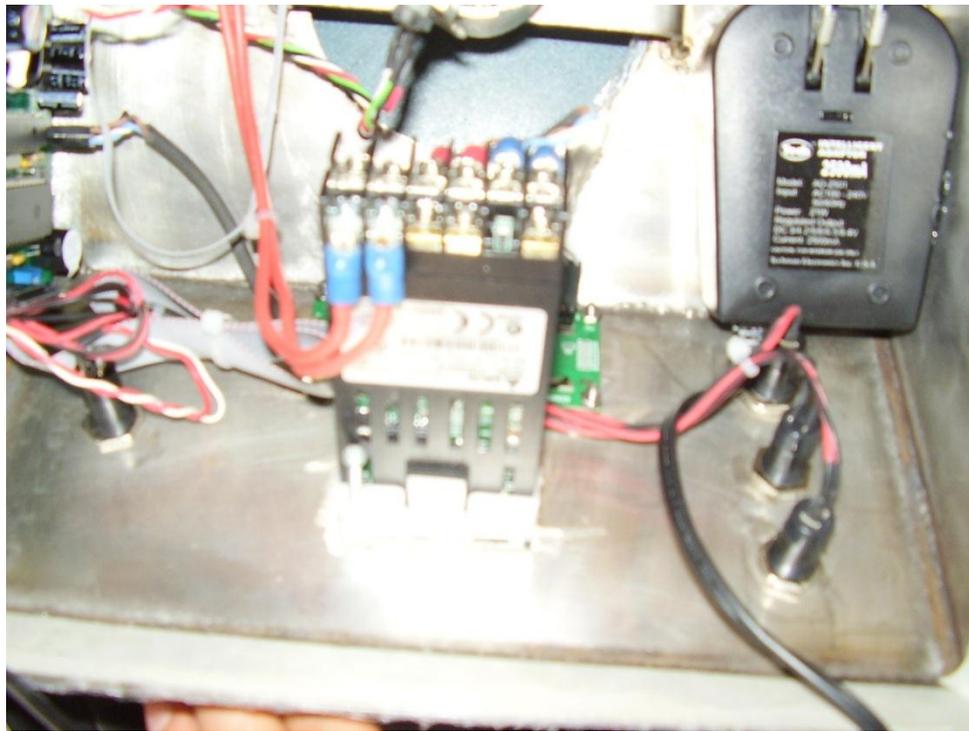


Figura 4.49 Colocación del controlador y de la fuente de energía del reactor

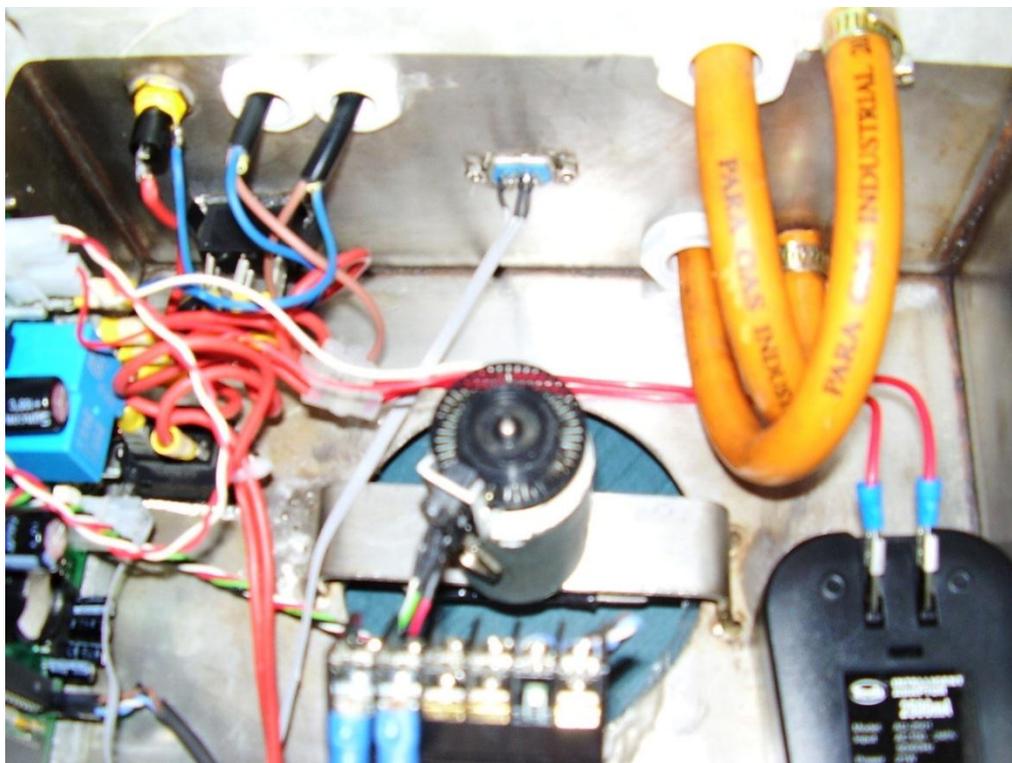


Figura 4.50 Conexión de mangueras de la bomba refrigerante hacia mangueras del reactor

CONCLUSIONES Y RECOMENDACIONES.

Conclusiones

En el desarrollo de los procesos químicos se necesita además de mezclas en cantidades exactas (en este caso el 5-hidroximetilfurfural) un lugar o ambiente controlado que pueda garantizar la estabilidad de la reacción y permitir a futuro una reproducción de dicho proceso de ser requerido. Esto conlleva a la utilización de reactores químicos que tengan estas características, siempre considerando que existen una amplia gama, en formas, tamaños y dependiendo del proceso que se vaya a realizar se busca el que cumpla con todos los requerimientos particulares.

Ahora considerando la factibilidad o complejidad que pueda tener cualquier tipo de proceso está en que tan experimentado sea el diseñador que lo va a llevar a cabo para lograr su objetivo, buscando las diferentes vías o maneras de cómo obtener un resultado óptimo, fiable, inmediato y en lo posible a bajos costos y que sobretodo garantice su análisis. Para ello se debe tener un conocimiento previo en el área en que se está incursionando, evitando así una variedad de errores que se puedan cometer por falta de conocimiento del tema, de ser requerido se deberá buscar un apoyo foráneo que de las pautas necesarias para proseguir con el análisis o, de requerirse un auto aprendizaje, tomarse el tiempo que sea necesario para la ejecución del mismo.

Al lograr el conocimiento necesario tanto de requerimientos como de funcionamiento del sistema que se pretende controlar, se procederá al desarrollo y la implementación del mismo considerando un proceso lógico, pudiendo ser como el que se expone a continuación, no obstante esto depende de cada diseñador.

- La identificación del sistema.
- Considerar variables externas que influyen en el sistema.
- Diseño del sistema de control.
- Simulaciones del sistema de control.
- Implementación.
- Ajustes.

Al tener un proceso ordenado también se debe buscar la elección correcta de nuestro controlador, para ello debemos de seguir un criterio que nos permita escoger y conocer el tipo o los tipos de controladores que se quiera implementar, algunos de esos criterios pueden ser:

- Garantizar la estabilidad y su robustez ante perturbaciones o errores de moldeamiento.
- Tiene que ser eficiente y realizable.

- Ser de fácil implementación.
- Tener costos asequibles para la realización.

Después de tener presente varios criterios y haber tomado una decisión se podría realizar la implementación de nuestros controladores, siempre teniendo presente como último paso la realización de las pruebas, las mismas que se deberían de hacer en diferentes condiciones, para poder conocer el comportamiento real que pueda tener a diferentes horas o temperaturas, sin llegar a los extremos ya que esto puede dañar seriamente todo nuestro sistema o incluso puede ser perjudicial para quien lo controla.

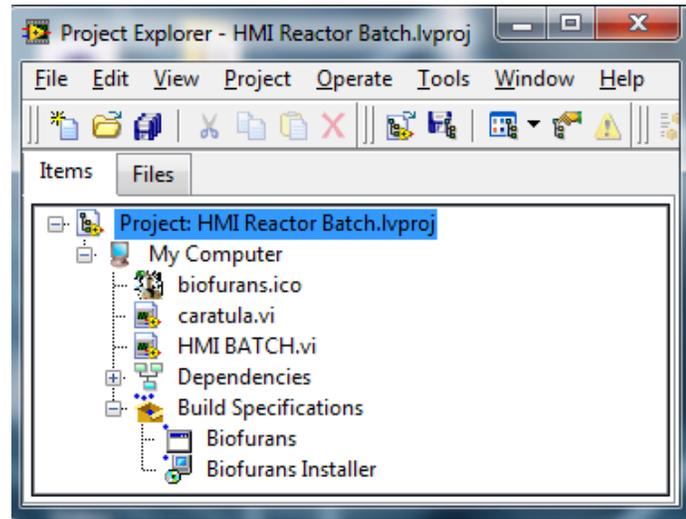
Recomendaciones

- Al realizar cualquier tipo de control de un sistema se debe de tener muy en claro todo lo que se va a utilizar dentro del mismo, las funciones que queremos que desempeñe cada actuador o sensor dentro de este, debido que al no tener claro nos puede repercutir en el momento de puesta en marcha de nuestro sistema en conjunto, es por ello que se debe de realizar un análisis amplio de los actuadores, sensores, el sistema en donde se los pretende poner, tipos de materiales a soportar, condiciones en las que va a operar, entre otras, para lograr determinar con eficiencia cual es el más idóneo para la adquisición del mismo.
- Al no tener productos que garanticen las características constructivas del mismo, o que no hagan mención a su comportamiento, se está sujeto a cambios o anomalías que pueda sufrir el mismo sin previo aviso, en este caso el comportamiento de la niquelina ha sido fluctuante y no se ha podido analizar un parámetro con eficacia, por ello si se aprecia en el capítulo 4, en diferentes pruebas realizadas hay una variación significativa en algunos de los casos y en otros se mantiene un comportamiento constante, para salvaguardar estos inconvenientes se podría optar por otra clase de sistemas de control, que sean más robustos o que generen un auto aprendizaje del comportamiento del sistema, pudiendo ser estos con lógica difusa o a su vez algo más complejo como las redes neuronales.
- Los criterios a seguir para la realización de los controladores deben de ser los adecuados, a más de tener un conocimiento de su aplicación, pues en algunos casos se necesitaría un control retroalimentado para una estabilización correcta o simplemente un control abierto donde simplemente se necesite que se cumpla una determinada función pero no una estabilización del sistema, lo cual disminuirá costos de implementación y tiempo en su desarrollo.

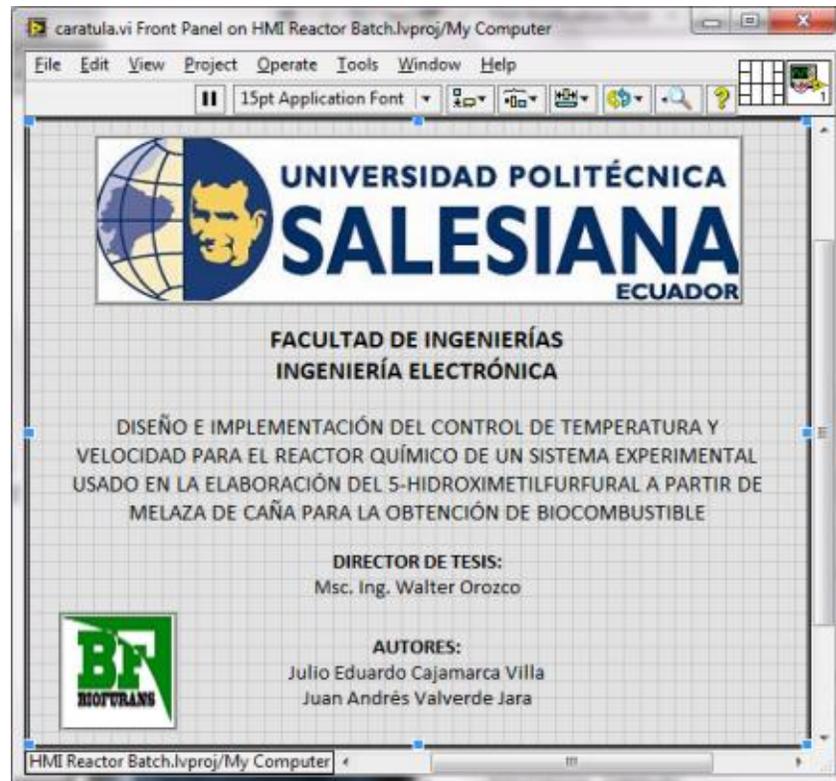
ANEXOS

ANEXO A CÓDIGO FUENTE DEL SOFTWARE HMI³²

Vista del proyecto “HMI Reactor Batch.lvproj”

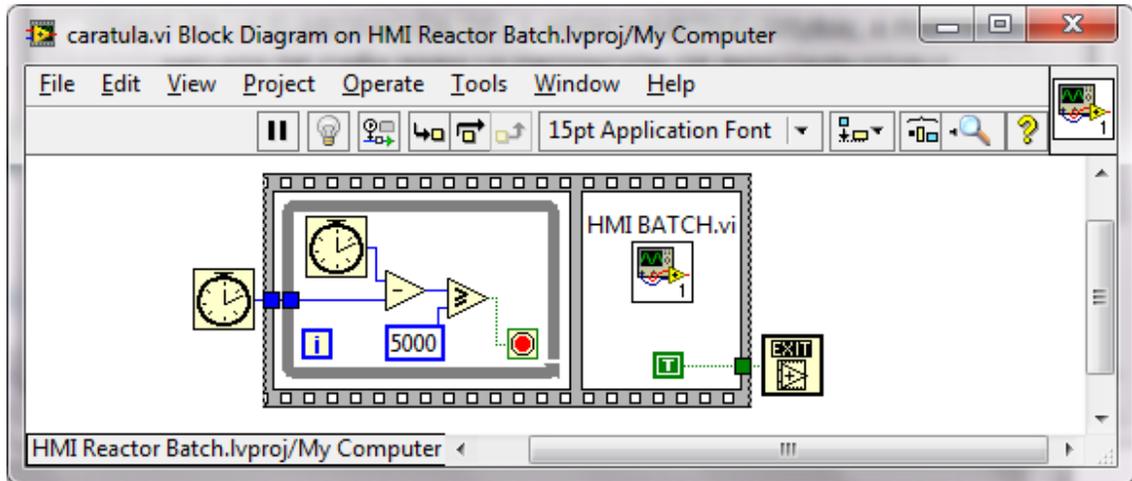


Panel frontal de “caratula.vi”



³² Para mayor detalle referirse al CD-ROM adjunto.

Diagrama de bloques de "catátula.vi"



Panel frontal de "HMI BATCH.vi"

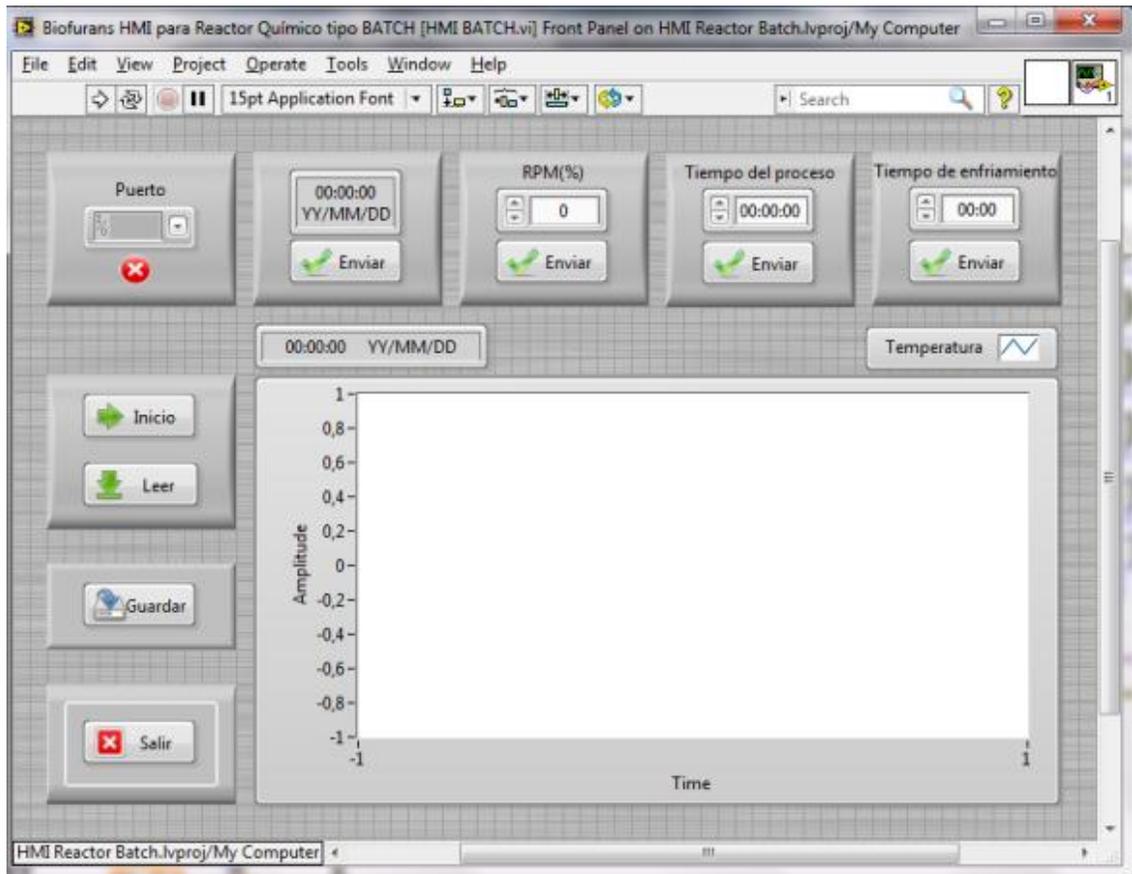


Diagrama de bloques de "HMI BATCH.vi" para "encontrar el instrumento."

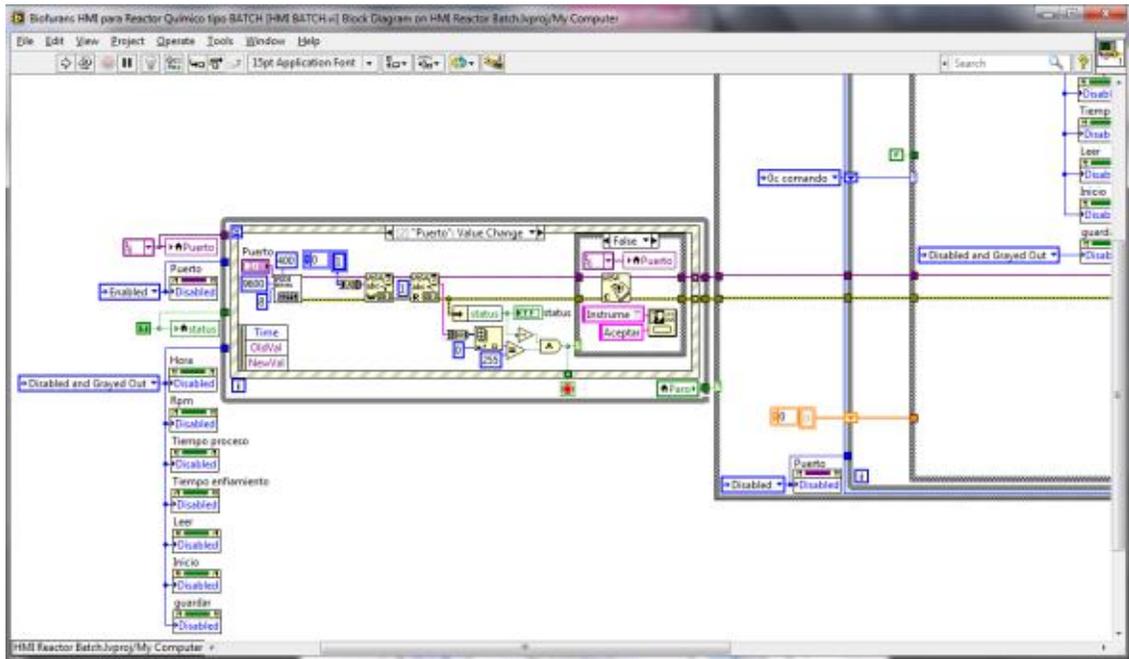


Diagrama de bloques de "HMI BATCH.vi" para "Comando".

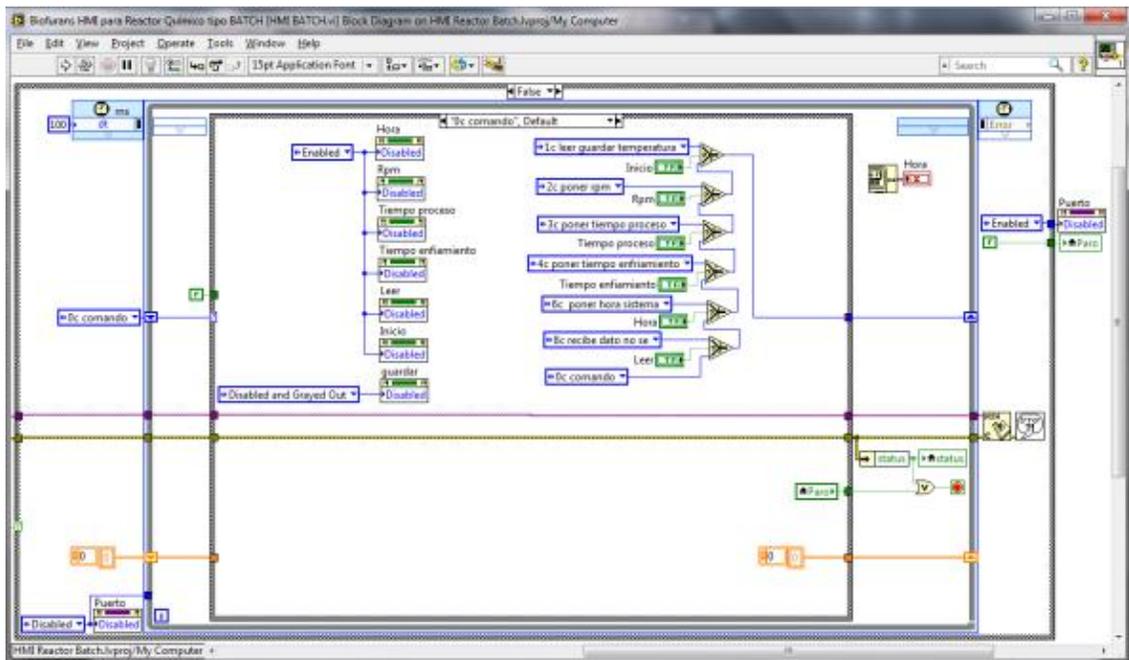


Diagrama de bloques de "HMI BATCH.vi" para "Leer guardar temperatura".

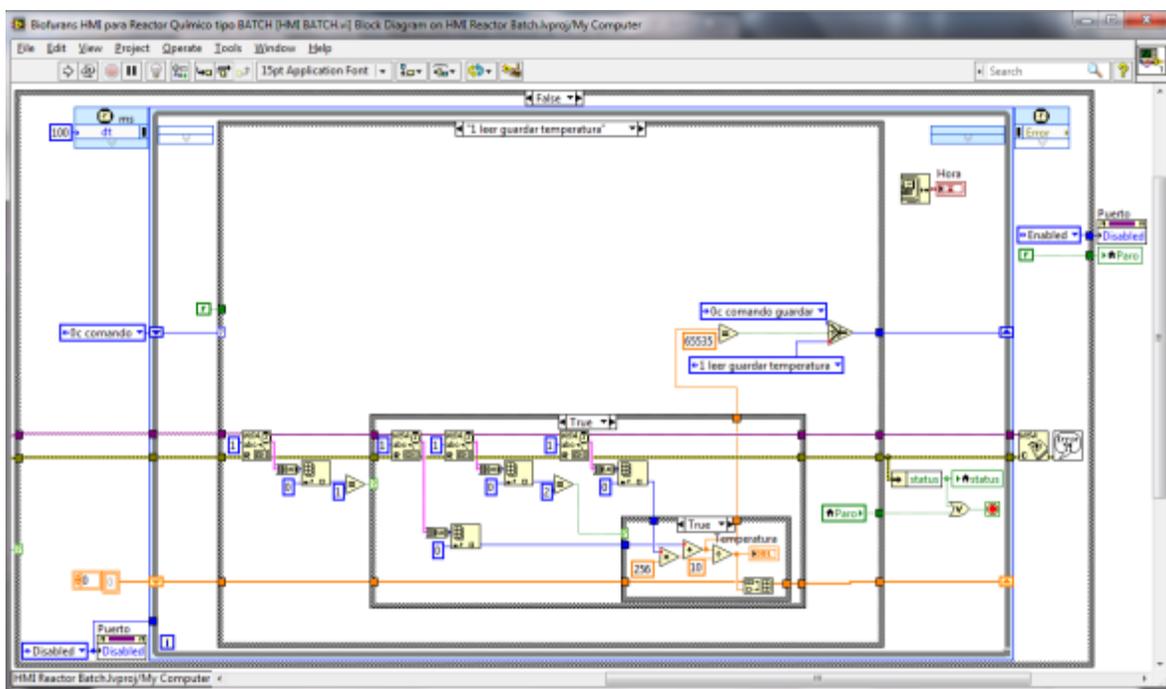


Diagrama de bloques de "HMI BATCH.vi" para "Poner RPM"

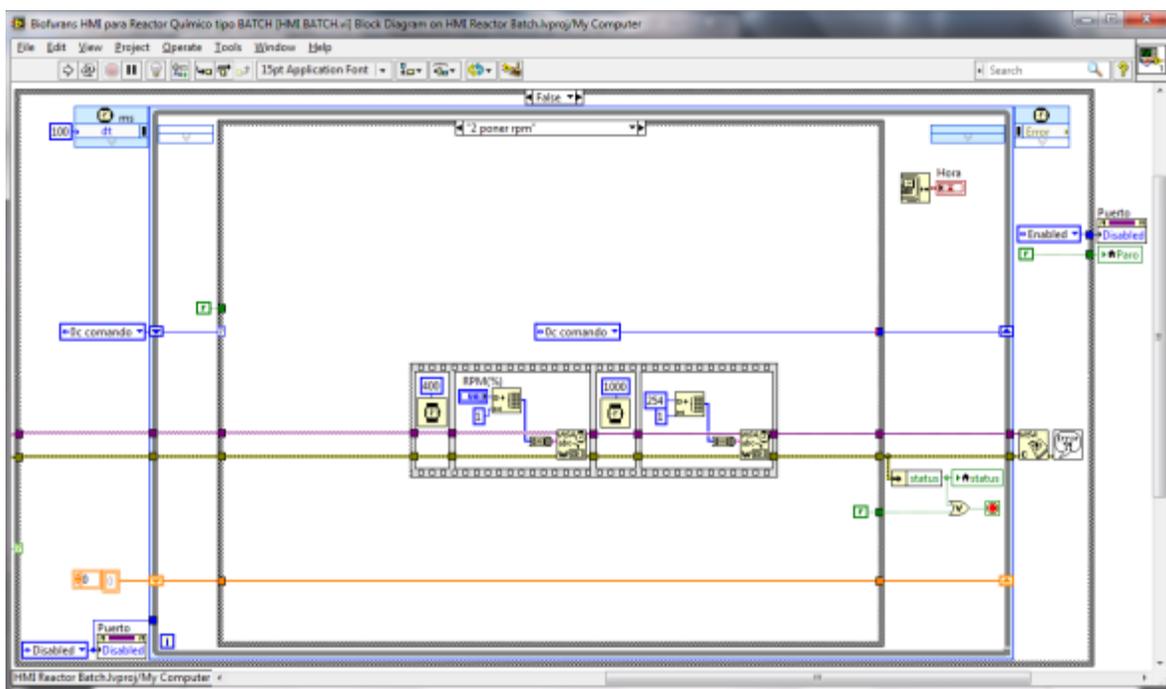


Diagrama de bloques de "HMI BATCH.vi" para "Poner tiempo proceso"

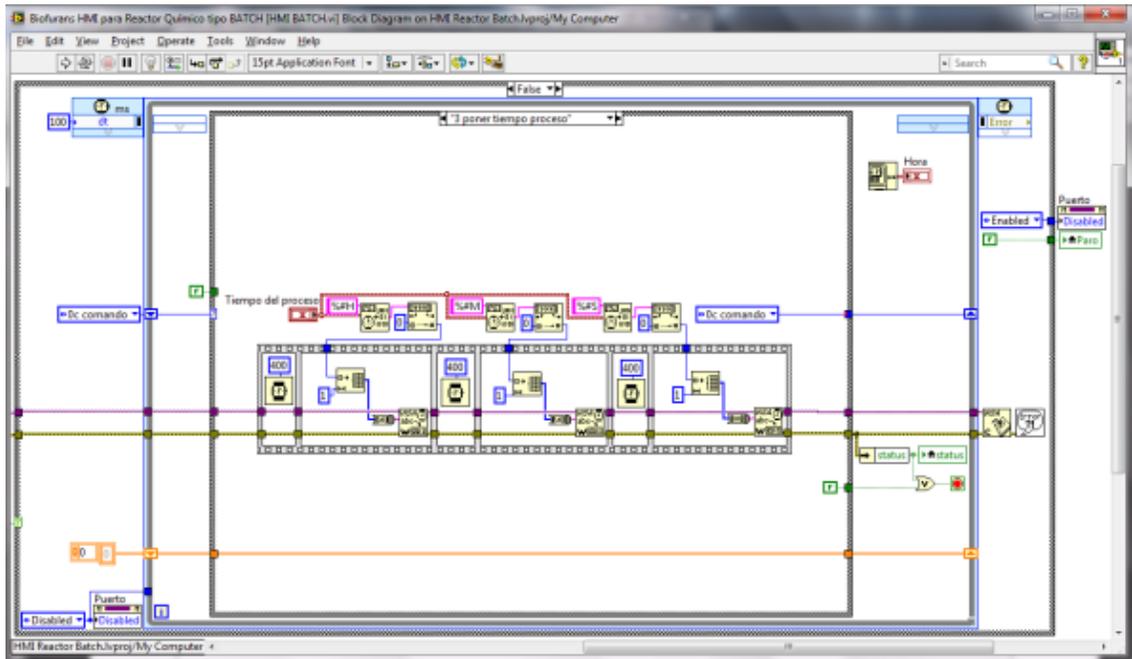


Diagrama de bloques de "HMI BATCH.vi" para "Poner tiempo enfriamiento"

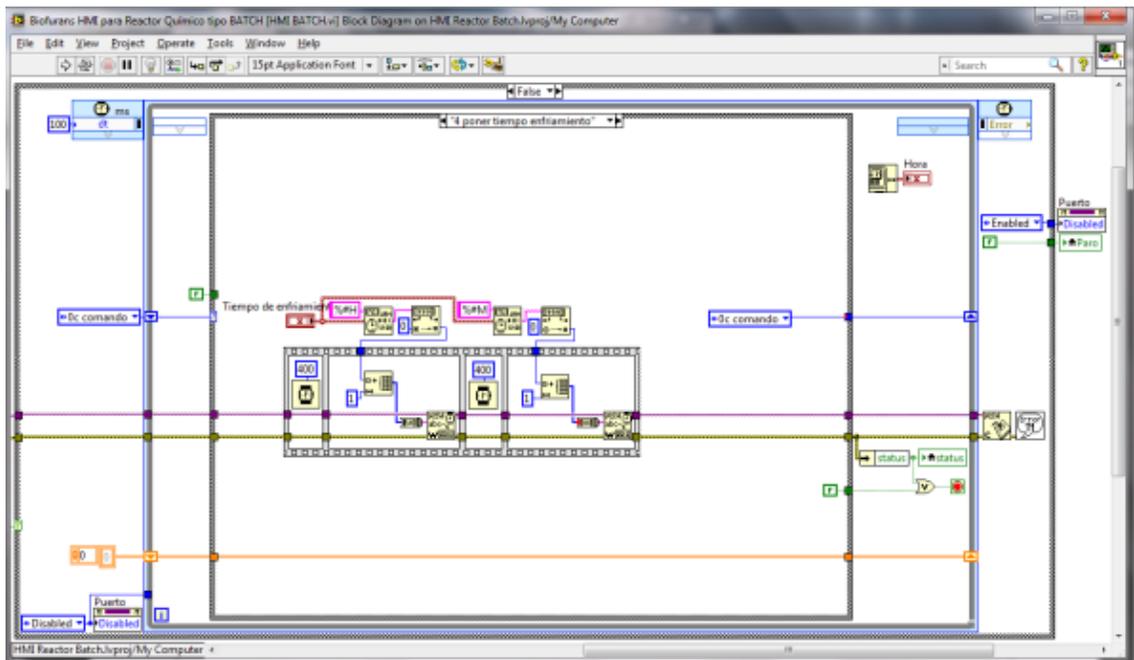


Diagrama de bloques de "HMI BATCH.vi" para "Poner hora del sistema".

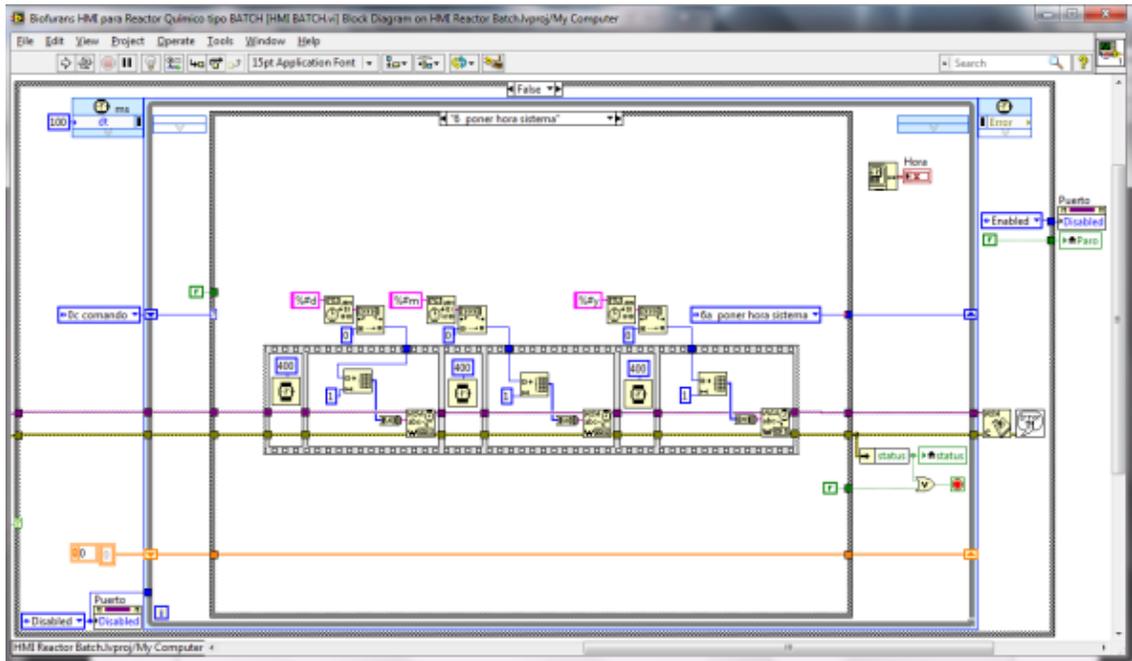


Diagrama de bloques de "HMI BATCH.vi" para "Leer memoria".

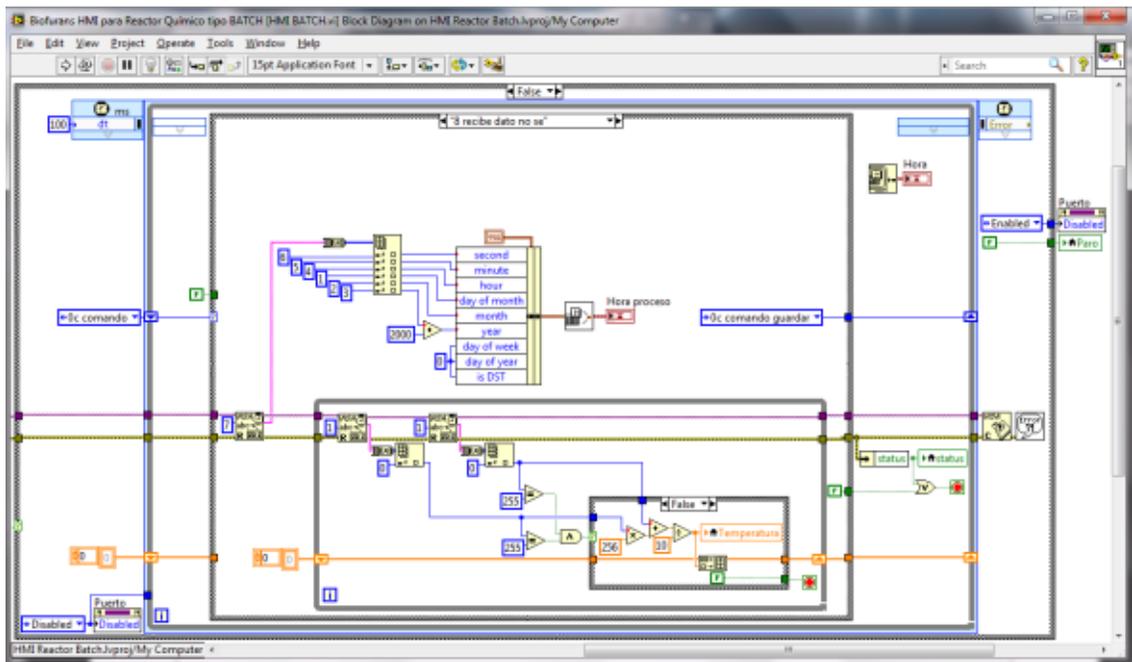


Diagrama de bloques de "HMI BATCH.vi" para "comando guardar"

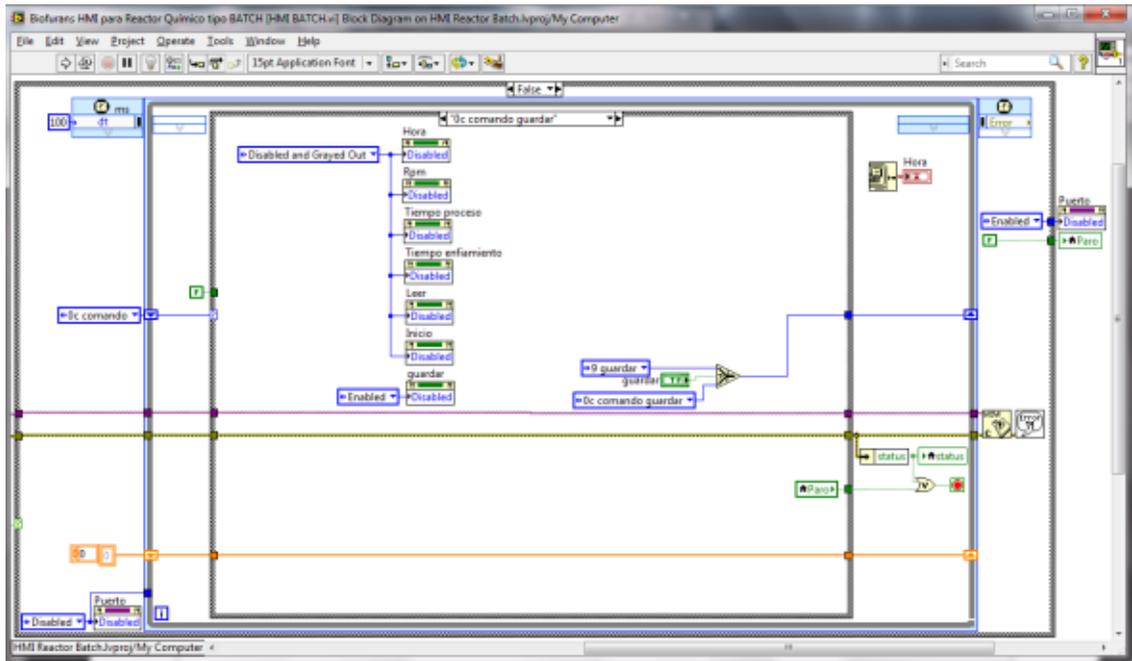
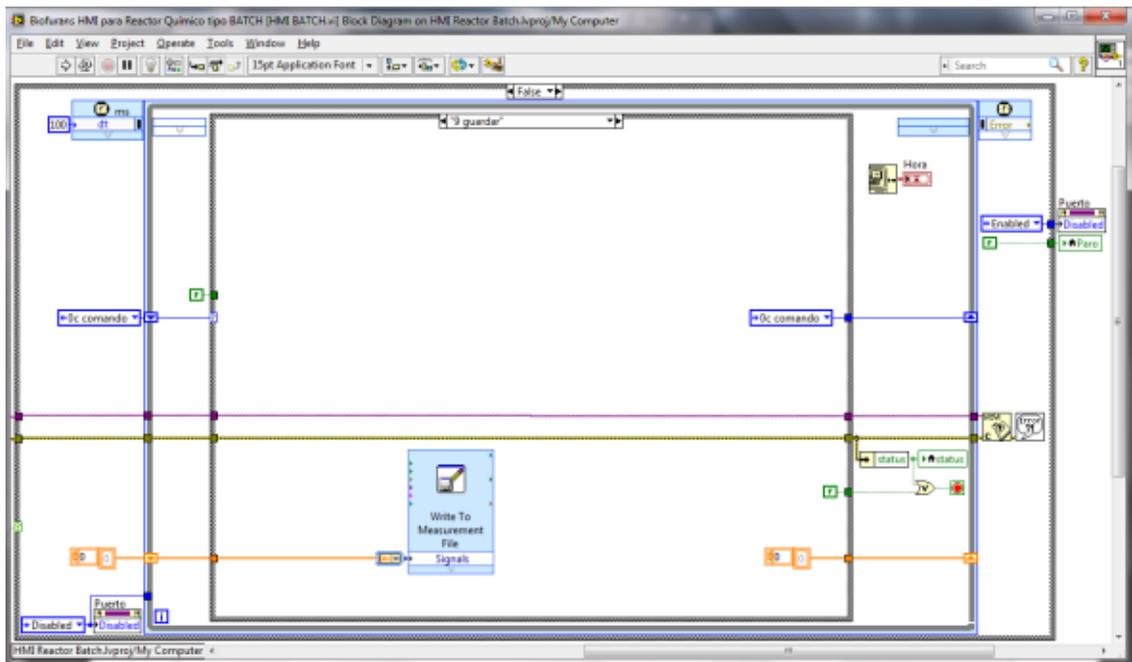


Diagrama de bloques de "HMI BATCH.vi" para "guardar"



ANEXO B

CÓDIGO FUENTE PROGRAMA PRINCIPAL³³

```

1: //Asignacion de nombres a los puertos
2: #define ENTER PORTE.f0
3: #define UP PORTE.f1
4: #define DOWN PORTE.f2
5: #define LED1_SPI PORTC.f0
6: #define LED2_SPI PORTC.f1
7: #define PITO PORTC.f2
8: #define RELE PORTC.f5
9: #define onOff_delta PORTB.F7
10: #define DIN2_SPI PORTB.F6
11: #define DIN1_SPI PORTB.F5
12: #define CLK2_SPI PORTB.F4
13: #define CLK1_SPI PORTB.F3
14: #define DOUT_SPI PORTB.F2
15: #define SEG_SPI PORTB.F1
16: #define CLK_RELOJ PORTB.F0
17: #define MenOn_off PORTA.F0
18: #define dato_ini PORTA.F1
19: #define PARO PORTA.F5
20:
21: // Software I2C connections
22: sbit Soft_I2C_Scl at RC3_bit;
23: sbit Soft_I2C_Sda at RC4_bit;
24: sbit Soft_I2C_Scl_Direction at TRISC3_bit;
25: sbit Soft_I2C_Sda_Direction at TRISC4_bit;
26: // Lcd pinout settings
27: sbit LCD_D7 at RD7_bit;
28: sbit LCD_D6 at RD6_bit;
29: sbit LCD_D5 at RD5_bit;
30: sbit LCD_D4 at RD4_bit;
31: sbit LCD_EN at RD3_bit;
32: sbit LCD_RS at RD2_bit;
33: // Pin direction
34: sbit LCD_D7_Direction at TRISD7_bit;
35: sbit LCD_D6_Direction at TRISD6_bit;
36: sbit LCD_D5_Direction at TRISD5_bit;
37: sbit LCD_D4_Direction at TRISD4_bit;
38: sbit LCD_EN_Direction at TRISD3_bit;
39: sbit LCD_RS_Direction at TRISD2_bit;
40: //Declaración de rutinas
41: void main (void);
42: void set_temp(void);
43: void config(void);
44: //Declaracion de rutina logo
45: void CustomChar1(char pos_row, char pos_char);
46: void CustomChar2(char pos_row, char pos_char);
47: void CustomChar3(char pos_row, char pos_char);
48: void CustomChar4(char pos_row, char pos_char);
49: void CustomChar5(char pos_row, char pos_char);
50: void CustomChar6(char pos_row, char pos_char);

```

³³ Para mayor detalle referirse al CD-ROM adjunto.

```
51: void CustomChar7(char pos_row, char pos_char);
52: void CustomChar8(char pos_row, char pos_char);
53: //Declaracion de rutina SPI
54: void spiMotor(char sendRpm);
55: void spi16receive(void);
56: void spi16send(int SPI_temp);
57: void spi8receive(void);
58: void spi8send(char spi_byte);
59: //Declaracion de rutina paro
60: void Paro_emergencia(void);
61: //Declaracion de rutina i2c
62: void Set_Time(char seconds, char minutes, char hours, char day, char
month, char year)
;
63: void Transform_Time(void);
64: void Display_Time(void);
65: void Read_Time(void);
66: void Read_Mem(int address);
67: void Write_Mem_clock(int address);
68: //Declaracion de rutina Visualizacion
69: void visual_menu(char menu, char x_lcd, char y_lcd);
70: void visual_contador(void);
71: //Declaracion de rutina comandos
72: void rutina_1(void);
73: void rutina_2(void);
74: void rutina_3(void);
75: void rutina_4(void);
76: void rutina_5(void);
77: void rutina_6(void);
78: void rutina_7(void);
79: void rutina_8(void);
80:
81: //Declacion de variables
82: char rpm_set=0;
83: char rpm=0;
84: char rd_dato=0;
85: char txt1[9];
86: char menu=1;
87: char valida=0;
88: char clock_max=0;
89: char dato_selec=0;
90: char x_lcd=1, y_lcd=1;
91: char temp_ok=0;
92: char seg_time=0, min_time=0, hours_time=0;
93: char seg_crono=0, min_crono=0, hours_crono=0;
94: char seg_set=0, min_set=0, hours_set=0;
95: char seg_setenf=20, min_setenf=0;
96: char menSi_no=12;
97: char setDia, setMes, setAnio;
98: char menu_pc=0;
99: char pc_set[9];
100: char SPI_i;
101: char i=0;
102: char flanco_seg=0;
103: char flanco=0;
104: char option=0;
```

```

105: char low_dat=0, high_dat=0;
106: char high_address1=0, low_address1=0, high_address2=0,
low_address2=0;
107: char text[9];
108: char seconds, minutes, hours, day, day_num, month, year;
109: int contador=0;
110: int SPI_dato[17];
111: int num_dato=0;
112: int SPI_read_bajo=0;
113: int SPI_read_alto=0;
114: unsigned int temperatura=0;
115: int temp_set=1500;
116: //Declaracion de constantes de logo
117: const char character1[] = {31,15,0,3,3,3,3,3};
118: const char character2[] = {31,31,7,25,29,5,29,25};
119: const char character3[] = {31,31,31,1,1,31,17,29};
120: const char character4[] = {31,31,31,31,31,31,31,31};
121: const char character5[] = {3,3,3,3,3,0,0,0};
122: const char character6[] = {25,29,5,29,25,0,0,0};
123: const char character7[] = {28,16,16,16,16,0,0,0};
124: const char character8[] = {31,31,31,15,15,7,3,1};
125:
126: //Programa principal.
127: void main (void) {
128: config();
129: dato_ini=1; //portb.f5 del temp
130: MenOn_off=0;
131:
132: Lcd_Cmd(_LCD_CLEAR); // Clear display
133: Lcd_Cmd(_LCD_CURSOR_OFF);
134: Lcd_Out(1,6,"BIOFURANS");
135: CustomChar1(1,1);
136: CustomChar2(1,2);
137: CustomChar3(1,3);
138: CustomChar4(1,4);
139: CustomChar5(2,1);
140: CustomChar6(2,2);
141: CustomChar7(2,3);
142: CustomChar8(2,4);
143: for (rpm=0; rpm<=2; rpm++) PITO=1, delay_ms(500), PITO=0, delay_ms(500);
144: for (rpm=0; rpm<=15; rpm++) {
145: delay_ms(200);
146: Lcd_Cmd(_LCD_SHIFT_LEFT);
147: }
148: valida=1;
149: set_temp();
150: visual_menu(menu,1,1);
151:
152: while(1) {
153: if(DOWN==1) {
154: delay_ms(300);
155: menu++;
156: switch(menu) {
157: case 1: x_lcd=1,y_lcd=1;break;
158: case 2: x_lcd=2,y_lcd=1;break;
159: case 3: x_lcd=1,y_lcd=1;break;

```

```

160: case 4: x_lcd=2,y_lcd=1;break;
161: case 5: x_lcd=1,y_lcd=1;break;
162: case 6: x_lcd=2,y_lcd=1;break;
163: case 7: x_lcd=1,y_lcd=1;break;
164: }
165: if(menu>=7)menu=7;
166: visual_menu(menu,x_lcd,y_lcd);
167: }
168: if(UP==1){
169: delay_ms(300);
170: menu--;
171: switch(menu){
172: case 1: x_lcd=1,y_lcd=1;break;
173: case 2: x_lcd=2,y_lcd=1;break;
174: case 3: x_lcd=1,y_lcd=1;break;
175: case 4: x_lcd=2,y_lcd=1;break;
176: case 5: x_lcd=1,y_lcd=1;break;
177: case 6: x_lcd=2,y_lcd=1;break;
178: case 7: x_lcd=1,y_lcd=1;break;
179: }
180: if(menu<1){menu=1;}
181: visual_menu(menu,x_lcd,y_lcd);
182: }
183: if(ENTER==1)switch(menu){
184: ////////////////////////////////////Inicio programa
185: case 1:
186: if((0==seg_set)&&(0==min_set)&&(0==hours_set)){
187: Lcd_Cmd(_LCD_CLEAR); // Clear display
188: Lcd_Cmd(_LCD_CURSOR_OFF);
189: Lcd_Out(1,5,"ERROR");
190: Lcd_Out(2,1,"INGRESE TIEMPO");
191: delay_ms(2000);
192: rutina_3();
193: visual_menu(1,1,1),menu=1;
194: delay_ms(500);
195: break;
196: }
197: valida=1;
198: MenOn_off=0;
199: Lcd_Cmd(_LCD_CLEAR); // Clear LCD display
200: Lcd_Cmd(_LCD_CURSOR_OFF); // Turn cursor off
201: Read_Time();
202: Transform_Time();
203: Display_Time();
204: delay_ms(2000);
205: Lcd_Cmd(_LCD_CLEAR); // Clear display
206: Lcd_Cmd(_LCD_CURSOR_OFF);
207: Lcd_Out(1,1,"Desea Guardar");
208: Lcd_Out(2,1,"datos: Si No");
209: while (valida==1)
210: {
211: if(DOWN==1)menSi_no=12;
212: if(UP==1)menSi_no=8;
213: if(ENTER==1)valida=0;
214: Lcd_Out(2,menSi_no,"");
215: Lcd_Cmd(_LCD_BLINK_CURSOR_ON);

```

```
216: delay_ms(100);
217: }
218: if(menSi_no==8)Write_Mem_clock(0),delay_ms(10),MenOn_off=1
;//guarda datos en memoria
219: rutina_1();
220: MenOn_off=0;
221: x_lcd=1;
222: delay_ms(500);
223: break;
224: case 2:
225: rutina_2();
226: x_lcd=1;
227: delay_ms(500);
228: break;
229: case 3:
230: rutina_3();
231: x_lcd=1;
232: delay_ms(500);
233: break;
234: case 4:
235: rutina_4();
236: x_lcd=1;
237: delay_ms(500);
238: break;
239: case 5:
240: rutina_5();
241: x_lcd=1;
242: delay_ms(500);
243: break;
244: case 6:
245: rutina_6();
246: x_lcd=1;
247: delay_ms(500);
248: break;
249: case 7:
250: rutina_7();
251: x_lcd=1;
252: delay_ms(500);
253: break;
}
254: if(PARO==1)Paro_emergencia();
255: }
256: }
257: //configuraion inicial de parametros
258: void config(void){
259: ADCON1 = 0x06; //Digital out porta
260: TRISA = 0b00100000;
261: PORTA = 0;
262: TRISB = 0b01100001;
263: PORTB = 0;
264: TRISC = 0b10000000;
265: PORTC = 0;
266: TRISD = 0b00000000;
267: PORTD = 0;
268: TRISE = 0b00000111;
269: PORTE = 0;
```

```
270:
271: Lcd_Init();
272: Lcd_Cmd(_LCD_CLEAR); // Clear display
273: Lcd_Cmd(_LCD_CURSOR_OFF); // Cursor off
274:
275: UART1_Init(9600);
276:
277: delay_ms(100);
278: //Set_Time(00,00,13,1,1,12);
279:
280: Soft_I2C_Init();
281: Soft_I2C_Start();
282: Soft_I2C_Write(0xD0);
283: Soft_I2C_Write(7);
284: Soft_I2C_Write(0x10);
285: Soft_I2C_Stop();
286:
287: return;
288: }
289: //configuraion inicial de logo
290: void CustomChar1(char pos_row, char pos_char) {
291: char i;
292: Lcd_Cmd(64);
293: for (i = 0; i<=7; i++) Lcd_Chrcp(character1[i]);
294: Lcd_Cmd(_LCD_RETURN_HOME);
295: Lcd_Chrcp(pos_row, pos_char, 0);
296: }
297: //configuraion inicial de logo
298: void CustomChar2(char pos_row, char pos_char) {
299: char i;
300: Lcd_Cmd(72);
301: for (i = 0; i<=7; i++) Lcd_Chrcp(character2[i]);
302: Lcd_Cmd(_LCD_RETURN_HOME);
303: Lcd_Chrcp(pos_row, pos_char, 1);
304: }
305: //configuraion inicial de logo
306: void CustomChar3(char pos_row, char pos_char) {
307: char i;
308: Lcd_Cmd(80);
309: for (i = 0; i<=7; i++) Lcd_Chrcp(character3[i]);
310: Lcd_Cmd(_LCD_RETURN_HOME);
311: Lcd_Chrcp(pos_row, pos_char, 2);
312: }
313: //configuraion inicial de logo
314: void CustomChar4(char pos_row, char pos_char) {
315: char i;
316: Lcd_Cmd(88);
317: for (i = 0; i<=7; i++) Lcd_Chrcp(character4[i]);
318: Lcd_Cmd(_LCD_RETURN_HOME);
319: Lcd_Chrcp(pos_row, pos_char, 3);
320: }
321: //configuraion inicial de logo
322: void CustomChar5(char pos_row, char pos_char) {
323: char i;
324: Lcd_Cmd(96);
325: for (i = 0; i<=7; i++) Lcd_Chrcp(character5[i]);
```

```
326: Lcd_Cmd(_LCD_RETURN_HOME);
327: Lcd_Chr(pos_row, pos_char, 4);
328: }
329: //configuraion inicial de logo
330: void CustomChar6(char pos_row, char pos_char) {
331: char i;
332: Lcd_Cmd(104);
333: for (i = 0; i<=7; i++) Lcd_Chr_CP(character6[i]);
334: Lcd_Cmd(_LCD_RETURN_HOME);
335: Lcd_Chr(pos_row, pos_char, 5);
336: }
337: //configuraion inicial de logo
338: void CustomChar7(char pos_row, char pos_char) {
339: char i;
340: Lcd_Cmd(112);
341: for (i = 0; i<=7; i++) Lcd_Chr_CP(character7[i]);
342: Lcd_Cmd(_LCD_RETURN_HOME);
343: Lcd_Chr(pos_row, pos_char, 6);
344: }
345: //configuraion inicial de logo
346: void CustomChar8(char pos_row, char pos_char) {
347: char i;
348: Lcd_Cmd(120);
349: for (i = 0; i<=7; i++) Lcd_Chr_CP(character8[i]);
350: Lcd_Cmd(_LCD_RETURN_HOME);
351: Lcd_Chr(pos_row, pos_char, 7);
352: }
353:
354: //Rutinas de programa
355:
356: void set_temp(void) {
357:
358: Lcd_Cmd(_LCD_CLEAR);
359: Lcd_Cmd(_LCD_CURSOR_OFF);
360: Lcd_Out(1,3,"Temp. Inicial");
361: Lcd_Chr(2,9,'. ');
362: temp_set=1000;
363: while(valida==1) {
364: if(PARO==1) Paro_emergencia();
365: if((DOWN==1) && (temp_set>=0)) delay_ms(50), temp_set--;
366: if((UP==1) && (temp_set<2000)) delay_ms(50), temp_set++;
367: if(ENTER==1) {
368: SEG_SPI=1;
369: spil6send(temp_set);
370: delay_ms(10);
371: dato_ini=0; //portb.f5 del temp
372: MenOn_off=1;
373: delay_ms(200);
374: spil6receive();
375: SEG_SPI=0;
376: dato_ini=0;
377: MenOn_off=0;
378: valida=0;
379: }
380: temperatura=temp_set/10;
381: inttostr(temperatura,txt1);
```

```
382: Lcd_Out(2,3,txt1);
383: temperatura=temp_set%10;
384: bytetostr(temperatura,txt1);
385: Lcd_chr(2,10,temperatura+48);
386: }
387: temperatura=SPI_read_bajo+(SPI_read_alto*256);
388: temp_set=temperatura;
389: delay_ms(200);
390: Lcd_Out(1,3,"Temp. Seteada");
391: temperatura=temp_set/10;
392: inttostr(temperatura,txt1);
393: Lcd_Out(2,3,txt1);
394: temperatura=temp_set%10;
395: bytetostr(temperatura,txt1);
396: Lcd_chr(2,10,temperatura+48);
397: delay_ms(1000);
398: MenOn_off=0,dato_ini=0,valida=0;
399: return;
400: }
401:
402: void rutina_1(void) {
403:
404: contador=0;
405: temp_ok=0;
406: flanco=0;
407: flanco_seg=0;
408: seg_crono=0;
409: min_crono=0;
410: hours_crono=0;
411: seg_time=0;
412: min_time=0;
413: hours_time=0;
414:
415: RELE=0;
416:
417: Lcd_Cmd(_LCD_CLEAR);
418: Lcd_Out(1,5,"Inicio");
419: Lcd_Out(2,3,"DE PROCESO");
420:
421: for(rpm=2;rpm<=rpm_set;rpm++) spiMotor(rpm*2.5),delay_ms(30);
422: delay_ms(150);
423: onOff_delta=1;
424: valida=1;
425: delay_ms(100);
426: onOff_delta=0;
427:
428: for(rpm=0;rpm<=5;rpm++) PITO=1,delay_ms(50),PITO=0,delay_ms(50);
429:
430: while(valida==1) {
431: if(PARO==1) Paro_emergencia();
432: visual_contador();
433: }
434: contador=0;
435: seg_crono=0;
436: min_crono=0;
437: hours_crono=0;
```

```
438:
439: onOff_delta=1;
440: delay_ms(100);
441: onOff_delta=0;
442:
443: Lcd_Cmd(_LCD_CLEAR);
444: Lcd_Out(1,1,"Proceso");
445: Lcd_Out(2,1,"Terminado");
446:
447: for(rpm=0;rpm<=3;rpm++) PITO=1,delay_ms(50),PITO=0,delay_ms(50);
448:
449: RELE=1;
450:
451: delay_ms(100);
452: Lcd_Init();
453: Lcd_Cmd(_LCD_CLEAR);
454: Lcd_Cmd(_LCD_CURSOR_OFF);
455: Lcd_Out(1,3,"ENFRIAMIENTO");
456: Lcd_Out(2,6,"INICIADO");
457:
458: delay_ms(300);
459: seg_set=seg_setenf;
460: min_set=min_setenf;
461: hours_set=0;
462:
463: valida=1;
464: temp_ok=1;
465: option=0;
466:
467: while(valida==1){
468: visual_contador();
469: delay_ms(100);
470: if(PARO==1)Paro_emergencia();
471: }
472: UART1_Write(0x01);
473: UART1_Write(0xFF);
474: UART1_Write(0x02);
475: UART1_Write(0xFF);
476: RELE=0;
477:
478: Lcd_Init();
479: Lcd_Cmd(_LCD_CLEAR);
480: Lcd_Cmd(_LCD_CURSOR_OFF);
481:
482: onOff_delta=1;
483: delay_ms(100);
484: onOff_delta=0;
485:
486: Lcd_Out(1,3,"ENFRIAMIENTO");
487: Lcd_Out(2,6,"TERMINADO");
488: delay_ms(2000);
489: Lcd_Cmd(_LCD_CLEAR);
490: Lcd_Out(1,1,"Time:");
491: Lcd_Out(2,1,"# dato:");
492: Lcd_Chr(1,11,':');
493: Lcd_Chr(1,14,':');
```

```

494: Lcd_Chr(1,9, (hours_time/10) + 48);
495: Lcd_Chr(1,10, (hours_time%10) + 48);
496: Lcd_Chr(1,12, (min_time / 10) + 48);
497: Lcd_Chr(1,13, (min_time % 10) + 48);
498: Lcd_Chr(1,15, (seg_time / 10) + 48);
499: Lcd_Chr(1,16, (seg_time % 10) + 48);
500: inttostr(num_dato,txt1);
501: Lcd_Out(2,8,txt1);
502:
503: for(rpm=0;rpm<=2;rpm++) PITO=1,delay_ms(200),PITO=0,delay_ms(200);
504: if(menu==6)menu=0;
505: if(menu!=0)visual_menu(1,1,1),menu=1;
506:
507: seg_set=0;
508: min_set=0;
509: hours_set=0;
510: spiMotor(2);
511: if(menSi_no==8){
512: EEPROM_Write(0x04,(num_dato&0xFF00)>>8);//posmen Alto
513: delay_ms(20);
514: EEPROM_Write(0x05,(num_dato&0x00FF));//posmen Bajo
515: delay_ms(20);
516: }
517: delay_ms(2000);
518: return;
519: }
520:
521: void rutina_2(void) {
522: Lcd_Cmd(_LCD_CLEAR);
523: Lcd_Cmd(_LCD_CURSOR_OFF);
524: Lcd_Out(1,4,"SET RPM:");
525: Lcd_Out(2,1,"%: Rpm:");
526: delay_ms(100);
527: valida=1;
528: contador=1;
529: rpm=1;
530: rpm_set=0;
531: while(valida==1){
532: delay_ms(110);
533:
534: if((UP==1)&&(i<20))i++,rpm=i*5,delay_ms(50);
535: if((DOWN==1)&&(i>0))i--,rpm=i*5,delay_ms(50);
536: if(ENTER==1)valida=0,delay_ms(500);
537: if(PARO==1)Paro_emergencia();
538:
539: if(menu==0){
540: while(menu_pc==1){
541: if(UART1_Data_Ready())rpm=UART1_Read(),menu_pc=2;
542: }
543: while(menu_pc==2){
544: if(UART1_Data_Ready())rpm_set=UART1_Read(),menu_pc=0;
545: }
546: }
547: if(contador==0)valida=0;
548: bytetostr(rpm,txt1);
549: Lcd_Out(2,3,txt1);

```

```
550: spiMotor(rpm*2.5);
551: IntToStr(SPI_read_bajo*12.5,txt1);
552: Lcd_Out(2,11,txt1);
553: if(rpm_set==254){
554: valida=0;
555: delay_ms(2000);
556: spiMotor(rpm*2.5);
557: IntToStr(SPI_read_bajo*12.5,txt1);
558: Lcd_Out(2,11,txt1);
559: delay_ms(3000);
560: }
561: }
562: rpm_set=rpm;
563: spiMotor(2);
564: if(menu!=0)visual_menu(1,1,1),menu=1;
565: return;
566: }
567:
568: void rutina_3(void){
569: Lcd_Cmd(_LCD_CLEAR);
570: Lcd_Cmd(_LCD_CURSOR_OFF);
571: Lcd_Out(1,1,"Set Tiempo Proce");
572: Lcd_Out(2,1,"SET: hh:mm:ss");
573: delay_ms(500);
574: seg_crono=0;
575: min_crono=0;
576: hours_crono=0;
577: rpm=0;
578: valida=1;
579: menu_pc=1;
580: clock_max=0;
581: while((valida>=1)&&(valida<=3)){
582: if((UP==1)&&(rpm<clock_max))rpm++,delay_ms(200);//incremento
583: if((DOWN==1)&&(rpm>0))rpm--,delay_ms(200);//decremento
584: if(ENTER==1)valida++,rpm=0,delay_ms(200);
585: if(menu==0){
586: menu=10;
587: while(menu_pc==1){
588: if(UART1_Data_Ready())hours_set=UART1_Read(),menu_pc=2;
589: }
590: while(menu_pc==2){
591: if(UART1_Data_Ready())min_set=UART1_Read(),menu_pc=3;
592: }
593: while(menu_pc==3){
594: if(UART1_Data_Ready())seg_set=UART1_Read(),menu_pc=0;
595: }
596: }
597: switch(valida){
598: case 1:
599: if(menu!=10) hours_set=rpm;
600: Lcd_Chr(2,6,(hours_set / 10) + 48);
601: Lcd_Chr(2,7,(hours_set % 10) + 48);
602: clock_max=12;
603: break;
604:
605: case 2:
```

```

606: if(menu!=10)min_set=rpm;
607: Lcd_Chr(2,9,(min_set / 10) + 48);
608: Lcd_Chr(2,10,(min_set % 10) + 48);
609: clock_max=60;
610: break;
611:
612: case 3:
613: if(menu!=10)seg_set=rpm;
614: Lcd_Chr(2,12,(seg_set / 10) + 48);
615: Lcd_Chr(2,13,(seg_set % 10) + 48);
616: clock_max=60;
617: break;
618: }
619: if(menu==10)valida++,delay_ms(100);
620: }
621: if(menu==10)menu=0;
622: if(menu!=0)visual_menu(1,1,1),menu=1;
623: return;
624: }
625:
626: void rutina_4(void) {
627: portc.f0=0;
628: portc.f1=0;
629: Lcd_Cmd(_LCD_CLEAR);
630: Lcd_Cmd(_LCD_CURSOR_OFF);
631: delay_ms(100);
632: Lcd_Out(1,1,"Set Proceso Enfri");
633: Lcd_Out(2,1,"SET: mm:ss");
634: delay_ms(100);
635: seg_setenf=0;
636: min_setenf=0;
637: rpm=0;
638: valida=1;
639: clock_max=0;
640: menu_pc=1;
641: while((valida>=1)&&(valida<=2))
642: {
643: if((UP==1)&&(rpm<clock_max))rpm++,delay_ms(200);//incremento
644: if((DOWN==1)&&(rpm>0))rpm--,delay_ms(200);//decremento
645: if(ENTER==1)valida++,rpm=0,delay_ms(500);
646: if(menu==0){
647: menu=10;
648: while(menu_pc==1){
649: if(UART1_Data_Ready())min_setenf=UART1_Read(),menu_pc=2;
650: }
651: while(menu_pc==2){
652: if(UART1_Data_Ready())seg_setenf=UART1_Read(),menu_pc=0;
653: }
654: }
655: switch(valida){
656: case 1:
657: if(menu!=10)min_setenf=rpm;
658: Lcd_Chr(2,9,(min_setenf / 10) + 48);
659: Lcd_Chr(2,10,(min_setenf % 10) + 48);
660: clock_max=60;
661: break;

```

```
662:
663: case 2:
664: if(menu!=10) seg_setenf=rpm;
665: Lcd_Chr(2,12,(seg_setenf / 10) + 48);
666: Lcd_Chr(2,13,(seg_setenf % 10) + 48);
667: clock_max=60;
668: break;
669: }
670: if(menu==10) valida++,delay_ms(100);
671: }
672: if(menu==10) menu=0;
673: if(menu!=0) visual_menu(1,1,1),menu=1;
674: return;
675: }
676:
677: void rutina_5(void) {
678: valida=1;
679: Lcd_Cmd(_LCD_CLEAR); // Clear LCD display
680: Lcd_Cmd(_LCD_CURSOR_OFF); // Turn cursor off
681: while(valida==1) {
682: Read_Time();
683: Transform_Time();
684: Display_Time();
685: delay_ms(400);
686: if(ENTER==1) valida=0,delay_ms(500);
687: }
688: if(menu!=0) visual_menu(1,1,1),menu=1;
689: return;
690: }
691:
692: void rutina_6(void) {
693: portc.f0=0;
694: portc.f1=0;
695: Lcd_Cmd(_LCD_CLEAR);
696: Lcd_Cmd(_LCD_CURSOR_OFF);
697: Lcd_Out(1,1,"Date:DD.MM.YYYY.");
698: Lcd_Out(2,1,"Time:HH:MM:SS");
699: Read_Time();
700: Transform_time();
701: delay_ms(500);
702: clock_max=0;
703: rpm=0;
704: valida=1;
705: dato_selec=1;
706: while((valida>=1) && (valida<=6))
707: {
708: if((UP==1) && (rpm<clock_max)) rpm++,delay_ms(200); //incremento
709: if((DOWN==1) && (rpm>0)) rpm--,delay_ms(200); //decremento
710: if(ENTER==1) valida++,rpm=0,delay_ms(500);
711: if(menu==0) {
712: menu=10;
713: while(menu_pc==1) {
714: if(UART1_Data_Ready()) day=UART1_Read(),menu_pc=2;
715: }
716: while(menu_pc==2) {
717: if(UART1_Data_Ready()) month=UART1_Read(),menu_pc=3;
```

```
718: }
719: while(menu_pc==3) {
720: if(UART1_Data_Ready()) year=UART1_Read(), menu_pc=4;
721: }
722: while(menu_pc==4) {
723: if(UART1_Data_Ready()) hours=UART1_Read(), menu_pc=5;
724: }
725: while(menu_pc==5) {
726: if(UART1_Data_Ready()) minutes=UART1_Read(), menu_pc=6;
727: }
728: while(menu_pc==6) {
729: if(UART1_Data_Ready()) seconds=UART1_Read(), menu_pc=0;
730: }
731: }
732: switch(valida) {
733: case 1:
734: if(dato_selec==1) rpm=day, dato_selec=2;
735: if(menu!=10) day=rpm;
736: Lcd_Chr(1, 6, (day / 10) + 48);
737: Lcd_Chr(1, 7, (day % 10) + 48);
738: clock_max=31;
739: break;
740:
741: case 2:
742: if(dato_selec==2) rpm=month, dato_selec=3;
743: if(menu!=10) month=rpm;
744: Lcd_Chr(1, 9, (month / 10) + 48);
745: Lcd_Chr(1, 10, (month % 10) + 48);
746: clock_max=12;
747: break;
748:
749: case 3:
750: if(dato_selec==3) rpm=year, dato_selec=4;
751: if(menu!=10) year=rpm;
752: Lcd_Out(1, 12, "20");
753: Lcd_Chr(1, 14, (year/10) + 48);
754: Lcd_Chr(1, 15, (year % 10) + 48);
755: clock_max=60;
756: break;
757: case 4:
758: if(dato_selec==4) rpm=hours, dato_selec=5;
759: if(menu!=10) hours=rpm;
760: Lcd_Chr(2, 6, (hours / 10) + 48);
761: Lcd_Chr(2, 7, (hours % 10) + 48);
762: clock_max=23;
763: break;
764:
765: case 5:
766: if(dato_selec==5) rpm=minutes, dato_selec=6;
767: if(menu!=10) minutes=rpm;
768: Lcd_Chr(2, 9, (minutes / 10) + 48);
769: Lcd_Chr(2, 10, (minutes % 10) + 48);
770: clock_max=60;
771: break;
772:
773: case 6:
```

```
774: if(dato_selec==6)rpm=seconds,dato_selec=0;
775: if(menu!=10)seconds=rpm;
776: Lcd_Chr(2,12,(seconds / 10) + 48);
777: Lcd_Chr(2,13,(seconds % 10) + 48);
778: clock_max=60;
779: break;
780: }
781: if(menu==10)valida++,delay_ms(50);
782: }
783: if(menu==10)menu=0;
784: if(menu!=0)visual_menu(1,1,1),menu=1;
785: Set_Time(seconds,minutes,hours,day,month,year);
786: Lcd_Cmd(_LCD_CLEAR); // Clear LCD display
787: Lcd_Cmd(_LCD_CURSOR_OFF); // Turn cursor off
788: Read_Time();
789: Transform_Time();
790: Display_Time();
791: delay_ms(2000);
792: Lcd_Out(1,1,"DATOS SETEADOS");
793: if(menu==10)menu=0;
794: if(menu!=0)visual_menu(1,1,1),menu=1;
795: return;
796: }
797:
798: void rutina_7(void) {
799: char acepta=0;
800: menu=1;
801: UART1_Write(255);
802: Lcd_Cmd(_LCD_CLEAR);
803: Lcd_Cmd(_LCD_CURSOR_OFF);
804: Lcd_Out(1,1," Esperando ");
805: Lcd_Out(2,1,"Conexion puerto ");
806: delay_ms(500);
807: PITO=1,delay_ms(200),PITO=0;
808:
809: while(acepta==0) {
810: if(PARO==1)Paro_emergencia();
811: if(ENTER==1)menu=0;
812: if(UART1_Data_Ready()) {
813: menu=UART1_Read();
814: if(menu!=1) {
815: Lcd_Cmd(_LCD_CLEAR);
816: Lcd_Cmd(_LCD_CURSOR_OFF);
817: Lcd_Out(1,1," ERROR ");
818: Lcd_Out(2,1," Conexion ");
819: delay_ms(500);
820: Lcd_Cmd(_LCD_CLEAR);
821: Lcd_Cmd(_LCD_CURSOR_OFF);
822: Lcd_Out(1,1," Esperando ");
823: Lcd_Out(2,1,"Conexion puerto ");
824: }
825: if(menu==1) {
826: PITO=1,delay_ms(200),PITO=0;
827: UART1_Write(0xFF);
828: PITO=1,delay_ms(200),PITO=0;
829: delay_ms(200);
```

```

830: Lcd_Cmd(_LCD_CLEAR);
831: Lcd_Cmd(_LCD_CURSOR_OFF);
832: Lcd_Out(1,1," Puerto ");
833: Lcd_Out(2,1," Conectado ");
834: for(rpm=0;rpm<=3;rpm++) PITO=1, delay_ms(50), PITO=0, delay_ms(200
0);
835: acepta=1;
836: }
837: }
838: }
839: menu=0;
840: Lcd_Cmd(_LCD_CLEAR);
841: Lcd_Cmd(_LCD_CURSOR_OFF);
842: Lcd_Out(1,1,"Esperando comando");
843: Lcd_Out(2,1," de HMI ");
844: while(menu!=254) {
845: if(ENTER==1) {
846: menu=254;
847: for(rpm=0;rpm<=2;rpm++) PITO=1, delay_ms(50), PITO=0, delay_ms(50);
848: delay_ms(500);
849: }
850: if(UART1_Data_Ready()==1)
851: {
852: menu = UART1_Read();
853: switch(menu) {
854: //////////////////////////////////////////Inicio programa
855: case 1:
856: menu=0;
857: valida=1;
858: MenOn_off=0; //poner donde quito el otro
859: rutina_1();
860: Lcd_Init();
861: Lcd_Cmd(_LCD_CLEAR);
862: Lcd_Cmd(_LCD_CURSOR_OFF);
863: Lcd_Out(1,1,"Conexion PC");
864: for(rpm=0;rpm<=3;rpm++) PITO=1, delay_ms(50), PITO=0, delay_ms(20);
865:
866: break;
867: case 2:
868: menu=0;
869: menu_pc=1;
870: rutina_2();
871: Lcd_Init();
872: Lcd_Cmd(_LCD_CLEAR);
873: Lcd_Cmd(_LCD_CURSOR_OFF);
874: Lcd_Out(1,1,"Conexion PC");
875: for(rpm=0;rpm<=3;rpm++) PITO=1, delay_ms(50), PITO=0, delay_ms
s(20);
876:
877: break;
878: case 3:
879: menu=0;
880: rutina_3();
881: Lcd_Init();
882: Lcd_Cmd(_LCD_CLEAR);
883: Lcd_Cmd(_LCD_CURSOR_OFF);

```

```
884: Lcd_Out(1,1,"Conexion PC");
885: for(rpm=0;rpm<=3;rpm++) PITO=1, delay_ms(50), PITO=0, delay_ms(20);
886:
887: break;
888: case 4:
889: menu=0;
890: rutina_4();
891: Lcd_Init();
892: Lcd_Cmd(_LCD_CLEAR);
893: Lcd_Cmd(_LCD_CURSOR_OFF);
894: Lcd_Out(1,1,"Conexion PC");
895: for(rpm=0;rpm<=3;rpm++) PITO=1, delay_ms(50), PITO=0, delay_ms(20);
896:
897: break;
898: case 5:
899: menu=0;
900: menu_pc=1;
901: rutina_6();
902: Lcd_Init();
903: Lcd_Cmd(_LCD_CLEAR);
904: Lcd_Cmd(_LCD_CURSOR_OFF);
905: Lcd_Out(1,1,"Conexion PC");
906: for(rpm=0;rpm<=3;rpm++) PITO=1, delay_ms(50), PITO=0, delay_ms(20);
907:
908: break;
909: case 6:
910: menu=0;
911: rutina_8();
912: Lcd_Init();
913: Lcd_Cmd(_LCD_CLEAR);
914: Lcd_Cmd(_LCD_CURSOR_OFF);
915: Lcd_Out(1,1,"Conexion PC");
916: for(rpm=0;rpm<=3;rpm++) PITO=1, delay_ms(50), PITO=0, delay_ms(20);
917:
918: break;
919: }
920: }
921:
922: }
923: if(menu!=0) visual_menu(1,1,1), menu=1;
924: return;
925: }
926:
927: void rutina_8(void) {
928: num_dato=(EEPROM_Read(0x04))<<8; //posmen Alto
929: delay_ms(20);
930: num_dato=(num_dato|EEPROM_Read(0x05)); //posmen Bajo
931: delay_ms(20);
932: Read_Mem((num_dato*2)+7);
933: return;
934: }
935:
936: void visual_menu(char menu, char x_lcd, char y_lcd) {
937:
938: Lcd_Cmd(_LCD_CLEAR); // Clear display
939: Lcd_Cmd(_LCD_CURSOR_OFF);
```

```
940: if((menu==1) || (menu==2)) {
941: Lcd_Out(1,1,"1.Inicio");
942: Lcd_Out(2,1,"2.Set RPM");
943: }
944: if((menu==3) || (menu==4)) {
945: Lcd_Out(1,1,"3.Set Tiempo Pro");
946: Lcd_Out(2,1,"4.Set Tiempo Enf");
947: }
948: if((menu==5) || (menu==6)) {
949: Lcd_Out(1,1,"5.Hora");
950: Lcd_Out(2,1,"6.Set Hora");
951: }
952: if((menu==7) || (menu==8)) {
953: Lcd_Out(1,1,"7.Conexcion PC");
954: }
955: Lcd_Out(x_lcd,y_lcd,"");
956: Lcd_Cmd(_LCD_BLINK_CURSOR_ON);
957: return;
958: }
959:
960: void visual_contador(void) {
961:
962: flanco_seg=0;
963: if(PARO==1) Paro_emergencia();
964: if(ENTER==1) valida=0;
965: if((CLK_RELOJ==flanco) && (valida==1)) {
966: if((flanco==0) && (flanco_seg==0)) {
967: flanco=1;
968: flanco_seg=1;
969: }
970: if((flanco==1) && (flanco_seg==0)) {
971: flanco=0;
972: flanco_seg=1;
973: }
974: if((flanco==0) && (temp_ok==0)) {
975: if (seg_time==60) {seg_time=0;min_time++,PITO=1;}
976: if (min_time==60) {min_time=0;hours_time++;}
977: spi16receive();//recive dato esclavo envia uart
978: temperatura=SPI_read_bajo+(SPI_read_alto*256);
979: PITO=0;
980: if(temperatura>=temp_set) temp_ok=1,option=0;
981: if(menu==0) {
982: UART1_Write(0x01);
983: UART1_Write(SPI_read_bajo);
984: UART1_Write(0x02);
985: UART1_Write(SPI_read_alto);
986: }
987: if(seg_time<=20) {
988: if(option==0) {
989: Lcd_Init();
990: Lcd_Cmd(_LCD_CLEAR);
991: Lcd_Cmd(_LCD_CURSOR_OFF);
992: Lcd_Out(1,3,"Temperatura:");
993: option=1;
994: }
995: temperatura=SPI_read_bajo+(SPI_read_alto*256);
```

```
996: inttostr(temp_set/10,txt1);
997: Lcd_Out(2,8,txt1);
998: inttostr(temperatura/10,txt1);
999: Lcd_Out(2,1,txt1);
1000: }
1001: if((seg_time>20)&&(seg_time<30)){
1002: spiMotor(rpm_set*2.5);
1003: if(option==1){
1004: Lcd_Cmd(_LCD_CLEAR);
1005: Lcd_Cmd(_LCD_CURSOR_OFF);
1006: Lcd_Out(1,2,"Velocidad RPM:");
1007: Lcd_Out(2,1,"%: Rpm:");
1008: option=2;
1009: }
1010: bytetostr(rpm_set,txt1);
1011: Lcd_Out(2,3,txt1);
1012: IntToStr(SPI_read_bajo*12.5,txt1);
1013: Lcd_Out(2,11,txt1);
1014: }
1015: if (seg_time>30){
1016: if(option==2){
1017: Lcd_Init();
1018: Lcd_Cmd(_LCD_CLEAR);
1019: Lcd_Cmd(_LCD_CURSOR_OFF);
1020: Lcd_Out(1,1,"Time:");
1021: Lcd_Out(2,1,"# dato:");
1022: option=0;
1023: }
1024: Lcd_Chr(1,11,':');
1025: Lcd_Chr(1,14,':');
1026: Lcd_Chr(1,9,(hours_time/10)+48);
1027: Lcd_Chr(1,10,(hours_time%10)+48);
1028: Lcd_Chr(1,12,(min_time/10)+48);
1029: Lcd_Chr(1,13,(min_time%10)+48);
1030: Lcd_Chr(1,15,(seg_time/10)+48);
1031: Lcd_Chr(1,16,(seg_time%10)+48);
1032: inttostr(num_dato,txt1);
1033: Lcd_Out(2,8,txt1);
1034: }
1035: seg_time++;
1036: num_dato++;
1037: }
1038: if((flanco==0)&&(temp_ok==1)){
1039: if (seg_crono==60){seg_crono=0;min_crono++,PITO=1;}
1040: if (min_crono==60){min_crono=0;hours_crono++;}
1041: if (seg_time==60){seg_time=0;min_time++;}
1042: if (min_time==60){min_time=0;hours_time++;}
1043: if (option==0){
1044: Lcd_Init();
1045: Lcd_Cmd(_LCD_CLEAR);
1046: Lcd_Cmd(_LCD_CURSOR_OFF);
1047: Lcd_Out(1,1,"SetTime:");
1048: Lcd_Chr(1,11,':');
1049: Lcd_Chr(1,14,':');
1050: Lcd_Out(2,4,"Time:");
1051: Lcd_Chr(2,11,':');
```

```

1052: Lcd_Chr(2,14,':');
1053: option=1;
1054: }
1055: Lcd_Chr(1,9, (hours_set/10) + 48);
1056: Lcd_Chr(1,10, (hours_set%10) + 48);
1057: Lcd_Chr(1,12, (min_set / 10) + 48);
1058: Lcd_Chr(1,13, (min_set % 10) + 48);
1059: Lcd_Chr(1,15, (seg_set / 10) + 48);
1060: Lcd_Chr(1,16, (seg_set % 10) + 48);
1061:
1062: Lcd_Chr(2,9, (hours_crono/10) + 48);
1063: Lcd_Chr(2,10, (hours_crono%10) + 48);
1064: Lcd_Chr(2,12, (min_crono / 10) + 48);
1065: Lcd_Chr(2,13, (min_crono % 10) + 48);
1066: Lcd_Chr(2,15, (seg_crono / 10) + 48);
1067: Lcd_Chr(2,16, (seg_crono % 10) + 48);
1068:
if((seg_crono==seg_set)&&(min_crono==min_set)&&(hours_crono==hours_set
t)){valida=0;}
1069: seg_crono++;
1070: num_dato++;
1071: seg_time++;
1072: spi16receive();//recive dato esclavo envia uart
1073: PITO=0;
1074: if(menu==0){
1075: UART1_Write(0x01);
1076: UART1_Write(SPI_read_bajo);
1077: UART1_Write(0x02);
1078: UART1_Write(SPI_read_alto);
1079: }
1080: }
1081: }
1082: return;
1083: }
1084:
1085: void spiMotor(char sendRpm){
1086: SEG_SPI=1;
1087: delay_ms(1);
1088: spi8send(sendRpm);
1089: delay_us(10);
1090: spi8receive();
1091: delay_ms(1);
1092: SEG_SPI=0;
1093: delay_ms(1);
1094: return;
1095: }
1096: ///// send date to motor /////
1097: void spi8send(char SPI_SEdat){
1098: SPI_i=0;
1099: while(SPI_i<=7)
1100: {
1101: LED2_SPI=1;
1102: if((SPI_SEdat&1)==1)//Escribo un 1
1103: {
1104: DOUT_SPI=1;
1105: CLK1_SPI=1;

```

```

1106: delay_ms(5);
1107: CLK1_SPI=0;
1108: delay_ms(5);
1109: }
1110: else //Escribo un 0
1111: {
1112: DOUT_SPI=0;
1113: CLK1_SPI=1;
1114: delay_ms(5);
1115: CLK1_SPI=0;
1116: delay_ms(5);
1117: }
1118: SPI_SEdat= SPI_SEdat>>1;
1119: SPI_i++;
1120: LED2_SPI=0;
1121: }
1122: CLK1_SPI=0;DOUT_SPI=0;
1123:
1124: return;
1125: }
1126: ///// Recibo dato de motor
1127: void spi8receive(void) {
1128: SPI_read_bajo=0;
1129: SPI_i=0;
1130: while(SPI_i<=7)
1131: {
1132: CLK1_SPI=1;
1133: LED1_SPI=1;
1134: delay_ms(1);
1135: SPI_dato[SPI_i]=DIN1_SPI;
1136: CLK1_SPI=0;
1137: LED1_SPI=0;
1138: delay_ms(1);
1139: SPI_i++;
1140: }
1141:
SPI_read_bajo=SPI_dato[0]*1+SPI_dato[1]*2+SPI_dato[2]*4+SPI_dato[3]*8+S
PI_dat
to[4]*16+SPI_dato[5]*32+SPI_dato[6]*64+SPI_dato[7]*128;
1142: CLK1_SPI=0;DOUT_SPI=0;SPI_i=0;
1143: return;
1144: }
1145: ///// send date to temp /////
1146: void spi16send(int SPI_temp){
1147: SPI_i=0;
1148: while(SPI_i<=15)
1149: {
1150: LED2_SPI=1;
1151: if(( SPI_temp&1)==1) //Escribo un 1
1152: {
1153: DOUT_SPI=1;
1154: CLK2_SPI=1;
1155: delay_ms(1);
1156: CLK2_SPI=0;
1157: delay_ms(1);
1158: }

```

```

1159: else //Escribo un 0
1160: {
1161: DOUT_SPI=0;
1162: CLK2_SPI=1;
1163: delay_ms(1);
1164: CLK2_SPI=0;
1165: delay_ms(1);
1166: }
1167: SPI_temp= SPI_temp>>1;
1168: SPI_i++;
1169: delay_ms(1);
1170: LED2_SPI=0;
1171: }
1172: CLK2_SPI=0,DOUT_SPI=0;
1173: return;
1174: }
1175: ///// Recibo dato de temp
1176: void spi16receive(void) {
1177: SEG_SPI=1;
1178: delay_us(10);
1179: SPI_i=0;
1180: while(SPI_i<=15)
1181: {
1182: CLK2_SPI=1;
1183: LED2_SPI=1;
1184: delay_ms(1);
1185: SPI_dato[SPI_i]=DIN2_SPI;
1186: LED2_SPI=DIN2_SPI;
1187: CLK2_SPI=0;
1188: LED2_SPI=0;
1189: delay_ms(1);
1190: SPI_i++;
1191: }
1192: delay_us(10);
1193: SEG_SPI=0;
1194:
SPI_read_bajo=SPI_dato[0]*1+SPI_dato[1]*2+SPI_dato[2]*4+SPI_dato[3]*8+S
PI_dat
to[4]*16+SPI_dato[5]*32+SPI_dato[6]*64+SPI_dato[7]*128;
1195:
SPI_read_alto=SPI_dato[8]*1+SPI_dato[9]*2+SPI_dato[10]*4+SPI_dato[11]*8
+SPI_d
dato[12]*16+SPI_dato[13]*32+SPI_dato[14]*64+SPI_dato[15]*128;
1196: return;
1197: }
1198:
1199: //----- Reads time and date information from RTC
(PCF8583)
1200: void Read_Time() {
1201: Soft_I2C_Start(); // Issue start signal
1202: Soft_I2C_Write(0xD0);
1203: Soft_I2C_Write(0); // Start from address 2
1204: Soft_I2C_Start(); // Issue repeated start signal
1205: Soft_I2C_Write(0xD1); // Address PCF8583 for reading R/W=1
1206:
1207: seconds = Soft_I2C_Read(1); // Read seconds byte

```

```

1208: minutes = Soft_I2C_Read(1); // Read minutes byte
1209: hours = Soft_I2C_Read(1); // Read hours byte
1210: day_num = Soft_I2C_Read(1);
1211: day = Soft_I2C_Read(1);
1212: month = Soft_I2C_Read(1); // Read weekday/month byte
1213: year = Soft_I2C_Read(0);
1214: Soft_I2C_Stop(); // Issue stop signal
1215: }
1216:
1217: //----- Formats date and time
1218: void Transform_Time() {
1219: seconds = ((seconds & 0xF0) >> 4)*10 + (seconds & 0x0F);
1220: minutes = ((minutes & 0xF0) >> 4)*10 + (minutes & 0x0F);
1221: hours = ((hours & 0xF0) >> 4)*10 + (hours & 0x0F);
1222: year = ((year & 0xF0) >> 4)*10 + (year & 0x0F);
1223: day = ((day & 0x30) >> 4)*10 + (day & 0x0F);
1224: month = ((month & 0x10) >> 4)*10 + (month & 0x0F);
1225: }
1226:
1227: //----- Output values to LCD
1228: void Display_Time() {
1229: Lcd_Out(1,1,"Date:"); // Prepare and output static text on LCD
1230: Lcd_Chr(1,8,'. ');
1231: Lcd_Chr(1,11,'. ');
1232: Lcd_Chr(1,16,'. ');
1233: Lcd_Out(2,1,"Time:");
1234: Lcd_Chr(2,8,': ');
1235: Lcd_Chr(2,11,': ');
1236: Lcd_Out(1,12,"201"); // start from year 2010
1237:
1238: Lcd_Chr(1, 6, (day / 10) + 48);
1239: Lcd_Chr(1, 7, (day % 10) + 48);
1240: Lcd_Chr(1, 9, (month / 10) + 48);
1241: Lcd_Chr(1,10, (month % 10) + 48);
1242: Lcd_Chr(1,14, (year/10) + 48);
1243: Lcd_Chr(1,15, (year %10) + 48);
1244:
1245: Lcd_Chr(2, 6, (hours / 10) + 48);
1246: Lcd_Chr(2, 7, (hours % 10) + 48);
1247: Lcd_Chr(2, 9, (minutes / 10) + 48);
1248: Lcd_Chr(2,10, (minutes % 10) + 48);
1249: Lcd_Chr(2,12, (seconds / 10) + 48);
1250: Lcd_Chr(2,13, (seconds % 10) + 48);
1251: }
1252:
1253: void Set_Time(char seconds,char minutes,char hours,char day,char
month,char year)
{
1254: Soft_I2C_Start();
1255: Soft_I2C_Write (0xD0); // Address of DS1307
1256: Soft_I2C_Write (0); // Start from address 0
1257: Soft_I2C_Write (Dec2Bcd(seconds));
1258: Soft_I2C_Write (Dec2Bcd(minutes)); // Read seconds byte
1259: Soft_I2C_Write (Dec2Bcd(hours)); // Read seconds byte
1260: Soft_I2C_Write (7);
1261: Soft_I2C_Write (Dec2Bcd(day)); // Read seconds byte

```

```

1262: Soft_I2C_Write (Dec2Bcd(month)); // Read seconds byte
1263: Soft_I2C_Write (Dec2Bcd(year)); // Read seconds byte
1264: Soft_I2C_Stop();
1265: }
1266:
1267: void Write_Mem_clock(int address){
1268: //address high and low
1269: high_address1 =(address&0xFF00)>>8;
1270: low_address1 =(address&0x00FF);
1271: //write in 24ls512
1272: Soft_I2C_Start(); // Issue start signal
1273: Soft_I2C_Write(0xA0); // Address (A0) 24ls512 write
1274: Soft_I2C_Write(high_address1); // Address 24ls512
1275: Soft_I2C_Write(low_address1); // Address 24ls512
1276: Soft_I2C_Write(day_num);
1277: Soft_I2C_Write(day);
1278: Soft_I2C_Write(month);
1279: Soft_I2C_Write(year);
1280: Soft_I2C_Write(hours);
1281: Soft_I2C_Write(minutes);
1282: Soft_I2C_Write(seconds);
1283: Soft_I2C_Stop();
1284: return;
1285: }
1286:
1287: void Read_Mem(int address){
1288: int nulldat=511;
1289: portc.f0=0;
1290: portc.f1=0;
1291: contador=0;
1292: Lcd_Cmd( LCD_CLEAR);
1293: Lcd_Cmd( LCD_CURSOR_OFF);
1294: lcd_out(1,1,"Numero de datos:");
1295: inttostr(address,txt1);
1296: lcd_out(2,1,txt1);
1297: delay_ms(100);
1298:
1299: Soft_I2C_Start(); // Issue start signal
1300: Soft_I2C_Write(0xA0); // Address (A0) 24ls512 write
1301: Soft_I2C_Write(0); // Address 24ls512
1302: Soft_I2C_Write(0); // Address 24ls512
1303: Soft_I2C_Start(); // Issue repeated start signal
1304: Soft_I2C_Write(0xA1);
1305: while(contador<address){
1306: SPI_read_bajo = Soft_I2C_Read(1); // Read seconds byte
1307:
if((contador!=nulldat)&&(contador!=nulldat+1))UART1_Write(SPI_read_bajo
);
1308: if(contador==nulldat+1)nulldat=nulldat+512,address=address-2;
1309: contador++;
1310: }
1311: SPI_read_bajo = Soft_I2C_Read(00);
1312: Soft_I2C_Stop();
1313: UART1_Write(0xFF);
1314: UART1_Write(0xFF);
1315: inttostr(contador,txt1);

```

```
1316: lcd_out(2,8,txt1);
1317: delay_ms(100);
1318: return;
1319: }
1320:
1321: void Paro_emergencia(void) {
1322: MenOn_off=1;
1323: dato_ini=1;
1324: spi16receive();
1325: MenOn_off=0;
1326: dato_ini=0;
1327: Lcd_Init();
1328: Lcd_Cmd(_LCD_CLEAR);
1329: Lcd_Cmd(_LCD_CURSOR_OFF);
1330: delay_ms(100);
1331: Lcd_Out(1,3,"REINICIE EL");
1332: Lcd_Out(2,6,"SISTEMA");
1333: delay_ms(1000);
1334: Lcd_Cmd(_LCD_TURN_OFF);
1335:
for (rpm=0; rpm<=3; rpm++) RELE=1, PITO=1, delay_ms(1000), RELE=0, PITO=0, delay
_ms(
1000);
1336: RELE=1;
1337: delay_ms(100);
1338: Lcd_Init();
1339: Lcd_Cmd(_LCD_TURN_ON);
1340: Lcd_Cmd(_LCD_CLEAR);
1341: Lcd_Cmd(_LCD_CURSOR_OFF);
1342: Lcd_Out(1,3,"REINICIE EL");
1343: Lcd_Out(2,6,"SISTEMA");
1344: while (1) RELE=1, PITO=1, delay_ms(500), PITO=0, delay_ms(5000);
1345:
1346: return;
1347: }
```

ANEXO C

CÓDIGO FUENTE PROGRAMA CONTROL DE VELOCIDAD³⁴

```

2: #define encoder portb.f4
3: char SPI_dato[9];
4: char SPI_REdat=0;
5: char SPI_SEdat=0;
6: char SPI_recibo=0;
7:
8: char pwm=0;
9: int ppT=0;
10: int pulsos=0;
11: char cont=0;
12:
13: void main (void);
14: void interrupt (void);
15: void config (void);
16: void SPI_sclavere(void);
17:
18: void interrupt(void) {
19:   INTCON.GIE=0; //Disable global interrupts
20:   if(PIR1.TMR1IF) {
21:     PIR1.TMR1IF = 0;
22:     TMR1H=0xCF;
23:     TMR1L=0x2C;
24:     ppT=pulsos;
25:     pulsos=0;
26:   }
27:   if(INTCON.RBIF){ //encoder
28:     if(portb.f4==0)pulsos++;
29:     INTCON.RBIF=0;
30:   }
31:   if(INTCON.INTF){
32:     SPI_SEdat=ppT; //dato que envio RPM
33:     SPI_sclavere();
34:     pwm=SPI_recibo*1.95; //dato que recibo RPM
35:     PWM1_Set_Duty(pwm);
36:     EEPROM_Write(0x00,SPI_recibo);
37:     INTCON.INTF = 0;
38:   }
39:   INTCON.GIE=1;
40:   return;
41: }
42:
43: void main(void) {
44:   config(void);
45:   PWM1_Start();
46:   pwm=EEPROM_Read(0x00);
47:   while(1) {
48:     delay_ms(1000);
49:     asm CLRWDI;
50:   }
51: }

```

³⁴ Para mayor detalle referirse al CD-ROM adjunto.

```

52:
53: void config(void) {
54: PCON.OSCF = 1; //4mhz
55: Delay_us(50); // Retardo para estabilizar el oscilador LP
56: CMCON = 0x07; //Turn comparators off and enable pins for I/O
    functions
57: TRISA = 0b10110101; //config port
58: PORTA = 0;
59: TRISB = 0B11110111; //config port
60: PORTB = 0; //initialize port
61:
62: //configure TMR0
63: OPTION_REG = 0x00;
64: OPTION_REG.TOCS=0;
65: OPTION_REG.PSA=1; //Asignacion a WDT
66: OPTION_REG.PS2=1; /**prescaler 1:128
67: OPTION_REG.PS1=1; /**
68: OPTION_REG.PS0=1; /**
69: asm CLRWDT;
70:
71: T1CON = 0B00011101;
72: TMR1H=0xCF;
73: TMR1L=0x2C;
74: PIE1.TMR1IE = 1; //TMR1IE: Hab. Timer1 Overflow interrupt.
75: PIR1.TMR1IF = 0; //Hago la bandera 0
76:
77: //configure PWM use the TMR2
78: PWM1_Init(15600);
79:
80: //configure interrupts
81: INTCON.GIE = 1; //enable interrupts
82: INTCON.PEIE = 1; //Enables peripheral interrupts
83: INTCON.INTE = 1; // habilito la interrupcion de RBO
84: INTCON.RBIE = 1; //RB port change interrupt
85: INTCON.RBIF = 0; //RB banderas de ocurrencia RB<7:4>
86: INTCON.INTF = 0; //RBO bandera en cero
87: return;
88: }
89:
90: ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
91: /// RECIBE CONFIGUTACION CONFIGURACION
92: ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
93: void SPI_sclavere(void)
94: {
95: char count_dato=0;
96: while(porta.f0==1)
97: {
98: while((porta.f0==1) && (count_dato<8)) {
99: while((portb.f0==1) && (porta.f0==1)) {
100: SPI_dato[count_dato]=porta.f2;
101: porta.f3=1;
102: };
103: while((portb.f0==0) && (porta.f0==1)) {porta.f3=0;}
104: count_dato++;
105: }
106: count_dato=0;

```

```
107: while ((porta.f0==1) && (count_dato<8)) {
108: while ((portb.f0==1) && (porta.f0==1)) {
109: porta.f1=0b1&SPI_SEdat;
110: porta.f3=1;
111: }
112: count_dato++;
113: SPI_SEdat=SPI_SEdat>>1;
114: while ((portb.f0==0) && (porta.f0==1)) {porta.f3=0;}
115: }
116: }
117: SPI_recibo=SPI_dato[0]*1+SPI_dato[1]*2+SPI_dato[2]*4+SPI_dato[3]*8+
SPI_dato[4]*16+
SPI_dato[5]*32+SPI_dato[6]*64+SPI_dato[7]*128;
118: return;
119: }
```

ANEXO D

CÓDIGO FUENTE PROGRAMA CONTROL DE TEMPERATURA.

```

1: //-----//
2: // Configuro nombres de puerto I2C //
3: //-----//
4: // Software I2C connections
5: sbit Soft_I2C_Scl at RC3_bit;
6: sbit Soft_I2C_Sda at RC4_bit;
7: sbit Soft_I2C_Scl_Direction at TRISC3_bit;
8: sbit Soft_I2C_Sda_Direction at TRISC4_bit;
9: // End Software I2C connections
10: //////////////////////////////////////////////////
11: //configuration of SPI
12: #define CLK2_SPI portb.f0;
13: #define SEG_SPI portb.f1;
14: #define DIN2_SPI portb.f2;
15: #define DOUT_SPI portb.f3;
16: #define LEDT_SPI portb.f4;
17: #define dato_ini portb.f5;
18: //configuration of port activation system
19: #define onOff portc.f0
20: #define menOn_off portc.f1
21: #define CLK_RELOJ portc.f2
22: #define enable_modbus portc.f5
23: ///////
24:
25: void interrupt(void);
26: void main(void);
27: void config(void);
28: void Write_Mem(int address, int dat);
29: void modbus_TX(char adress, char function, int starReg, int
numReg);
30: void modbus_RX(void);
31:
32: char count_dato=0;
33: char valida=0;
34: char contMenH=0;
35: char contMenL=0;
36: int cont_men=4;
37: int prac=0;
38: char dato[15];
39: int valor=0;
40: int send_dat=0;
41: char flanco=1;
42: char flanco_seg=0;
43: char seg_onOff=0;
44: char emergencia=0;
45: char configura=0;
46: char SPI_dato[16];
47: int SPI_read_bajo=0;
48: int SPI_read_alto=0;
49: int SPI_read=0;
50: char low_dat=0, high_dat=0, check_low=0, check_high=0;
51: int high_address1=0, low_address1=0;

```

```

52: char seg=0;
53: char calcula_LRC(char dat[17]);
54: char asciiToHex(char dato);
55:
56:
57: //-----//
58: // Variables de modbus //
59: //-----//
60: //-----//
61: char dat[17], resp[13], txt[9], evenHex[6], calcLrc, i, mdat, x;
62: int m_dato;
63: //-----//
64:
65:
66: //-----//
67: /// INTERRUPCIONES ///
68: //-----//
69: void interrupt(void)
70: {
71: INTCON.GIE=0;
72: if (INTCON.INTF)
73: {
74: char count_dato=0;
75: while(portb.f1==1)
76: { if(configura==0){
77: while((portb.f1==1)&&(count_dato<16)){
78: while((portb.f0==1)&&(portb.f1==1)){SPI_dato[count_dato]=portb.
f3,portb.f4=1;}
79: while((portb.f0==0)&&(portb.f1==1)){portb.f4=0;}
80: count_dato++;
81: }
82:
SPI_read_bajo=SPI_dato[0]*1+SPI_dato[1]*2+SPI_dato[2]*4+SPI_dato[3]*8+
SPI_dato[4]*16+SPI_dato[5]*32+SPI_dato[6]*64+SPI_dato[7]*128;
83:
SPI_read_alto=SPI_dato[8]*1+SPI_dato[9]*2+SPI_dato[10]*4+SPI_dato[11]*
8+SPI_dato[12]*16+SPI_dato[13]*32+SPI_dato[14]*64+SPI_dato[15]*128;
84: SPI_read=SPI_read_bajo+SPI_read_alto*256;
85: send_dat=SPI_read;
86: }
87: count_dato=0;
88: while((portb.f1==1)&&(count_dato<16)){
89: while((portb.f0==1)&&(portb.f1==1)){
90: portb.f2=0b1&send_dat,portb.f4=1;
91: }
92: count_dato++;
93: send_dat=send_dat>>1;
94: while((portb.f0==0)&&(portb.f1==1)){portb.f4=0;}
95: }
96: portb.f5=0;
97: }
98: if((portb.f5==1)&&(portc.f1==1)) emergencia=1;
99: INTCON.INTF = 0;
100: }
101: INTCON.GIE=1;
102: return;

```

```
103: }
104:
105: void main(void) {
106: config(void);
107: SPI_dato[0]=1;
108: SPI_dato[1]=0;
109: SPI_dato[2]=1;
110: valida=1;
111: delay_ms(500);
112: if(portb.f5==1) {
113: while(valida==1) {
114: if((portb.f5==0) && (portc.fl==1)) {
115: configura=1;
116: }
117:
118: if((portb.f5==0) && (portc.fl==0)) modbus_TX(1,6,18177,SPI_read), valida=0,
119: configura=1;
120: }
121: EEPROM_Write(0x00,0x00); //encendido
122: delay_ms(20);
123: EEPROM_Write(0x01,0x00); //apagado
124: delay_ms(20);
125: EEPROM_Write(0x02,0x00); //enfriamiento
126: delay_ms(20);
127: EEPROM_Write(0x03,0x00); //valida
128: delay_ms(20);
129: EEPROM_Write(0x04,0x00); //posmen
130: delay_ms(20);
131: EEPROM_Write(0x05,0x04); //posmen
132: }
133: modbus_TX(1,6,18211,0);
134: if(EEPROM_Read(0x00)) {
135: modbus_TX(1,6,18211,1);
136: //asm CLRWDT;
137: delay_ms(500);
138: valida=1;
139: porta.f0=0;
140: porta.fl=1;
141: seg_onOff=1;
142: } //encendido
143: if(EEPROM_Read(0x01)) {
144: modbus_TX(1,6,18211,0);
145: //asm CLRWDT;
146: delay_ms(500);
147: valida=1;
148: porta.f0=0;
149: porta.fl=1;
150: seg_onOff=2;
151: } //apagado
152: if(EEPROM_Read(0x02)) {
153: modbus_TX(1,6,18211,0);
154: // asm CLRWDT;
155: delay_ms(500);
156: valida=0;
157: porta.f0=1;
158: porta.fl=0;
```

```
157: seg_onOff=2;
158: }//apagado
159: if (emergencia)modbus_TX(1,6,18211,0);
160: contMenH=EEPROM_Read(0x04);
161: delay_ms(20);
162: contMenL=EEPROM_Read(0x05);//valida
163: //cont_men=(contMenH<<8)|contMenL;
164:
165: flanco=1;
166: m_dato=1;
167: seg_onOff=0;
168: while(1){
169: flanco_seg=0;
170: if (emergencia)modbus_TX(1,6,18211,0);
171: if((onOff==1)&&(seg_onOff==0)){
172: //adress,function,star register,num register ->on delta
173: modbus_TX(1,6,18211,1);
174: asm CLRWDT;
175: delay_ms(500);
176: valida=1;
177: porta.f0=0;
178: porta.f1=1;
179: seg_onOff=1;
180: //prac=43690;
181: EEPROM_Write(0x00,0x01);
182: delay_ms(20);
183: EEPROM_Write(0x01,0x00);
184: delay_ms(20);
185: EEPROM_Write(0x02,0x00);
186: delay_ms(20);
187: EEPROM_Write(0x03,0x01);//valida
188: }
189: if((onOff==1)&&(seg_onOff==1)){
190: //adress,function,star register,num register ->off delta
191: //envio posicion de memoria
192: modbus_TX(1,6,18211,0);
193: asm CLRWDT;
194: delay_ms(500);
195: seg_onOff=2;
196: //prac=56797;
197: EEPROM_Write(0x00,0x00);
198: delay_ms(20);
199: EEPROM_Write(0x01,0x01);
200: delay_ms(20);
201: EEPROM_Write(0x02,0x00);
202: delay_ms(20);
203: EEPROM_Write(0x03,0x01);//valida
204: }
205: if((onOff==1)&&(seg_onOff==2)){
206: //adress,function,star register,num register ->off delta
207: //envio posicion de memoria
208: asm CLRWDT;
209: delay_ms(500);
210: valida=0;
211: porta.f0=1;
212: porta.f1=0;
```

```

213: seg_onOff=0;
214: EEPROM_Write(0x00,0x00);
215: delay_ms(20);
216: EEPROM_Write(0x01,0x00);
217: delay_ms(20);
218: EEPROM_Write(0x02,0x01);
219: delay_ms(20);
220: EEPROM_Write(0x03,0x01);//valida
221: delay_ms(20);
222: EEPROM_Write(0x04,0x00);//valida
223: delay_ms(20);
224: EEPROM_Write(0x05,0x04);//valida
225: }
226: if((CLK_RELOJ==flanco) &&(valida==1)){
227:
228: if((flanco==0) &&(flanco_seg==0)){
229: flanco=1;
230: flanco_seg=1;
231: }
232: if((flanco==1) &&(flanco_seg==0)){
233: flanco=0;
234: flanco_seg=1;
235: }
236: if(flanco==0){
237: modbus_TX(1,3,4096,1);// adress,function, star register, num reg
gister
238: modbus_RX();
239: asm CLRWDT;
240: send_dat=m_dato;
241: if(menOn_off==1){
242:
243: Write_Mem(cont_men,m_dato);//poner el dato que voy a grabar
244: cont_men++;
245: contMenH=(cont_men & 0xFF00)>>4;
246: contMenL=(cont_men & 0x00FF);
247: //EEPROM_Write(0x04,contMenH);//posicion de memoria alta
248: delay_ms(20);
249: //EEPROM_Write(0x05,contMenL);//posicion de memoria baja
250:
251: }
252: }
253: }
254: }
255: }
256:
257: //CONFIGURACION INICIAL DE LOS PARAMETROS
258: void config(void)
259: {
260: ADCON1 = 7; //Digital out porta
261: TRISA = 0b11111100; //config port
262: PORTA = 1; //initialize port
263: porta.f0=1;
264: porta.f1=0;
265: TRISB = 0b00101011; //config port
266: PORTB = 0; //initialize port
267: TRISC = 0b10000111; //config port

```

```

268: PORTC = 0; //initialize port
269:
270:
271: UART1_Init(9600); // Initialize UART module at 9600 bps
272:
273: INTCON.GIE=1;
274: INTCON.INTF = 0;
275: INTCON.INTE=1;
276: OPTION_REG.INTEDG=1;
277: PEIE_bit = 1; // enable peripheral interrupts
278:
279: Delay_ms(100);
280:
281: //configure TMR0
282: /*OPTION_REG = 0x00;
283: OPTION_REG.TOCS=0;
284: OPTION_REG.PSA=1; //Asignacion a WDT
285: OPTION_REG.PS2=1; //---prescaler 1:128
286: OPTION_REG.PS1=1; //---
287: OPTION_REG.PS0=1; //---
288: asm CLRWDWT;*/
289: return;
290: }
291:
292: void Write_Mem(int address, int dat){
293: char veri=0;
294:
295: address=(address*2)-1;
296: //address high and low
297: high_address1 = (address&0xFF00)>>8; //revisar esto
298: low_address1 = (address&0x00FF);
299:
300: if((low_address1==127)&&(seg==0)) low_address1=255;
301: if((low_address1==255)&&(seg==1)) low_address1=127,high_address1++;
302: if(low_address1<=127) seg=0;
303: if(low_address1>127) seg=1;
304:
305: //SEPARO EL DATO INT EN 2 CHAR
306: high_dat = (dat&0xFF00)>>8; //revisar esto
307: low_dat = (dat&0x00FF);
308: //i2c grabado
309: Soft_I2C_Start(); // Issue start signal
310: Soft_I2C_Write(0xA0); // Address (A0) 24ls512 write
311: Soft_I2C_Write(high_address1); // Address 24ls512
312: Soft_I2C_Write(low_address1);
313: Soft_I2C_Write(high_dat);
314: Soft_I2C_Write(low_dat); // Address 24ls512
315: Soft_I2C_Stop();
316: delay_ms(10);
317: //i2c lectura
318: Soft_I2C_Start(); // Issue start signal
319: Soft_I2C_Write(0xA0); // Address (A0) 24ls512 write
320: Soft_I2C_Write(high_address1); // Address 24ls512
321: Soft_I2C_Write(low_address1); // Address 24ls512
322: Soft_I2C_Start(); // Issue start signal
323: Soft_I2C_Write(0xA1); // Address (A0) 24ls512 write

```

```

324: check_high=Soft_I2C_Read(1);
325: check_low =Soft_I2C_Read(0);
326: Soft_I2C_Stop();
327:
328: while ((high_dat!=check_high)&&(low_dat!=check_low)&&(veri<=5)) {
329: //i2c grabado
330: Soft_I2C_Start(); // Issue start signal
331: Soft_I2C_Write(0xA0); // Address (A0) 24ls512 write
332: Soft_I2C_Write(high_address1); // Address 24ls512
333: Soft_I2C_Write(low_address1); // Address 24ls512
334: Soft_I2C_Write(high_dat);
335: Soft_I2C_Write(low_dat); // Address 24ls512
336: Soft_I2C_Stop();
337: delay_ms(10);
338: //i2c lectura
339: Soft_I2C_Start(); // Issue start signal
340: Soft_I2C_Write(0xA0); // Address (A0) 24ls512 write
341: Soft_I2C_Write(high_address1); // Address 24ls512
342: Soft_I2C_Write(low_address1); // Address 24ls512
343: Soft_I2C_Start(); // Issue start signal
344: Soft_I2C_Write(0xA1); // Address (A0) 24ls512 write
345: check_high=Soft_I2C_Read(1);
346: check_low =Soft_I2C_Read(0);
347: Soft_I2C_Stop();
348: veri++;
349: }
350: return;
351: }
352: //SUBROUTINA DEL MODBUS DE TRANSMISION
353: void modbus_TX(char adress,char function, int starReg, int
numReg) {
354: dat[0]=':', dat[15]=0x0D, dat[16]=0x0A;
355: LongWordToHex(adress,txt), dat[1]=txt[6], dat[2]=txt[7];
356: LongWordToHex(function,txt), dat[3]=txt[6], dat[4]=txt[7];
357: LongWordToHex(starReg,txt);
358: for (i=4;i<=7;i++) dat[i+1]=txt[i];
359: LongWordToHex(numReg,txt);
360: for (i=4;i<=7;i++) dat[i+5]=txt[i];
361: calcLrc=calcula_Lrc(dat);
362: LongWordToHex(calcLrc,txt), dat[13]=txt[6], dat[14]=txt[7];
363: enable_modbus=1;
364: for (i=0;i<=17;i++) UART1_Write(dat[i]);
365: enable_modbus=0;
366: return;
367: }
368: //SUBROUTINA DEL MODBUS DE RECEPCION
369: void modbus_RX(void) {
370: for (i=0;i<=16;i++) dat[i]=0x00, resp[i]=0x00;
371: if (UART1_DATA_READY()) UART1_Read_Text(dat,"ok",15);
372: dat[15]= dat[13], dat[16]= dat[14];
373: dat[13]= dat[15], dat[14]= dat[16];
374: for (i=0;i<=14;i++) dat[i]=dat[i+2];
375: dat[0]=0x3A, dat[11]= 0x00,dat[12]= 0x00;
376: calcLrc=calcula_Lrc(dat);
377: LongWordToHex(calcLrc,txt), dat[15]=txt[6], dat[16]=txt[7];
378: for (i=7;i<=10;i++) resp[i]=asciiToHex(dat[i]);

```

```
379: m_dato=0;
380: m_dato=resp[10]+resp[9]*16+resp[8]*256+resp[7]*4096;
381: if ((dat[13]==dat[15])&&(dat[14]==dat[16])){
382: return ;
383: }
384: else {
385: m_dato=65535;
386: return;
387: }
388: return;
389: }
390: //CALCULA EL LRC DE LA TRAMA PARA COMPROBAR SI ES CORRECTA
391: char calcula_Lrc(char dat[17]){
392: x=0, calcLrc=0x00;
393: for (i=1;i<=12;i++) resp[i]=asciiToHex(dat[i]);
394: for (i=1;i<=11;i=2+i){
395: evenHex[x]=resp[i]<<4;
396: evenHex[x]=evenHex[x]|resp[i+1];
397: x++;
398: }
399: for (i=0;i<=5;i++)calcLrc=evenHex[i]+calcLrc;
400: calcLrc=(0xFF-calcLrc)+0x01;
401: return(calcLrc);
402: }
403: //CONVERSION DE ASCII A HEX
404: char asciiToHex(char mdatο){
405: switch (mdatο){
406: case 0x30: mdatο = 0x00; break; // 0
407: case 0x31: mdatο = 0x01; break; // 1
408: case 0x32: mdatο = 0x02; break; // 2
409: case 0x33: mdatο = 0x03; break; // 3
410: case 0x34: mdatο = 0x04; break; // 4
411: case 0x35: mdatο = 0x05; break; // 5
412: case 0x36: mdatο = 0x06; break; // 6
413: case 0x37: mdatο = 0x07; break; // 7
414: case 0x38: mdatο = 0x08; break; // 8
415: case 0x39: mdatο = 0x09; break; // 9
416: case 0x41: mdatο = 0x0A; break; // A
417: case 0x42: mdatο = 0x0B; break; // B
418: case 0x43: mdatο = 0x0C; break; // C
419: case 0x44: mdatο = 0x0D; break; // D
420: case 0x45: mdatο = 0x0E; break; // E
421: case 0x46: mdatο = 0x0F; break; // F
422: }
423: return(mdatο);
424: }
```

ANEXO E DATA SHEET 18F452



PIC18FXX2

28/40-pin High Performance, Enhanced FLASH Microcontrollers with 10-Bit A/D

High Performance RISC CPU:

- C compiler optimized architecture/instruction set
 - Source code compatible with the PIC16 and PIC17 instruction sets
- Linear program memory addressing to 32 Kbytes
- Linear data memory addressing to 1.5 Kbytes

Device	On-Chip Program Memory		On-Chip RAM (bytes)	Data EEPROM (bytes)
	FLASH (bytes)	# Single Word Instructions		
PIC18F242	16K	8192	768	256
PIC18F252	32K	16384	1536	256
PIC18F442	16K	8192	768	256
PIC18F452	32K	16384	1536	256

- Up to 10 MIPS operation:
 - DC - 40 MHz osc./clock input
 - 4 MHz - 10 MHz osc./clock input with PLL active
- 16-bit wide instructions, 8-bit wide data path
- Priority levels for interrupts
- 8 x 8 Single Cycle Hardware Multiplier

Peripheral Features:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter with 8-bit period register (time-base for PWM)
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules. CCP pins that can be configured as:
 - Capture input: capture is 16-bit, max. resolution 6.25 ns (Tcy/16)
 - Compare is 16-bit, max. resolution 100 ns (Tcy)
 - PWM output: PWM resolution is 1- to 10-bit, max. PWM freq. @: 8-bit resolution = 156 kHz, 10-bit resolution = 39 kHz
- Master Synchronous Serial Port (MSSP) module, Two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I²C™ Master and Slave mode

Peripheral Features (Continued):

- Addressable USART module:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

Analog Features:

- Compatible 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during SLEEP
 - Linearity ≤ 1 LSB
- Programmable Low Voltage Detection (PLVD)
 - Supports interrupt on-Low Voltage Detection
- Programmable Brown-out Reset (BOR)

Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced FLASH program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory
- FLASH/Data EEPROM Retention: > 40 years
- Self-reprogrammable under software control
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own On-Chip RC Oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options including:
 - 4X Phase Lock Loop (of primary oscillator)
 - Secondary Oscillator (32 kHz) clock input
- Single supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins

CMOS Technology:

- Low power, high speed FLASH/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Industrial and Extended temperature ranges
- Low power consumption:
 - < 1.6 mA typical @ 5V, 4 MHz
 - 25 μ A typical @ 3V, 32 kHz
 - < 0.2 μ A typical standby current

ANEXO F DATA SHEET 16F876A³⁵



PIC16F87XA

28/40/44-Pin Enhanced Flash Microcontrollers

Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input
DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory,
Up to 368 x 8 bytes of Data Memory (RAM),
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin
PIC16CXXX and PIC16FXXX microcontrollers

Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,
can be incremented during Sleep via external
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™
(Master mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver
Transmitter (USART/SCI) with 9-bit address
detection
- Parallel Slave Port (PSP) – 8 bits wide with
external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for
Brown-out Reset (BOR)

Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital
Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
 - Two analog comparators
 - Programmable on-chip voltage reference
(VREF) module
 - Programmable input multiplexing from device
inputs and internal voltage reference
 - Comparator outputs are externally accessible

Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash
program memory typical
- 1,000,000 erase/write cycle Data EEPROM
memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™)
via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC
oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

CMOS Technology:

- Low-power, high-speed Flash/EEPROM
technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I ² C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

³⁵ Para mayor información visitar: <http://www.microchip.com/>

ANEXO G DATA SHEET 16F648A³⁶



MICROCHIP PIC16F627A/628A/648A

18-pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology

High-Performance RISC CPU:

- Operating speeds from DC – 20 MHz
- Interrupt capability
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes
- 35 single-word instructions:
 - All instructions single cycle except branches

Special Microcontroller Features:

- Internal and external oscillator options:
 - Precision internal 4 MHz oscillator factory calibrated to $\pm 1\%$
 - Low-power internal 48 kHz oscillator
 - External Oscillator support for crystals and resonators
- Power-saving Sleep mode
- Programmable weak pull-ups on PORTB
- Multiplexed Master Clear/Input-pin
- Watchdog Timer with independent oscillator for reliable operation
- Low-voltage programming
- In-Circuit Serial Programming™ (via two pins)
- Programmable code protection
- Brown-out Reset
- Power-on Reset
- Power-up Timer and Oscillator Start-up Timer
- Wide operating voltage range (2.0-5.5V)
- Industrial and extended temperature range
- High-Endurance Flash/EEPROM cell:
 - 100,000 write Flash endurance
 - 1,000,000 write EEPROM endurance
 - 40 year data retention

Low-Power Features:

- Standby Current:
 - 100 nA @ 2.0V, typical
- Operating Current:
 - 12 μ A @ 32 kHz, 2.0V, typical
 - 120 μ A @ 1 MHz, 2.0V, typical
- Watchdog Timer Current:
 - 1 μ A @ 2.0V, typical
- Timer1 Oscillator Current:
 - 1.2 μ A @ 32 kHz, 2.0V, typical
- Dual-speed Internal Oscillator:
 - Run-time selectable between 4 MHz and 48 kHz
 - 4 μ s wake-up from Sleep, 3.0V, typical

Peripheral Features:

- 16 I/O pins with individual direction control
- High current sink/source for direct LED drive
- Analog comparator module with:
 - Two analog comparators
 - Programmable on-chip voltage reference (VREF) module
 - Selectable internal or external reference
 - Comparator outputs are externally accessible
- Timer0: 8-bit timer/counter with 8-bit programmable prescaler
- Timer1: 16-bit timer/counter with external crystal/clock capability
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Capture, Compare, PWM module:
 - 16-bit Capture/Compare
 - 10-bit PWM
- Addressable Universal Synchronous/Asynchronous Receiver/Transmitter USART/SCI

Device	Program Memory	Data Memory		I/O	CCP (PWM)	USART	Comparators	Timers 8/16-bit
	Flash (words)	SRAM (bytes)	EEPROM (bytes)					
PIC16F627A	1024	224	128	16	1	Y	2	2/1
PIC16F628A	2048	224	128	16	1	Y	2	2/1
PIC16F648A	4096	256	256	16	1	Y	2	2/1

³⁶ Para mayor información visitar: <http://www.microchip.com/>

ANEXO H

MANUAL DEL CONTROL DE TEMPERATURA INDUSTRIAL DELTA DTB³⁷



estándares eléctricos relevantes. (Valores nominales: 250 V, 15 A máx.). Debería facilitarse un dispositivo de desconexión adecuado cerca del controlador en la instalación final.

2. PANTALLA, LEDs Y TECLAS



Pantalla PV

Muestra el valor actual de la temperatura de proceso, o el nombre del parámetro

Pantalla SV

Muestra la consigna, valor leído del parámetro operativo, variable manipulada o valor consignado al parámetro.

AT

LED de auto-tuning. Parpadea cuando el proceso de auto-tuning está en marcha.

OUT1/OUT2

LEDs de salidas. Se encienden cuando la salida correspondiente está activada.



Tecla de confirmación / cambio de menú

Pulse esta tecla para cambiar de menú, y para confirmar un valor de ajuste.



Tecla de navegación entre parámetros

Pulse esta tecla para moverse entre parámetros.

°C, °F

LEDs de unidades de temperatura. °C = Celsius (centígrados); °F = Fahrenheit

ALM1 – ALM3

LEDs de salida de alarma, se encienden cuando la alarma correspondiente está activada.



Tecla abajo

Pulse esta tecla para decrementar los valores mostrados en la pantalla SV. Manténgala apretada para acelerar los decrementos.



Tecla arriba

Pulse esta tecla para incrementar los valores mostrados en la pantalla SV. Manténgala apretada para acelerar los incrementos.

3. ESPECIFICACIONES

<i>Tensión de entrada</i>	100 a 240 Vac, 50/60Hz
<i>Rango de tensión de funcionamiento</i>	85% a 110% de la tensión nominal
<i>Consumo de energía</i>	5 VA máx
<i>Protección de memoria</i>	EEPROM 4 Kbit memoria no volátil (100.000 escrituras)
<i>Tipo de display</i>	2 líneas x 4 caracteres de 7-segmentos de LEDs Pantalla PV: color rojo, Pantalla SV: color verde
<i>Tipos de sensor</i>	Termopares: K, J, T, E, N, R, S, B, L, U, TXK RTD de platino de 3 hilos: Pt100, JPt100 Entrada analógica: 0~5 V, 0~10 V, 0~20 mA, 4~20 mA, 0~50 mV
<i>Modo de control</i>	PID, ON/OFF, Control de programa manual o PID (Rampa / Llano)
<i>Salida de control</i>	Salida de relé: SPDT (SPST: tamaños 1/16 DIN y 1/32 DIN), máxima carga 250 Vac, 5A carga resistiva Salida de pulsos de tensión: 14 Vdc, máxima corriente de salida 40 mA Salida de corriente: 4 ~ 20 mA DC (resistencia de carga: máx. 600Ω) Salida lineal de tensión: 0~5 V, 0~10 V
<i>Precisión del valor mostrado</i>	0 o 1 dígito a la derecha del punto decimal (seleccionable)
<i>Frecuencia de muestreo</i>	Entrada analógica: 150 ms/scan Termopar o RTD de platino: 400 ms/scan
<i>Comunicación RS-485</i>	Protocolo MODBUS ASCII / RTU

³⁷ Para mayor información visitar:

[http://www.mecmod.com/downloads/DELTA/TC%20Controladores%20de%20Temperatura%20\(DT\)/](http://www.mecmod.com/downloads/DELTA/TC%20Controladores%20de%20Temperatura%20(DT)/)



<i>Resistencia a la vibración</i>	10 a 55 Hz, 10 m/s ² durante 10 min, en cada una de las direcciones X, Y, Z
<i>Resistencia al impacto</i>	Máx. 300 m/s ² , 3 veces en cada uno de los 3 ejes, 6 direcciones
<i>Temperatura ambiente</i>	0 °C a +50 °C
<i>Temperatura de almacenamiento</i>	-20 °C a +65 °C
<i>Altitud</i>	2000 m o menos
<i>Humedad relativa</i>	35% a 80% (sin condensación)

4. TIPOS DE SENSOR Y RANGOS DE TEMPERATURA

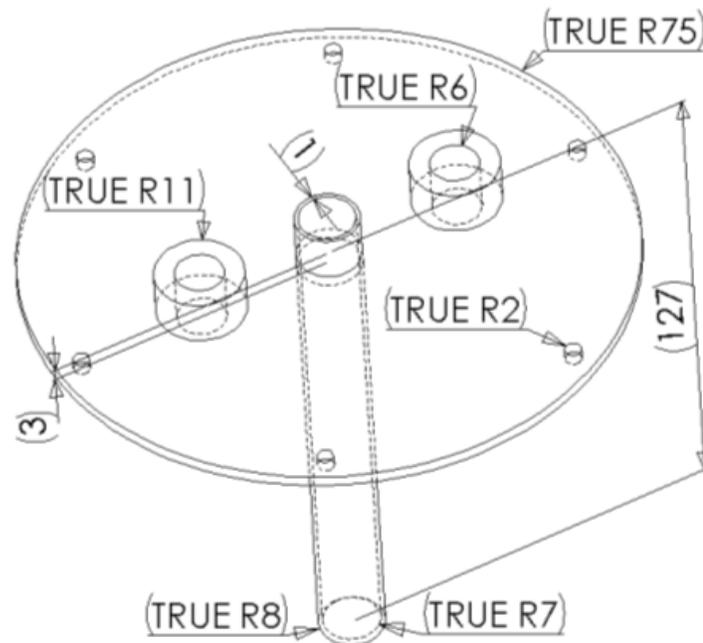
<i>Tipo de entrada de sensor de temperatura</i>	<i>Valor del parámetro</i>	<i>Display LED</i>	<i>Rango de temperatura</i>
Entrada analógica 0 ~ 50 mV	17	ñU	-999 ~ 9999
Entrada analógica 4 ~ 20 mA	16	ñR4	-999 ~ 9999
Entrada analógica 0 ~ 20 mA	15	ñR0	-999 ~ 9999
Entrada analógica 0 ~ 10 V	14	U10	-999 ~ 9999
Entrada analógica 0 ~ 5 V	13	US	-999 ~ 9999
Resistencia de platino (Pt100)	12	Pt	-200 ~ 600 °C
Resistencia de platino (JPt100)	11	JPt	-20 ~ 400 °C
Termopar tipo TXK	10	tU	-200 ~ 800 °C
Termopar tipo U	9	U	-200 ~ 500 °C
Termopar tipo L	8	L	-200 ~ 850 °C
Termopar tipo B	7	b	100 ~ 1800 °C
Termopar tipo S	6	S	0 ~ 1700 °C
Termopar tipo R	5	r	0 ~ 1700 °C
Termopar tipo N	4	n	-200 ~ 1300 °C
Termopar tipo E	3	E	0 ~ 600 °C
Termopar tipo T	2	t	-200 ~ 400 °C
Termopar tipo J	1	J	-100 ~ 1200 °C
Termopar tipo K	0	K	-200 ~ 1300 °C

Nota 1: Si se selecciona entrada de corriente (0-20 mA, 4-20 mA) como tipo de sensor de temperatura, se deberá conectar una resistencia de precisión externa de 250 Ω.

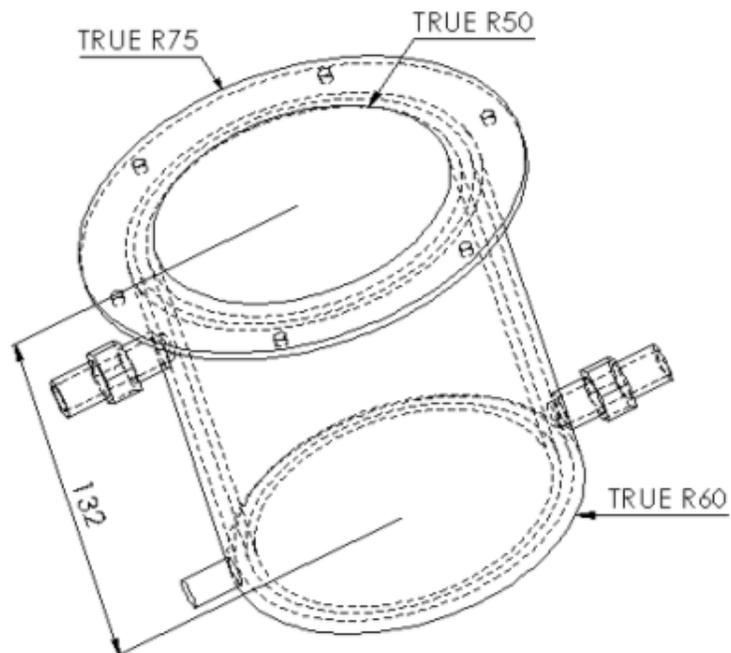
Nota 2: Deberá modificarse el parámetro **SP** (ver menú operación) si el usuario desea especificar la posición de la coma decimal. (Esta posibilidad no existe para los termopares de tipo B, S, R).

ANEXO I
PLANOS DEL REACTOR QUÍMICO AGITADO TIPO BATCH³⁸

Tapa del reactor químico agitado tipo batch.

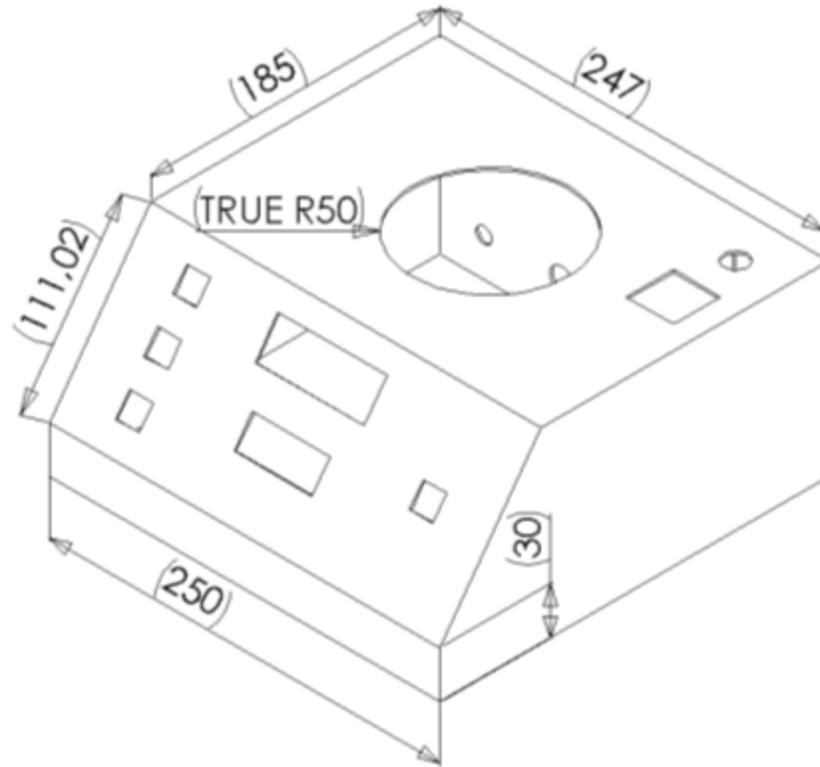


Envase del reactor químico agitado tipo batch.



³⁸ Para mayor detalle referirse al CD-ROM adjunto.

Base del reactor químico agitado tipo batch.



ANEXO J CÁLCULOS DE PRESIÓN DEL REACTOR³⁹

El factor de seguridad que utilizaremos para procesos químicos es de 2,5, con este dato y con un análisis realizado en el software **Inventor 2012**, se dice que la presión que podría soportar el reactor batch es de 145,64 PSI

Tabla E.1 Factor de seguridad y presión

Factor de Seguridad	de Presion (Mpa)	Presión Interna (psi)
3,4	0,5	72,825
2,43	0,7	101,955
2,5	1	145,65
1,69	1,25	182,0625

En la tabla E.1, está colocado el factor de seguridad que quiere decir, la relación que existe entre el esfuerzo máximo y el esfuerzo de fluencia.

El esfuerzo máximo es la relación que existe entre la fuerza aplicada y el área sobre la cual se aplica esa fuerza.

El esfuerzo de fluencia es el límite en el cual el material (en este caso acero inoxidable), está teóricamente en la zona lineal, cuya pendiente ($0 < m < 1$) está definida por el módulo de Young o módulo de elasticidad.

A continuación se muestra parcialmente lo realizado en el software **Inventor 2012**, para la obtención del resultado de la presión, si se desea la información completa recurrir al cd adjunto.

³⁹ Cálculos realizados por Andrés Sarmiento Cajamarca, a quien se le agradece por su colaboración en la realización de las simulaciones y cálculos de presión del reactor químico tipo batch.

Autodesk®

Analyzed File:	total.iam
Autodesk Inventor Version:	2010 (Build 140223002, 223)
Creation Date:	03/02/2012, 22:09
Simulation Author:	andres sarmiento
Summary:	

Project Info

Summary

Author andres sarmiento

Project

Part Number	total
Designer	andres sarmiento
Cost	0,00 €
Date Created	03/02/2012

Status

Design Status WorkInProgress

Physical

Mass	2,94314 kg
Area	272263 mm ²
Volume	380854 mm ³
Center of Gravity	x=55,2577 mm y=0,571082 mm z=0 mm

Simulation:1

General objective and settings:

Design Objective	Single Point
Simulation Type	Static Analysis
Last Modification Date	03/02/2012, 22:08
Detect and Eliminate Rigid Body Modes	No

Advanced settings:

Avg. Element Size (fraction of model diameter)	0,1
Min. Element Size (fraction of avg. size)	0,2
Grading Factor	1,5
Max. Turn Angle	60 deg

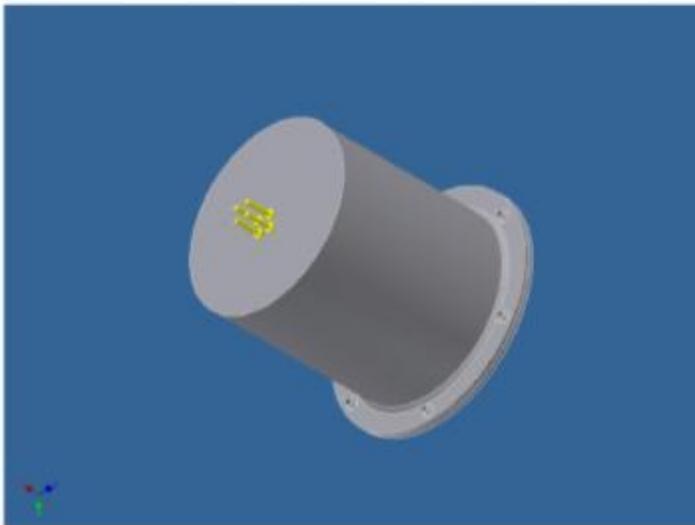
Create Curved Mesh Elements	No
Ignore Small Geometry	No

Operating conditions

Pressure:1

Load Type	Pressure
Magnitude	1,000 MPa

Selected Face(s)



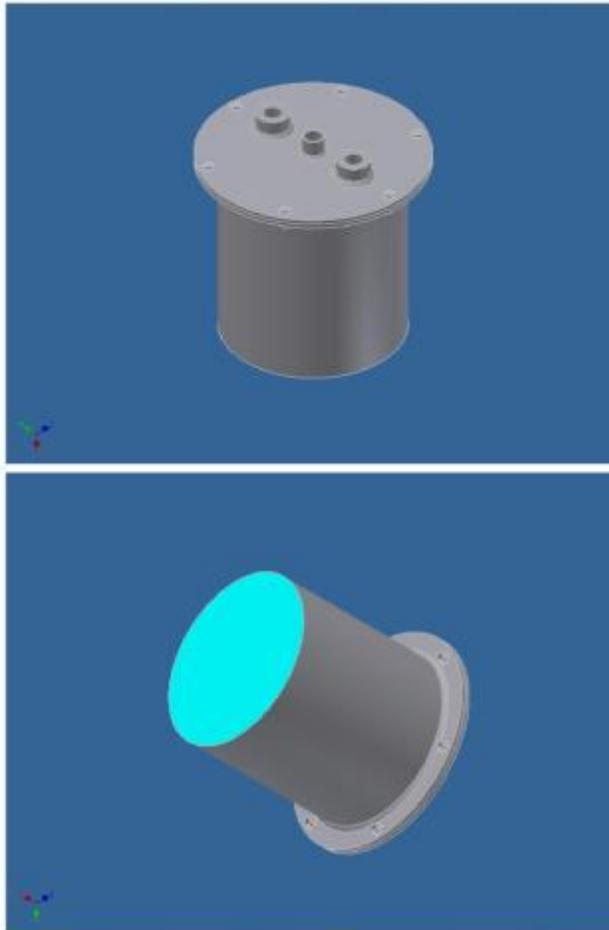
Pressure:2

Load Type	Pressure
Magnitude	1,000 MPa

Selected Face(s)

05/02/12

Stress Analysis Report Presion = 1 Mpa



▣ Results

▣ Reaction Force and Moment on Constraints

Constraint Name	Reaction Force		Reaction Moment	
	Magnitude	Component (X,Y,Z)	Magnitude	Component (X,Y,Z)
Fixed Constraint:1	9473,54 N	9473,54 N	0,05781 N m	-0,0120003 N m
		-0,0608953 N		-0,0363859 N m
		0,0591262 N		-0,0432904 N m

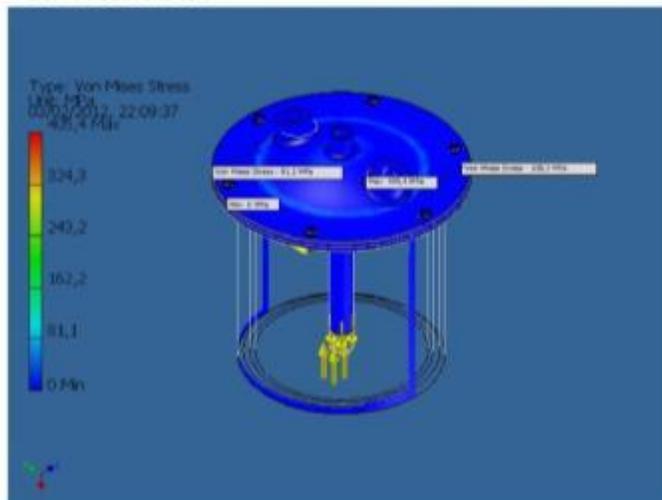
▣ Result Summary

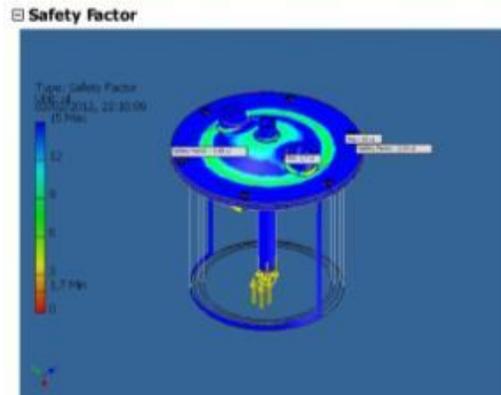
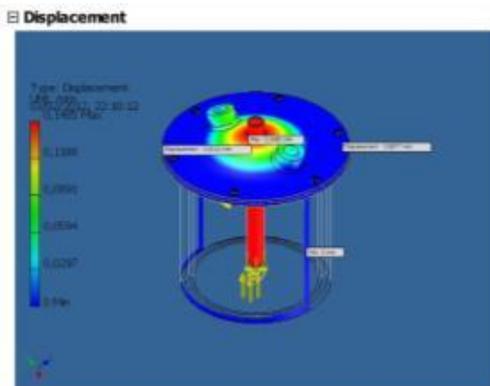
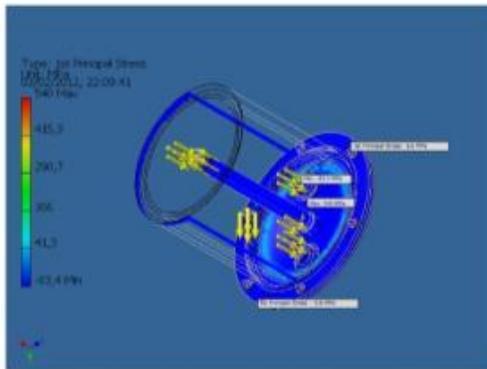
Name	Minimum	Maximum
Volume	380792 mm ³	
Mass	2,94267 kg	
Von Mises Stress	0,00135311 MPa	405,387 MPa
1st Principal Stress	-83,4033 MPa	540,038 MPa
3rd Principal Stress	-324,757 MPa	101,023 MPa

Displacement	0 mm	0,148476 mm
Safety Factor	1,69961 ul	15 ul
Stress XX	-286,529 MPa	518,233 MPa
Stress XY	-89,9597 MPa	100,008 MPa
Stress XZ	-50,9017 MPa	43,5148 MPa
Stress YY	-124,273 MPa	166,362 MPa
Stress YZ	-49,5653 MPa	51,7049 MPa
Stress ZZ	-113,853 MPa	173,047 MPa
X Displacement	-0,148476 mm	0,0000271498 mm
Y Displacement	-0,052337 mm	0,0523277 mm
Z Displacement	-0,00737939 mm	0,00736391 mm
Equivalent Strain	0,0000000052294 ul	0,00187056 ul
1st Principal Strain	-0,0000194722 ul	0,00224576 ul
3rd Principal Strain	-0,00160591 ul	0,0000438037 ul
Strain XX	-0,00111112 ul	0,00211179 ul
Strain XY	-0,00123119 ul	0,00122617 ul
Strain XZ	-0,000689307 ul	0,000696757 ul
Strain YY	-0,000620276 ul	0,000761314 ul
Strain YZ	-0,000304537 ul	0,000317683 ul
Strain ZZ	-0,000530256 ul	0,000526851 ul
Contact Pressure	0 MPa	416,085 MPa
Contact Pressure X	-309,982 MPa	415,738 MPa
Contact Pressure Y	-201,485 MPa	179,964 MPa
Contact Pressure Z	-146,404 MPa	150,863 MPa

☐ Figures

☐ Von Mises Stress





BIBLIOGRAFÍA

- [1] Areny, R. P. (2005). *Sensores y acondicionadores de señal* (Cuarta ed.). Barcelona: Alfaomega, Marcombo.
- [2] Arian. (31 de Julio de 2008). Recuperado el 9 de Mayo de 2011, de www.arian.cl/espanol/index.htm: <http://www.arian.cl/downloads/nt-004.pdf>
- [3] Bakshi, V., & Bakshi, U. (2007). *Linear Control Systems*. Recuperado el 1 de Agosto de 2011, de Google Books: http://books.google.com.ec/books?id=EhQ3-XsUgxMC&printsec=frontcover&hl=en&source=gbs_atb#v=onepage&q&f=false
- [4] Binder, J. B., & Raines, R. T. (21 de Enero de 2009). *Simple Chemical Transformation of Lignocellulosic Biomass into Furans for Fuels and Chemicals*. Obtenido de Journal of the American Chemical Society.: <http://pubs.acs.org/journal/jacsat>
- [5] Caccavale, F., Iamarino, M., Pierri, F., & Tufano, V. (2011). *Control and monitoring of chemical batch reactors*. Londres: Springer-Verlag.
- [6] Carletti, E. J. (18 de Mayo de 2009). Recuperado el 13 de Mayo de 2011, de <http://robots-argentina.com.ar>: http://robots-argentina.com.ar/Sensores_general.htm
- [7] Copeland, B. (s.f.). Retrieved Noviembre 1, 2011, from http://www.eng.uwi.tt/depts/elec/staff/copeland/ee27B/Ziegler_Nichols.pdf
- [8] Departamento de diseño mecánico. (s.f.). *Facultad de Ingenieria*. Recuperado el 12 de Mayo de 2011, de <http://www.fing.edu.uy/>: http://www.fing.edu.uy/iimpi/academica/grado/instindustrial/teorico/080306-Sensores-parte_II.temperatura.pdf
- [9] Diaz, H. M. (s.f.). *Galeon*. Recuperado el 14 de Julio de 2011, de Galeon: <http://www.galeon.com/hamd/pdf/reactorquimico.pdf>
- [10] Dorf, R. C., & Bishop, R. H. (2005). *Sistemas de control moderno*. Madrid (España): Peason Prentice Hall.
- [11] Edicions UPC. (1999). *Teoría de control y diseño electrónico*. Unión Europea: Ediciones UPC.
- [12] *El Prisma*. (s.f.). Recuperado el 14 de Julio de 2011, de El Prisma: http://www.elprisma.com/apuntes/ingenieria_quimica/reactorquimico/default2.asp

- [13] *EXTECH INSTRUMENTS*. (s.f.). Recuperado el 3 de Agosto de 2011, de http://www.shopextech.com/crm_uploads/RPM10_UMsp.pdf
- [14] Fogler, H. (2001). *Elementos de Ingeniería de las Reacciones Químicas* (Tercera edición ed.). México: Person Education.
- [15] García Jaimes, L. E. (2010). *Control digital. Teoría y práctica* (Tercera ed.). Medellín.
- [16] Golnaragui, F., & Kuo, B. C. (2010). *Automatic control systems*. United States of América: John Wiley & Sons, Inc.
- [17] Grau Vilalta, M. D. (18 de Noviembre de 1999). *Estudio del comportamiento de reactores discontinuos y semicontinuos: Modelización y comprobación experimental*. Recuperado el 23 de Julio de 2011, de Tesis Doctorals en Xarxa. Universitat Politècnica de Catalunya. Departament d'Enginyeria Química: <http://tdx.cat/handle/10803/6463>
- [18] Hsu, H. P. (1995). *Signals and systems*. United States of America: McGraw-Hill.
- [19] *IFEnt*. (s.f.). Recuperado el 22 de Julio de 2011, de <http://www.ifent.org/Lecciones/PTC/ptc.htm>
- [20] Iñi, L. (2 de Julio de 2010). *liso.net*. Recuperado el 5 de Julio de 2011, de <http://liosonet.net/?p=592>
- [21] J.M. Coulson, J. R. (2004). *Ingeniería química*, (Vol. III). España: Editorial Reverte S.A.
- [22] Jimenez, M. (14 de Julio de 2006). *Comunicaciones Industriales*. Recuperado el 13 de Junio de 2011, de www.dte.upct.es: http://www.dte.upct.es/personal/manuel.jimenez/docencia/GD6_Comunic_Ind/pdfs/Tema%207.pdf
- [23] Johnson, M. A., & Moradi, M. H. (2005). *PID Control. Identification and Design Methods*. Londres: Springer.
- [24] Kern, D. Q. (1999). *Procesos de transferencia de calor*. México: Compañía editorial continental S.A. de C.V.
- [25] Krishnaswamy, K. (2007). *Process Control*. Recuperado el 5 de Agosto de 2011, de Google Books: http://books.google.com.ec/books?id=Lha8HBVRYX0C&printsec=frontcover&hl=en&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

- [26] Kuo, B. C. (1996). *Sistemas de control automático* (Séptima ed.). Naucalpan de Juárez, Edo. de México: Prentice-Hall Hispanoamericana S.A.
- [27] Kuo, B. C. (1996). *Sistemas de control automático* (Séptima ed.). Naucalpan de Juárez, Edo. de México: Prentice-Hall Hispanoamericana S.A.
- [28] Levenspiel, O. (1993). *Flujo de fluidos e intercambio de calor*. España: Reverté.
- [29] Levenspiel, O. (2004). *Ingeniería de las Reacciones Químicas* (Tercera edición ed.). México: Limusa Wiley.
- [30] Levenspiel, O. (2002). *El minilibro de los reactores químicos*. Sevilla: Reverté S.A.
- [31] LIMPI. (s.f.). Recuperado el 25 de Julio de 2011, de http://www.fing.edu.uy/iimpi/academica/grado/instindustrial/teorico/080306-Sensores-parte_II.temperatura.pdf
- [32] Mann, U. (2009). *Principles of chemical reactor analysis and design*. New Jersey: John Wiley & Sons, Inc Publication.
- [33] Mayné, J. (2003). Recuperado el 6 de Mayo de 2011, de www.bairesrobotics.com.ar: http://www.bairesrobotics.com.ar/data/sensores_2003.pdf
- [34] Mendiburu Díaz, H. A. (2005). *Diseño de un Neurocontrolador Dinámico (DBP) aplicado a un Reactor Químico Continuo (CSTR)*. Lima: Pontificia Universidad Católica del Perú .
- [35] Moncada Albitres, M. L. (s.f.). *Plantas químicas*. Recuperado el 12 de Julio de 2011, de <http://www.plantasquimicas.com>
- [36] Muñoz, D. R. (23 de Marzo de 2005). Recuperado el 9 de Mayo de 2011, de www.uv.es: http://www.uv.es/ramirez/Docencia/LO/Notas_termopares.pdf
- [37] Muñoz, D. R. (2005). *Vniversitat de Valencia*. Recuperado el 9 de Mayo de 2011, de http://www.uv.es/ramirez/Docencia/LO/Notas_termopares.pdf
- [38] National Instruments. (29 de Marzo de 2011). *National Instruments*. Recuperado el 1 de Junio de 2011, de NI Developer Zone: <http://zone.ni.com/devzone/cda/tut/p/id/3782>
- [39] Ogata, K. (1996). *Sistemas de control en tiempo discreto*. Naucalpan de Juárez, Edo. de México: Prentice-Hall Hispanoamericana S.A.

- [40] Ogata, K. (1998). *Ingeniería de control moderna*. México: Prentice-Hall, Inc.
- [41] Ordaz, G. V., Romero, R. R., Ortiz, F. E., & Montúfar, M. A. (2006). *Medición y control de procesos industriales* (Primera ed.). México D.F.
- [42] Pagola y de las Heras, F. L. (2007 de Junio de 2007). *Universidad Pontificia Comillas*. Recuperado el 31 de Julio de 2011, de Departamento de Electrónica y Automática: <http://www.dea.icaei.upco.es/pagola/Material/modelado.pdf>
- [43] Perez, I. A., Perez Hidalgo, I., & Perez Berenguer, B. (2008). *Departamento de Electrónica Automática y Bioingeniería - Facultad de Ingeniería - Universidad Nacional de San Juan*. Recuperado el 31 de 7 de 2011, de CONTROL I (Bioingeniería): <http://dea.unsj.edu.ar/control1b/teoria/unidad1y2.pdf>
- [44] Pérez, O. P. (20 de Abril de 2010). *Desarrollo de un auto Robot utilizando material reciclable*. Recuperado el 6 de Agosto de 2011, de <http://es.scribd.com/doc/30718664/Documentacion-de-Interfaces-1>
- [45] Perry, H. (1973). *Biblioteca del Ingeniero Químico* (Quinta edición ed.). México: McGraw-Hill.
- [46] Sendrós, C. F. (Noviembre de 2004). *Departamete de Ingeniería, Electrónica, Electrica y Automática*. Recuperado el 18 de Julio de 2011, de <http://cde05.etse.urv.es/pub/pdf/675pub.pdf>
- [47] Shannon, R. E. (1988). *Simulación de Sistemas. Diseño, desarrollo e implementación*. México: Editorial Trillas.
- [48] Solé, A. C. (1997). *Instrumentación industrial* (Sexta ed.). Barcelona: Marcombo, S.A.
- [49] *Taringa*. (s.f.). Recuperado el 5 de Junio de 2011, de http://www.taringa.net/posts/info/8040814/_Que-es-el-efecto-Seebeck_.html
- [50] *Thermibel*. (s.f.). Recuperado el 20 de Julio de 2011, de <http://www.thermibel.be/documents/pt100.xml?lang=es>
- [51] Vilalt, M. D. (Septiembre de 1999). Recuperado el 10 de Julio de 2011, de http://tdx.cat/bitstream/handle/10803/6463/01_grauVilalta_capitol_1.pdf?sequence=6
- [52] Villavicencio, M. (2011). *Escribir en la Universidad*. Cuenca: Universidad de Cuenca (Servigraf).
- [53] Visioli, A. (2006). *Practical PID control*. Londres: Springer.

- [54] West instruments de México. (s.f.). *Manual de encoders*. Recuperado el 20 de Mayo de 2011, de WESTMEXICO: <http://www.westmexico.com.mx/pfd/dynapar/catalogos/4.-Manual%20de%20Aplicacion%20de%20Encoders.pdf>
- [55] Yurkanis, P. (2008). *Química Orgánica*. México: Pearson.
- [56] Zambrano Pérez, V. D. (Septiembre de 2009). *DSPACE*. Recuperado el 5 de Mayo de 2011, de EE DSpace Escuela Politécnica Nacional. Facultad de Ingeniería Eléctrica y Electrónica (ECUADOR): <http://bieec.epn.edu.ec:8180/dspace/bitstream/123456789/1407/3/T11367%20CAP%20III.pdf>