



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DIFUSO PARA EL CONTROL DEL CAUDAL Y LA TEMPERATURA DEL AGUA PARA CALENTADORES SOLARES DE USO DOMÉSTICO.”

JOHNNY PAUL FREIRE DIAZ

**Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo,
presentado ante el Instituto de Posgrado y Educación Continua de la ESPOCH,
como requisito parcial para la obtención del grado de:**

**MAGISTER EN SISTEMAS DE CONTROL
Y AUTOMATIZACIÓN INDUSTRIAL**

RIOBAMBA - ECUADOR

MARZO 2018

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

CERTIFICACIÓN:

EL TRIBUNAL DEL TRABAJO DE TITULACIÓN CERTIFICA QUE:

El Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, denominado “Diseño e implementación de un sistema difuso para el control del caudal y la temperatura del agua para calentadores solares de uso doméstico.” de responsabilidad del Ingeniero Johnny Paúl Freire Díaz, ha sido prolijamente revisado y se autoriza su presentación.

Ing. Wilson Armando Zúñiga Vinuesa

PRESIDENTE

FIRMA

Ing. José Enrique Guerra Salazar

DIRECTOR DE TESIS

FIRMA

Ing. Nelson Santiago Chuquín Vasco

MIEMBRO DEL TRIBUNAL

FIRMA

Ing. Franklin Geovanni Moreno Montenegro

MIEMBRO DEL TRIBUNAL

FIRMA

Riobamba, Marzo 2018

DERECHOS INTELECTUALES

Yo, Johnny Paúl Freire Díaz, soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual del mismo pertenece a la Escuela Superior Politécnica de Chimborazo.

Johnny Paúl Freire Díaz
CÉDULA 060274619-0

DEDICATORIA

Al Grupo 24 Horas de Alcohólicos Anónimos Riobamba, sin el cual no hubiera sido posible para mi ningún logro personal, mucho menos académico. A mi esposa Maritza y a mi hijo Ignacio, por la fuerza que me dan para seguir adelante y la razón de mi vida. A mis padres y hermanos, por haber confiado en mí, más que yo mismo. A mis abuelos, por haberme inculcado con su ejemplo, el valor de la honradez y del trabajo.

AGRADECIMIENTO

A la Escuela Superior Politécnica de Chimborazo.

Al Msc. José Guerra, Director de Tesis y a los Msc. Nelson Chuquín y Msc. Franklin Moreno, por la paciencia y conocimientos demostrados en el desarrollo de la presente investigación.

Al Profesor Hubert J. Farnsworth, a los Doctores Emmett Lathrop Brown, Víctor Frankenstein y Who, a Anthony Edward Stark y Bruce Wayne, por ser la fuente de inspiración para este proyecto.

CONTENIDO

	Páginas
RESUMEN.....	XIV
SUMMARY.....	XV
CAPITULO I	
1. INTRODUCCIÓN.....	1
1.1 Planteamiento del problema	1
1.2 Justificación	2
1.3 Objetivos	3
1.3.1 <i>Objetivo General</i>.....	3
1.3.2 <i>Objetivos Específicos</i>	3
CAPITULO II	
2. MARCO TEÓRICO.....	5
2.1 Antecedentes del problema	5
2.2 Teoría de control	6
2.2.1 <i>Planta</i>.....	6
2.2.2 <i>Controlador</i>	7
2.2.3 <i>Sensor</i>	7
2.2.4 <i>Actuador</i>	7
2.2.5 <i>Transmisor</i>.....	7
2.2.6 <i>Función de transferencia</i>	8
2.2.7 <i>Sistemas de primer orden</i>	8
2.3 Lógica difusa.....	9
2.4 Protocolos de comunicación.....	18
2.5 Calentadores solares domésticos	19

CAPÍTULO III

3.	Metodología de la investigación.....	20
3.1	Análisis del sistema actual	20
3.2	Requerimientos del sistema	22
3.3	Concepción de la arquitectura general del sistema.....	23
3.3.1	<i>Módulo de sensores</i>	24
3.3.2	<i>Módulo de Control</i>	25
3.3.3	<i>Módulo de Actuadores</i>	25
3.3.4	<i>Módulo de interfaz</i>	25
3.3.5	<i>Módulo de Alimentación</i>	25
3.3.6	<i>Módulo Monitoreo</i>	26
3.4	Selección de los elementos de <i>hardware</i> del Sistema	26
3.4.1	<i>Módulo de sensores</i>	26
3.4.1.1	<i>Sensor de temperatura</i>	28
3.4.1.2	<i>Caudalímetro o Sensor de Flujo</i>	29
3.4.2	<i>Módulo de Control</i>	30
3.4.3	<i>Módulo de Actuadores</i>	31
3.4.3.1	<i>Grifo Monomando Cerámico</i>	31
3.4.3.2	<i>Servo motor Hi-Tec HS-755HB</i>	34
3.4.3.3	<i>Válvula Motórica Reguladora</i>	35
3.4.3.4	<i>Electroválvula Solenoide</i>	36
3.4.4	<i>Módulo de Interfaz</i>	38
3.4.5	<i>Módulo de Alimentación</i>	38
3.5	Diagramas de conexión del sistema	40
3.6	Selección del <i>software</i> de desarrollo	41
3.6.1	<i>Software Arduino</i>	41
3.6.2	<i>Matlab</i>	42
3.6.3	<i>Simulink</i>	43

3.7	Procesamiento de la información de los módulos.....	43
3.7.1	Procesamiento de la información de temperatura	44
3.7.2	Procesamiento de la información de caudal	46
3.7.3	Comunicación entre placas Arduino.....	47
3.8	Implementación del <i>software</i> del sistema.....	48
3.8.1	Estados del sistema:	48
3.8.2	Bloque de control difuso de temperatura	51
3.8.3	Bloque de control difuso de caudal	55
3.8.4	Bloque Interface	60

CAPÍTULO IV

4.	Validación del prototipo del sistema	63
4.1	Caracterización de las mediciones realizadas por el módulo de sensores	63
4.2.1	Calibración sensor de temperatura ds18b20	64
4.2.1	Calibración sensor de caudal YF-S201	65
4.2	Simulación del sistema	66
4.3	Pruebas y puesta a punto del controlador difuso de temperatura	69
4.4	Análisis de resultados del control difuso de temperatura.....	71
4.5	Análisis de resultados del control difuso de caudal.....	72
4.6	Costos del prototipo	72
	CONCLUSIONES.....	74
	RECOMENDACIONES.....	76

BIBLIOGRAFÍA

ANEXOS

INDICE DE TABLAS

Tabla 1-3	Características del sensor YF-S201	30
Tabla 2-3	Relación datos/temperatura para el termómetro digital DS18b20	45
Tabla 3-3	Función de las entradas y salidas del bloque de estados	50
Tabla 4-3	Notación de las variables de entrada del control de temperatura	53
Tabla 5-3	Notación de la variable de salida del control de temperatura	54
Tabla 6-3	Reglas de control del sistema propuesto.....	55
Tabla 7-3	Caudales de trabajo del sistema.....	56
Tabla 8-3	Entradas y salidas del bloque interface.....	60
Tabla 1-4	Comparativa del instrumento patrón y el sensor Ds18b20.....	57
Tabla 2-4	Conteo de pulsos por volumen y valor resultante de la constante K.....	60
Tabla 3-4	Costo de elementos del prototipo.....	67

INDICE DE FIGURAS

Figura 1 - 2	Tipos de planta de acuerdo al número de entradas.....	6
Figura 2 - 2	Esquema de un sistema de control básico.....	7
Figura 3 - 2	Sistema en el dominio de Laplace.....	8
Figura 4 - 2	Respuesta de un sistema de primer orden simple ante una entrada escalón.....	9
Figura 5 - 2	Visión de una variable de acuerdo a la Lógica Difusa y a la Lógica.....	11
Figura 6 - 2	Función Trapezoidal.....	14
Figura 7 - 2	Función Triangular.....	14
Figura 8- 2	Configuración básica de un sistema tipo Mamdani.....	16
Figura 9 - 2	Diagrama de conexión de dispositivos I2C.....	19
Figura 1 - 3	Esquema hidráulico del sistema propuesto.....	22
Figura 3 - 2	Diagrama dell sistema propuesto.....	24
Figura 3 - 3	Computador utilizado como módulo de monitoreo.....	26
Figura 4 - 3	Arduino Nano.....	27
Figura 5- 3	Esquema del módulo de recolección de datos.....	28
Figura 6 - 3	Sensor DS18B20 con encapsulado a prueba de agua.....	28
Figura 7 - 3	Caudalímetro YF-S201.....	29
Figura 8 - 3	Cartucho de un.....	32
Figura 9 - 3	Posiciones de las superficies de.....	32
Figura 10 - 3	Partes del Grifo monomando Franz Viegenger E108.05/B1.....	33
Figura 11 - 3	Funciones de un grifo monomando.....	33
Figura 12 - 3	Servo motor acoplado al grifo monomando.....	34
Figura 13 - 3	Servo motor Hi-Tec HS-755HB.....	35
Figura 14 - 3	Válvula CWX-25S CR-05.....	36
Figura 15 - 3	Diagrama de conexiones de la válvula.....	36
Figura 16 - 3	Electro válvula Solenoide 12V DIN 15.....	37
Figura 17 - 3	LCD (Liquid Cristal Display) de 20.....	38
Figura 18 - 3	Fuente ATX utilizada como módulo de.....	39
Figura 19 - 3	Driver L298N para motores.....	40
Figura 20 - 3	Diagrama de conexiones del sistema.....	41
Figura 21 - 3	Formato del registro de temperatura del sensor.....	45
Figura 22 - 3	Parámetros del Bloque I2C que.....	46
Figura 23 - 3	Bloque Caudalímetro en Simulink y las señales que emite.....	47
Figura 24 - 3	Conexión de dos Arduino Nano y Arduino Mega por medio de I2C.....	48
Figura 25 - 3	Máquina de estados finitos del sistema.....	50
Figura 26 - 3	Esquema del sistema de control difuso de temperatura.....	51
Figura 27 - 3	Gráfica de la temperatura y su derivada en.....	52

Figura 28 - 3	Situaciones posibles del error y su derivada.	52
Figura 29 - 3	(a) Parámetros del controlador difuso. (b) Funciones de pertenencia variable de entrada “error”. (c) Funciones de pertenencia variable entrada “derror”	53
Figura 30 - 3	Funciones de pertenencia de la variable de salida “angulo”	54
Figura 31 - 3	Variables y métodos incluidos en el controlador difuso de caudal.	56
Figura 32-3	Función de pertenencia de la señal de entrada “errorcaudal”	57
Figura 33 - 3	Función de pertenencia de las variables de entrada.....	57
Figura 34 - 3	Función de pertenencia de la señal de salida.....	58
Figura 35 - 3	Entrada, salida y sus correspondientes cuatro funciones de pertenencia del	59
Figura 36 - 3	Estructura de control difuso de velocidad del motor de la válvula de caudal.	60
Figura 37 - 3	(a) Bloque interface (b) subsistema contenido en el bloque interface	61
Figura 38 - 3	Bloque contador_flujo y la máquina de estados que procesa su información.	62
Figura 1 - 4	Resultados del ajuste lineal del sensor de temperatura.	63
Figura 2 - 4	Bloque del sensor de temperatura con el polinomio de ajuste.....	65
Figura 3 - 4	Curva de reacción de la planta.	67
Figura 4 - 4	Modelo de la planta en Simulink	68
Figura 5 - 4	Primeros resultados de las simulaciones	68
Figura 6 - 4	Respuesta simulada de la planta a una entrada de tipo escalón,	69
Figura 7 - 4	Banco de pruebas.....	69
Figura 8 - 4	Tablero de control definitivo del sistema (a) abierto y cerrado (b)	70
Figura 9 - 4	Prototipo definitivo del sistema.	70

RESUMEN

El objetivo de la presente investigación fue el diseño e implementación de un sistema difuso para el control del caudal y la temperatura del agua para calentadores solares de uso doméstico. Para el desarrollo de hardware se usó la plataforma *Arduino* y para el software el paquete *Simulink*. El sistema mezcla agua proveniente de dos tomas independientes, una de agua fría de un reservorio o de la red local de agua potable, y una toma de agua caliente provista por un calentador solar doméstico. Se implementó una interfaz física que permite al usuario especificar los valores de temperatura y caudal e informa sobre el estado del sistema. Se usó lógica difusa para el control de las variables temperatura y caudal de la planta, con la cual se pudo mantener el valor de la temperatura indicada por el usuario con un rango de $\pm 2\%$ y el caudal con el $\pm 3\%$ de error en relación con el set point. El desperdicio de agua que ocurre desde el momento en que se abre la llave hasta que el agua caliente empieza a salir por la ducha se evita a través de una tubería de retorno hacia el reservorio principal. Para proyectos similares se recomienda la conformación de un equipo multidisciplinario, que solucione, como en la actual investigación, requerimientos electrónicos, mecánicos, de programación y control automático.

Palabras clave: TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA, CONTROL AUTOMÁTICO, ARDUINO (SOFTWARE - HARDWARE), SIMULINK (SOFTWARE), CALENTADORES SOLARES, LÓGICA DIFUSA.

ABSTRACT

The objective of the present investigation was the design and implementation of a diffuse system for the control of the flow and the water temperature for solar heaters for domestic use. For the hardware development the Arduino platform used and for the software, Simulink package used. The system mixes water for two independent outlets, one from cold water from a reservoir or for the local drinking water network, and one hot water outlet provided by the domestic solar heater. A physical interface implemented that allows the user to specify the temperature and flow values and report on the status of the system. The diffuse logic used to control variables temperature and flow of the plant, plant, with which it maintained the value of the temperature indicated by the user with a range of $\pm 2\%$ and the flow rate with the $\pm 3\%$ glitch in relation to the setpoint. The waste of water that occurs from the moment the tap opened until the hot water starts to flow out of the shower prevented through a return pipe to the main reservoir. For similar projects, the creation of a multidisciplinary team, it recommends that solve, as in the current research, electronic, mechanical, programming and automatic control requirements.

Keywords: TECHNOLOGY AND SCIENCES OF ENGINEERING, AUTOMATIC CONTROL, ARDUINO (SOFTWARE-HARDWARE), SIMULINK (SOFTWARE), SOLAR HEATERS, DIFFUSIVE LOGIC.



CAPÍTULO I

1. INTRODUCCIÓN

Este capítulo aborda el Planteamiento del problema de investigación y su justificación, cuyo objetivo es diseñar e implementar un sistema difuso para el control del caudal y la temperatura del agua para calentadores solares de uso doméstico.

1.1 Planteamiento del problema

En el país los sistemas de control usados a nivel domiciliario, rara vez buscan optimizar los recursos utilizados, como el agua, la electricidad o el aspecto económico. Si bien es cierto que en la industria en mayor o menor grado se adoptan políticas que ayudan a controlar y optimizar el consumo de los recursos energéticos, es pertinente que el ciudadano común aplique normas similares en su hogar. Es necesario que busque disminuir el consumo de energía y el uso de recursos renovables, procesos que pueden ser supervisados en su totalidad y con gran facilidad por sistemas de control automático.

En las viviendas que cuentan con un termostato para calentar el agua, se controla únicamente el encendido o apagado del sistema, pero no se vigila la temperatura de salida del agua. En el caso de calentadores de agua solares, estos tienen gran eficiencia para entregar agua caliente, obteniendo en días poco soleados temperaturas por encima de 40°C y en días con luz solar abundante, consiguen temperaturas superiores a los 75°C. Por lo regular, los calentadores de agua se ajustan a temperaturas superiores a 55° C a fin de prevenir el desarrollo de bacterias nocivas en el suministro de agua.

Una forma de controlar el caudal y temperatura de agua caliente que se quiere es por medio de grifería de uno o dos mandos, que permite mezclar el agua hasta conseguir la temperatura deseada, proceso con el cual se genera un considerable desperdicio del líquido vital.

Esto, tomando en cuenta la eficiencia, la comodidad y ante todo el aprovechamiento óptimo de recursos, no es la mejor alternativa, dado que difícilmente un persona puede regular óptimamente los mandos para tener una salida a temperatura estable, ésta puede ser muy baja o muy alta, afecta

al confort del usuario o su integridad física, puesto que el agua caliente puede producir graves quemaduras al poco tiempo de exposición, especialmente en niños y personas de la tercera edad. El principio del control automático usando inteligencia artificial permite una mejor gestión de la temperatura de un fluido minimiza la pérdida de agua y de energía al ajustar en forma la temperatura al nivel deseado.

Por tal motivo el presente trabajo, propone implementar un sistema difuso para el control automático de caudal y la temperatura del agua para calentadores solares de uso doméstico que elimine el desperdicio de agua generado en los sistemas convencionales en el proceso de obtención de la temperatura deseada por un usuario.

1.2 Justificación

La lógica clásica asigna a las proposiciones simples o compuestas solamente uno de dos valores de verdad, falso o verdadero, no admite posición intermedia entre estos dos valores, de lo cual sí presume la lógica difusa. A este respecto, (Ponce Cruz, 2010) manifiesta: Las computadoras manejan datos precisos que se reducen a cadenas de unos (1) y ceros (0) y proposiciones que son ciertas y falsas. El cerebro humano puede razonar con información que involucra incertidumbre o juicios de valor como: “el aire es frío” o “la velocidad es rápida”. Además, las personas tienen un sentido común que les permite razonar en un mundo donde las cosas son parcialmente ciertas. La lógica difusa es una rama de la Inteligencia Artificial que le permite a una computadora analizar información del mundo real en una escala entre lo falso y verdadero. (Cabrera Almache, 2013)

Se propone en la presente investigación el empleo de la Lógica Difusa, una de las ramas de la inteligencia artificial, que ofrece salidas de forma veloz y precisa, disminuyendo así las transiciones de estados en el entorno físico que controla (Del Cerro Sánchez, 2016)

Si existe un conocimiento del proceso, es posible modelarlo mediante lógica difusa, ya que muchos procesos difícilmente pueden ser expresados con ecuaciones diferenciales. Los sistemas basados en lógica difusa son fáciles de diseñar, modificar y mantener. La reducción de tiempo de desarrollo y mantenimiento es muy relevante para su uso industrial. El control difuso permite diseñar soluciones de alta calidad que eviten las patentes existentes en otros sistemas de control. En Japón este tipo de controladores se asocia a modernidad, alta calidad y tecnológicamente potente. (González Morcillo, 2011).

Para la implementación del sistema propuesto se empleará para la recolección de datos caudalímetros electrónicos y sensores de temperatura de fluidos para tener lecturas con un alto grado de precisión. Esta investigación servirá además como modelo aplicable a entornos similares.

Con este proyecto se pretende desarrollar tecnología acorde a las necesidades de nuestro medio que brinde el confort y la seguridad necesaria a la población y optimice los recursos. Se enmarca en el objetivo tres del Plan Nacional del Buen vivir, que en su objetivo 3 dice “Mejorar la calidad de vida de la población”. (Semplades, 2013-2017). y en el objetivo 3.9 que plantea “Garantizar el acceso a una vivienda adecuada, segura y digna” , además de “Promover la construcción de viviendas y equipamientos sustentables que optimicen el uso de recursos naturales y utilicen la generación de energía a través de sistemas alternativos.” (Semplades, 2013-2017).

El sistema propuesto hace uso de la transferencia tecnológica para aportar al correcto aprovechamiento del agua en un domicilio, además pretende ser una herramienta de ayuda para optimizar el uso de recursos renovables como la energía solar y la protección de recursos no renovables como el agua.

1.3 Objetivos

1.3.1 Objetivo General

Diseñar e implementar un sistema difuso para el control del caudal y la temperatura del agua para calentadores solares de uso doméstico.

1.3.2 Objetivos Específicos

- Determinar la cantidad de agua que se desperdicia en sistemas de calentamiento tradicional hasta obtener la temperatura deseada por el usuario.
- Estudiar los principios en los que se basan los sistemas existentes en el mercado para el control de temperatura y caudal de agua para uso domiciliario.
- Identificar las variables involucradas y analizarlas desde el punto de vista de la lógica difusa.

- Realizar el proceso de defuzificación de las variables analizadas en un sistema controlado electrónicamente.
- Seleccionar los dispositivos necesarios para la construcción del sistema propuesto.
- Diseñar e implementar el sistema difuso de control de temperatura y caudal.

CAPÍTULO II

2. MARCO TEÓRICO

En este capítulo se exponen los antecedentes del problema, revisando investigaciones previas relacionadas con el tema de la presente tesis. También se exponen conceptos generales de teoría de control, lógica difusa y el protocolo de comunicaciones I2C. Contiene además la descripción de los componentes de hardware necesarios para implementar el sistema de control de temperatura.

2.1 Antecedentes del problema

En el país se han realizado varias investigaciones encaminadas al control de temperatura y de caudal, pero ninguno destinado a controlar el agua proveniente de un calentador solar y minimizar su desperdicio. No se analiza el ahorro de agua que se genera al utilizar un sistema de control. Los calentadores solares ofrecen un gran ahorro en el hogar, ya que aproximadamente el 80 por ciento del consumo de gas en una casa corresponde al calentador de gas. Son capaces de absorber hasta el 95 por ciento de la radiación solar para calentar el agua, garantizando agua caliente hasta en días nublados. (Eco Vita, 2016).

Por otro lado existen dispositivos denominados válvulas termostáticas que permiten suministrar agua a una temperatura constante, mezclan agua caliente y fría, generalmente con un diferencial de temperatura de al menos 7° C. Asimismo, la válvula compensa las variaciones de presión y temperatura que pudieran producirse y regula tanto la entrada de agua fría como la de agua caliente. Dicho control ejercido en su totalidad por el termostato interno, es automático y viene determinado por la posición prefijada en la válvula. (Watts Industries, 2016). Es un dispositivo totalmente mecánico, que no permite un fácil ajuste de la temperatura por parte del usuario.

Considerando que al ducharse se gasta entre 80 y 120 litros de agua, que un baño de tina requiere alrededor de 200 litros de agua y una llave abierta puede significar un gasto de 12 litros de agua por minuto (Espinosa Sarria, 2007), se puede concluir que la implementación del sistema propuesto optimizará el consumo de agua, pues se podrá controlar por parte del usuario la temperatura de salida del agua para su uso de una manera rápida y sencilla, disminuyendo la cantidad de agua que se desperdicia hasta lograr dicha temperatura.

2.2 Teoría de control

El funcionamiento de cualquier sistema, sea este electrónico, mecánico o incluso biológico, depende de las características e interacciones entre sus componentes. Se denomina sistema de control al constituido por un conjunto de elementos conectados de tal forma que permiten reglamentar o administrar otro sistema llamado planta o proceso. (Bahón & Raya Giner, 2004). Los sistemas de control manipularan de alguna manera, una o varias entradas (acción de control) para influenciar en el comportamiento de una o más salidas, y así poder controlarlo y predecirlo con independencia de las perturbaciones que ataquen al sistema (Miranda, 2012). En estos sistemas se encuentran varios elementos y señales características que deben ser diferenciados para comprender su funcionamiento.

2.2.1 Planta

Es la parte del sistema a controlar se lo denomina proceso. En la planta se produce la transformación de la energía suministrada que se desea controlar, en general la energía eléctrica suministrada al actuador se transforma en energía mecánica –motor-, en energía potencial –depósito de líquido-, cinética –caudal impulsado por una tubería-, etc. (Gordillo Guambaña, 2014).

En la Figura 1-2 se observan dos tipos de planta de acuerdo al número de entradas. En el ejemplo se muestra una casa en la que se realiza un cambio de temperatura mediante una chimenea encendida, que sería la entrada, y la salida sería la temperatura en el interior de la vivienda; este tipo de sistema se denomina SISO (Single Input Single Output). Si se incrementa un ventilador para bajar la temperatura de la casa se tiene una entrada más y se convierte sistema MISO (Multiple Input Single Output).

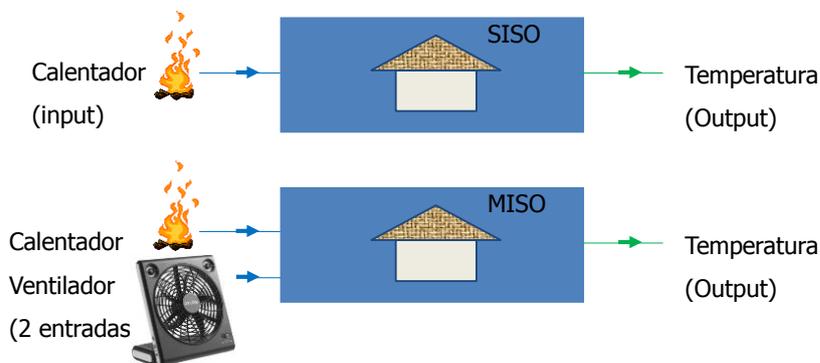


Figura 1 - 2 Tipos de planta de acuerdo al número de entradas.

Fuente: (Aimagin, 2017)

2.2.2 Controlador

Proporciona el estímulo o señal al proceso para controlar su comportamiento y alcanzar la salida deseada. Se conoce también como regulador. En la Figura 2 - 2 se muestra un esquema de un sistema de control básico, donde se aprecia un controlador encargado de regular la velocidad del ventilador y la potencia de un calentador para que la temperatura de una casa sea la requerida por el usuario, mediante la señal Set Point. La temperatura de la casa es medida por un sensor.

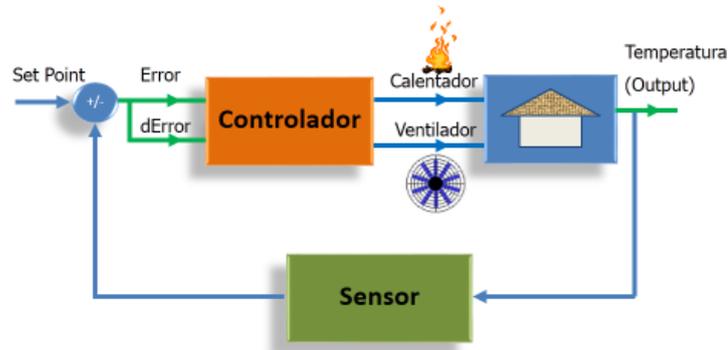


Figura 2 - 2 Esquema de un sistema de control básico.

Fuente: (Aimagin, 2017)

2.2.3 Sensor

Es el elemento que permite supervisar la salida de la planta, en un lazo de control comúnmente va en la realimentación, como se muestra también en la Figura 2 - 2. Los sensores transforman una variable física - temperatura, posición, flujo, velocidad- en una señal eléctrica.

2.2.4 Actuador

Es elemento final de control, la parte de proceso que provoca físicamente la dinámica de todo el sistema. Normalmente consta de un adaptador de potencia y del elemento de actuación.

2.2.5 Transmisor

Recoge la señal del sensor, la acondiciona y la compara con la señal de consigna (Set Point), para generar una señal de error que ingresará al regulador; en general son elementos que no cambian el proceso.

Un sistema queda especificado si es posible conocer la función de salida que corresponde a cada posible función de entrada. El proceso de análisis del sistema se hace con la finalidad de construir,

probar y validar un modelo. Se puede obtener dos tipos de modelos: los axiomáticos y los empíricos, los primeros describen un sistema en base a ecuaciones físicas que explican las interioridades dinámicas del sistema, mientras que los segundos relacionan la entrada y salida utilizando el concepto de caja negra. De cualquier forma, el objetivo es el mismo obtener una expresión matemática llamada función de transferencia del sistema, tal que, explique cuantitativamente el comportamiento dinámico del sistema ante cualquier tipo de excitación temporal. (Valdivia Miranda, 2012)

2.2.6 Función de transferencia

Se describe mediante una ecuación diferencial lineal e invariante en el tiempo se define como el cociente entre la transformada de Laplace de salida (función de respuesta) y la transformada de Laplace de la entrada (función de excitación); bajo la suposición de que todas las condiciones iniciales son cero. (Moreno, Garrido, & Balaguer, 2003). Utilizar el dominio de Laplace es un método ampliamente utilizado para resolver ecuaciones matemáticamente, pero dentro del control automático se asocia con el concepto de función de transferencia. En general un proceso descrito a través de una función $g(t)$ recibe una entrada $u(t)$ y genera una salida $y(t)$. Si se lleva estas señales al dominio de Laplace se tiene una entrada $U(s)$ que genera una salida $Y(s)$, como se muestra en el sistema en el dominio de Laplace de la Figura 3-2.

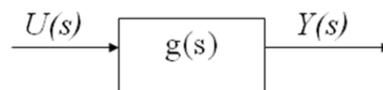


Figura 3 - 2 Sistema en el dominio de Laplace

Fuente: Freire Paúl, 2017

De modo que

$$Y(s) = g(s) \times U(s) . \quad (1)$$

2.2.7 Sistemas de primer orden

Se caracterizan por tener un solo elemento capaz de almacenar energía y por lo tanto son representados por ecuaciones diferenciales ordinarias de primer orden. Estos sistemas pueden ser de tipo eléctrico, mecánicos (rotacionales, y traslacionales), hidráulicos, térmicos o híbridos. (Ogata, Dinámica de sistemas, 1987)

En la ecuación general aparece solamente la derivada primera del lado izquierdo (el de la variable de estado). Se reducen al formato siguiente:

$$\tau \frac{dy}{dt} + y = ku \quad (2)$$

Donde k se denomina ganancia del proceso y τ es la constante de tiempo del sistema.

En el caso de la presente investigación, el sistema a controlar es de tipo térmico, por lo cual se analiza como un sistema de primer orden.

En los sistemas de primer orden simple, los valores característicos pueden ser determinados experimentalmente ante la respuesta de una entrada en escalón unitario, como se muestra en la Figura 4 - 2 . La ganancia estática, G_0 , será el valor final de la señal de salida y la constante de tiempo, T_0 , está dada por el tiempo en que alcanza 0.632 veces el valor de G_0 o tres veces su valor coincidirá con el tiempo de establecimiento, T_s , esto es, el tiempo en alcanzar la señal 0.95k. (Universidad Politécnica de Madrid, Dpto. Electrónica, Automática e Informática Industrial)

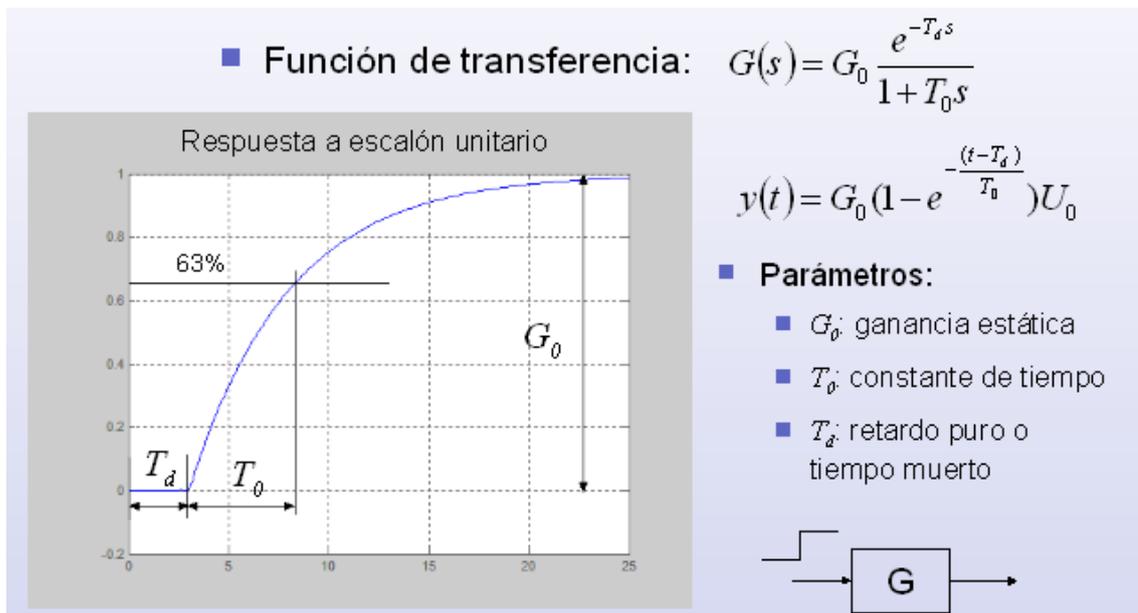


Figura 4 - 2 Respuesta de un sistema de primer orden simple ante una entrada escalón

Fuente: (Tumero, 2017)

2.3 Lógica difusa

Simula la manera en que el cerebro capta información y reacciona a estímulos externos, aplicando la teoría básica de conjuntos, permitiendo agrupar la información en subconjuntos asignando un

grado de pertenencia a un elemento. La teoría clásica de conjuntos se basa en el principio básico de que un elemento pertenece o no a un conjunto. La idea básica de un conjunto difuso es que un elemento forma parte de un conjunto con un determinado grado de pertenencia. De este modo una proposición no es totalmente cierta o falsa. Este grado de verdad se expresa mediante un entero en el intervalo $[0; 1]$.

Los sistemas de este tipo son muy recomendables en aquellos problemas muy complejos donde no existe un modelo matemático simple asociado. Igualmente en procesos que obedecen a un comportamiento no lineal, la solución difusa plantea grandes ventajas puesto que requiere que el conocimiento experto sea expresado lingüísticamente, requisito que es normalmente fácil de obtener. (González Morcillo, 2011)

Los sistemas expertos de control difuso basados en reglas, conocidos como controladores borrosos o FLC (Fuzzy Logic Controllers), o también, sistemas de inferencia borrosa o FIS (Fuzzy Inference Systems), son sin duda la aplicación más extendida de este tipo de lógica.

Zadeh propone en 1965 por primera vez la noción de Conjunto Difuso. Este hecho marca el principio de una nueva teoría denominada Teoría de Conjuntos Difusos. El Principio de Incompatibilidad (Zadeh L. , 1965) dice que la descripción del comportamiento de un sistema complejo no puede realizarse de forma absolutamente precisa. Como solución Zadeh propone la necesidad de herramientas capaces de manejar de forma fiel y confiable la información imprecisa, que deben tener dos características:

Representación de la información imprecisa: Se propone el empleo de la Teoría de conjuntos difusos. Se describe la experiencia de los sistemas complejos en sus relaciones entrada-salida mediante proposiciones condicionales del tipo Si-Entonces (Ejemplo: Si la temperatura del agua es muy alta entonces se agrega agua fría)

Inferencia sobre información imprecisa: Para combinar esta información y obtener nuevos hechos, Zadeh establece la necesidad de un método de inferencia general e introduce lo que se conoce como Regla Composicional de Inferencia. La inferencia es un proceso de propagación de restricciones difusas.

De esta forma un sistema puede tomar una decisión con valores incompletos o información incierta. El razonamiento exacto es un caso particular del razonamiento aproximado. Cualquier sistema lógico puede ser fuzzificado. Mediante lógica difusa se puede expresar el conocimiento

humano del comportamiento de un sistema de ingeniería. La lógica binaria clásica puede verse como un caso particular de lógica difusa.

Los conceptos se asocian a conjuntos difusos (asociando los valores de pertenencia) en un proceso llamado fuzzificación. Una vez con los valores fuzzificados calculados se trabaja con reglas lingüísticas y obtener una salida, que podrá seguir siendo difusa o defuzzificada para obtener un valor discreto crisp o clásico.

Como ejemplo se toma la visión de la lógica clásica para definir a una persona como “ALTO”, si este supera 1.80m de altura, caso contrario se la define como “NO ALTO” (Figura 5 - 2). Este límite rígido no corresponde a la realidad, puesto que si la persona mide 1.79m, será considerada “NO ALTO”, a pesar de serlo solo por un cm. Se hace necesario contemplar la variable altura desde el punto de vista de la lógica difusa.



Figura 5 - 2 Visión de una variable de acuerdo a la Lógica Difusa y a la Lógica Clásica

Fuente: (Vivanco, 2017)

Un conjunto difuso se define en general como un conjunto con límites difusos. Sea X el Universo del discurso, y sus elementos se denotan como x. En la teoría clásica de conjuntos se define un conjunto C sobre X mediante la función característica de C como f_c . (González Morcillo, 2011)

$$f_c(x) \begin{cases} 1 & \text{cuando } x \in C \\ 0 & \text{cuando } x \notin C \end{cases} \quad (3)$$

Este conjunto lleva al universo X a un conjunto de dos elementos, donde la función $f_c(x)$ es 1 si el elemento x pertenece al conjunto C y 0 si el elemento x no pertenece al conjunto C. Generalizando esta función para que los valores asignados a los elementos del conjunto caigan en

un rango particular y así indicar el grado de pertenencia de los elementos a ese conjunto, se tiene una función de pertenencia de un conjunto difuso. La función de pertenencia μ_A por la que se define un conjunto difuso A, es:

$$\mu_A = X \rightarrow [0,1] \quad (4)$$

Donde $\mu_A(x) = 1$ si x está totalmente en A, $\mu_A(x) = 0$ si x no está en A y $0 < \mu_A(x) < 1$ si x está parcialmente en A. Este valor entre 0 y 1 representa el grado de pertenencia de un elemento x a un conjunto A.

Así, el intervalo de la ecuación (4) es de números reales e incluye los extremos. Aunque $[0,1]$ es el rango de valores más utilizado para representar funciones de pertenencia, cualquier conjunto arbitrario con alguna ordenación total o parcial podría ser utilizado.

La programación clásica no es suficiente en cuando un sistema no ha sido programado explícitamente para realizar una acción y es necesario que la realice. Ante esta situación el sistema debe razonar en función de los hechos y reglas que conoce. Cuando el número de hechos y reglas aumenta, el sistema tiene que poder verificar gran cantidad de hechos que nacen en las etapas de razonamiento. Se hace necesario entonces el concepto de regla difusa.

Para trabajar con conocimiento expresado en forma de proposiciones enunciadas en lenguaje natural se utiliza razonamiento aproximado. Por ejemplo “La temperatura tiene un valor negativo grande”.

Se debe transformar esta expresión usando variables lingüísticas al lenguaje natural, para lo cual:

1. Se selecciona un símbolo para representar la variable física. T para “temperatura”.
2. Se elige un símbolo para representar el valor particular. NG “negativo grande” de la variable física “temperatura”.
3. La expresión en lenguaje natural pasa a ser: T es NG

Con lo que se obtiene una proposición atómica difusa. La interpretación de la expresión en el ejemplo está dada por la pertenencia de la variable física temperatura T al conjunto difuso NG, $\mu_{PG}(t)$, donde t denota un valor arbitrario del universo del discurso U. De esta forma se determina en qué grado la expresión es satisfecha por un valor específico de la variable T.

Se puede combinar proposiciones difusas a través de conectores lingüísticos tales como “y”, “o” y “no” para proposiciones difusas más complejas “T es X y V es Y”, “T es no X”, etc. El grado de satisfacción de una expresión es un conjunto difuso por lo que estos conectores deben utilizar operadores de conjuntos difusos.

Una regla difusa se expresa simbólicamente como:

SI <proposición difusa> ENTONCES <proposición difusa>

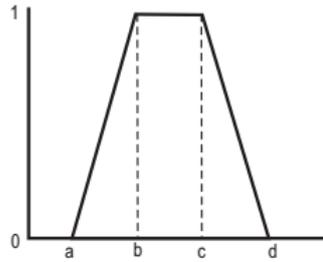
Donde <proposición difusa> puede ser una proposición difusa atómica o compuesta. Se define una proposición sencilla de este tipo mediante:

p: SI X es A ENTONCES Y es B

A diferencia de los sistemas de reglas clásicos en donde si el antecedente es cierto, el consecuente es también cierto, en los sistemas fuzzy el antecedente y consecuente de una regla pueden tener múltiples partes. Si el antecedente difuso es cierto con cierto grado de pertenencia, todas las reglas se ejecutan parcialmente, y el consecuente es cierto en cierto grado.

En teoría es posible usar para la fuzzyficación de variables físicas cualquier función que se encuentre en el intervalo entre 0 y 1, aunque se han adoptado de manera clásica en la mayoría de los trabajos las funciones de tipo trapezoidal, triangular, gaussiana, sigmoidea, pi y singleton. Esta última hace referencia a que un único valor de pertenencia es igual a 1 en un solo punto y el resto es igual a cero (Vargas Toro, 2012). En la presente investigación se hace uso de la función trapezoidal y triangular que son las que mejor representan los conjuntos de las variables involucradas en el sistema a controlar.

Función trapezoidal: En la Figura 6 – 2 se tiene cuatro valores que pueden ser descritos como a, b, c, y d. Esta función tiene un valor de 0 cuando no se encuentra entre a y d, un valor de 1 cuando x esta entre b y c, mientras que los rangos entre a y b, y entre c y d toma valores entre 0 y 1. Se utiliza cuando las variables físicas presentan intervalos de pertenencia total al conjunto difuso, e intervalos en los que la variación del nivel de pertenencia es lineal con respecto al cambio de la variable física.

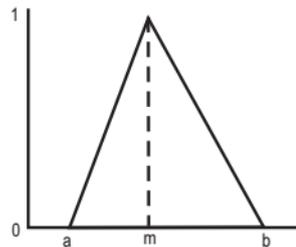


$$Función \mu A(x) = \begin{cases} 0, & \text{si } x < a \text{ ó } x > d \\ \left[\frac{x-a}{b-a} \right], & \text{si } a < x \leq b \\ 1, & \text{si } b \leq x \leq c \\ \left[\frac{d-x}{d-c} \right], & \text{si } c \leq x \leq d \end{cases}$$

Figura 6 - 2 Función Trapezoidal

Fuente: (Maguiña Pérez, 2010)

Función Triangular: Definida mediante el límite inferior a , el superior b y el valor modal m , tal que $a < m < b$ como se muestra en la Figura 7 - 2. Estas funciones son utilizadas para fuzzyficar variables físicas que no se mantienen fijas en un valor, pero que tienen una relación lineal como por ejemplo la velocidad de un móvil, o variables, que por su naturaleza, varían muy rápido en frecuencia, pero con pequeños cambios en amplitud, como la temperatura en un mezclador. La función no necesariamente debe ser simétrica.



$$Función \mu A(x) = \begin{cases} 0, & \text{si } x \leq a \\ \left[\frac{x-a}{m-a} \right], & \text{si } a < x \leq m \\ \left[\frac{b-x}{b-m} \right], & \text{si } m \leq x \leq b \\ 0, & \text{si } x \geq b \end{cases}$$

Figura 7 - 2 Función Triangular

Fuente: Fuente: (Maguiña Pérez, 2010).

Inferencia Difusa: Es un proceso de obtener un valor de salida para un valor de entrada empleando la teoría de conjuntos difusos. Se interpretan las reglas del tipo SI-ENTONCES para

los valores actuales de las variables de entrada para obtener los resultados de las variables lingüísticas de salida. (Maguiña Pérez, 2010)

El mecanismo de inferencia representa el núcleo del sistema, ya que agrupa toda la lógica de inferencia borrosa del sistema, del uso de las reglas durante ésta, determinar que reglas están activadas, etc. (Aja Fernández, 2003).

Existen varios modelos de inferencia, de los cuales los más utilizados son el de Mamdani y el de TSK (Takagi, Sugeno y Kang).

Inferencia de Mamdani: Desarrollado por Ebrahim Mamdani en 1975, es uno de los más utilizados ya que es más intuitivo, y se adapta mejor al lenguaje humano. El proceso se realiza en cuatro pasos:

1. Fuzificación de las variables de entrada.
2. Evaluación de las reglas.
3. Agregación de las salidas de las reglas.
4. Defuzificación.

El formato de estas reglas es:

$$\text{SI } u_1 \text{ es } A_1 \text{ Y } u_2 \text{ es } A_2 \text{ Y...Y } u_n \text{ es } A_n \text{ ENTONCES } v \text{ es } B$$

Donde los u_i y v son variables lingüísticas, y los A_i y B representan los valores lingüísticos (términos lingüísticos asociados a conjuntos borrosos) que dichas variables pueden tomar. Cada una de las reglas tiene un lado izquierdo formado por uno o varios antecedentes y un lado derecho (consecuente), por ejemplo “SI la temperatura es alta y la presión es media ENTONCES bajar un poco el caudal de combustible”.

La Inferencia en sistemas Mamdani, según (Maguiña Pérez, 2010), comprende dos fases básicas:

Cálculo de la parte SI de las reglas: Evalúa el grado de certeza o activación del antecedente de cada regla para los valores actuales de las variables de entrada. En esta fase se usan generalmente los operadores Mínimo y Producto para evaluar el “AND” que conecta las proposiciones del antecedente.

Cálculo de la parte ENTONCES de las reglas: Partiendo del grado de activación se determina la conclusión de la regla. Se asigna a cada variable de salida del consecuente el conjunto borroso correspondiente modificado en el grado de certeza o activación. La función de pertenencia del conjunto modificado se representa por $\mu_{LD(n)}(v)$, donde “n” es la regla evaluada y “v” es la variable de salida. La modificación del conjunto borroso de salida se realiza mediante la implicación borrosa.

Cada regla se dispara dependiendo del estado actual de las variables de entrada del sistema y la estructura de los antecedentes de cada regla, como se muestra en la Figura 8 - 2, para un sistema tipo Mandani.

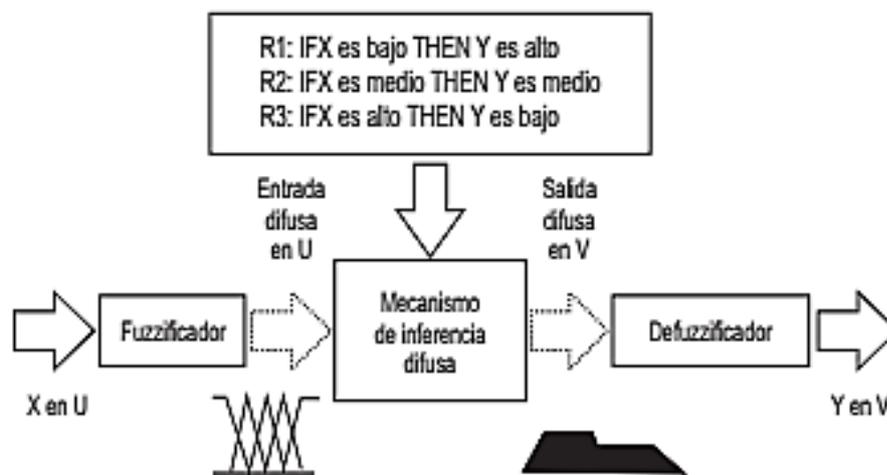


Figura 8- 2 Configuración básica de un sistema tipo Mamdani.

Fuente: Fuente: (Maguiña Pérez, 2010)

Implicación borrosa: Definidos dos conjuntos borrosos A y B en los universos de discurso U y V, respectivamente, una implicación difusa de A en B, $A \rightarrow B$, es una relación difusa en $U \times V$. Existen muchas implicaciones borrosas, entre ellas la de Zadeh y Larsen, pero la de Mamdani es una de las de más sencilla implementación. Se implementa usando el mínimo:

$$\mu_{A \rightarrow B}(x,y) = \min \{ \mu_A(x), \mu_B(y) \} \quad (5)$$

Agregación

Regla 1: SI u_1 es A_1 Y u_2 es A_2 Y...Y u_n es A_n
ENTONCES v es B_1

Regla 2: SI u_1 es A_1 Y u_2 es A_2 Y...Y u_n es A_n

ENTONCES v es B2

.....

Regla M: SI u1 es A1 Y u2 es A2 Y...Y un es An
ENTONCES v es BM

El conjunto borroso global de salida está dado por la unión o agregación borrosa de los conjuntos borrosos resultantes B_j para cada regla:

$$B = B1 \oplus B2 \oplus \dots \oplus BM \quad (6)$$

Donde \oplus significa operador máximo. Aunque hay otros operadores disponibles, el máximo es el más usado para efectuar esta operación.

Desborrosificación: Es el proceso de convertir los valores borrosos de las variables de salida resultantes del proceso de inferencia, en información precisa expresada mediante un valor nítido, pasando de lo difuso a lo real. Esta etapa en los sistemas Mamdani sirve para dar una información precisa a la planta o actuadores, ya el conjunto borroso de salida para estos sistemas no se puede aplicar directamente.

Se presenta una breve descripción de algunos de los principales métodos de desborrosificación, tomando como base para la descripción de estos métodos los conceptos dados por (Mugica Alvarez, 2007) sobre este tema.

Centro de Máximos (COM): Se consideran los k términos lingüísticos de la variable borrosa de salida que resultan de la aplicación de la inferencia, y sus valores máximos de pertenencia, c_{max_k} , tal que $\mu(c_{max_k}) = 1.0$. A los c_{max_k} se les conoce también como los “valores típicos” de la variable.

La contribución de cada regla al valor de salida nítido y_{COM} es ponderada mediante la altura α_j de la j-ésima conclusión de la siguiente manera:

$$y_{COM} = \frac{\sum_K c_{max_k} * \alpha_j}{\sum_j \alpha_j} \quad (7)$$

Suma de los Centros de Áreas (SCOA): El método de Suma de los Centros de Área permite tomar en cuenta el perfil de las funciones de pertenencia, no contemplado en el método anterior, lo que hace que el costo computacional aumente de manera significativa. Se considera entonces

individualmente el centro de gravedad (área cuando es una sola salida) de los conjuntos borrosos obtenidos en la aplicación de la inferencia y, posteriormente, se halla la media ponderada del conjunto total, que resulta de la agregación de dichos conjuntos borrosos. Esto puede ser calculado, si la función de pertenencia $\mu_B(y)$ es continua, con:

$$\frac{\int_S y \sum \mu_B(y) dy}{\int_S \mu_B(y) dy} \quad (8)$$

En esta expresión matemática “y” es la variable de integración y “S” es el soporte del conjunto borroso global. Este método presenta la limitación de que el área de superposición entre una conclusión y otra es tomada en cuenta dos veces lo que puede producir efectos indeseables, dependiendo del caso que se aborde.

Centro de Área (COA): La idea de este método es similar al anterior pero corrige el efecto de superposición de las áreas entre una conclusión y otra, que se producía en el método SCOA. Esto se hace simplemente tomando en cuenta las áreas sólo una vez; para ello se considera únicamente el perfil de la función de pertenencia que resulte mayor en cada punto.

Esto puede ser calculado, si la función de pertenencia $\mu_B(y)$ es continua, con:

$$\frac{\int_S \mu_B(y) y dy}{\int_S \mu_B(y)} \quad (9)$$

Aunque las expresiones matemáticas de los métodos COA y SCOA pueden parecer muy similares, el costo computacional del COA es bastante superior ya que requiere calcular o bien los puntos de intersección entre cada una de las funciones que intersectan, o bien, algo aún más costoso, evaluar todas las funciones punto a punto y seleccionar el de mayor valor en cada caso.

2.4 Protocolos de comunicación

El bus I2C (Inter Integrated Circuits bus) diseñado por Philips, es un protocolo de comunicación en serie y sincrónica entre micro controladores, memorias y otros dispositivos digitales. Transmite información entre dispositivos a una velocidad, de alrededor de 100 Kbits por segundo, usando dos líneas de señal y un común o masa. Una de las señales del bus utiliza pulsos de reloj para marcar el tiempo, y la otra se utiliza para intercambiar datos. En la Figura 9 – 2 se muestra un diagrama de conexión de dispositivos I2C.

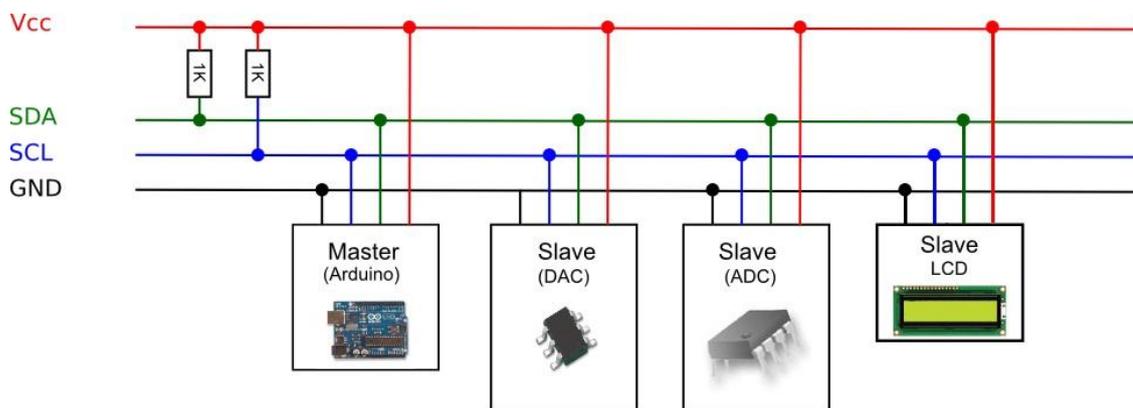


Figura 9 - 2 Diagrama de conexión de dispositivos I2C

Fuente: (PROMETEC, 2016)

Todos los componentes se conectan en paralelo a las dos líneas del Bus, SDA y SCL. Se usan resistencias Pullup conectadas a SDA y SCL debido a que el bus es activo bajo (La señal activa es un 0) La norma propone un sistema de arbitraje para transferir el control en el caso de que exista más de un Master, pero solo uno a la vez puede ser el master y los demás se configuran como esclavos. Un dispositivo conectado al bus I2C tiene una dirección única de 7 bits. Cada mensaje y orden transmitido al bus, incluye esta dirección, para saber cuál de los dispositivos es el receptor del mensaje. Se debe revisar la información técnica del fabricante del componente para conocer su dirección, pero si no se dispone, existen programas que determinan la dirección de un dispositivo conectado al bus. Teóricamente se pueden conectar $2^7 = 128$ dispositivos. (PROMETEC, 2016)

2.5 Calentadores solares domésticos

Existen varios tipos de calentadores solares, pero entre los más eficientes se encuentran los de tubos al vacío. Cada tubo forma un sistema cerrado de evaporación-condensación, que suele ser un tubo metálico largo y fino herméticamente cerrado, el cual contiene un líquido en equilibrio con su vapor (gas) a determinada presión (vacío) y temperatura. Si la temperatura aumenta, aumenta la fase gaseosa; y si disminuye, aumenta la fase líquida. Existen de diferentes dimensiones. Los de un metro cuadrado de área de captación calientan, como promedio, 150 litros de agua a 45°C , lo que es suficiente para una vivienda de cuatro a cinco habitantes, para el baño, cocina y el lavado de la ropa. Producen, como promedio, 3 KWh cada día como energía térmica, o sea, 1 MWh al año. La presión (vacío) dentro del tubo se selecciona de tal forma que empiece la evaporación a 25°C , lo que garantiza el funcionamiento del colector solar aún con baja radiación. (Bérriz, 2011)

CAPÍTULO III

3. METODOLOGÍA DE LA INVESTIGACIÓN

En este capítulo se describe el funcionamiento del sistema actual. Son expuestos además los requerimientos del sistema y los criterios adoptados en la elección del hardware y software del prototipo, en base a los objetivos planteados para el funcionamiento del mismo. Se describen también los módulos que componen el sistema y los procesos que realizan.

3.1 Análisis del sistema actual

Se recolectaron muestras de los valores de las variables del sistema durante 7 días, con la finalidad de determinar cual es la cantidad de agua consumida durante una ducha con el caudal total del agua abierto y el tiempo que el usuario tarda en estabilizar la temperatura del líquido.

Como se puede apreciar en el Gráfico 3-1 que muestra las curvas de temperatura en función del tiempo con control manual de la ducha por parte de dos usuarios diferentes (a) y (b), es muy difícil alcanzar una temperatura estable manipulando el grifo monomando manualmente. Se asume que el usuario busca una temperatura de aproximadamente 36°C, por ser muy cercana a la temperatura normal del cuerpo humano.

Una vez que el usuario abre una válvula monomando de la ducha, por lo general la pone en la posición de apertura total de agua caliente hasta que percibe que la temperatura del agua empieza a ascender. Este tiempo puede ser de pocos segundos hasta varios minutos, dado que dentro de la tubería de agua caliente existe agua que no circulaba por algún tiempo y ya se encuentra a temperatura baja, y ese volumen de líquido debe ser evacuado antes de que el agua caliente empiece a salir. En ese momento empieza el proceso de control en el cual si la sensación térmica es muy alta, se gira el mando en dirección anti horaria para que ingrese agua fría. En el caso contrario se girará la válvula en sentido horario para incrementar la cantidad de agua caliente en la mezcla. Este serie de pasos se repite cada vez que se siente un cambio no solicitado de temperatura, cuando el agua se torna muy caliente o fría, hasta finalizar el proceso de duchado.

En función de las mediciones realizadas, se obtiene en promedio un caudal de ocho l/min., para una afusión que tomó tres minutos. El tiempo que tarda el agua caliente en llegar a la ducha varía alrededor de los 30 segundos en este caso, pero depende de la distancia del reservorio a la salida

de agua. En las tomas de agua más lejanas se realizaron mediciones que alcanzaron 4 minutos hasta que el agua alcance los 36°C, que significa que se desperdiciaron de 4 a 32 litros de agua.

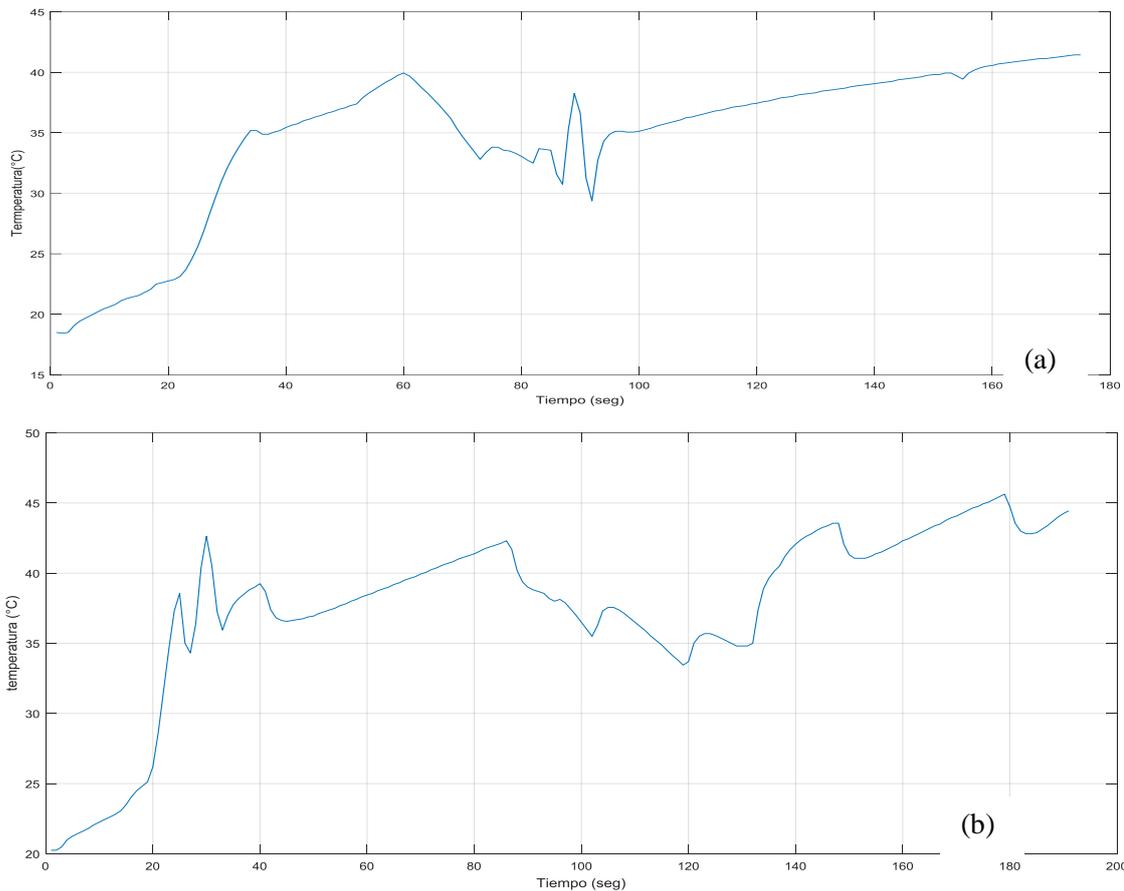


Gráfico 3-1 (a) y (b) Curvas de temperatura en función del tiempo con control manual de la ducha por parte de dos usuarios diferentes.

Realizado por Freire, Paúl, 2017

Observando las curvas obtenidas, hay momentos en que la temperatura está muy por debajo o sobre los 36°C, en los que el usuario se retira de la caída de agua dado que el agua está muy caliente o muy fría. Se vuelve difícil cuantificar cual es el desperdicio en esos momentos, dado que depende de la habilidad de cada usuario de calibrar la temperatura de agua y la temperatura que considere tolerable, pero considerando que el rango de temperatura aceptable está entre 33°C y 40°C, ese nivel se logra durante cortos intervalos de tiempo. En total se gastó 28 litros. Se puede hablar entonces de un desperdicio de alrededor de ocho litros en esta ducha en particular de tres minutos. Esto representa el 28% del total de agua utilizada.

Según (FACUA, 2017), una reconocida organización española de defensa al consumidor, la media de consumo de agua en la ducha se encuentra en 95 litros, en una ducha de cinco minutos.

Con estos datos, haciendo una estimación optimista en el que el desperdicio de agua por persona es de 10 litros por ducha, no solo por los factores expuestos anteriormente, sino también por un tiempo excesivo de baño, en un hogar de tres personas se malgastarían 30 litros diarios. En un año se alcanza los 10950 litros. En nuestro medio el ahorro económico no es significativo, ya que el metro cúbico del líquido vital cuesta 0.49 usd, pero en países europeos como es el caso de Alemania, el metro cúbico cuesta 1.91 euros.

De acuerdo a la Empresa Municipal de Agua Potable y Alcantarillado de la ciudad de Riobamba EP-EMAPAR, esta urbe cuenta con alrededor 225 mil habitantes. Con la misma estimación, con un manejo adecuado del recurso, se lograría un ahorro de 2'463750 metros cúbicos de agua.

Se debe tomar en cuenta que el beneficio no es solo económico y de comodidad del usuario al no preocuparse de cambios no deseados de temperatura, sino más que nada ambiental.

3.2 Requerimientos del sistema

El sistema mezclará agua caliente y fría proveniente de dos tomas independientes, una de agua fría, proveniente de un reservorio o de la red local de agua potable, y una toma de agua caliente que es provista por un calentador solar doméstico. El esquema hidráulico del sistema se muestra en la Figura 1-3.

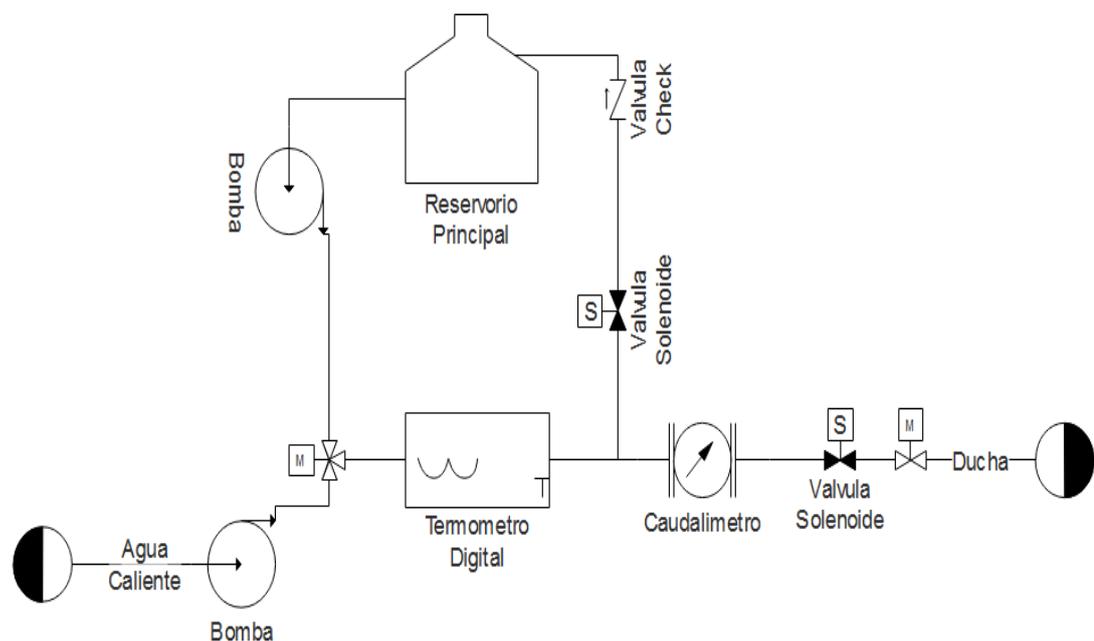


Figura 1 - 3 Esquema hidráulico del sistema propuesto.

Fuente: Freire, Paúl, 2017

El agua caliente podría obtenerse también de un calefón de gas o eléctrico, dado que las instalaciones de esos dispositivos son de similares características. Se requiere:

- Mantener el valor de la temperatura deseada por el usuario con un rango mínimo de error. Se propone un error de +/- 2% en relación con el set point.
- Evitar el desperdicio de agua que ocurre desde el momento en que se abre la llave hasta que el agua caliente empieza a salir por la ducha.
- Tener una interface física que permita al usuario especificar los valores de temperatura y caudal deseados.
- Monitorear la temperatura, caudal y volumen del agua utilizada.
- Emitir mensajes que permitan conocer al usuario el estado del sistema.

El sistema constará de dispositivos que permitan censar las variables a controlar, que son temperatura y caudal. Los datos aportados por estos deberán ser procesados para ingresar a la lógica de control, que enviará los parámetros necesarios a los actuadores (servomotores y electroválvulas) para alcanzar o mantener los valores requeridos por el usuario a través de un panel de control. El sensor de temperatura debe ser sumergible y debe poder censar temperaturas en el rango de los 32°C a 40°C, que es el rango requerido de la salida. El caudal de agua de salida será de máximo 10 l/min, que es el caudal de agua domiciliario.

3.3 Concepción de la arquitectura general del sistema

En la Figura 3 – 2 se muestra un diagrama general del sistema propuesto. Este constará de 6 módulos, Módulo de sensores, módulo de control, módulo de actuadores, módulo interfaz, módulo de alimentación y módulo de monitoreo.

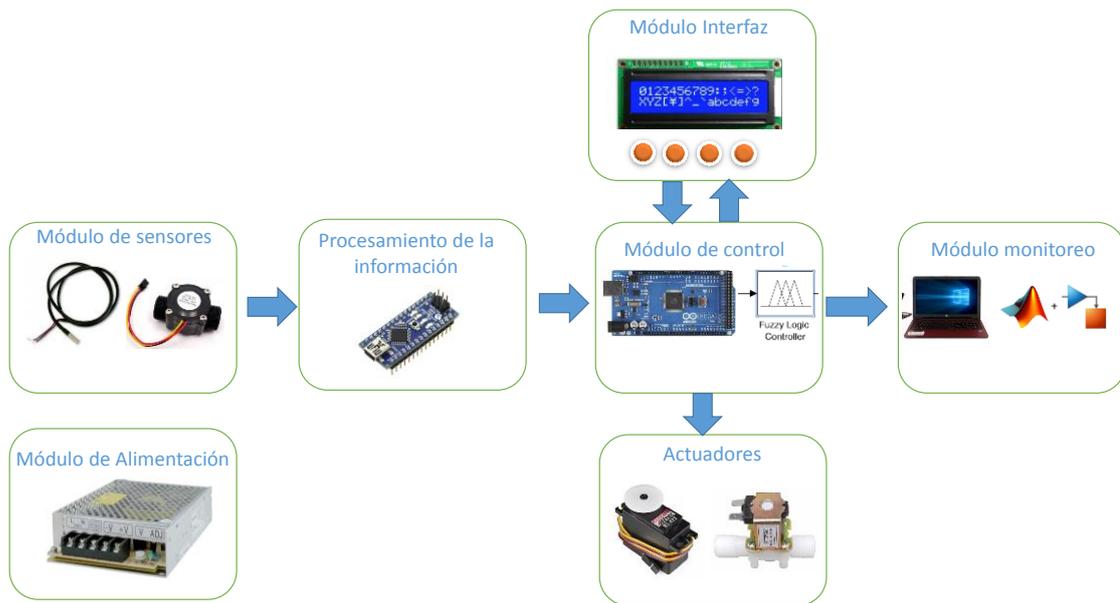


Figura 3 - 2 Diagrama del sistema propuesto

Fuente: Freire, Paul, 2017

El *software* en el que se desarrollará el sistema debe manejar lógica difusa de forma que se pueda realizar cambios en los controladores de una manera ágil para la optimización del mismo. Debe también ser compatible con una placa de desarrollo y soportar comunicación entre sus componentes.

3.3.1 Módulo de sensores

Compuesto por un sensor de temperatura y un caudalímetro, cuya función es registrar la variación de la temperatura del agua a la salida del sistema y para registrar el flujo de líquido que pasa por el elemento final (ducha), respectivamente.

Entre los sensores más comunes empleados para medir temperatura con instrumentación electrónica se tienen: RTDs, termistores, sensores de circuito integrado (IC), Una termocupla produce un voltaje que está relacionado con la temperatura, así mismo en una resistencia metálica se aprovecha el fenómeno de variación de la resistencia con la temperatura para producir una señal de voltaje que sea proporcional a la temperatura.

Un sensor de flujo es un dispositivo que permite medir la cantidad de agua que atraviesa una tubería. El caudal es la relación entre volumen y tiempo. Las unidades en el sistema internacional son m^3/s , siendo otras unidades habituales l/s y l/min . El caudal depende de diversos factores, principalmente de la sección de tubería y la presión de suministro.

3.3.2 Módulo de Control

Constituido por la placa de desarrollo. Se encarga de evaluar las señales de los sensores, determinar cuál es el estado en que se encuentra el sistema, administrar los mensajes que serán mostrados al usuario a través del módulo de interfaz, realizar los procesos de fuzzificación y defuzzificación y enviar los valores necesarios a los actuadores. Debe tener la capacidad de comunicarse en tiempo real con el módulo de monitoreo.

3.3.3 Módulo de Actuadores

Constituido por la válvula monomando a la cual se deberá adaptar un servomotor que permita controlar su posición, que a su vez controla la mezcla de agua caliente y fría, una válvula motórica reguladora para el caudal, una electro válvula encargada de abrir o cerrar el paso de agua a la ducha y una electroválvula que se abre en caso de necesitar recirculación.

3.3.4 Módulo de interfaz

La interfaz es donde el usuario se comunica con la máquina, para controlar su funcionamiento. Dichas acciones se realizarán a través de botones, ubicados en la parte externa del dispositivo. Sus funciones son las siguientes:

- Encendido y apagado.
- Configuración de los Set Point de temperatura y caudal.
- Informar la temperatura actual del agua
- Mostrar el volumen de agua utilizado.

3.3.5 Módulo de Alimentación

Compuesto por los dispositivos encargados de entregar los voltajes y corrientes necesarias para el funcionamiento de los elementos que conforman el prototipo. El principal dispositivo de este módulo será una fuente, que transformará el voltaje de la red eléctrica en uno o varios voltajes.

3.3.6 Módulo Monitoreo

Constituido por un ordenador (Figura 3 - 3) en el cual se dispone del *software* Simulink, en el cual se puede visualizar en tiempo real los valores de las variables durante la ejecución del programa de control. Este módulo solo es usado durante la fase de programación y pruebas del sistema, ya que el prototipo una vez optimizado funcionará sin necesidad de un computador.

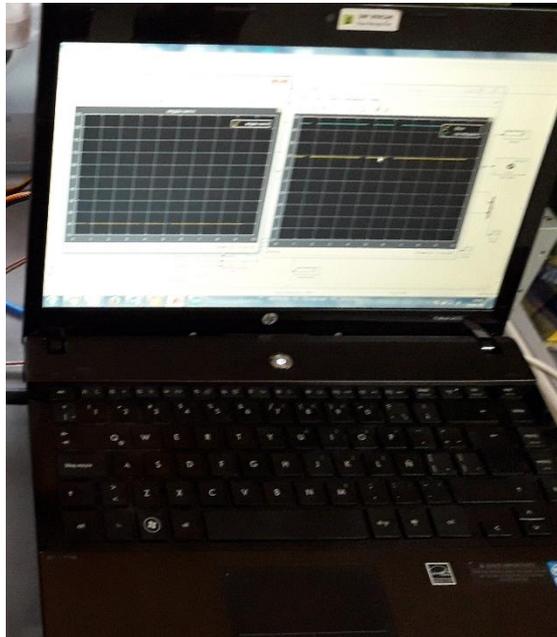


Figura 3 - 3 Computador utilizado como módulo de monitoreo

Fuente: Freire, Paul, 2017

3.4 Selección de los elementos de *hardware* del Sistema

Definidos los requerimientos de diseño y concebida la arquitectura general, se presentan a continuación los criterios de selección los elementos que compondrán el hardware del sistema.

3.4.1 Módulo de sensores

Compuesto por sensores de temperatura y caudal. La señal de cada uno de ellos es recogida por una placa de desarrollo Arduino Nano V3.0, que correrán programas que procesen la información proveniente del sensor de temperatura y el sensor de caudal, la acondicionen para ser usada por el programa principal y la envíen mediante el protocolo de comunicaciones I2C. El Arduino Nano es una tarjeta basada en el ATmega328. Posee las mismas funcionalidades que un Arduino UNO, pero en un tamaño reducido. Para programarla es necesario un cable Mini USB.



Figura 4 - 3 Arduino Nano.

Fuente: (Ingeniería MCI Ltda. , 2017)

Características:

- Microcontrolador: ATmega328
- Voltaje de operación: 5V
- Voltaje de alimentación (Recomendado): 7-12V
- I/O Digitales: 14 (6 son PWM)
- Memoria Flash: 32KB
- EEPROM: 1KB
- Frecuencia de trabajo: 16MHz
- Dimensiones: 0.73" x 1.70"

Para recolectar la información inicial del sistema existente de la ducha sin ningún tipo de control automático, se diseñó e implementó un módulo de recolección de datos que se muestra en la Figura 5 - 2, conformado por un Arduino Nano, que procesa la información de temperatura y caudal entregada por los sensores, y la graba en una memoria SD para así poder llevar una estadística del líquido utilizado en cada afusión, la variación de la temperatura en cada ducha, y el tiempo utilizado para la misma. Se registró también los datos del sistema sometido a una entrada

escalón, con los cuales se obtuvo la función de transferencia con la que se trabajó las simulaciones iniciales.

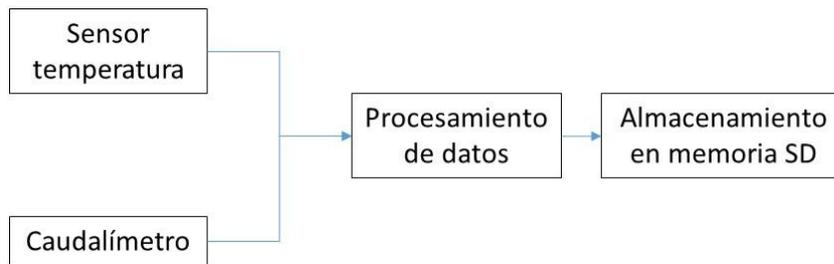


Figura 5- 3 Esquema del módulo de recolección de datos.

Fuente: Freire, Paul 2013

3.4.1.1 Sensor de temperatura

Dado que sus prestaciones satisfacen los requerimientos del sistema, se opta por el sensor DS18B20 de Maxim Integrated, mostrado en la Figura 6 – 3



Figura 6 - 3 Sensor DS18B20 con encapsulado a prueba de agua

Fuente: Freire, Paul, 2017

Sus principales características, según su hoja de datos son:

- Se comunica con el protocolo OneWire, que requiere solo un cable (DQ) para comunicarse con un microprocesador.
- Mide temperaturas en el rango de -55°C a $+123^{\circ}\text{C}$.
- $\pm 0.5^{\circ}\text{C}$ de precisión en el rango de -10°C a $+85^{\circ}\text{C}$.
- Resolución programable de 9 a 12 bits.
- No requiere componentes externos

- Disponible en encapsulado a prueba de agua
- En modo de alimentación parásita requiere solo dos pines de operación (DQ y GND).
- Bajo costo

3.4.1.2 Caudalímetro o Sensor de Flujo

En instalaciones domésticas diámetros habituales de tuberías 1" (DN25), 3/4" (DN20) y 1/2" (DN15), siendo esta última la normal en grifos. La presión debería estar en el rango de 100 kPA (1 Kg/cm²) a 500kPA (5 Kg/cm²). Caudales habituales para instalaciones para tuberías de 1/2" (las normales en grifos) 0.1 l/s (6 l/min) y 0.2 l/s (12 l/min). Para tuberías de 3/4" se tiene caudales en torno a 20 l/min, y para tuberías de 1" en torno a 35 l/min.

Los caudalímetros como el YF-S201, están constituidos por una carcasa plástica, un rotor con paletas y un sensor de efecto Hall en su interior. Al atravesar el fluido el caudal hace girar el rotor. El rotor que tiene un pequeño imán adherido para poder registrar una vuelta cada vez que pasa por el sensor de efecto Hall, generándose pulsos de salida a una velocidad proporcional a la del flujo. En la parte posterior del sensor hay una flecha para indicar la dirección del flujo. En la Tabla 1-3 se muestra las características de este sensor.

El criterio de selección que predomina es el diámetro de la tubería, por lo que se escoge el sensor YF-S201 mostrado en la Figura 7 - 3, dado que su conexión es de 1/4", compatible con la instalación domiciliaria. Además su salida usa lógica TTL, compatible con la placa de desarrollo Arduino y el rango de velocidad de flujo que maneja es de 1 a 30 Litros/Minuto, que excede el flujo de la instalación domiciliaria.



Figura 7 - 3 Caudalímetro YF-S201

Fuente: Freire Paúl, 2017

No es un sensor de precisión, por lo que la frecuencia del pulso varían dependiendo de la velocidad de flujo, la presión del fluido y la orientación del sensor, necesítándose calibrarlo si se requiere más que un 10% de precisión.

Tabla 1-3: Características del sensor YF-S201

Parámetro	Valor
Tipo de Sensor	Efecto Hall
Voltaje Nominal	5 a 18VDC (Voltaje Mínimo Requerido 4,5V)
Máxima Corriente de operación	15mA @ 5V
Voltaje de Salida	5V TTL
Velocidad de Flujo	1 a 30 Litros/Minuto
Rango de Temperatura	-25 a +80°C
Rango de Humedad	35%-80% RH
Precisión	±10%
Máxima Presión de Agua	2.0 MPa (4351 psi)
Salida del Ciclo de Trabajo	50% +-10%
Pulsos por Litro	450
Durabilidad	Mínimo 300,000 ciclos
Largo del Cable	15cm
Conectores	nominales tubería de ½"
Diámetro externo	0,78"
Rosca	½"
Tamaño	2.5" x 1.4" x 1.4"

Fuente: (Mecatronium, 2017)

3.4.2 Módulo de Control

Se requiere de una placa de desarrollo capaz de correr un programa que requiere una cantidad moderada de procesos, con al menos 22 Entradas/Salidas, que son las que tendrá el prototipo a desarrollarse. Debe soportar protocolos de comunicación serial e I2C para comunicarse con otras placas y dispositivos. Tendrá que interactuar con un PC, así como correr un programa sin necesidad de un ordenador. Cumple con estas características la placa Arduino Mega.

Arduino Mega es una tarjeta de desarrollo *open-source*. Se usa para implementar objetos interactivos autónomos o puede comunicarse a un PC a través del puerto serial USB. Puede recibir alimentación por medio de un cable USB o con una fuente de alimentación externa de 9 hasta 12VDC. (Ingeniería MCI Ltda. , 2017).

Características:

- Microcontrolador ATmega2560.
- Voltaje de entrada de – 7-12V.
- 54 pines digitales de Entrada/Salida (14 de ellos son salidas PWM).
- 16 entradas análogas.
- 256k de memoria flash.
- Velocidad del reloj de 16Mhz.

3.4.3 Módulo de Actuadores

Se compone de elementos mecánicos y electromecánicos sujetos al sistema de control.

3.4.3.1 Grifo Monomando Cerámico

Es un tipo de grifo con un mando único que efectúa tanto la regulación de caudal como la mezcla de agua fría y caliente. Actualmente coexisten dos sistemas principales de monomando o monocomando para lograr la regulación del caudal y de la mezcla: mediante discos de cerámica o mediante una esfera metálica. En un sistema basado en discos cerámicos, la parte clave del sistema son dos piezas contenidas en un cartucho como el de la Figura 8 - 3, que permiten cizallar por completo el flujo del agua.

Contienen dos discos cerámicos de gran dureza, que presentan superficies perfectamente planas que deslizan entre sí. El principio de funcionamiento es sencillo: cuando se hace bascular el mando del grifo desde la posición de cerrado, la placa cerámica móvil se desliza de forma que su orificio coincide gradualmente con alguno de los dos orificios de la placa fija. De esta forma, aumenta gradualmente el caudal a medida que se va levantando la maneta. A su vez, cuando se hace girar la maneta, el orificio de la pieza móvil se orienta progresivamente con la posición de los dos orificios de la pieza fija, agua fría o agua caliente, como se observa en la Figura 9 - 3, que indica dos posiciones de las superficies de paso de agua del grifo., permitiendo regular la proporción entre ambas. Una vez mezclada, el agua sale hacia la boca del grifo por el tercer orificio.



Figura 8 - 3 Cartucho de un monomando.

Fuente: Freire Paul, 2017

La principal ventaja de este tipo de grifos de discos cerámicos es que, al carecer de piezas metálicas en su interior, son prácticamente inmunes a la corrosión. Su mecanismo no requiere el uso de juntas de goma móviles, evitando el goteo que inevitablemente se produce a largo plazo en las griferías con otros sistemas, lo que garantiza la durabilidad del sistema.

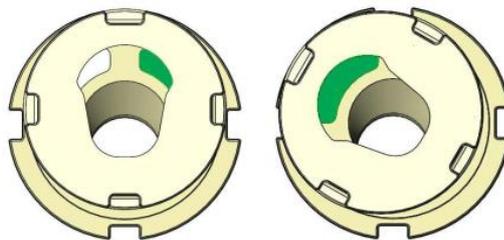


Figura 9 - 3 Posiciones de las superficies de paso de agua de un grifo monomando.

Fuente: (Pascual Fernandez, 2016)

Se escogió el FV Franz Viegner E108.05/B1, dado su bajo costo en relación a otros grifos con similares características, y además que es de una marca reconocida.

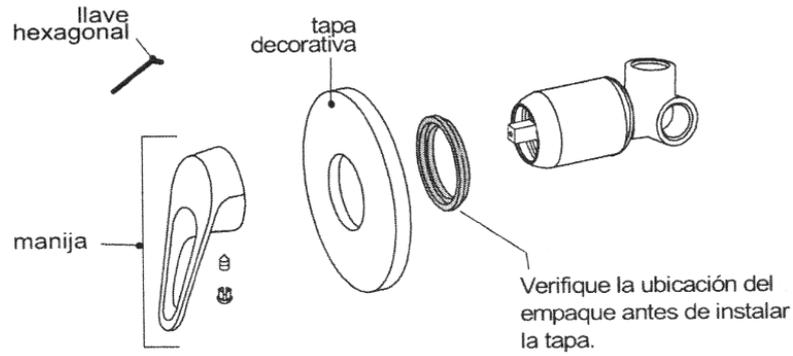


Figura 10 - 3 Partes del Grifo monomando Franz Viegener E108.05/B1.

Fuente: (Franz Viegener, 2016)

Se realizó una adaptación a la válvula para poder manipularla a través de un servo motor, para que se pueda controlar su apertura mediante el ángulo de giro enviado por el sistema. Se prescindió de algunos de los elementos tales como la manija, la tapa decorativa y el empaque, mostrados en la Figura 10 - 3 Partes del Grifo monomando Franz Viegener E108.05/B1. En lugar de la manija, se adaptó una pieza de metal que permite la regulación de temperatura.

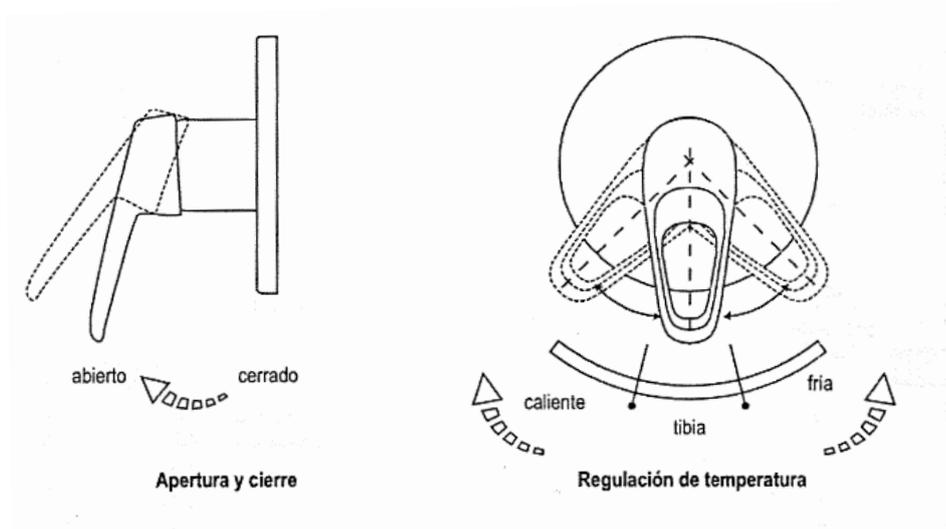


Figura 11 - 3 Funciones de un grifo monomando.

Fuente: (Franz Viegener, 2016)

En cuanto a la función apertura y cierre mostrada en la Figura 11 - 3, el grifo se deja en posición de abierto, para controlar el caudal con una válvula motórica reguladora. En la Figura 12 - 3 se muestra el servo motor acoplado al grifo monomando cerámico.



Figura 12 - 3 Servo motor acoplado al grifo monomando cerámico.

Fuente: Freire, Paul, 2017

3.4.3.2 Servo motor *Hi-Tec HS-755HB*

Es un motor de corriente continua que tiene la capacidad de ser controlado en posición. Es capaz de ubicarse en cualquier posición dentro de un rango de operación (generalmente de 180°) y mantenerse estable en dicha posición. Los servos se suelen utilizar en robótica, automática y modelismo debido a su gran precisión en el posicionamiento. (Candelas Herías & Corrales Ramón, 2007). Los servomotores o servos se utilizan en procesos industriales, en sistemas de radiocontrol, y en robótica debido a que su tiempo de reacción es corto.

Para variar el ángulo del grifo monomando que controlar la mezcla del agua caliente con la fría, se escogió el servo motor Hi-Tec HS-755HB mostrado en la Figura 13 - 3, que de acuerdo con su hoja de datos cuenta con las siguientes características:

- Torque de 11kg.cm a 5v.
- Rango de voltaje de operación de :4.8V a 6.0V
- Corriente con carga 1500 mA.
- Velocidad de operación de 0.28sec/60° sin carga
- Resistente al polvo y al agua
- Sus engranajes son de resina de alta resistencia.



Figura 13 - 3 Servo motor Hi-Tec HS-755HB

Fuente: Freire, Paúl, 2016

3.4.3.3 Válvula Motórica Reguladora

Para regular el caudal, se utiliza una válvula de bola motórica reguladora de origen chino CWX-25S CR-05 mostrada en la Figura 14 - 3. Cuenta con las siguientes características:

- Voltaje de trabajo de 12VDC
- Tamaño de la rosca de $\frac{1}{4}$ "
- Posibilidad de quedarse en una posición semi-abierta.
- Dispone de 5 cables para su control.
- La dirección de giro del motor para abrir o cerrar la válvula puede ser controlada electrónicamente por un Puente H.



Figura 14 - 3 Válvula CWX-25S CR-05

Fuente: Freire, Paul 2017

Como se muestra en el diagrama de conexión de la Figura 14 - 3, al polarizar los cables amarillo con positivo y el azul con negativo, la válvula gira para cerrarse, y el actuador automáticamente la apaga una vez que se cierra completamente. Al invertir la polaridad la válvula se abre, y un actuador la apaga una vez que alcanza una abertura total. Los cables verde y negro se conectan cuando la válvula está completamente abierta, los cables rojo y negro se conectan cuando la válvula está completamente cerrada.

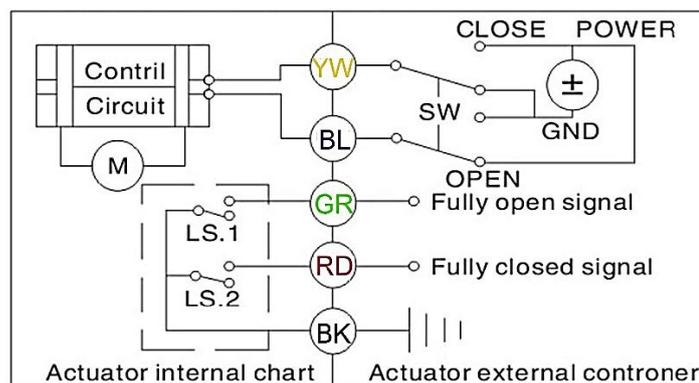


Figura 15 - 3 Diagrama de conexiones de la válvula CWX-25S CR-05.

Fuente: (aliexpress.com, 2017)

3.4.3.4 Electroválvula Solenoide

Se usarán en el proyecto dos electroválvulas solenoide, de s3milares características. Una se utilizará para permitir o no el paso de agua del suministro hasta la cisterna y la otra abre o cierra el caudal de salida de la ducha.

También conocida como válvula solenoide de uso general es una válvula que abre o cierra el paso de un fluido en un circuito. Mediante un campo magnético generado por una bobina en una base fija que atrae un émbolo, se controla la apertura o cerrado de la válvula. Las válvulas más conocidas son las de dos vías ya que tienen una entrada y una salida. Pueden ser normalmente cerradas (NC) o normalmente abiertas (NA). Las válvulas normalmente cerradas, no dejan pasar el fluido cuando están sin energía y energizadas se abren dejando pasar el fluido. Las válvulas normalmente abiertas dejan pasar el fluido cuando están en reposo y al momento de energizarlas se cierran impidiendo el paso. (Altecdust, 2017)

Se escogieron para el sistema dos electroválvulas solenoide genéricas como las de la Figura 16 - 3, fabricadas con metal y plástico. Cuentan con los siguientes parámetros:

- Voltaje de trabajo de 12 V DC.
- Corriente de 0.6 A.
- El diámetro de la manguera es de ½” para su entrada y salida
- Maneja una presión de máximo 116 PSI.
- La temperatura máxima 100°C.
- Normalmente cerrada, utiliza energía sólo cuando se lo requiera para reducir gastos, y se mantendrán cerradas en un apagado de emergencia aún sin energía de suministro local.



Figura 16 - 3 Electro válvula Solenoide 12V DIN 15

Fuente: Freire, Paúl, 2017

3.4.4 Módulo de Interfaz

Constituido por un LCD (*Liquid Cristal Display*) de 20 caracteres por 4 filas, mostrado en la **Figura 17 - 3** Figura 17 - 3, en el que se presentan los valores de la temperatura y caudal deseados (Set-Point), el estado del sistema, el volumen de agua utilizado y la temperatura a la que está saliendo el líquido en la ducha. También forman parte los pulsadores que permiten al usuario configurar el comportamiento del sistema.



Figura 17 - 3 LCD (Liquid Cristal Display) de 20 caracteres por 4 filas.

Fuente: Freire, Paul, 2017

Las principales características de un módulo LCD de este tipo, según (Brizzio , 2017) son:

- Pantalla de caracteres ASCII, además de caracteres japoneses, griegos y símbolos matemáticos.
- Se visualizan 20 caracteres por línea.
- Desplazamiento de caracteres bidireccional.
- Movimiento del cursor y cambio de su aspecto
- Se pueden programar 8 caracteres por el usuario
- Consumo de aproximadamente 7.5mW

3.4.5 Módulo de Alimentación

Se elige una fuente ATX (*Advanced Technology eXtended*) de computadora (Figura 18 - 3), de la cual se obtienen los diferentes voltajes requeridos por los dispositivos que conforman el sistema.

De acuerdo a los requerimientos de elementos de hardware seleccionados, es necesario contar con voltajes de 5V, con una corriente máxima de 3^a, y de 12 V con una corriente de 1A.



Figura 18 - 3 Fuente ATX utilizada como módulo de alimentación

Fuente: Freire, Paul 2017

Las características principales de la fuente elegida son:

- Encendido digital
- Poseen modo Stand By, es decir, consumen energía aunque el equipo esté apagado.
- Para 5V ofrecen una corriente máxima de 38 A.
- Para 12V ofrecen una corriente máxima de 20 A.

Se escogió para el control de la dirección de giro del motor que comanda la válvula motórica el Driver dual para motores basado en el chip L298N, ofrecido por ELECTRONILAB.CO., mostrado en la Figura 19 - 3, con las siguientes características:

- Permite controlar dos motores de corriente continua o un motor paso a paso bipolar de hasta 2 amperios.
- Permite controlar el sentido de giro y velocidad mediante señales TTL que se pueden obtener de micro controladores o tarjetas de desarrollo.

- Cuenta con diodos de protección y un regulador LM7805 que suministra 5V a la parte lógica del L298N.
- A través de jumpers habilita cada una de las salidas del módulo (A y B). La salida A está conformada por OUT1 y OUT2 y la salida B por OUT3 y OUT4. Los pines que las habilitan son ENA y ENB respectivamente. (Cruz, 2017)

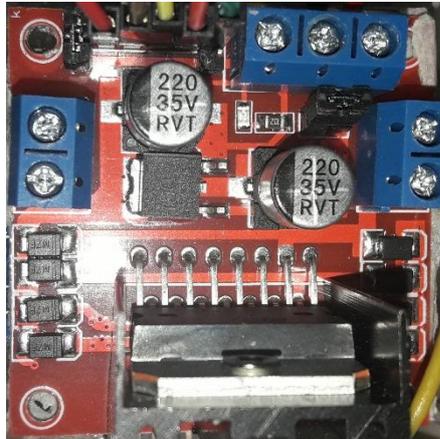


Figura 19 - 3 Driver L298N para motores DC y paso a paso.

Fuente: Freire Paúl, 2017

3.5 Diagramas de conexión del sistema

Luego de seleccionar los componentes electrónicos y sensores del sistema, se presenta el diagrama de conexión de ellos en la Figura 20 - 3.

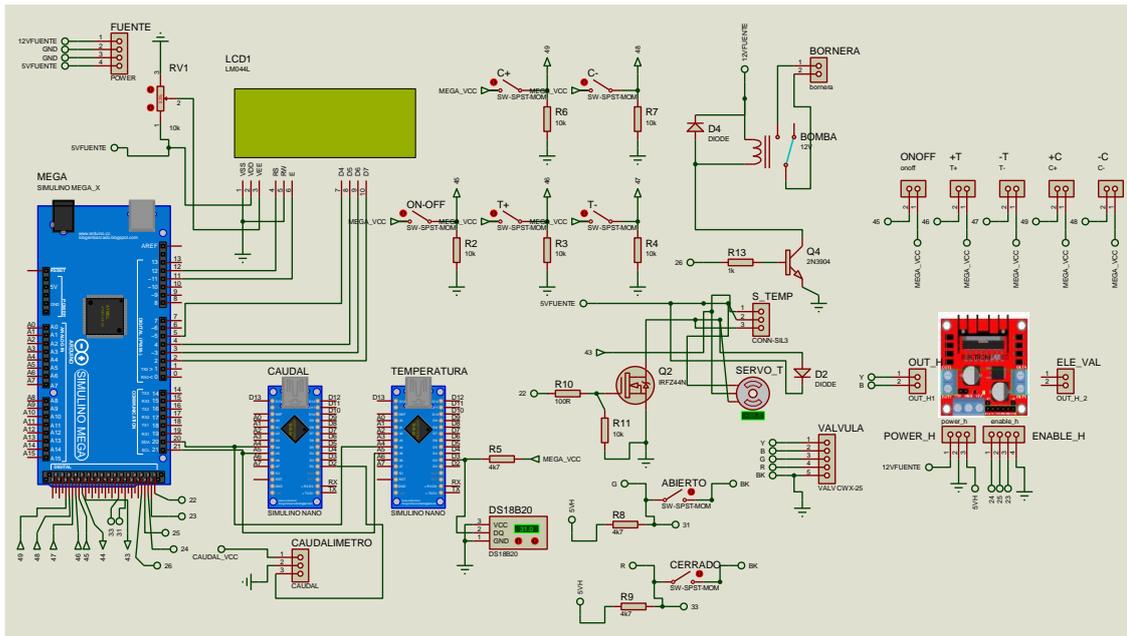


Figura 20 - 3 Diagrama de conexiones del sistema.

Fuente: Freire, Paúl, 2017

3.6 Selección del *software* de desarrollo

A continuación se analizan las características de las plataformas de desarrollo usadas para programar las tarjetas que controlarán el prototipo. En base a dichas características se decide que software es el adecuado para cumplir con los objetivos de cada módulo.

3.6.1 Software *Arduino*

Arduino consta con su propio entorno de programación, Arduino IDE, la cual usa un lenguaje basado en C, y pone a disposición ejemplos y librerías para la interacción con un gran número de dispositivos compatibles. Esta IDE es adecuada para programar las placas de desarrollo encargadas de interactuar con los sensores escogidos de temperatura y caudal, ya que existen librerías y ejemplos creados para este propósito. También tiene la capacidad de configurar y usar el protocolo I2C para la comunicación entre Arduinos.

La interfaz de programación de Arduino tiene una herramienta para comunicarse con la tarjeta a través del puerto serial en forma de texto, pero es muy limitada. Existen otras herramientas como PROCESSING, basada también en C, que maneja de una mejor manera estos datos, pero implica una programación en el dispositivo en el que se pretende visualizar la información, así como en la placa de desarrollo.

Si se pretende implementar algoritmos de lógica difusa en este IDE, cualquier variación en sus parámetros, requerirá cambios en el código del programa que pueden resultar engorrosos, dado que la lógica difusa deberá ser programada de acuerdo a sus algoritmos matemáticos que llegan a ser complejos de manejar con programación estructurada.

Existen numerosas herramientas que permiten el manejo de lógica difusa a través de una interfaz gráfica, pero no son compatibles con *Arduino*. La mejor alternativa que cumple con los requerimientos del *software* a desarrollar es Matlab a través de sus herramientas Simulink y Fuzzy Logic Designer. En la etapa de simulación, será útil también la aplicación PID tuner.

3.6.2 Matlab

Es una plataforma ampliamente utilizada para analizar y diseñar soluciones a problemas científicos y de ingeniería en prácticamente todos los ámbitos del conocimiento, desde salud hasta naves espaciales, pasando por telecomunicaciones y procesamiento de imágenes. El lenguaje Matlab está basado en matrices. Dispone de una gran cantidad de librerías de toolboxes preinstaladas le permiten empezar a trabajar inmediatamente con algoritmos esenciales para su dominio. Todas estas herramientas y prestaciones de esta plataforma están probadas y diseñadas rigurosamente para trabajar juntas. El código MATLAB se puede integrar con otros lenguajes, lo que le permite desplegar algoritmos y aplicaciones en sistemas web, empresariales o de producción.

El Fuzzy Logic Designer es una aplicación de Matlab que permite diseñar y probar sistemas de inferencia difusos para modelar sistemas de comportamiento complejo.

- Diseñar sistemas de inferencia Mamdani y Sugeno.
- Añadir o remover variables de entrada o salida.
- Especificar funciones de pertenencia de entrada y salida.
- Definir reglas if-then difusas.
- Seleccionar funciones de inferencia difusas para operaciones AND, OR, implicación, agregación, defuzificación.

- Ajustar valores de entrada y visualizar sus diagramas de inferencia asociados.
- Ver mapas de superficie para sistemas de inferencia difusos.
- Exportar sistemas de inferencia para el espacio de trabajo Matlab y ser utilizados por sus otras aplicaciones.

3.6.3 Simulink

El programa Simulink de Matlab presenta ventajas frente a otros programas matemáticos que podrían ser también utilizados para resolver las ecuaciones de los sistemas, tales como un entorno interactivo y un conjunto de librerías con bloques personalizables que permiten simular, implementar y probar una serie de sistemas variables con el tiempo. Además Simulink está integrado en Matlab y por ello es posible tener acceso a una amplia gama de herramientas que permiten desarrollar algoritmos, analizar y visualizar simulaciones. (Arce Rubio & Vianna Raffo, 2009).

Simulink permite la instalación de múltiples paquetes para la interacción con *hardware* externo. *Simulink Support Package for Arduino Hardware* es un paquete que adiciona la capacidad de crear y correr modelos en la placa de desarrollo Arduino. Añade bloques que permiten monitorear el estado de las entradas y salidas de la placa escogida, o lo que es lo mismo, de los sensores o actuadores conectados a éstas. Esto permite interactuar con los algoritmos implementados en Simulink y correrlos en el Arduino, bajo un entorno de programación gráfico para la placa y para el monitoreo de los resultados en un solo *software*.

Con los bloques de este paquete se puede manejar, entre otras cosas, servos, PWM, transmisión y recepción serial, entradas y salidas digitales del Arduino, programándolas y configurándolas directamente desde el entorno gráfico, por lo que es el *software* seleccionado para la aplicación.

3.7 Procesamiento de la información de los módulos

Es necesaria la comunicación de los sensores con la placa de desarrollo tanto para el sensor de temperatura como para el caudalímetro, tomando en cuenta que el software a utilizar para implementar el control difuso es Simulink. Usar programas que manejan los sensores corriendo en una placa de desarrollo independiente funciona de una manera óptima, por lo que una solución factible y de bajo costo es recoger y procesar los datos de cada uno de los sensores en placas independientes y comunicarlos mediante alguno de los protocolos soportados por Simulink con

el Arduino Mega, encargado de manejar las funciones de lógica difusa y las condiciones generales del sistema.

De principio, se podría pensar en utilizar comunicación serial, pero esta cuenta con algunos inconvenientes. Los más importantes:

- No existe ninguna garantía de que los dispositivos a comunicarse se encuentren a la misma velocidad, lo que puede hacer que se envíen datos de uno de los extremos y la otra parte no se entere.
- No se tiene ninguna manera de garantizar que la otra parte ha recibido el mensaje,

Los protocolos I2C y SPI, son las soluciones dadas por la industria para estos problemas. Analizadas las ventajas y desventajas de estos protocolos, se opta por I2C para la comunicación entre placas, debido a que:

- Utiliza menos líneas por dispositivo
- Arduino lo soporta de fábrica con una librería estándar, que utiliza dos de los pines analógicos para las funciones SDA (Datos) y SCL (Clock).
- Es sencillo configurar las direcciones I2C en las placas Arduino para realizar comunicación entre ellos.
- Limita la palabra de datos a transmitir a 8 bits, que es lo que se necesita para procesar y transmitir la información que proviene de los sensores.

3.7.1 Procesamiento de la información de temperatura

El sensor DS18B20 utiliza el protocolo de comunicación OneWire. La librería OneWire.h, una vez que toma a cargo de las operaciones de bajo nivel, permite que el usuario únicamente reciba y envíe bytes al dispositivo que usa este protocolo. Según la hoja de datos proporcionada por el fabricante, este sensor está calibrado en grados centígrados. La temperatura es colocada como un número de 16 bits con signo en complemento 2 en el registro de temperatura. El bit del signo (S) indica si la temperatura es positiva o negativa. (S=0 para positivo, S=1 para negativo), como se muestra en la Figura 21 - 3.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LS BYTE	2 ³	2 ²	2 ¹	2 ⁰	2 ⁻¹	2 ⁻²	2 ⁻³	2 ⁻⁴
	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
MS BYTE	S	S	S	S	S	2 ⁶	2 ⁵	2 ⁴

S = SIGN

Figura 21 - 3 Formato del registro de temperatura del sensor.

Fuente: (Maxim Integrated Products, Inc, 2015)

La Tabla 2-3 da ejemplos de salida digital y la correspondiente lectura de temperatura para la resolución de 12 bits, que es la que maneja por defecto el sensor.

Tabla 2-3: Relación datos/temperatura para el termómetro digital DS18B20

TEMPERATURA (°C)	SALIDA DIGITAL (BINARIA)	SALIDA DIGITAL (HEX)
125	0000 0111 1101 0000	07D0h
+85*	0000 0101 0101 0000	0550h
250,625	0000 0001 1001 0001	0191h
10,125	0000 0000 1010 0010	00A2h
0,5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0,5	1111 1111 1111 1000	FFF8h
-10,125	1111 1111 0101 1110	FF5Eh
-250,625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90

*El valor al encender o resetear el registro de temperatura es +84°C.

Fuente: (Maxim Integrated Products, Inc, 2015)

Con esta información se escribe un programa, utilizando el IDE de Arduino, que será cargado en uno de los Arduino Nano. Este programa obtiene la temperatura proveniente del termómetro digital, conectado al Pin 3, la almacena en una variable de dos Bytes y la envía vía I2C al Arduino Mega. Se enviará primero el Byte más significativo. En este mismo programa se configura el Arduino Nano para que sea reconocido como un dispositivo I2C con la dirección 6. Esta dirección es arbitraria, ya que puede ser cualquier valor desde 1 hasta 127, siempre y cuando no haya otro dispositivo en el mismo bus con la misma dirección, y sirve para que el Mega lo reconozca y solicite información del mismo.

En Simulink, se debe configurar un bloque I2C para recibir esta información. El bloque contiene los parámetros descritos en la Figura 22 - 3.

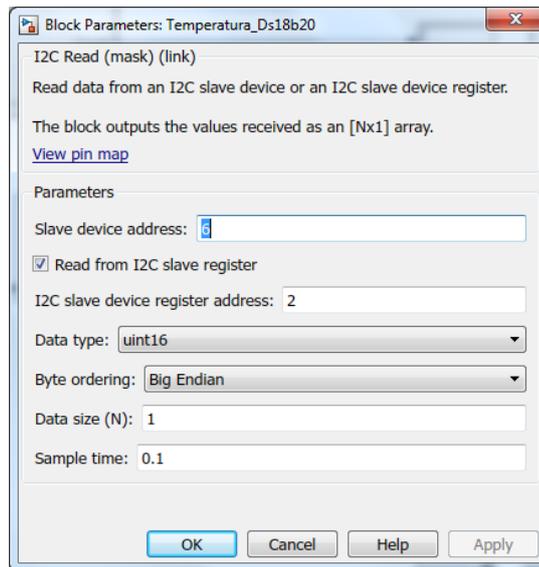


Figura 22 - 3 Parámetros del Bloque I2C que recibirá los valores de temperatura.

Fuente: Freire Paúl, 2017.

Se ha configurado la dirección del dispositivo en 6, que es de donde se requerirá información. Otro parámetro importante es el Orden de los Bytes, las dos opciones son:

Big Endian: El byte más significativo es enviado primero en el bus I2C. Esta es la opción por default.

Little Endian: El byte menos significativo es enviado primero en el bus I2C.

Se escogió Big Endian, ya que así se va a enviar la información desde los Arduino Nano. Una vez configurados estos valores, el bloque está listo para recibir la información de temperatura.

3.7.2 Procesamiento de la información de caudal

Para realizar la lectura del caudalímetro se usa una interrupción que cuenta los pulsos en un determinado intervalo, y dividiendo el número de pulsos entre el intervalo en segundos, se encuentra la frecuencia. La salida del sensor es una onda cuadrada cuya frecuencia es proporcional al caudal atravesado. Cada pulso en la salida del sensor equivale aproximadamente a 2.25 mililitros.

Para el paso de frecuencia a caudal se usa el factor K, que según la hoja de datos del fabricante es de 7.5 para este modelo. Para calcular el volumen de agua consumido, simplemente se hace una integración respecto del tiempo.

Como se ha descrito anteriormente, este dispositivo genera pulsos de salida a una velocidad proporcional a la del flujo. Para procesar de una manera óptima este tren de pulsos, es necesario utilizar “interrupciones”. Debe existir un determinado periodo de tiempo en el cual el micro controlador tendrá como única tarea leer una determinada entrada, con la finalidad de que no se pierda ninguno de dichos pulsos. De manera similar que con el sensor de temperatura, se incluyen un Arduino Nano que recoge las señales del caudalímetro, las procesa, y la envía el dato de frecuencia al Arduino Mega mediante protocolo i2c. En Simulink se implementa el bloque mostrado en la Figura 23 - 3, el cual recibe este valor de frecuencia y calcula el caudal y volumen.

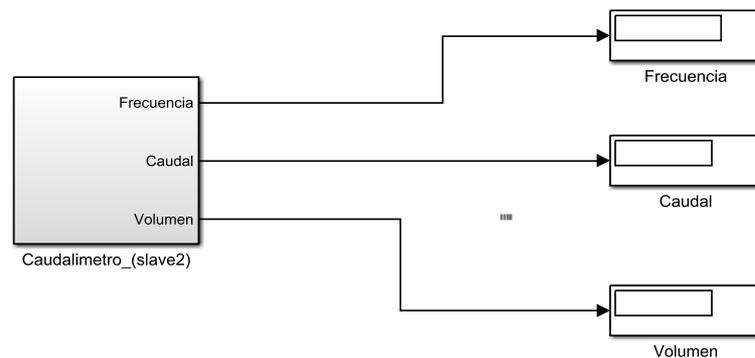


Figura 23 - 3 Bloque Caudalímetro en Simulink y las señales que emite.

Fuente: Freire Paúl, 2017

3.7.3 Comunicación entre placas Arduino

El esquema de conexión de dos Arduino Nano y Arduino Mega por medio de protocolo serial I2C se muestra Figura 24 - 3. Se cuenta con un Arduino Nano por sensor y un Arduino Mega que corre el *software* de control, por lo que actuará como Maestro (o Master). Los Arduino Nano se configuran como Esclavos (o Slave). Se debe enlazar las señales SDA (pin 20 en el Mega, A4 en los Nano) y la señal SCL (pin 21 Mega, A5 en los Nano). GND es común para todos los dispositivos.

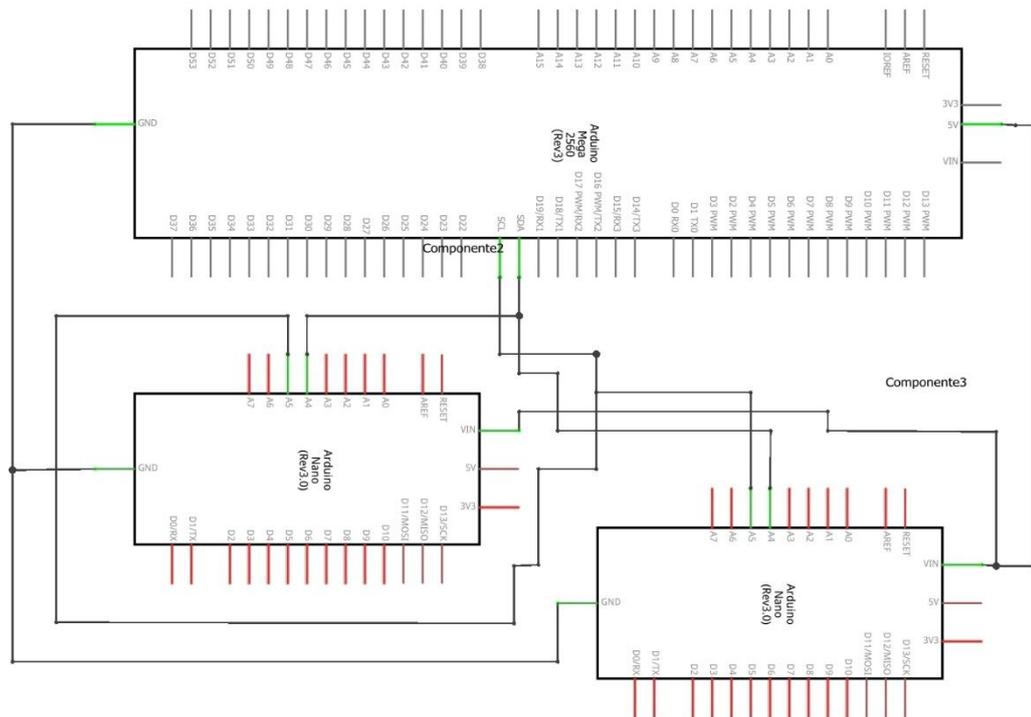


Figura 24 - 3 Conexión de dos Arduino Nano y Arduino Mega por medio de I2C

Fuente: Freire Paúl, 2017

En este conexionado es indistinto cual es el Maestro o el Esclavo. También hay que tomar en cuenta que el Arduino Nano también necesita una alimentación, que es brindada por el Arduino Mega, y este a su vez es energizado por medio del módulo de alimentación.

3.8 Implementación del *software* del sistema

Como se describió anteriormente, el módulo de control del sistema está diseñado en Simulink. Este *software* trabaja en base a bloques con distintas funcionalidades. Se han programado y configurado cada uno de los bloques para interactuar con el *hardware* del prototipo. Se describe a continuación los estados y los subsistemas implementados.

3.8.1 Estados del sistema:

Se distinguen durante el funcionamiento del sistema 4 estados diferentes en los que es necesario determinar el comportamiento de los actuadores en función de la información proveniente de los sensores. Se los ha nombrado Inicio, Arrancando_t, Regulando y Error, respectivamente.

Estado Inicio: El usuario no ha realizado ninguna petición al sistema por lo que éste se mantiene en espera. Se mantiene cerrado el caudal de agua hacia la ducha, y no se realiza tampoco ningún otro proceso.

Arrancando_t: El usuario ha inicializado el sistema, pero el agua fría existente en la tubería de agua caliente no se ha evacuado. Se compara el set point con la temperatura del agua a la salida del monomando y de acuerdo con el resultado el controlador difuso tomará la decisión de aumentar agua fría o caliente para la mezcla. Mientras el agua no esté en una temperatura con un error absoluto de 2°C con respecto al set point, no saldrá por la ducha y se la envía a recircular en dirección al reservorio principal.

Regulando: Se ingresa a este estado cuando el error absoluto de la temperatura con respecto al set point es menor a 2°C. En este estado el agua sale por la ducha.

Error: Se pasa a este estado en el caso de que el sistema no alcance el set point en un determinado tiempo. Esto puede ser debido a:

- No existe suministro de agua fría o caliente.
- Falla en alguno de los componentes del sistema.

De acuerdo a las mediciones realizadas, el sistema deberá tener un error absoluto menor o igual a cinco grados centígrados en menos de 60 segundos, de lo contrario pasará al estado de error. En cualquier estado del sistema, deberá ser posible detener el flujo de agua presionando el botón de encendido de la interfaz del usuario, que regresará al sistema al estado inicio.

Estos procesos son implementados en el *software* Simulink mediante una máquina de estados finitos que se muestra en la Figura 25 - 3.

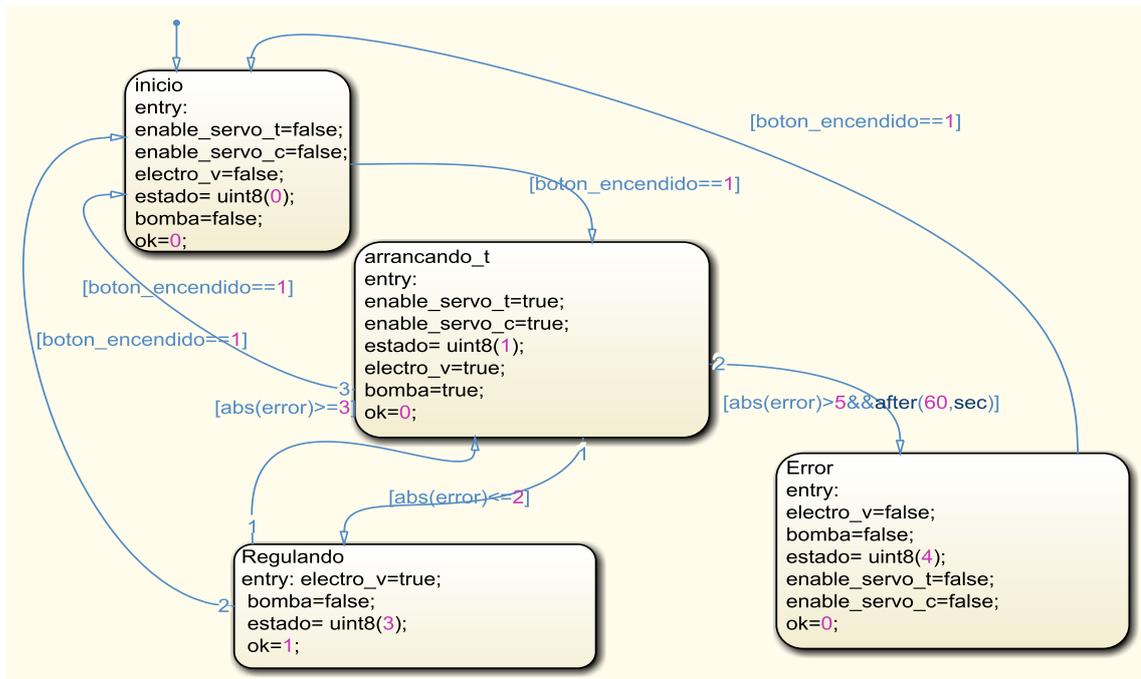


Figura 25 - 3 Máquina de estados finitos del sistema.

Fuente: Freire, Paúl, 2017

Esta máquina de estados finitos recibe y proporciona señales que permiten controlar el resto del sistema, descritas en la Tabla 3-3:

Tabla 3-3: Función de las entradas y salidas del bloque de estados

Nombre entrada	Función
boton_encendido:	Recibe una señal que indica si el usuario ha iniciado el proceso.
Error	Recibe la diferencia entre el set-point y la temperatura que en ese momento tiene el agua.
Nombre salida	Función
electro_v:	Enciende o apaga la electroválvula
Estado	Contiene el número de estado en que se encuentra el sistema
servo_temp	Ángulo en que se posiciona el servo encargado de regular la temperatura.
Enable_servo_t	Cuando es verdadera, energiza el servo motor que controla el giro de la válvula que controla la mezcla del agua caliente y fría.
Bomba	Habilita la re-circulación del agua hacia el reservorio.

Fuente: Freire, Paúl, 2017

3.8.2 Bloque de control difuso de temperatura

Durante la realización de un sistema de control, el primer paso consiste en identificar las variables de entrada y las variables a controlar. En el sistema de control difuso de temperatura planteado, cuyo esquema se presenta en la Figura 26 - 3, la variable de entrada es la temperatura deseada o Set Point, y las variables que servirán para la toma de decisiones del controlador serán el error de temperatura, que es la diferencia entre el Set Point y la temperatura actual del agua en el sistema. Se toma en cuenta la derivada del error, obtenida mediante el bloque **Discrete Derivative**, que es la rapidez con la que se está dando un cambio en el error. Los bloques **Saturation** limitan los valores que ingresarán al bloque de control difuso, para que no superen los valores contemplados en el controlador. El bloque **Discrete-Time Integrator** permite realizar una acumulación (integración) de los valores entregados por el controlador difuso, para calcular el valor del ángulo en que será posicionado el servo que controla la mezcla de agua fría con la caliente.

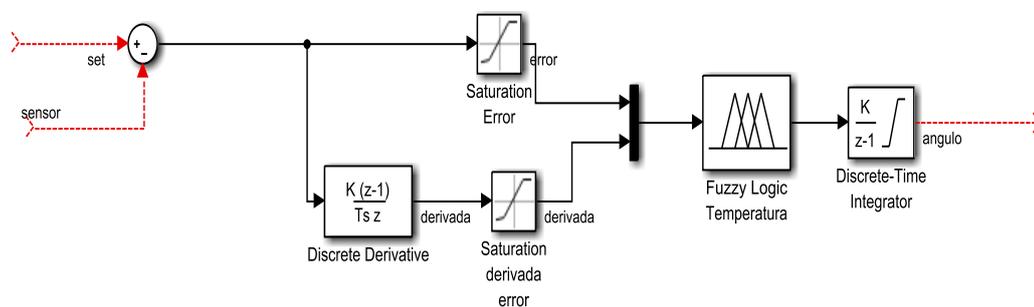


Figura 26 - 3 Esquema del sistema de control difuso de temperatura.

Fuente: Freire Paúl, 2017

En un sistema sujeto a control, por lo general se produce una fluctuación del valor de la variable a controlar en torno al Set Point, Figura 27 - 3 en donde se pueden observar las situaciones que se pueden presentar:

Error positivo: Cuando no se ha alcanzado el Set Point

Error 0: El Set Point es igual a la temperatura medida.

Error Negativo: Cuando se ha sobrepasado el Set Point

Derivada del Error Negativa: El valor del error está disminuyendo. Significa que la temperatura del agua está incrementándose, sobre o bajo el Set Point.

Derivada del Error 0: El error está cambiando de signo.

Derivada del Error Positiva: El valor del error está incrementándose. Significa que la temperatura del sistema se encuentra sobre o bajo el Set Point pero se encuentra descendiendo.

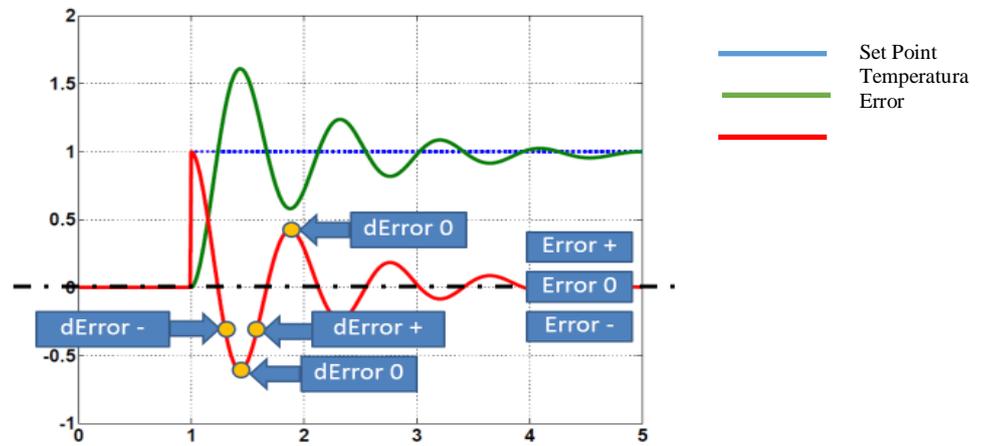


Figura 27 - 3 Gráfica de la temperatura y su derivada en función del tiempo.

Fuente: (Aimagin, 2017)

En base a estas consideraciones, se presentan nueve situaciones posibles del error y su derivada a ser analizadas, que se muestran en la Figura 28 - 3.

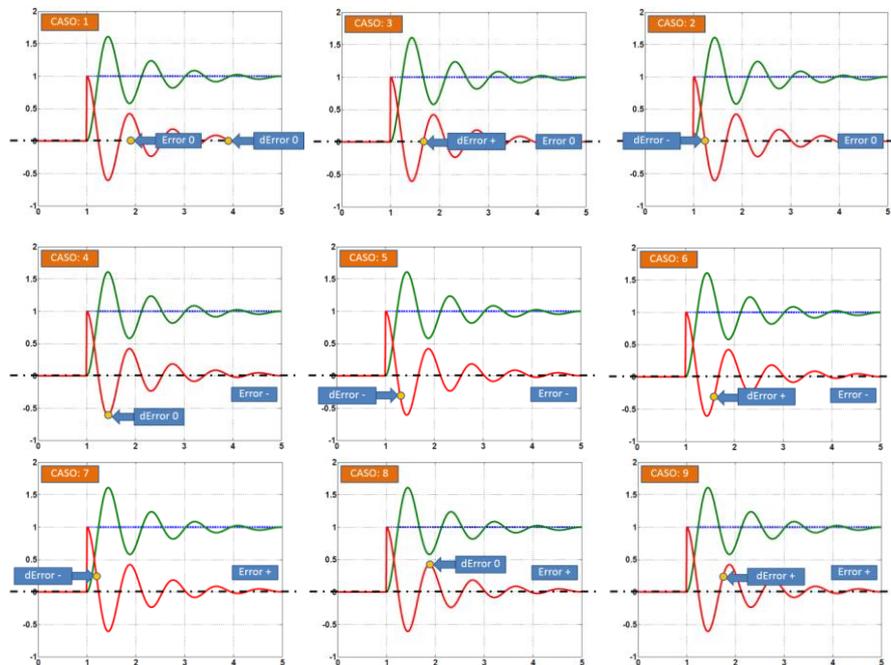


Figura 28 - 3 Situaciones posibles del error y su derivada.

Fuente: (Aimagin, 2017)

En la Tabla 4-3 se detalla la notación y valor lingüístico de las variables de entrada del control de temperatura. Para la implementación de este bloque, se utiliza la herramienta fuzzy logic designer de Matlab para crear un archivo .fis que sirve como parámetro al bloque fuzzy logic controller de Simulink. Se usa un sistema difuso tipo Mamdani, y el método del centroide para la defuzzificación. Las variables de entrada son “error” y “derror”. La variable de salida es “angulo” que representa el ángulo que deberá moverse el servo motor que controla la mezcla de agua caliente y fría.

Tabla 4-3: Notación de las variables de entrada del control de temperatura

Notación	Significado
N	Muy Negativo
NM	Negativo Medio
Z	Zero
PM	Positivo Medio
P	Muy Positivo

Fuente: Freire, Paúl, 2017.

En las Figura 29 - 3 se muestra los parámetros del controlador difuso, las funciones de pertenencia de la variable de entrada “error” y las funciones de pertenencia de la variable derivada del error “derror”, utilizadas para el controlador de temperatura empleado.

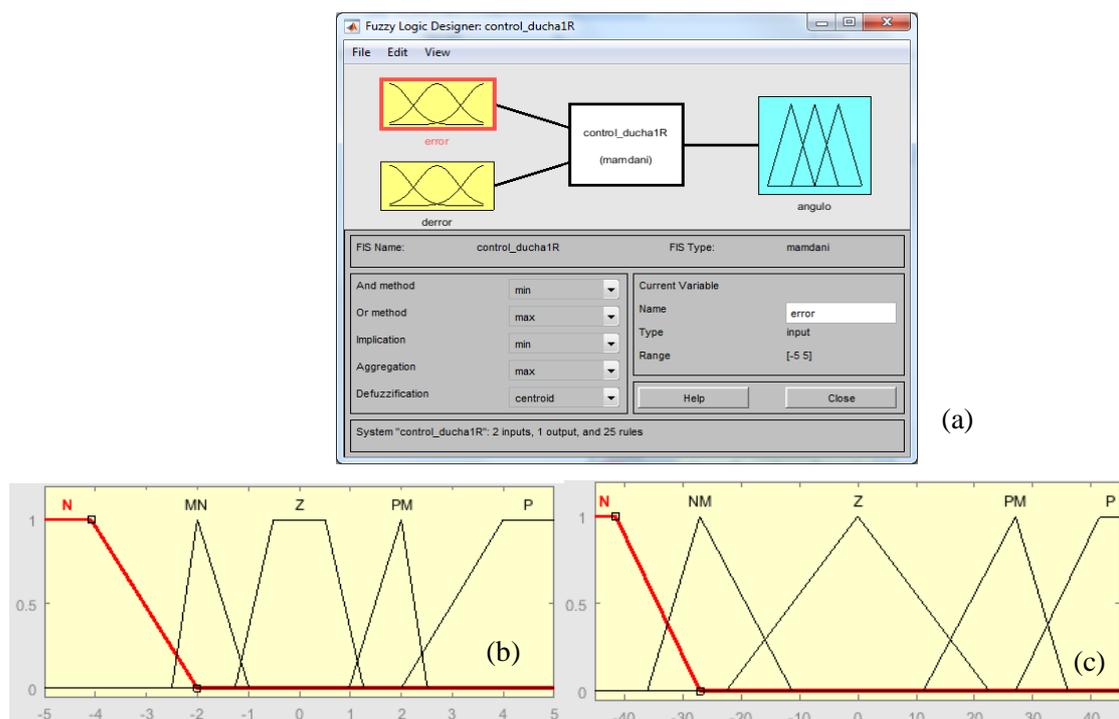


Figura 29 - 3 (a) Parámetros del controlador difuso. (b) Funciones de pertenencia variable de entrada “error”. (c) Funciones de pertenencia variable entrada “derror”

Fuente: Freire Paúl, 2017

La herramienta también permite visualizar las reglas aplicadas y el resultado de la defuzzificación, variando interactivamente las señales de entrada. Esto permitió evaluar los cambios necesarios en las reglas para mejorar el comportamiento del sistema.

Tabla 5-3: Notación de la variable de salida del control de temperatura

Notación	Significado
C	Cerrar
MC	Medio Cerrar
Z	Zero
MA	Medio Abrir
A	Abrir

Fuente: Freire, Paúl, 2017.

Para la variable de salida “ángulo” se definen cinco notaciones y valores lingüísticos, descritos en la Tabla 5-3 . Sus funciones de pertenencia se describen en la Figura 30 - 3.

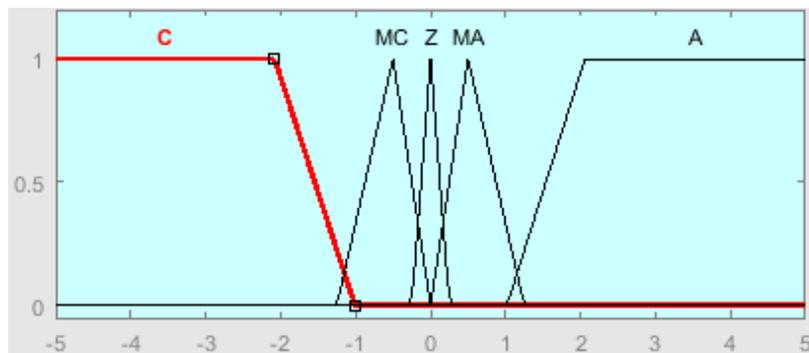


Figura 30 - 3 Funciones de pertenencia de la variable de salida “ángulo”

Fuente: Freire Paúl, 2017

Se tienen 25 reglas de control detalladas en la Tabla 6-3 que deberán ser introducidas en el editor de reglas del diseñador. De estas reglas y de las funciones de pertenencia depende el funcionamiento del sistema.

Tabla 6-3: Reglas de control del sistema propuesto

Número de Regla	Regla
1	If (error is N) and (derror a N) then (ángulo is C)
2	If (error is MN) and (derror is N) then (ángulo is C)
3	If (error is Z) and (derror is N) then (ángulo is MC)
4	If (error is PM) and (derror is N) then (ángulo is Z)
5	If (error is P) and (derror is N) then (ángulo is MA)
6	If (error is N) and (derror is NM) then (ángulo is C)
7	If (error is MN) and (derror is NM) then (ángulo is MC)
8	If (error is Z) and (derror is NM) then (ángulo is MC)
9	If (error is PM) and (derror is NM) then (ángulo is Z)
10	If (error s P) and (derror s NM) then (ángulo is MA)
11	If (error a N) and (derror is Z) then (ángulo is MC)
12	If (error s MN) and (derror is Z) then (ángulo is MC)
13	If (error is Z) and (derror is Z) then (ángulo is Z)
14	If (error is PM) and (derror is Z) then (ángulo is MA)
15	If (error s P) and (derror is Z) then (ángulo is MA)
16	If (error s N) and (derror is PM) then (ángulo is MC)
17	If (error is MN) and (derror is PM) then (ángulo is Z)
18	If (error is Z) and (derror is PM) then (ángulo is MA)
19	If (error is PM) and (derror is PM) then (ángulo is MA)
20	If (error a P) and (derror s PM) then (ángulo is A)
21	If (error is N) and (derror is P) then (ángulo is Z)
22	If (error is MN) and (derror is P) then (ángulo is Z)
23	If (error is Z) and (derror is P) then (ángulo is MA)
24	If (error s PM) and (derror is P) then (ángulo is MA)
25	If (error s P) and (derror is P) then (ángulo is A)

Fuente: Freire, Paúl, 2017

3.8.3 Bloque de control difuso de caudal

Se configura la interfaz del sistema para que permita ingresar 3 valores posibles de caudal de trabajo del sistema, que se detallan en la Tabla 7-3.

Para controlar el caudal se utiliza la válvula motórica CWX-25S. No posee un control de posición, pues dependiendo de los valores de sus líneas de control gira constantemente abriendo y cerrando el paso de agua. Se necesita entonces saber los valores entregados por las señales de fully open (cable verde) y fully closed (cable rojo) para determinar cuando la válvula se encuentra completamente abierta o cerrada. Se debe también conocer el valor del error entre el set point y el caudal registrado por el sensor de flujo. El controlador entregar el sentido de giro de la válvula o si debe detenerse en una determinada posición.

Tabla 7-3: Caudales de trabajo del sistema

Caudal (Interfaz)	Caudal (l/min)
MIN	2.5
MED	5.0
MAX	7

Fuente: Freire, Paúl, 2017

El controlador difuso que controla el sentido de giro del motor de la válvula de caudal se muestra en la Figura 31 - 3. Se trata simplemente de un control on/off, solo controla encendido y apagado de la válvula.

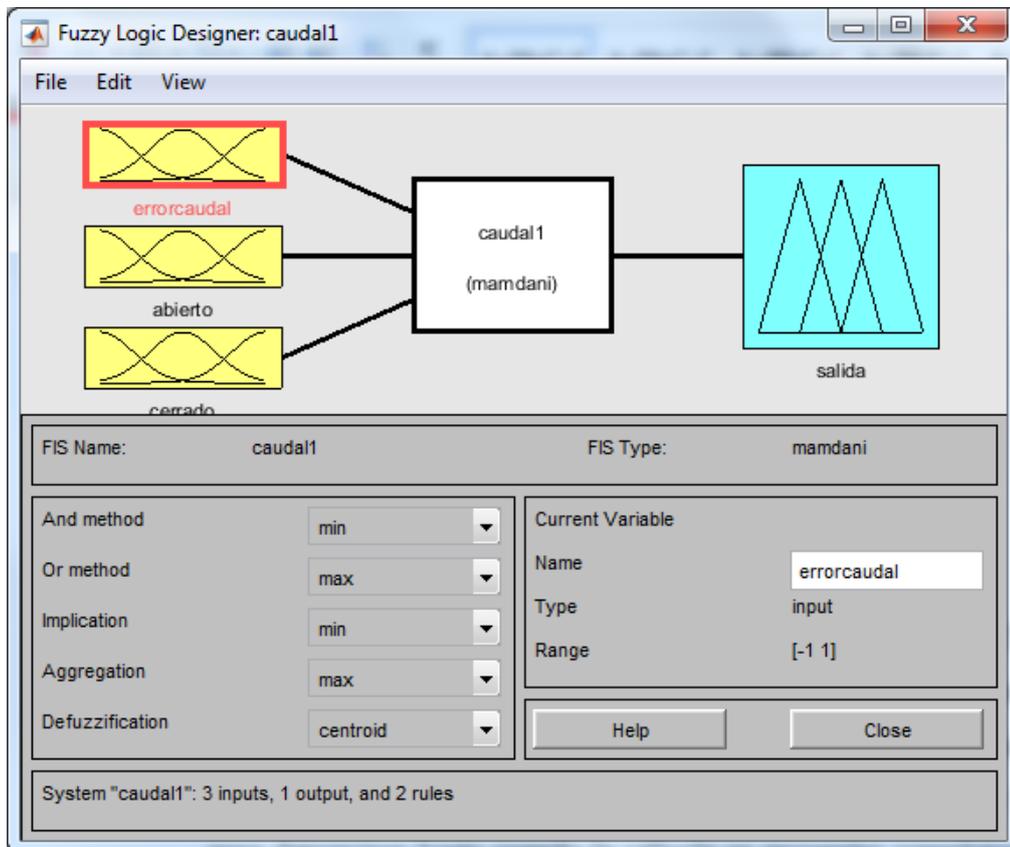


Figura 31 - 3 Variables y métodos incluidos en el controlador difuso de caudal.

Fuente: Freire, Paúl, 2017

Se definen funciones de pertenencia de tipo trapezoidal (trampf) y se les asigna tres valores lingüísticos a la entada “error caudal” (negativo, zero, positivo), como se muestra en la Figura 32 - 3.

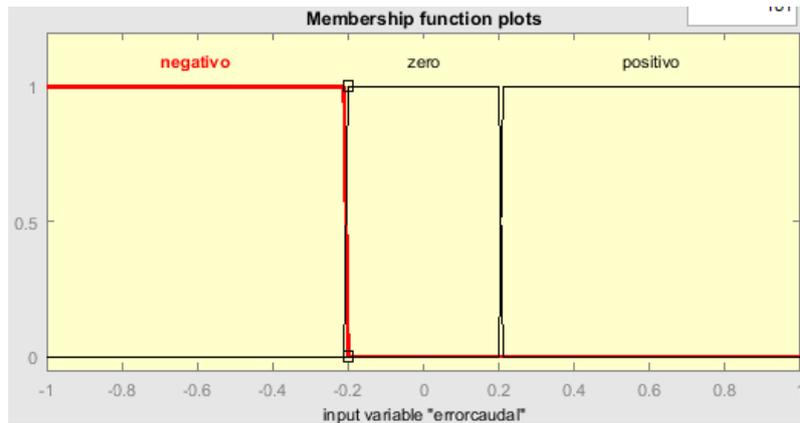


Figura 32-3 Función de pertenencia de la señal de entrada “errorcaudal”

Fuente: Freire, Paúl, 2017

Las variables “abierto” y “cerrado” entregan únicamente valores de verdadero y falso. Como se había descrito anteriormente, la lógica binaria es un caso particular de la lógica difusa. Se les asigna únicamente un valor lingüístico. (verdadero), como se observa en la Figura 33 - 3

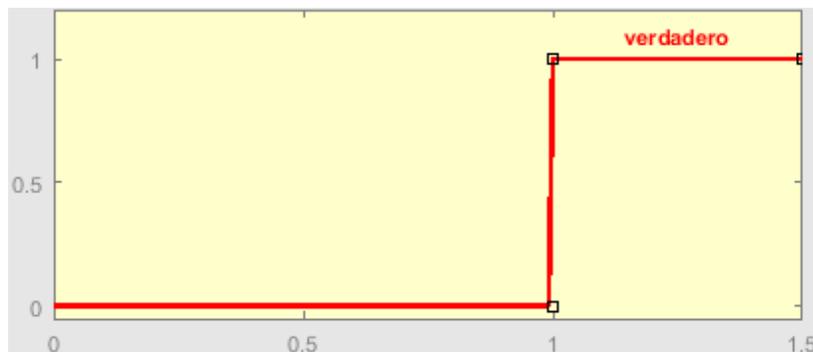


Figura 33 - 3 Función de pertenencia de las variables de entrada “abierto” y “cerrado”

Fuente: Freire, Paúl, 2017

Se configura la variable “salida” de tal forma que entregue el valor 1 si se debe abrir la válvula, 0 si debe detenerse y -1 si debe cerrarse. Los valores lingüísticos son “abrir” y “cerrar”.

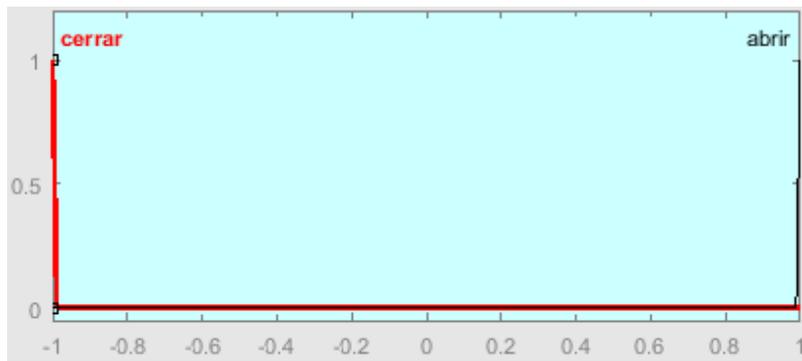


Figura 34 - 3 Función de pertenencia de la señal de salida

Fuente: Freire Paúl, 2017

En base a lo anterior, se realiza las reglas que permite controlar el caudal.

1. If (errorcaudal is negativo) and (cerrado is verdadero) then (salida is cerrar)
2. If (errorcaudal is positivo) and (abierto is verdadero) then (salida is abrir)

Una de las características de un control On/Off es el de presentar oscilaciones debido a los cruces por cero de la variable controlada. Para solucionar este problema se implementó un bloque de control difuso llamado “pwmfuzzy” que controla la rapidez de apertura y cerrado de la válvula, de tal manera que a medida que el error va disminuyendo, se reduce la velocidad del motor de la válvula usando la técnica de modulación por ancho de pulsos o PWM *pulse-width modulation*. Se define una entrada “error”, y una variable de salida output1, que entrega el valor de PWM calculado, tomando en cuenta cuatro funciones de pertenencia denominadas “muybajo”, “bajo”, “medio” y “alto”, como se muestra en la Figura 33 - 3.

Las cuatro reglas que rigen al controlador son las siguientes:

1. If (error is muybajo) then (output1 is muybajo) (1)
2. If (error is bajo) then (output1 is bajo) (1)
3. If (error is medio) then (output1 is medio) (1)
4. If (error is alto) then (output1 is alto) (1)

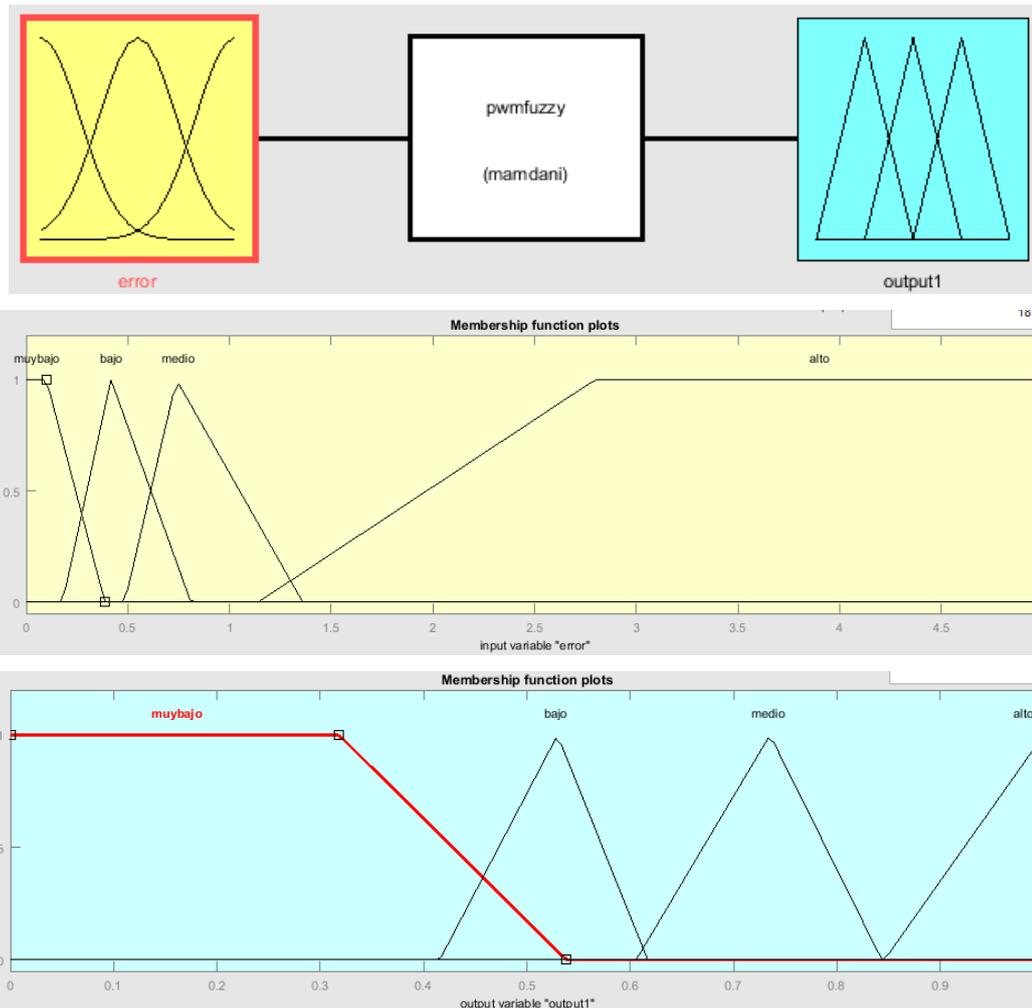


Figura 35 - 3 Entrada, salida y sus correspondientes cuatro funciones de pertenencia del bloque de control difuso “pwmfuzzy”

Fuente: Freire Paúl, 2017

El diagrama de bloques de la estructura de control difuso de velocidad del motor de la válvula de caudal se muestra en la Figura 36 - 3

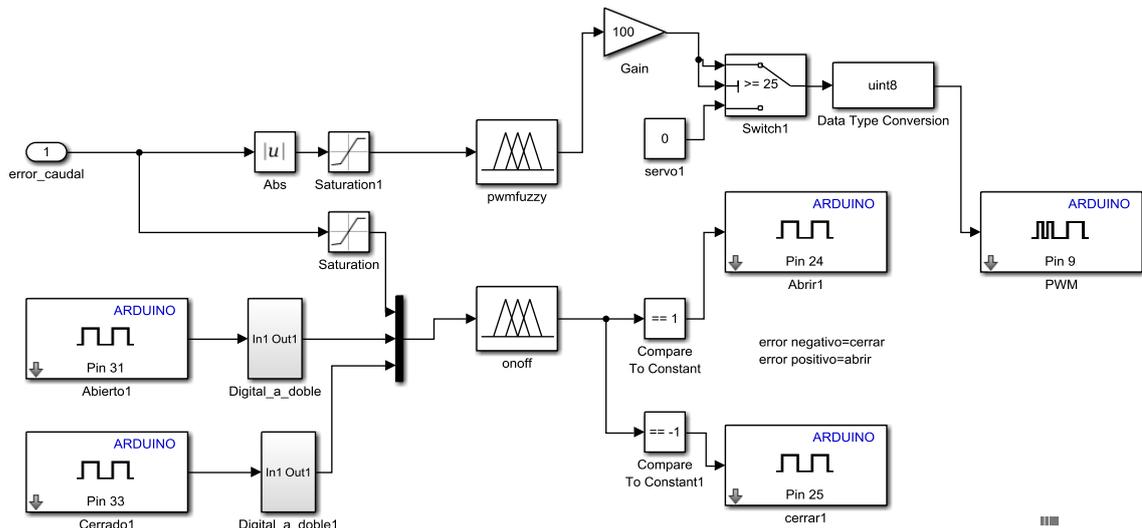


Figura 36 - 3 Estructura de control difuso de velocidad del motor de la válvula de caudal.

Fuente: Freire, Paúl, 2017

3.8.4 Bloque Interface

Este bloque permite presentar y recibir información del usuario. En la Tabla 8-3 se presentan las entradas y salidas del bloque interface. En su interior se compone de 3 bloques, que se muestran en la Figura 37 – 3.

Tabla 8-3: Entradas y salidas del bloque interface

Nombre entrada	Función
Estado	Estado en el que se encuentra el sistema
Temperatura	Señal proveniente del sensor de temperatura
Volumen	Señal proveniente del sensor de caudal
Nombre salida	Función
Set-point temp	Valor de temperatura requerida, ingresado por el usuario
Set-point caudal	Valor de caudal requerido, ingresado por el usuario

Fuente: Freire, Paúl, 2017

contador_temperatura, que recibe las señales de los botones para fijar el set point de la temperatura. Dicho bloque detecta el cambio de valor en los botones del módulo de interface, específicamente a través de verificar un flanco de subida en la señal. Estas señales son procesadas por un bloque llamado **contar**, programado en un bloque S-function Builder, para entregar a su salida el valor de temperatura pedido por el usuario. Este bloque además cuenta con los parámetros `min_` y `max_` que son los valores tope que se puede seleccionar.

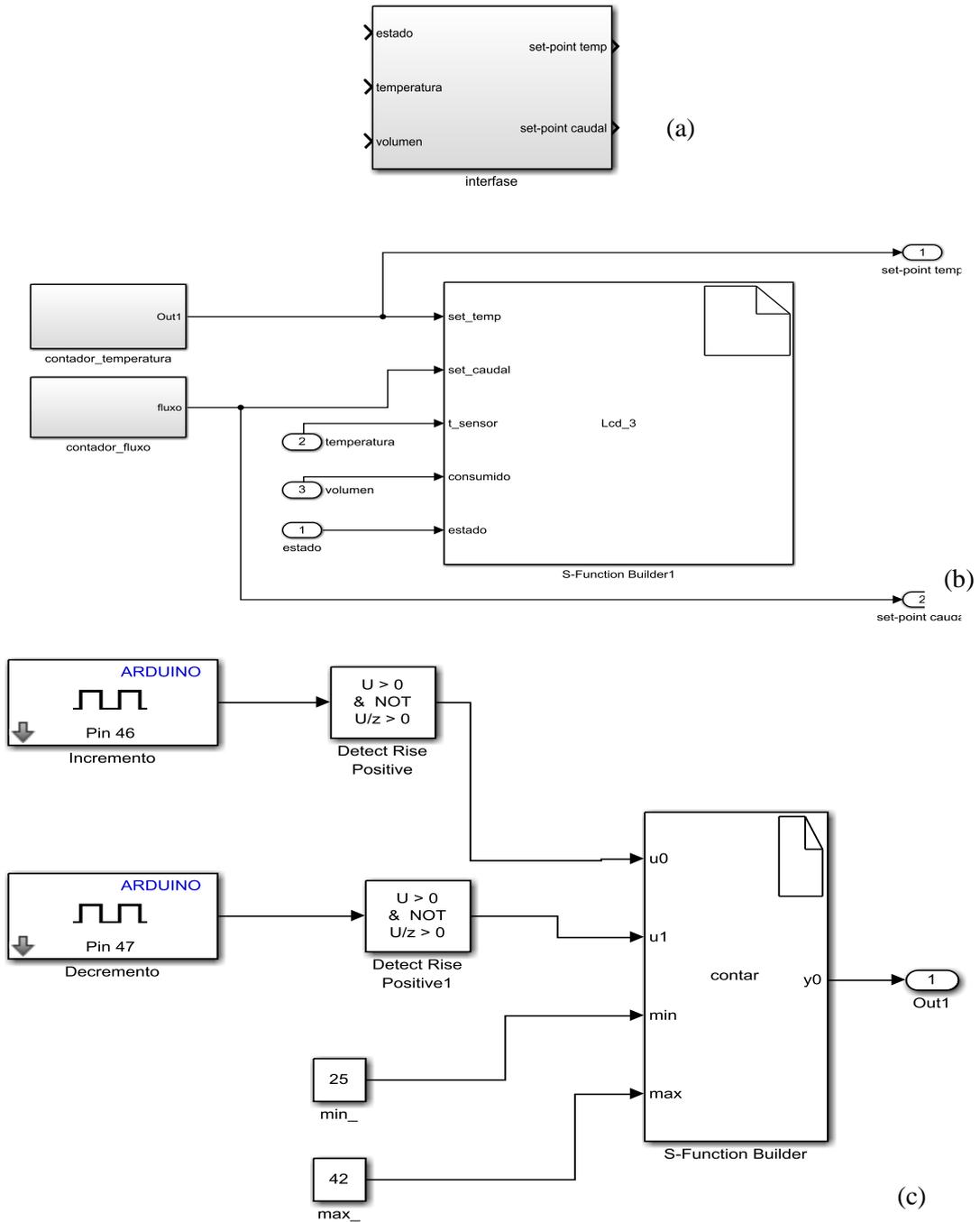


Figura 37 - 3 (a) Bloque interface (b) subsistema contenido en el bloque interface .
(c) Bloque contador de temperatura

Fuente: Freire Paúl, 2017

contador_fluxo, y la máquina de estados que procesa su información que se muestra en la Figura 38 - 3, se encarga de entregar al sistema el valor de caudal a circular por la ducha. Al igual que el bloque anterior, registra un flanco de subida en la señal proveniente de los botones de la interface física para incrementar o reducir el valor del caudal. Estas señales ingresan a una máquina de estados finitos que procesa la información.

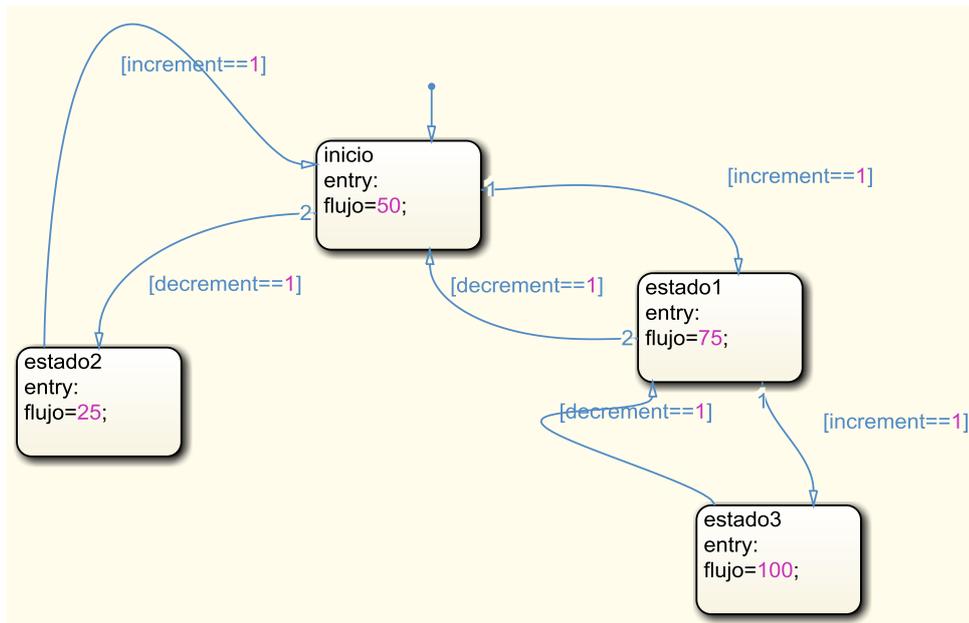
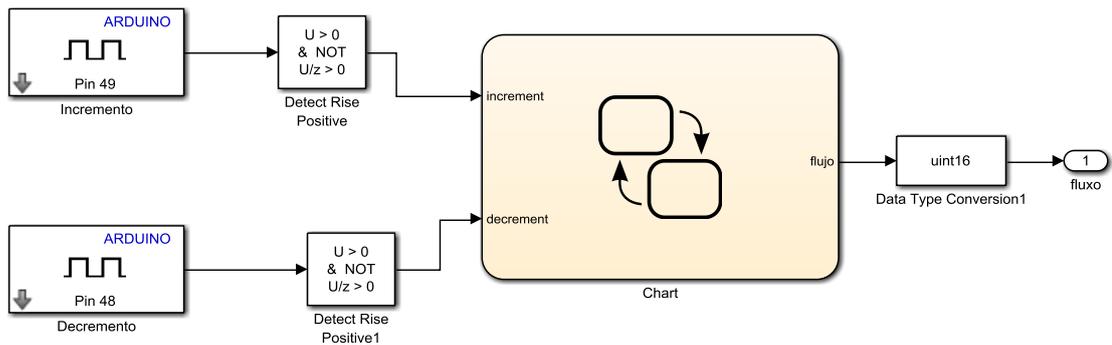


Figura 38 - 3 Bloque contador_flujo y la máquina de estados que procesa su información.

Fuente: Freire, Paúl, 2017

CAPÍTULO IV

4. VALIDACIÓN DEL PROTOTIPO DEL SISTEMA

En este capítulo se presentan los resultados alcanzados en las pruebas realizadas al prototipo del sistema. La evaluación de cada módulo del sistema se realizó integrando el *software* y *hardware* diseñado. Se caracterizó la respuesta de cada módulo en función de los requerimientos del proyecto, para ello se realizaron mediciones que corroboraron los valores de las variables y sus errores. Se verificó la correcta comunicación entre los componentes del sistema.

4.1 Caracterización de las mediciones realizadas por el módulo de sensores

Para comprobar que el módulo de sensores cumple con los resultados esperados se realizaron mediciones y se analizaron los resultados. Los sensores usados cuentan con sus respectivas hojas de datos, en las cuales se dan los parámetros necesarios para su utilización, pero es necesario verificarlos a través de una comparación con medidas estándar o patrones de referencia y, de existir algún error, ajustarlos por medio de tablas o curvas de corrección.

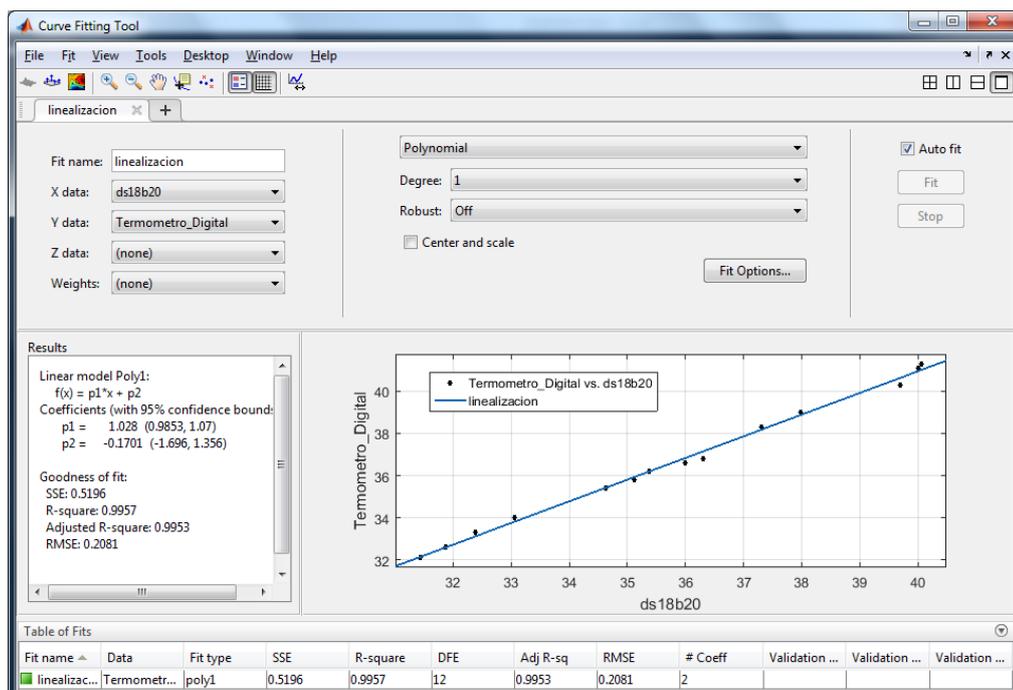


Figura 1 - 4 Resultados del ajuste lineal del sensor de temperatura.

Fuente: Freire, Paúl, 2017

4.2.1 Calibración sensor de temperatura ds18b20

Como equipo patrón se considera un termómetro digital diseñado para medir la temperatura corporal de seres humanos. La gama de temperatura en la cual trabaja el sensor es de 32.0°C a 42.9°C y según su hoja de datos, su error es de +/- 0.1°C.

Se procedió a sumergir el sensor de temperatura y el termómetro digital al mismo tiempo en un recipiente con agua, realizando mediciones en intervalos de un minuto, tiempo en el cual el agua se enfría ligeramente, hasta que la temperatura del líquido sale del rango de operación del termómetro. Es importante que los dos dispositivos se sumerjan a una misma distancia y que no toquen las paredes del recipiente para evitar errores de medición.

Como se observa en la Tabla 1-4 , los valores del instrumento patrón no coinciden con los valores registrados por el sensor de temperatura, y se hace necesario un ajuste lineal, que es un modelo matemático usado para aproximar la relación de dependencia entre una variable dependiente Y y las variables independientes Xi. Para hallar la ecuación lineal de la correlación entre la medición de los dos instrumentos de medida utilizados, se usó la herramienta Curve Fitting Tool de Matlab.

Tabla 1-4: Comparativa del instrumento patrón y el sensor Ds18b20

Termometro Digital	Sensor Ds18b20	Error
41.3	40.06	1.24
41.1	40	1.1
40.3	39.69	0.61
39	37.98	1.02
38.3	37.31	0.99
36.8	36.31	0.49
36.2	35.38	0.82
35.8	35.13	0.67
35.4	34.63	0.77
34	33.06	0.94
33.3	32.38	0.92
32.6	31.87	0.73
32.1	31.44	0.66
36.6	36	0.6

Fuente: Freire, Paúl, 2017

Según los resultados mostrados en la Figura 1 - 4, se requiere un polinomio de ajuste de la forma :

$$y=mx+b \quad (10)$$

Donde m y b son constantes reales, $m, b \in R$ y x es una variable real. La constante m es la pendiente de la recta y b es el punto de corte de la recta con el eje y . Se reemplazan en (10) los valores obtenidos para m, b e y , siendo entonces el polinomio de ajuste:

$$y = 1.028x - 0.1701 \quad (11)$$

Se obtiene que el coeficiente de determinación (R^2) es de 0.9957 que indica que la correlación entre las variables analizadas es alta y que la ecuación es válida.

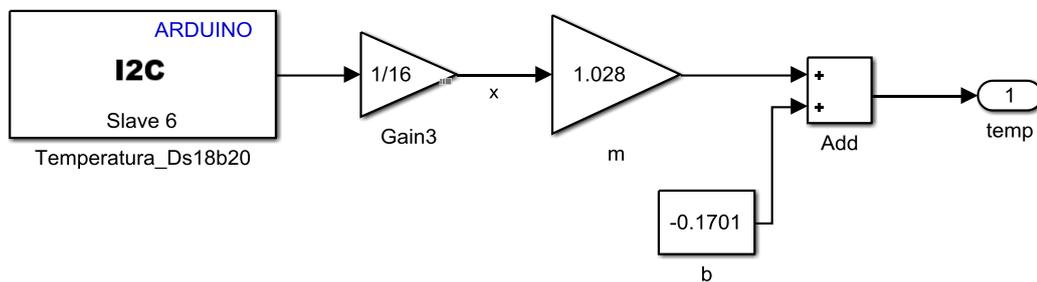


Figura 2 - 4 Bloque del sensor de temperatura con el polinomio de ajuste

Fuente: Freire, Paúl, 2017

4.2.1 Calibración sensor de caudal YF-S201

El sensor de caudal YF-S201 no es un sensor de precisión, por lo que se recomienda realizar una adecuada calibración del mismo. El fabricante provee el valor de una constante K , que permite pasar de Frecuencia de pulsos a caudal, la cual se obtiene de:

$$K = \frac{\text{número de pulsos}}{\text{Volumen} * 60} \quad (12)$$

Se puede medir exactamente la cantidad de pulsos y con un recipiente graduado se mide el volumen de agua. Se realizarón 10 conteos de pulsos para un volumen de 1 litro, 2 litros y 4 litros, calculando el valor de la constante K , y los resultados del conteo de pulsos para un volumen dado y el valor resultante de la constante K se muestran en la tabla 2-4 El valor promedio de K es:

$$K \cong 6.433 \quad (13)$$

El valor dado para la constante por el fabricante es de 7.5, difiere del obtenido experimentalmente, por lo que se utiliza el valor calculado de acuerdo a las mediciones.

Tabla 2-4: Conteo de pulsos por volumen y valor resultante de la constante K

Volumen (litros)	Pulsos	K
1	396	6.6
1	388	6.466666667
1	394	6.566666667
1	387	6.45
1	390	6.5
1	388	6.466666667
1	389	6.483333333
1	385	6.416666667
1	391	6.516666667
1	390	6.5
2	780	6.5
2	771	6.425
2	764	6.366666667
2	762	6.35
2	759	6.325
2	770	6.416666667
2	765	6.375
2	771	6.425
2	771	6.425
2	767	6.391666667
4	1510	6.291666667
4	1532	6.383333333
4	1538	6.408333333
4	1537	6.404166667
4	1542	6.425
4	1546	6.441666667
4	1540	6.416666667
4	1540	6.416666667
4	1538	6.408333333
4	1542	6.425

Fuente: Freire, Paúl, 2017

4.2 Simulación del sistema

Para obtener la función de transferencia del sistema, se aplicó una entrada de tipo escalón de la válvula mezcladora, abriendo completamente de forma rápida el agua caliente. Los datos

obtenidos fueron procesados con la herramienta PID tuner de Matlab para obtener un modelo de la planta, que servirá para realizar simulaciones y encontrar una función Fuzzy óptima, sin necesidad de poner a funcionar el dispositivo físico. Introducidos los datos, se obtiene la curva de reacción de la planta identificada que consta en la Figura 3 - 4.

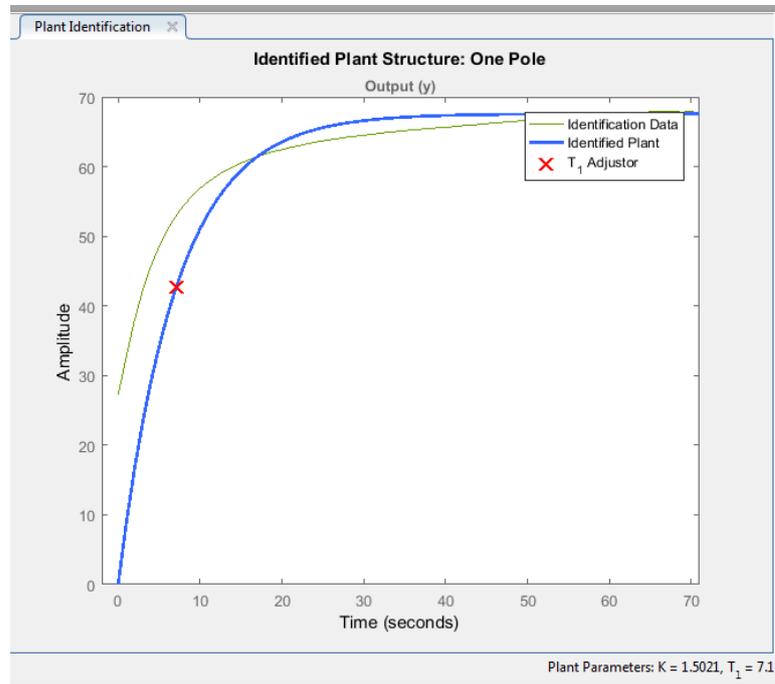


Figura 3 - 4 Curva de reacción de la planta.

Fuente: Freire, Paúl, 2017

La planta identificada no coincide plenamente con los datos recogidos por el módulo de supervisión, por lo que se intentará hacer una mejor aproximación con la herramienta de auto estimación de la que dispone esta aplicación, que aplica el método de los mínimos cuadrados. La función de transferencia obtenida de la planta es:

$$G(s) = \frac{1.4773}{7.8313s + 1} \quad (14)$$

Se construye el modelo de la planta en Simulink para obtener las simulaciones.

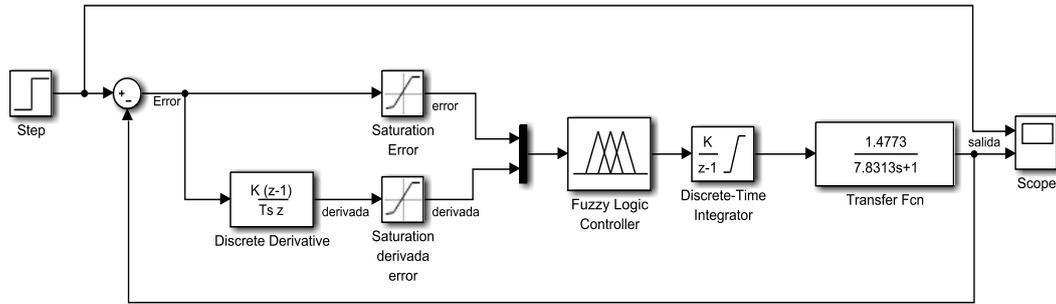


Figura 4 - 4 Modelo de la planta en Simulink

Fuente: Freire Paúl, 2017

Se prueba la respuesta a una entrada escalón usando varios controladores difusos creados en la herramienta Fuzzy Logic Designer. En las primeras pruebas se obtienen respuestas con fuertes oscilaciones. Estas oscilaciones se corrigen asignando valores más pequeños a la variable “ángulo” de salida del controlador, lo que causa que la válvula gire menos ante un cambio de temperatura, especialmente con valores pequeños de error. En la Figura 5 – 4 se observa los primeros resultados de las simulaciones y también el valor arrojado por el integrador y que será mapeado para ubicar el servo motor en la posición deseada para mantener la temperatura.

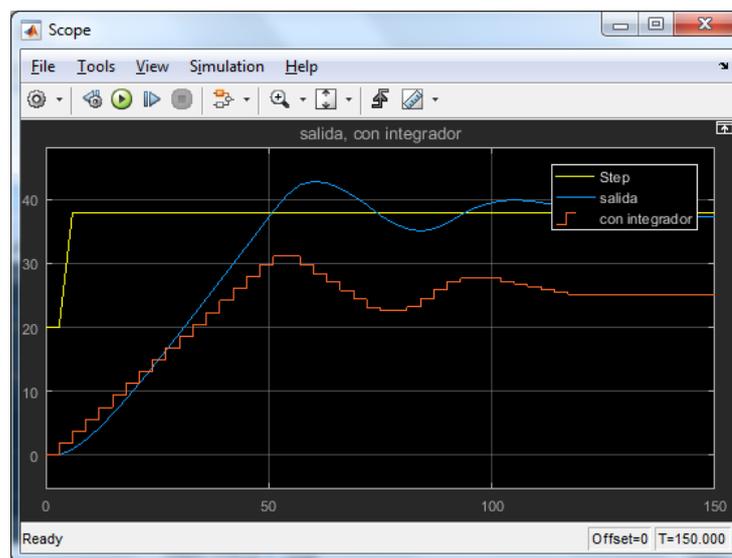


Figura 5 - 4 Primeros resultados de las simulaciones

Fuente: Freire Paúl, 2017

Tras varias modificaciones en los parámetros del controlador, se pudo obtener una curva de reacción mucho más estable, como se observa en la Figura 6 - 4. Con el controlador utilizado se procedió a verificar el funcionamiento real del sistema en un banco de pruebas.

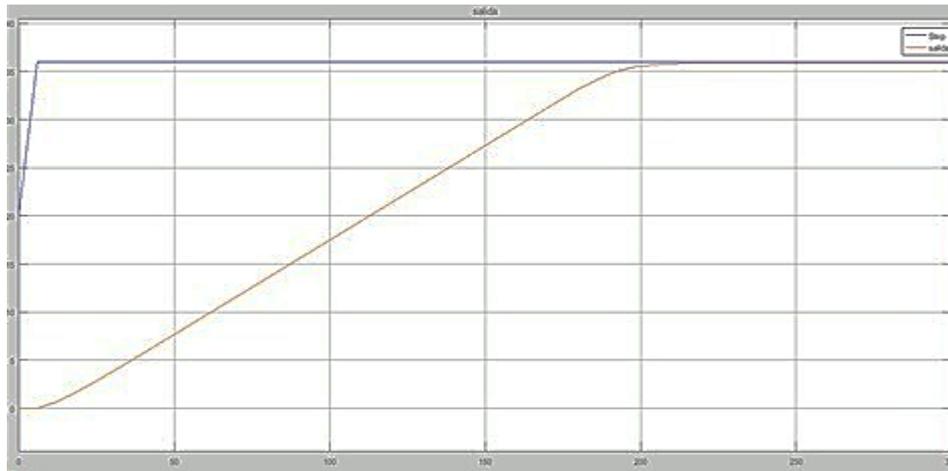


Figura 6 - 4 Respuesta simulada de la planta a una entrada de tipo escalón, luego de realizar ajustes en el controlador difuso inicial.

Fuente: Freire, Paúl, 2017

4.3 Pruebas y puesta a punto del controlador difuso de temperatura

Se construyó un banco de pruebas (Figura 7 - 4) y se monitoreó los valores de las variables en tiempo real. Se usó el módulo de monitoreo usado en las pruebas preliminares para registrar los cambios en las variables temperatura y caudal durante el funcionamiento del sistema.

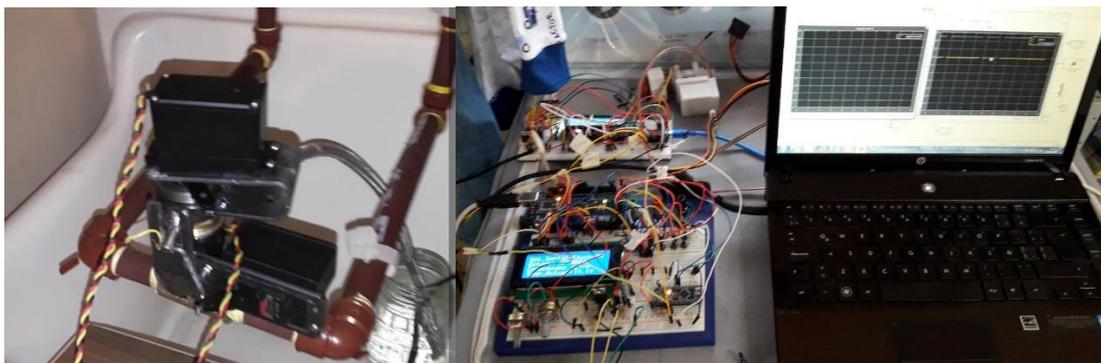


Figura 7 - 4 Banco de pruebas.

Fuente: Freire, Paúl, 2017

En las primeras pruebas se obtuvo una variación de $\pm 4^{\circ}\text{C}$ con respecto al set point con el controlador difuso empleado. Esto permitió tomar las correcciones necesarias en dicho controlador. Nuevamente se hizo necesario realizar modificaciones en los valores de pertenencia de las variables, procurando que cuando el error es bajo, el ángulo de giro de la válvula sea bajo. Se obtiene finalmente un controlador que estabiliza la temperatura al valor ingresado por el usuario con un error de $\pm 1^{\circ}\text{C}$, valor que se considera aceptable.

Se muestra en la Figura 8 - 4 el tablero de control del sistema. Se elaboró el circuito impreso en una máquina CNC y el soldado de los componentes se realizó manualmente. El circuito está diseñado de forma que se integra al Arduino Mega como un *Shield*. Los componentes se montan dentro de una caja plástica en donde se sellan las entradas de cables con silicona para impermeabilizar el sistema.

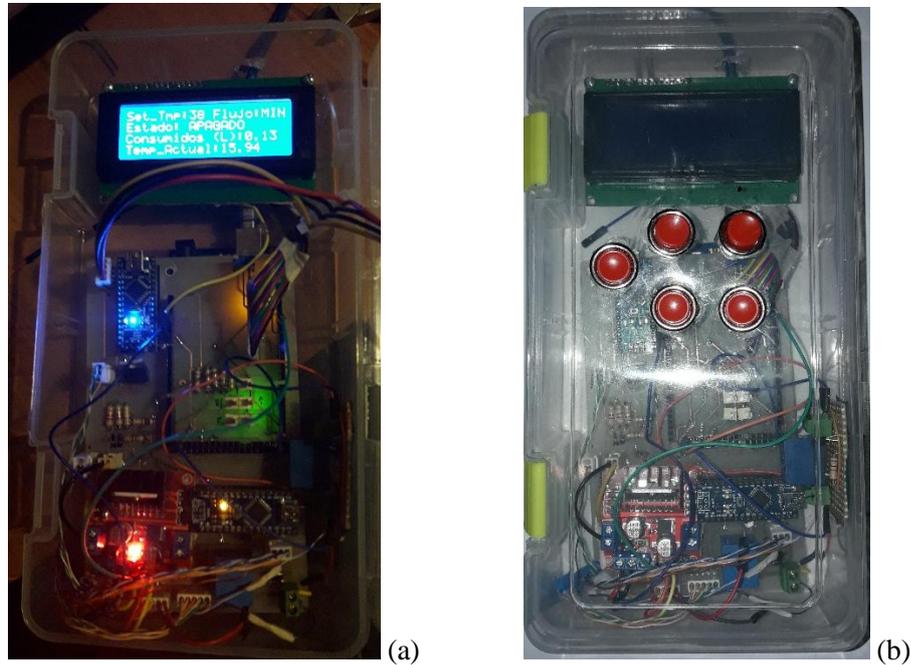


Figura 8 - 4 Tablero de control definitivo del sistema (a) abierto y cerrado (b)

Fuente: Freire, Paúl, 2017

Finalmente, se procedió a desmontar el banco de pruebas y a colocar los componentes en un cuarto de baño, para realizar las pruebas finales en el prototipo definitivo del sistema, que se muestra en la Figura 9 - 4 . Se obtienen los resultados que serán analizados para validar la presente investigación.



Figura 9 - 4 Prototipo definitivo del sistema.

Fuente: Freire, Paúl, 2017

4.4 Análisis de resultados del control difuso de temperatura

La curva de Temperatura en función del tiempo obtenida por el sistema para un set point de 36°C se muestra en el Gráfico 1-4. La validación del controlador se realizó considerando el máximo sobre impulso, $M_p(\%)$, el tiempo de establecimiento, T_s (s), y el error de estado estacionario, Ess. Se realizaron 5 pruebas, obteniendo similares resultados.

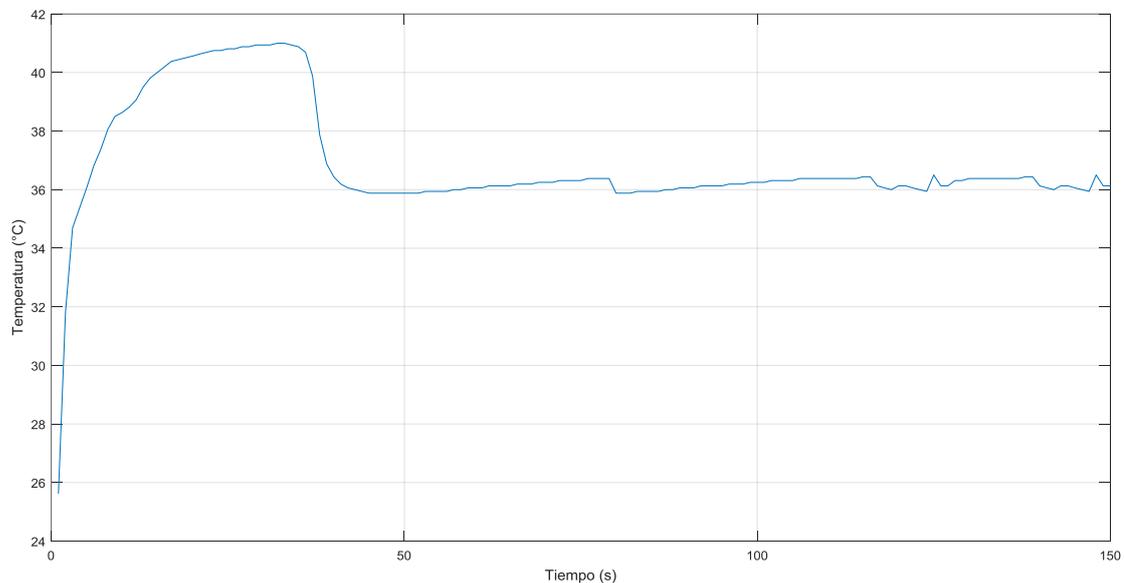


Gráfico 1 – 4 Gráfica de Temperatura en función del tiempo obtenida por el sistema para un set point de 36°C.

Fuente: Freire, Paúl, 2017.

De los datos registrados, se puede calcular M_p con la formula:

$$M_p(\%) = \frac{(\text{Valor Mximo Variable} - \text{set point}) * 100}{\text{set point}} \quad (15)$$

Donde Valor Mximo Variable es el pico mximo alcanzado por la variable bajo control, en este caso la temperatura. Aplicando (15) a los datos obtenidos se obtiene

$$M_p(\%) = 13,88\%$$

El tiempo de establecimiento T_s (s), es el tiempo en el cual la respuesta difiere del valor del set point en un rango de 2~5% (en valor absoluto). De acuerdo nuevamente a las mediciones este tiempo es de 39 segundos.

El error de estado estable es de 2.44%. Se considera que un sistema es estable si el error de estado estable está entre el 2 y el 5%.

4.5 Análisis de resultados del control difuso de caudal

La respuesta en el tiempo del controlador difuso de caudal se muestra en el Gráfico 2-4. La validación del controlador se realizó a través de cambios en el set point a los 100, 150, 175 y 260 segundos, para observar el seguimiento de la consigna por parte del controlador. Se observa que el error una vez que el sistema alcanza el valor ingresado, oscila entre cero y un litro por minuto. Esta diferencia de caudal no es percibida por el usuario. Existen picos y zonas en que el caudal no está fuera del rango del error descrito.

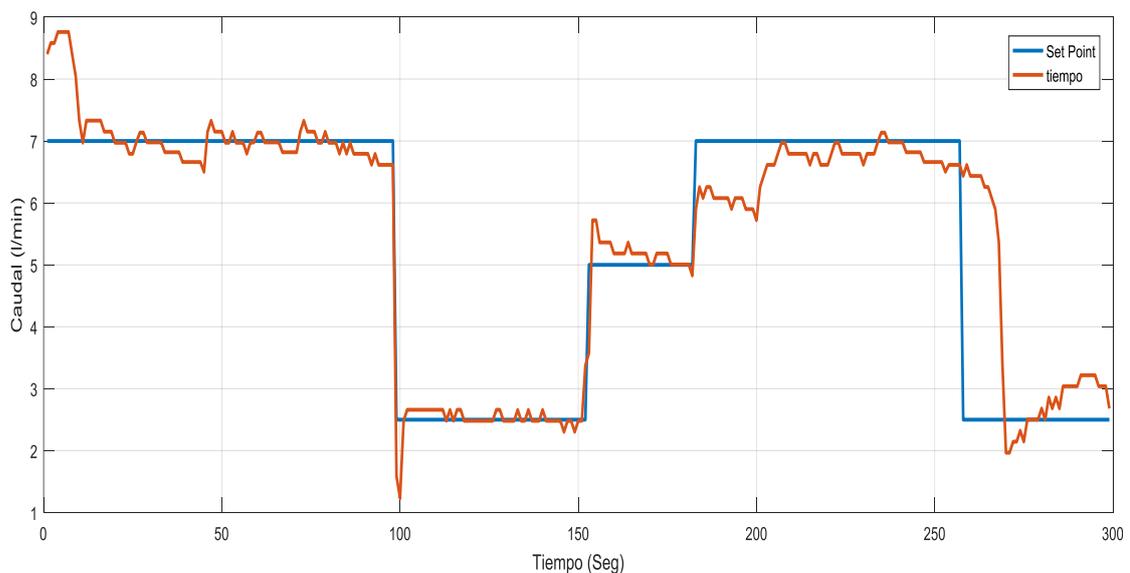


Gráfico 2-4 Respuesta en el tiempo del controlador difuso de caudal.

Fuente: Freire, Paúl, 2017.

Estas irregularidades se dan debido a que la bomba que suministra el caudal al domicilio se encuentra apagada, pues no detecta la variación de presión necesaria para arrancar, o cuando arranca produce un pico en el suministro de agua. Estas perturbaciones son compensadas rápidamente por el controlador.

4.6 Costos del prototipo

Se detalla en la tabla 3-4 el costo de los elementos que conforman el prototipo. El costo de una cabina de baño disponible en el mercado oscila entre los 300 a 1000 USD, por lo que la elaboración de un dispositivo comercial basado en la presente investigación, estaría a la par en

costos, debiendo además tomar en cuenta que estas cabinas comerciales no posee las mismas características de ahorro de agua.

Tabla 3-4: Costo de elementos del prototipo

Cantidad	Componente	Precio Unitario	Precio Total
1	Servo Hitec HS-755HB	40	40
1	Caudalímetro YF-S201	9	9
1	Sensor temperatura DS18B20	4	4
2	Electro válvula 12V	9	18
1	Valvula CWX-25S	70	70
1	Arduino Mega	15	15
2	Arduino Nano	7	14
1	Circuito impreso	30	30
1	Módulo Puente H	6	6
1	Fuente ATX	17	17
1	Componentes electrónicos	10	10
1	Componentes eléctricos	10	10
1	Grifo monomando FV	55	55
1	Ducha	20	20
1	Tubería termo resistente	10	10
1	Accesorios Plomería	10	10
		Valor	
		total:	338

CONCLUSIONES

- Se desperdicia por persona 10 litros de agua por ducha de 3 minutos en un sistema de calentamiento tradicional hasta obtener la temperatura deseada por el usuario. En un hogar de tres personas se malgastarían 30 litros diarios, que es el 28% del total de agua utilizada. En la ciudad de 225 mil habitantes, con un adecuado manejo del recurso, se lograría un ahorro de 2'463750 metros cúbicos de agua.
- En el mercado se encuentran varios dispositivos para el control de caudal y temperatura del agua, pero ninguno de ellos es programable o utiliza la lógica difusa.
- Se diseñó e implementó un sistema difuso para el control del caudal y la temperatura del agua para calentadores solares de uso doméstico, que permite optimizar el uso de líquido a través de recirculación y monitoreo de variables por parte de un sistema electro mecánico. Puede ser aplicado a cualquier sistema que posea tuberías independientes de agua caliente y fría, tales como los calefones eléctricos o a gas.
- El pico máximo de la variable temperatura, M_p (%) es de un 13%, que se considera un valor aceptable de acuerdo a los objetivos del proyecto. El tiempo de establecimiento T_s (s) , que es el tiempo en el cual la respuesta difiere del valor del set point en un rango de 2 al 5% es en promedio de 39 segundos, de acuerdo a cinco mediciones realizadas . Este valor depende de la distancia que tiene que cruzar el agua caliente desde el reservorio hasta la ducha.
- La defuzzificación de las variables se realiza mediante parámetros configurados en la interface gráfica de Simulink. Un controlador difuso tiene como principal ventaja que, de necesitarse un cambio, no es necesario volverlo a diseñar completamente, ya que observando el comportamiento del mismo, se podrá saber que cambios se debe hacer en las funciones de pertenencia o en las reglas, hasta obtener una respuesta adecuada.
- Fue necesario usar tarjetas de desarrollo independientes para procesar la información de temperatura y caudal de los sensores. En el caso de la temperatura, Simulink no soporta el protocolo Onewire, necesario para obtener información del sensor DS18B20. Para el caudal se requiere que la tarjeta de desarrollo soporte multitarea, no siendo esto posible para el Arduino Mega.

- Se diseñó un controlador on-off de caudal desde el punto de vista de la lógica difusa, teniendo en cuenta que la lógica binaria es un caso particular de la misma, con la ventaja de ser sencillo de implementar y controla de una mejor manera la oscilación propia de este tipo de controladores.
- El error de estado estable de la temperatura es de 2,44%. El error en estado estable del caudal oscila entre 0 y 1 l/min. Depende de perturbaciones como la apertura de otra llave en el domicilio, el encendido o apagado automático de la bomba del reservorio. El control difuso de caudal mostró una eficiente reacción ante estos eventos.
- Como trabajo futuro se propone un mejor manejo de la recirculación del agua, dado que en el prototipo desarrollado en esta investigación el agua regresa al reservorio a través de una tubería alterna a la domiciliaria. Esta recirculación podría diseñarse de tal forma que el agua retorne por la misma tubería de agua fría de la instalación existente.

RECOMENDACIONES

- De implementarse esta investigación como un dispositivo comercial, se deberá utilizar elementos que sean resistentes a la humedad, ya que el tablero estará dentro del cuarto de baño, donde existe vapor de agua que puede dañar los componentes electrónicos. Se recomienda para un futuro diseño que el tablero de control sea impermeable para que el usuario pueda realizar cambios en los parámetros del sistema durante el proceso de duchado.
- Es posible optimizar el uso de tarjetas de desarrollo utilizando otros protocolos de comunicación, haciendo que la información de los sensores sea procesada en una sola de ellas. Se recomienda buscar una solución de programación para que todos los sensores sean manejados en una sola tarjeta y ahorrar costos y espacio físico.
- Se recomienda la conformación de un equipo multidisciplinario, que aporte en la solución de requerimientos electrónicos, mecánicos, de programación y control automático.
- El obtener la función de transferencia de la planta que se desea controlar permite realizar simulaciones que evitan que se someta a continuo funcionamiento la planta real, y por consiguiente, desperdicio de tiempo e insumos.
- El software de este prototipo es fácilmente configurable dado que está creado en una interfaz gráfica, sin requerir cambios de hardware, por lo que es factible añadirle nuevas funciones o mejorar las actuales. Se recomienda la incorporación de alarmas, tiempos de duchado, cambios de interfaz, o un mejoramiento de los controladores difusos.

BIBLIOGRAFÍA

- Aimagin.** (2 de Enero de 2017). *aimagin.com*. Obtenido de https://www.aimagin.com/downloads/dl/file/id/186/fuzzy_logic_fio_std_lab_sheet.pdf
- Aja Fernández, S.** (2003). *Un nuevo marco matricial para la implementación de inferencia borrosa.Tesis de Doctorado*. Valladolid: Universidad de Valladolid,Escuela Técnica Superior de Ingenieros de Telecomunicación.
- aliexpress.com.** (3 de 9 de 2017). <https://es.aliexpress.com>. Obtenido de https://es.aliexpress.com/item/CWX-25S-Stainless-steel-Motorized-Ball-Valve-1-2-DN15-Water-control-Angle-valve-DC12V-2/1000001435640.html?aff_platform=aaf&cpt=1508950082854&sk=fUbQV7q&aff_trace_key=75e179d2f9ba41c79ce14979d383e7b6-1508950082854-00353-fUbQ
- Altecdust.** (6 de 6 de 2017). Obtenido de Sitio Web de Altec: <http://www.altecdust.com/sopORTE-tecnico/que-son-las-electrovalvulas>
- Arce Rubio, A., & Vianna Raffo, G.** (Marzo de 2009). *Manual de simulink para la asignatura de teoría de sistemas*. Sevilla: Universidad de Sevilla Escuela Superior de Ingenieros, Departamento de Ingeniería de Sistemas y Automática.
- Bahón, C., & Raya Giner, C.** (2004). *Tecnología de Sistemas de Control*. Barcelona: Edicions UPC.
- Bentley, J.** (1993). *Sistemas de Medición: Principios y Aplicaciones (SMPA)*. México: Continental.
- Bérriz, L.** (2011). Ventajas y desventajas de los calentadores solares. *Energía y tú,*.
- Brizzio , E.** (11 de 11 de 2017). <http://www.ezebrizzio.com.ar>. Obtenido de <http://www.ezebrizzio.com.ar/Ejemplos/LCD/Modulo%20LCD.pdf>
- Cabrera Almache, J.** (14 de 2 de 2013). Recuperado el 5 de 8 de 2016, de www.ucuenca.edu.ec: www.ucuenca.edu.ec/ojs/index.php/estoa/article/download/306/259

- Candelas Herías, F. A., & Corrales Ramón, J. A.** (2007). *Servomotores*. Alicante: Grupo de Automática, Robótica y Visión Artificial, Universidad de Alicante.
- Cejas Montero, J.** (8 de Enero de 2017). <http://www.monografias.com>. Obtenido de <http://www.monografias.com/trabajos95/master-direccion/master-direccion.shtml>
- Cruz, A.** (1 de octubre de 2017). *electronilab.co*. Obtenido de <https://electronilab.co/tutoriales/tutorial-de-uso-driver-dual-l298n-para-motores-dc-y-paso-a-paso-con-arduino/>
- Del Cerro Sánchez, T.** (1 de 8 de 2016). <http://www.it.uc3m.es>. Recuperado el 5 de 8 de 2016, de <http://www.it.uc3m.es/jvillena/irc/practicas/13-14/06.pdf>
- Eco Vita. (2016).** *Comercial EcoVita*. Recuperado el 18 de 01 de 2016, de EcoVITA Calentadores Solares : www.ecovita.mx
- Espinosa Sarria, M.** (2007). *Manual para el consumo responsable de agua potable*. Chile: La Cuarta.
- FACUA.** (13 de 10 de 2017). *FACUA*. Obtenido de www.facua.org
- Facultad de Ingeniería, Universidad de la República, Uruguay.** (03 de septiembre de 2017). www.fing.edu.uy. Obtenido de <https://www.fing.edu.uy/iq/cursos/dcp/>
- Franz Viegner.** (08 de 01 de 2016). Manual de instalación grifo monomando Franz Viegner E108.05/B1. Ecuador.
- Gallardo García, D.** (2016). *Apuntes de Arduino Nivel Enteradillo*. Sevilla: IES LAGUNA DE TOLLÓN.
- González Morcillo, C.** (2011). *Lógica Difusa, Una introducción práctica, Técnicas de Softcomputing*. La Mancha: Universidad de Castilla.
- Gordillo Guambaña, B. G.** (2014). *Sintonización mediante formulación teórica del regulador PID para el control del nivel de agua en depósitos que alimentan aducciones de gran longitud*. Valencia: Universidad Politécnica de Valencia.

- Ibrahim, D. (2002).** *Microcontroller Based Temperature Monitoring and Control*. Miami: Newnes.
- Ingeniería MCI Ltda. . (20 de 10 de 2017).** *MCI electronics*. Obtenido de MCI electronics: <http://arduino.cl/arduino-mega-2560/>
- Maguiña Pérez, R. A. (2010).** Sistemas de inferencia basados en Lógica Borrosa: Fundamentos y caso de estudio. (U. N. Marcos, Ed.) *Revista de investigación de sistemas e informática*, 91-104.
- Maxim Integrated Products, Inc. (3 de 12 de 2015).** *www.maximintegrated.com*. Obtenido de <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- Mecatronicum. (1 de 10 de 2017).** *Mecatronicum Chips*. Obtenido de <http://chips.mecatronicum.com/tutorial-sensor-de-flujo-yf-s201/>
- Miranda, C. (2012).** *Sistemas de control continuos y discretos*. Madrid: Paraninfo.
- Moreno, L., Garrido, S., & Balaguer, C. (2003).** *Ingeniería de Control Modelado, Análisis y control de sistemas dinámicos*. Barcelona: Ariel.
- Mugica Alvarez, F. (2007).** *Diseño Sistemático de Controladores Difusos usando Razonamiento Inductivo*. Cataluña: Universidad Politécnica de Cataluña.
- Ogata, K. (1987).** *Dinámica de sistemas*. México: Prentice-Hall Hispanoamericana, S.A.
- Ogata, K. (1998).** *Ingeniería de Control Moderna*. México: Prentice-Hall Hispanoamericana.
- Pascual Fernandez, J. L. (4 de 5 de 2016). <http://upcommons.upc.edu>. Obtenido de <http://upcommons.upc.edu/bitstream/handle/2099.1/2687/31097-1.pdf?sequence=1&isAllowed=y>
- Ponce Cruz, P. (2010).** *Inteligencia Artificial con aplicaciones a la ingeniería*. México D.F.: Alfaomega Grupo Editor, S.A. de C.V.
- PROMETEC. (3 de 10 de 2016).** *www.prometec.net*. Obtenido de www.prometec.net/bus-i2c/
Semplades. (2013-2017). *Plan Nacional para el Buen Vivir*. Quito.

Ser, C. (s.f.). *cadenaser.com*. Obtenido de http://cadenaser.com/ser/2014/01/20/sociedad/1390187610_850215.html

Tumero, P. (3 de Enero de 2017). *monografias.com*. Obtenido de <http://www.monografias.com/trabajos106/optimizacion-lazos-control/optimizacion-lazos-control2.shtml>

Universidad Politécnica de Madrid, Dpto. Electrónica, Automática e Informática Industrial

. (s.f.). *Página del departamento de ingeniería eléctrica, electrónica, automática y física aplicada*. Obtenido de <http://www.elai.upm.es/>: http://www.elai.upm.es/webantigua/spain/Asignaturas/Servos/Apuntes/6_AnaTemp_1_2.pdf

Valdivia Miranda, C. (2012). *Sistemas de control continuos y discretos*. Madrid: Paraninfo.

Vargas Toro, D. V. (2012). Control difuso de la temperatura del agua de la ducha en el hogar para: Vía Vargas y Asociados Consultores. *Control difuso de la temperatura del agua de la ducha en el hogar para: Vía Vargas y Asociados Consultores*. Quito: Universidad Tecnológica Equinoccial.

Vivanco, L. (5 de Enero de 2017). *www.emaze.com*. Obtenido de <https://www.emaze.com/@ALWRCQQW>

Watts Industries. (06 de 1 de 2016). *Válvulas mezcladoras termostáticas*. Recuperado el 6 de 6 de 2016, de Watts Industries: www.wattsindustries.com

Zadeh, L. (1965). Fuzzy set. *Information and Control*. En L. Zadeh, *Fuzzy set. Information and Control* (págs. 338–353).

Zadeh, L. (1990). FUZZY SETS AND SYSTEMS*. *International Journal of General Systems*, 17, 129-138.

ANEXO 1 Hojas técnicas de los elementos que conforman el sistema.

DataSheet CWX-25S Valve

TECHNICAL PARAMETERS								
Series	Max torque	Operating time	Working voltage	Max power	Enclosure	Max pressure	Medium temperature	Ambient temperature
CWX-25	2N.m	75sec	DCSV	~2W	IP65	1.6Mpa	0-100℃	-20℃~+45℃
			AC/DC3-24V					
			AC85-265V					

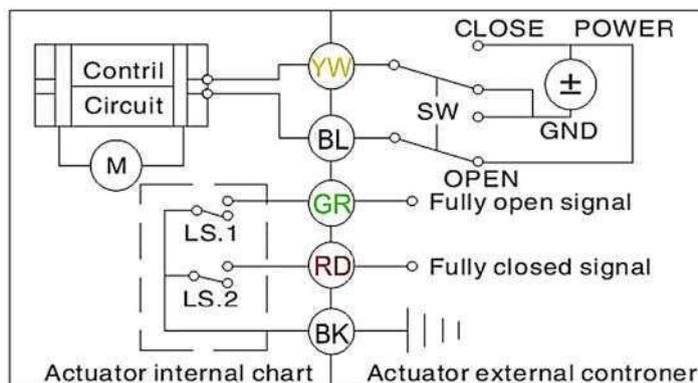
	FEATURES	APPLICATION
	Mini dimension suit for small equipment	HVAC
	Patented technology-floating seal structure	Water treatment
	Long service life:80000-100000 strokes	Chemical process
	Low working current, suitable for battery driving	Small equipment for automatic Control
	Manual override and position indicator	Replacing solenoid valve, particularly when solenoid can't work reliably
Metal gearbox and POM gear		
High flow capability		

PARTS		MATERIAL
Actuator housing		ABS
Valve body		88304.318
Valve ball		88304.318
Stem		88304.318
Ball seat		PTFE
Seal		EPDM

Valve size(mm)	Size	D1/D2	Φd	L1	L2	L3	F	h	H1	H2	H3	b	d	W(kg)
Both Female screw	DN15	G1/2"	16	11	11	64	27	4.6	16	22	66	73	81	0.38
	DN20	G3/4"	16	10	10	82	30	4.6	18	22	66	73	81	0.42
	DN25	G1"	20	16	16	73	38	4.6	22	26	66	73	81	0.53

WIRING				
CR01(2 wires)	CR02(3wires)	CR03(3wires)	CR04(2wires)	CR05(5wires)
Standard control	Two point control	One point control	Auto-return	Signal feedback

INDUSTRY USE FOR MANUAL OVERRIDE
 It is only permitted to use when power-off.
 Lift the hand-wheel, and turn it left or right until the valve is in place.
 When the black line on the indicator is horizontal, the valve is on; when vertical, the valve is off.
 Press down hand-wheel after using manual override in order to work properly when power-on.



DS18B20 Programmable Resolution 1-Wire Digital Thermometer

Absolute Maximum Ratings

Voltage Range on Any Pin Relative to Ground: -0.5V to +5.0V Storage Temperature Range: -55°C to +125°C
 Operating Temperature Range: -55°C to +125°C Solder Temperature: Refer to the IPC/JEDEC J-87D-020 Specification.

These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

DC Electrical Characteristics

(-55°C to +125°C; V_{DD} = 3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V _{DD}	Local power (Note 1)	+3.0		+5.5	V
Pullup Supply Voltage	V _{PU}	Parasite power	+3.0		+5.5	V
		Local power (Notes 1, 2)	+3.0		V _{DD}	
Thermometer Error	t _{ERR}	-10°C to +85°C			±0.5	°C
		-55°C to +125°C			±2	
Input Logic-Low	V _{IL}	(Notes 1, 4, 5)	-0.3		+0.8	V
Input Logic-High	V _{IH}	Local power	+2.2		The lower of 5.5 or V _{DD} + 0.3	V
		Parasite power (Notes 1, 6)	+3.0			
Sink Current	I _L	V _{I/O} = 0.4V	4.0			mA
Standby Current	I _{DDS}	(Notes 7, 8)		750	1000	nA
Active Current	I _{DD}	V _{DD} = 5V (Note 9)		1	1.5	mA
DQ Input Current	I _{DQ}	(Note 10)		5		µA
Drift		(Note 11)		±0.2		°C

- Note 1: All voltages are referenced to ground.
- Note 2: The Pullup Supply Voltage specification assumes that the pullup device is ideal, and therefore the high level of the pullup is equal to V_{PU}. In order to meet the V_{IH} spec of the DS18B20, the actual supply rail for the strong pullup transistor must include margin for the voltage drop across the transistor when it is turned on; thus: V_{PU_ACTUAL} = V_{PU_IDEAL} + V_{TRANSISTOR}.
- Note 3: See typical performance curve in Figure 1.
- Note 4: Logic-low voltages are specified at a sink current of 4mA.
- Note 5: To guarantee a presence pulse under low voltage parasite power conditions, V_{I,MAX} may have to be reduced to as low as 0.5V.
- Note 6: Logic-high voltages are specified at a source current of 1mA.
- Note 7: Standby current specified up to +70°C. Standby current typically is 3µA at +125°C.
- Note 8: To minimize I_{DDS}, DQ should be within the following ranges: GND ≤ DQ ≤ GND + 0.3V or V_{DD} - 0.3V ≤ DQ ≤ V_{DD}.
- Note 9: Active current refers to supply current during active temperature conversions or EEPROM writes.
- Note 10: DQ I/O is high (high-Z state).
- Note 11: Drift data is based on a 1000-hour stress test at +125°C with V_{DD} = 5.5V.

DS18B20 Programmable Resolution 1-Wire Digital Thermometer

General Description

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with non-volatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

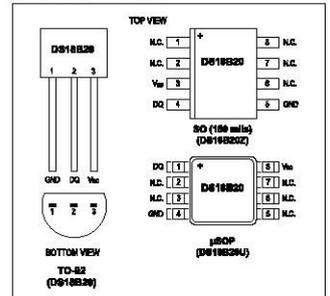
Applications

- Thermostatic Controls
- Industrial Systems
- Consumer Products
- Thermometers
- Thermally Sensitive Systems

Benefits and Features

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Reduce Component Count with Integrated Temperature Sensor and EEPROM
 - Measure Temperatures from -55°C to +125°C (-67°F to +257°F)
 - ±0.5°C Accuracy from -10°C to +85°C
 - Programmable Resolution from 9 Bits to 12 Bits
 - No External Components Required
- Parasite Power Mode Requires Only 2 Pins for Operation (DQ and GND)
- Simplified Distributed Temperature-Sensing Applications with Multitrip Capability
 - Each Device Has a Unique 64-Bit Serial Code Stored in On-Board ROM
- Flexible User-Definable Nonvolatile (NV) Alarm Settings with Alarm Search Command Identifies Devices with Temperature Outside Programmed Limits
- Available in 8-Pin SO (160 mils), 8-Pin µSOP, and 3-Pin TO-92 Packages

Pin Configurations



Ordering information appears at end of data sheet.

1-Wire is a registered trademark of Maxim Integrated Products, Inc.



DS18B20 Programmable Resolution 1-Wire Digital Thermometer

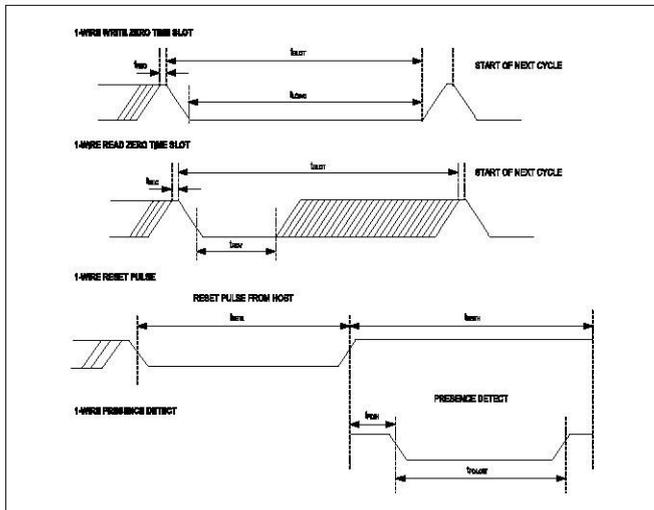


Figure 2. Timing Diagram

Pin Description

PIN		NAME	FUNCTION
SO	µSOP	TO-92	
1, 2, 8, 7, 8	2, 3, 5, 6, 7	—	No Connection
3	8	3	V _{DD} Optional V _{DD} . V _{DD} must be grounded for operation in parasite power mode.
4	1	2	DQ Data Input/Output. Open-drain 1-Wire interface pin. Also provides power to the device when used in parasite power mode (see the Powering the DS18B20 section.)
5	4	1	GND Ground

DS18B20 Programmable Resolution 1-Wire Digital Thermometer

AC Electrical Characteristics—NV Memory

(-55°C to +125°C; V_{DD} = 3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
NV Write Cycle Time	t _{WR}			2	10	ms
EEPROM Write	t _{EEWR}	-55°C to +55°C		50k		writes
EEPROM Data Retention	t _{EEDR}	-55°C to +55°C		10		years

AC Electrical Characteristics

(-55°C to +125°C; V_{DD} = 3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Temperature Conversion Time	t _{CONV}	9-bit resolution			83.75	ms
		10-bit resolution			187.5	
		11-bit resolution			375	
		12-bit resolution			750	
Time to Strong PUMP On	t _{SPON}	Start convert T command issued		10		µs
Time Slot	t _{SLT}	(Note 12)	60	120		µs
Recovery Time	t _{REG}	(Note 12)	1			µs
Write 0 Low Time	t _{LOW0}	(Note 12)	60	120		µs
Write 1 Low Time	t _{LOW1}	(Note 12)	1	16		µs
Read Data Valid	t _{RDV}	(Note 12)		15		µs
Reset Time High	t _{RESTH}	(Note 12, 13)	480			µs
Reset Time Low	t _{RESTL}	(Note 12, 13)	480			µs
Presence-Detect High	t _{PDHIGH}	(Note 12)	15	60		µs
Presence-Detect Low	t _{PDLOW}	(Note 12)	60	240		µs
Capacitance	C _{INDUT}			26		pF

Note 12: See the timing diagrams in Figure 2.
 Note 13: Under parasite power, if t_{RESTL} > 900µs, a power-on reset can occur.

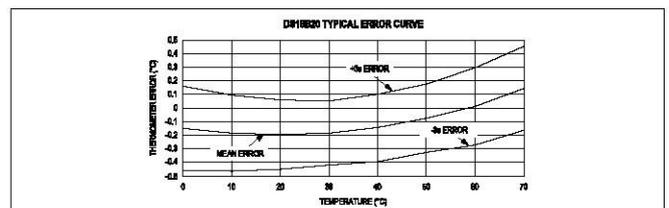


Figure 1. Typical Performance Curve

The DS18B20 output temperature data is calibrated in degrees Celsius; for Fahrenheit applications, a lookup table or conversion routine must be used. The temperature data is stored as a 16-bit sign-extended two's complement number in the temperature register (see Figure 4). The sign bits (S) indicate if the temperature is positive or negative; for positive numbers S = 0 and for negative numbers S = 1. If the DS18B20 is configured for 12-bit resolution, all bits in the temperature register will contain valid data. For 11-bit resolution, bit 0 is undefined. For 10-bit resolution, bits 1 and 0 are undefined, and for 9-bit resolution bits 2, 1, and 0 are undefined. Table 1 gives examples of digital output data and the corresponding temperature reading for 12-bit resolution conversions.

LS BYTE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	2 ⁻¹	2 ⁻²
MS BYTE	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
	S	S	S	S	S	2 ⁴	2 ⁵	2 ⁶

S = SIGN

Figure 4. Temperature Register Format

Table 1. Temperature/Data Relationship

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)
+125	0000 0111 1101 0000	07D0h
+65°	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0181h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

*The power-on reset value of the temperature register is +85°C.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
S	2 ⁴	2 ⁵	2 ⁶	2 ⁷	2 ⁸	2 ⁹	2 ⁰

Figure 5. T_H and T_L Register Format

Overview

Figure 3 shows a block diagram of the DS18B20, and pin descriptions are given in the Pin Description table. The 64-bit ROM stores the device's unique serial code. The scratchpad memory contains the 2-byte temperature register that stores the digital output from the temperature sensor. In addition, the scratchpad provides access to the 1-byte upper and lower alarm trigger registers (T_H and T_L) and the 1-byte configuration register. The configuration register allows the user to set the resolution of the temperature-to-digital conversion to 9, 10, 11, or 12 bits. The T_H, T_L, and configuration registers are nonvolatile (EEPROM), so they will retain data when the device is powered down.

The DS18B20 uses Maxim's exclusive 1-Wire bus protocol that implements bus communication using one control signal. The control line requires a weak pullup resistor since all devices are linked to the bus via a 3-state or open-drain port (the DQ pin in the case of the DS18B20). In this system, the microprocessor (the master device) identifies and addresses devices on the bus using each device's unique 64-bit code. Because each device has a unique code, the number of devices that can be addressed on one bus is virtually unlimited. The 1-Wire bus protocol, including detailed explanations of the commands and "time slots," is covered in the 1-Wire Bus System section.

Another feature of the DS18B20 is the ability to operate without an external power supply. Power is instead supplied through the 1-Wire pullup resistor through the

DQ pin when the bus is high. The high bus signal also charges an internal capacitor (C_{pp}), which then supplies power to the device when the bus is low. This method of deriving power from the 1-Wire bus is referred to as "parasite power." As an alternative, the DS18B20 may also be powered by an external supply on V_{DD}.

Operation—Measuring Temperature

The core functionality of the DS18B20 is its direct-to-digital temperature sensor. The resolution of the temperature sensor is user-configurable to 9, 10, 11, or 12 bits, corresponding to increments of 0.5°C, 0.25°C, 0.125°C, and 0.0625°C, respectively. The default resolution at power-up is 12-bit. The DS18B20 powers up in a low-power idle state. To initiate a temperature measurement and A-to-D conversion, the master must issue a Convert T [44h] command. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18B20 returns to its idle state. If the DS18B20 is powered by an external supply, the master can issue "read time slots" (see the 1-Wire Bus System section) after the Convert T command and the DS18B20 will respond by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done. If the DS18B20 is powered with parasite power, this notification technique cannot be used since the bus must be pulled high by a strong pullup during the entire temperature conversion. The bus requirements for parasite power are explained in detail in the Powering the DS18B20 section.

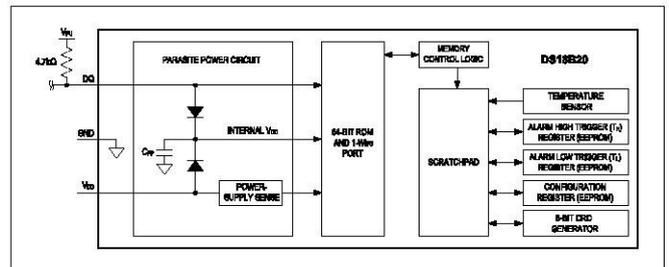


Figure 3. DS18B20 Block Diagram

64-BIT Lasered ROM code

Each DS18B20 contains a unique 64-bit code (see Figure 8) stored in ROM. The least significant 8 bits of the ROM code contain the DS18B20's 1-Wire family code: 28h. The next 48 bits contain a unique serial number. The most significant 8 bits contain a cyclic redundancy check (CRC) byte that is calculated from the first 55 bits of the ROM code. A detailed explanation of the CRC bits is provided in the CRC Generation section. The 64-bit ROM code and associated ROM function control logic allow the DS18B20 to operate as a 1-Wire device using the protocol detailed in the 1-Wire Bus System section.

Memory

The DS18B20's memory is organized as shown in Figure 9. The memory consists of an SRAM scratchpad with nonvolatile EEPROM storage for the high and low alarm trigger registers (T_H and T_L) and configuration register. Note that if the DS18B20 alarm function is not used, the T_H and T_L registers can serve as general-purpose memory. All memory commands are described in detail in the DS18B20 Function Commands section.

Byte 0 and byte 1 of the scratchpad contain the LSB and the MSB of the temperature register, respectively. These bytes are read-only. Bytes 2 and 3 provide access to T_H and T_L registers. Byte 4 contains the configuration regis-

ter data, which is explained in detail in the Configuration Register section. Bytes 5, 6, and 7 are reserved for internal use by the device and cannot be overwritten.

Byte 8 of the scratchpad is read-only and contains the CRC code for bytes 0 through 7 of the scratchpad. The DS18B20 generates this CRC using the method described in the CRC Generation section.

Data is written to bytes 2, 3, and 4 of the scratchpad using the Write Scratchpad [4Eh] command; the data must be transmitted to the DS18B20 starting with the least significant bit of byte 2. To verify data integrity, the scratchpad can be read (using the Read Scratchpad [BEh] command) after the data is written. When reading the scratchpad, data is transferred over the 1-Wire bus starting with the least significant bit of byte 0. To transfer the T_H, T_L, and configuration data from the scratchpad to EEPROM, the master must issue the Copy Scratchpad [46h] command.

Data in the EEPROM registers is retained when the device is powered down; at power-up the EEPROM data is reloaded into the corresponding scratchpad locations. Data can also be reloaded from EEPROM to the scratchpad at any time using the Recall E2 [B8h] command. The master can issue read time slots following the Recall E2 command and the DS18B20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done.

8-BIT CRC		48-BIT SERIAL NUMBER				8-BIT FAMILY CODE (28h)			
MSB	LSB	MSB	MSB	LSB	MSB	MSB	MSB	LSB	

Figure 8. 64-Bit Lasered ROM Code

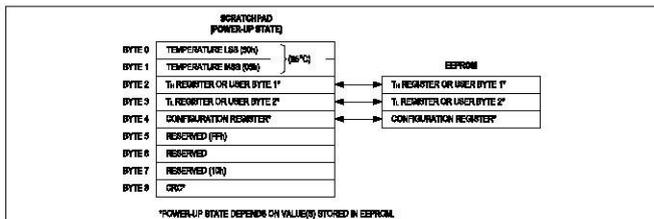


Figure 9. DS18B20 Memory Map

or equal to T_L or higher than or equal to T_H, an alarm condition exists and an alarm flag is set inside the DS18B20. This flag is updated after every temperature measurement; therefore, if the alarm condition goes away, the flag will be turned off after the next temperature conversion.

The master device can check the alarm flag status of all DS18B20s on the bus by issuing an Alarm Search [ECH] command. Any DS18B20s with a set alarm flag will respond to the command, so the master can determine exactly which DS18B20s have experienced an alarm condition. If an alarm condition exists and the T_H or T_L settings have changed, another temperature conversion should be done to validate the alarm condition.

Powering the DS18B20

The DS18B20 can be powered by an external supply on the V_{DD} pin, or it can operate in "parasite power" mode, which allows the DS18B20 to function without a local external supply. Parasite power is very useful for applications that require remote temperature sensing or that use very space constrained. Figure 3 shows the DS18B20's parasite-power control circuitry, which "steals" power from the 1-Wire bus via the DQ pin when the bus is high. The stolen charge powers the DS18B20 while the bus is high, and some of the charge is stored on the parasite power capacitor (C_{pp}) to provide power when the bus is low. When the DS18B20 is used in parasite power mode, the V_{DD} pin must be connected to ground.

In parasite power mode, the 1-Wire bus and GPP can provide sufficient current to the DS18B20 for most operations as long as the specified timing and voltage requirements are met (see the DC Electrical Characteristics and AC Electrical Characteristics). However, when the DS18B20 is performing temperature conversions or copying data from the scratchpad memory to EEPROM, the operating current can be as high as 1.5mA. This current can cause an unacceptable voltage drop across the weak 1-Wire pullup resistor and is more current than can be supplied

by C_{pp}. To ensure that the DS18B20 has sufficient supply current, it is necessary to provide a strong pullup on the 1-Wire bus whenever temperature conversions are taking place or data is being copied from the scratchpad to EEPROM. This can be accomplished by using a MOSFET to pull the bus directly to the rail as shown in Figure 6. The 1-Wire bus must be switched to the strong pullup within 10µs (max) after a Convert T [44h] or Copy Scratchpad [4Eh] command is issued, and the bus must be held high by the pullup for the duration of the conversion (CONV) or data transfer (t_{WR} = 10ms). No other activity can take place on the 1-Wire bus while the pullup is enabled.

The DS18B20 can also be powered by the conventional method of connecting an external power supply to the V_{DD} pin, as shown in Figure 7. The advantage of this method is that the MOSFET pullup is not required, and the 1-Wire bus is free to carry other traffic during the temperature conversion time.

The use of parasite power is not recommended for temperatures above +100°C since the DS18B20 may not be able to sustain communications due to the higher leakage currents that can exist at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that the DS18B20 be powered by an external power supply.

In some situations the bus master may not know whether the DS18B20s on the bus are parasite powered or powered by external supplies. The master needs this information to determine if the strong bus pullup should be used during temperature conversions. To get this information, the master can issue a Skip ROM [CCh] command followed by a Read Power Supply [B4h] command, followed by a "read time slot". During the read time slot, parasite powered DS18B20s will let the bus remain high. If the bus is pulled low, the master knows that it must supply the strong pullup on the 1-Wire bus during temperature conversions.

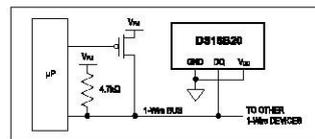


Figure 6. Supplying the Parasite-Powered DS18B20 During Temperature Conversion

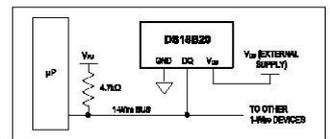


Figure 7. Powering the DS18B20 with an External Supply

1-Wire Bus System

The 1-Wire bus system uses a single bus master to control one or more slave devices. The DS18B20 is always a slave. When there is only one slave on the bus, the system is referred to as a "single-drop" system; the system is "multidrop" if there are multiple slaves on the bus.

All data and commands are transmitted least significant bit first over the 1-Wire bus.

The following discussion of the 1-Wire bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-Wire signaling (signal types and timing).

Hardware Configuration

The 1-Wire bus has by definition only a single data line. Each device (master or slave) interfaces to the data line via an open-drain or 3-state port. This allows each device to "release" the data line when the device is not transmitting data so the bus is available for use by another device. The 1-Wire port of the DS18B20 (the DQ pin) is open drain with an internal circuit equivalent to that shown in Figure 12.

The 1-Wire bus requires an external pullup resistor of approximately 5kΩ, thus, the idle state for the 1-Wire bus is high. If for any reason a transaction needs to be suspended, the bus MUST be left in the idle state if the transaction is to resume. Infringe recovery time can occur between bits so long as the 1-Wire bus is in the inactive (high) state during the recovery period. If the bus is held low for more than 480µs, all components on the bus will be reset.

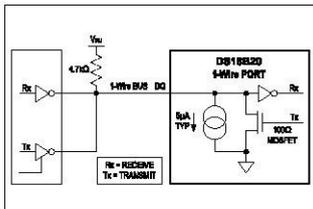


Figure 12. Hardware Configuration

Transaction Sequence

The transaction sequence for accessing the DS18B20 is as follows:

- Step 1. Initialization
- Step 2. ROM Command (followed by any required data exchange)
- Step 3. DS18B20 Function Command (followed by any required data exchange)

It is very important to follow this sequence every time the DS18B20 is accessed, as the DS18B20 will not respond if any steps in the sequence are missing or out of order. Exceptions to this rule are the Search ROM [F0h] and Alarm Search [ECh] commands. After issuing either of these ROM commands, the master must return to Step 1 in the sequence.

Initialization

All transactions on the 1-Wire bus begin with an initialization sequence. The initialization sequence consists of a reset pulse transmitted by the bus master followed by presence pulse(s) transmitted by the slave(s). The presence pulse lets the bus master know that slave devices (such as the DS18B20) are on the bus and are ready to operate. Timing for the reset and presence pulses is detailed in the 1-Wire Signaling section.

ROM Commands

After the bus master has detected a presence pulse, it can issue a ROM command. These commands operate on the unique 64-bit ROM codes of each slave device and allow the master to single out a specific device if many are present on the 1-Wire bus. These commands also allow the master to determine how many and what types of devices are present on the bus or if any device has experienced an alarm condition. There are five ROM commands, and each command is 8 bits long. The master device must issue an appropriate ROM command before issuing a DS18B20 function command. A flowchart for operation of the ROM commands is shown in Figure 13.

Search Rom [F0h]

When a system is initially powered up, the master must identify the ROM codes of all slave devices on the bus, which allows the master to determine the number of slaves and their device types. The master learns the ROM codes through a process of elimination that requires the master to perform a Search ROM cycle (i.e., Search ROM command followed by data exchange) as many times as necessary to identify all of the slave devices.

Copy Scratchpad [48h]

This command copies the contents of the scratchpad TH, TL and configuration registers (bytes 2, 3 and 4) to EEPROM. If the device is being used in parasitic power mode, within 10µs (max) after this command is issued the master must enable a strong pullup on the 1-Wire bus for at least 10ms as described in the Powering the DS18B20 section.

Recall E² [B8h]

This command recalls the alarm trigger values (TH and TL) and configuration data from EEPROM and places the data in bytes 2, 3, and 4, respectively, in the scratchpad memory. The master device can issue read time slots

following the Recall E² command and the DS18B20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done. The recall operation happens automatically at power-up, so valid data is available in the scratchpad as soon as power is applied to the device.

Read Power Supply [B4h]

The master device issues this command followed by a read time slot to determine if any DS18B20s on the bus are using parasitic power. During the read time slot, parasitic powered DS18B20s will pull the bus low, and externally powered DS18B20s will let the bus remain high. See the Powering the DS18B20 section for usage information for this command.

Table 3. DS18B20 Function Command Set

COMMAND	DESCRIPTION	PROTOCOL	1-WIRE BUS ACTIVITY AFTER COMMAND IS ISSUED	NOTES
TEMPERATURE CONVERSION COMMANDS				
Convert T	Initiate temperature conversion.	44h	DS18B20 transmits conversion status to master (not applicable for parasitic-powered DS18B20s).	1
MEMORY COMMANDS				
Read Scratchpad	Reads the entire scratchpad including the CRC byte.	BEh	DS18B20 transmits up to 9 data bytes to master.	2
Write Scratchpad	Writes data into scratchpad bytes 2, 3, and 4 (TH, TL, and configuration registers).	4Eh	Master transmits 3 data bytes to DS18B20.	3
Copy Scratchpad	Copies TH, TL, and configuration register data from the scratchpad to EEPROM.	48h	None	1
Recall E ²	Recalls TH, TL, and configuration register data from EEPROM to the scratchpad.	B8h	DS18B20 transmits recall status to master.	
Read Power Supply	Signals DS18B20 power supply mode to the master.	B4h	DS18B20 transmits supply status to master.	

Note 1: For parasitic-powered DS18B20s, the master must enable a strong pullup on the 1-Wire bus during temperature conversions and copies from the scratchpad to EEPROM. No other bus activity may take place during this time.

Note 2: The master can interrupt the transmission of data at any time by issuing a reset.

Note 3: All three bytes must be written before a reset is issued.

Configuration Register

Byte 4 of the scratchpad memory contains the configuration register, which is organized as illustrated in Figure 10. The user can set the conversion resolution of the DS18B20 using the R0 and R1 bits in this register as shown in Table 2. The power-up default of these bits is R0 = 1 and R1 = 1 (12-bit resolution). Note that there is a direct tradeoff between resolution and conversion time. Bit 7 and bits 0 to 4 in the configuration register are reserved for internal use by the device and cannot be overwritten.

CRC Generation

CRC bytes are provided as part of the DS18B20's 64-bit ROM code and in the 8th byte of the scratchpad memory. The ROM code CRC is calculated from the first 66 bits of the ROM code and is contained in the most significant byte of the ROM. The scratchpad CRC is calculated from the data stored in the scratchpad, and therefore it changes when the data in the scratchpad changes. The CRCs provide the bus master with a method of data validation when data is read from the DS18B20. To verify that data has been read correctly, the bus master must re-calculate the CRC from the received data and then compare the value to either the ROM code CRC (for ROM reads) or to the scratchpad CRC (for scratchpad reads). If the calculated CRC matches the read CRC, the data has been

received error free. The comparison of CRC values and the decision to continue with an operation are determined entirely by the bus master. There is no circuitry inside the DS18B20 that prevents a command sequence from proceeding if the DS18B20 CRC (ROM or scratchpad) does not match the value generated by the bus master.

The equivalent polynomial function of the CRC (ROM or scratchpad) is:

$$\text{CRC} = X^8 + X^6 + X^4 + 1$$

The bus master can re-calculate the CRC and compare it to the CRC values from the DS18B20 using the polynomial generator shown in Figure 11. This circuit consists of a shift register and XOR gates, and the shift register bits are initialized to 0. Starting with the least significant bit of the ROM code or the least significant bit of byte 0 in the scratchpad, one bit at a time should be shifted into the shift register. After shifting in the 50th bit from the ROM or the most significant bit of byte 7 from the scratchpad, the polynomial generator will contain the recalculated CRC. Next, the 8-bit ROM code or scratchpad CRC from the DS18B20 must be shifted into the circuit. At this point, if the re-calculated CRC was correct, the shift register will contain all 0s. Additional information about the Maxim 1-Wire cyclic redundancy check is available in Application Note 27: Understanding and Using Cyclic Redundancy Checks with Maxim 1-Wire Products.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0	R1	R0	1	1	1	1	1

Figure 10. Configuration Register

Table 2. Thermometer Resolution Configuration

R1	R0	RESOLUTION (BITS)	MAX CONVERSION TIME
0	0	9	93.75ms (t _{CONV9})
0	1	10	187.5ms (t _{CONV4})
1	0	11	375ms (t _{CONV2})
1	1	12	750ms (t _{CONV})

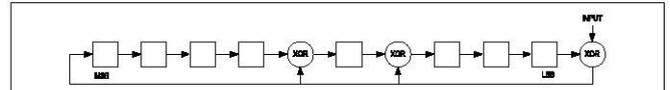


Figure 11. CRC Generator

If there is only one slave on the bus, the simpler Read ROM [33h] command can be used in place of the Search ROM procedure. For a detailed explanation of the Search ROM procedure, refer to Application Note 537: Book of Buttons® Standards. After every Search ROM cycle, the bus master must return to Step 1 (Initialization) in the transaction sequence.

Read Rom [33h]

This command can only be used when there is one slave on the bus. It allows the bus master to read the slave's 64-bit ROM code without using the Search ROM procedure. If the command is used when there is more than one slave present on the bus, a data collision will occur when all the slaves attempt to respond at the same time.

Match Rom [55h]

The match ROM command followed by a 64-bit ROM code sequence allows the bus master to address a specific slave device on a multidrop or single-drop bus. Only the slave that exactly matches the 64-bit ROM code sequence will respond to the function command issued by the master; all other slaves on the bus will wait for a reset pulse.

Skip Rom [CCh]

The master can use this command to address all devices on the bus simultaneously without sending out any ROM code information. For example, the master can make all DS18B20s on the bus perform simultaneous temperature conversions by issuing a Skip ROM command followed by a Convert T [44h] command.

Note that the Read Scratchpad [BEh] command can follow the Skip ROM command only if there is a single slave device on the bus. In this case, time is saved by allowing the master to read from the slave without sending the device's 64-bit ROM code. A Skip ROM command followed by a Read Scratchpad command will cause a data collision on the bus if there is more than one slave since multiple devices will attempt to transmit data simultaneously.

Alarm Search [ECh]

The operation of this command is identical to the operation of the Search ROM command except that only slaves with a set alarm flag will respond. This command allows the master device to determine if any DS18B20s experienced an alarm condition during the most recent temperature conversion. After every Alarm Search cycle (i.e., Alarm Search command followed by data exchange), the bus

master must return to Step 1 (Initialization) in the transaction sequence. See the Operation—Alarm Signaling section for an explanation of alarm flag operation.

DS18B20 Function Commands

After the bus master has used a ROM command to address the DS18B20 with which it wishes to communicate, the master can issue one of the DS18B20 function commands. These commands allow the master to write to and read from the DS18B20's scratchpad memory, initiate temperature conversions and determine the power supply mode. The DS18B20 function commands, which are described below, are summarized in Table 3 and illustrated by the flowchart in Figure 14.

Convert T [44h]

This command initiates a single temperature conversion. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18B20 returns to its low-power idle state. If the device is being used in parasitic power mode, within 10µs (max) after this command is issued the master must enable a strong pullup on the 1-Wire bus for the duration of the conversion (t_{CONV}) as described in the Powering the DS18B20 section. If the DS18B20 is powered by an external supply, the master can issue read time slots after the Convert T command and the DS18B20 will respond by transmitting a 0 while the temperature conversion is in progress and a 1 when the conversion is done. In parasitic power mode this notification technique cannot be used since the bus is pulled high by the strong pullup during the conversion.

Write Scratchpad [4Eh]

This command allows the master to write 3 bytes of data to the DS18B20's scratchpad. The first data byte is written into the TH register (byte 2 of the scratchpad), the second byte is written into the TL register (byte 3), and the third byte is written into the configuration register (byte 4). Data must be transmitted least significant bit first. All three bytes MUST be written before the master issues a read, or the data may be corrupted.

Read Scratchpad [BEh]

This command allows the master to read the contents of the scratchpad. The data transfer starts with the least significant bit of byte 0 and continues through the scratchpad until the 8th byte (byte 8 – CRC) is read. The master may issue a reset to terminate reading at any time if only part of the scratchpad data is needed.

DS18B20 Operation Example 1

In this example there are multiple DS18B20s on the bus and they are using parasite power. The bus master initializes a temperature conversion in a specific DS18B20 and then reads its scratchpad and recalculates the CRC to verify the data.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20s respond with presence pulse.
Tx	55h	Master issues Match ROM command.
Tx	64-bit ROM code	Master sends DS18B20 ROM code.
Tx	44h	Master issues Convert T command.
Tx	DQ line held high by strong pullup	Master applies strong pullup to DQ for the duration of the conversion (t _{CONV}).
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20s respond with presence pulse.
Tx	55h	Master issues Match ROM command.
Tx	64-bit ROM code	Master sends DS18B20 ROM code.
Tx	BEh	Master issues Read Scratchpad command.
Rx	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.

DS18B20 Operation Example 2

In this example there is only one DS18B20 on the bus and it is using parasite power. The master writes to the TH, TL, and configuration registers in the DS18B20 scratchpad and then reads the scratchpad and recalculates the CRC to verify the data. The master then copies the scratchpad contents to EEPROM.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20 responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	4Eh	Master issues Write Scratchpad command.
Tx	3 data bytes	Master sends three data bytes to scratchpad (TH, TL, and config).
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20 responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	BEh	Master issues Read Scratchpad command.
Rx	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20 responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	48h	Master issues Copy Scratchpad command.
Tx	DQ line held high by strong pullup	Master applies strong pullup to DQ for at least 10ms while copy operation is in progress.

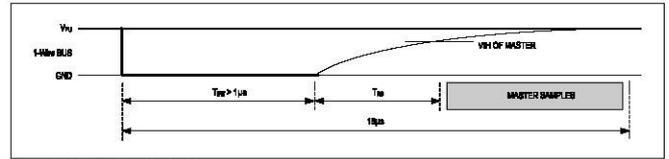


Figure 17. Detailed Master Read 1 Timing

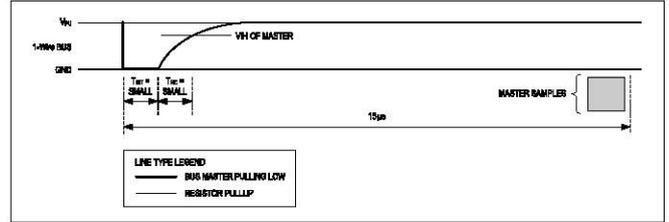


Figure 18. Recommended Master Read 1 Timing

Related Application Notes

The following application notes can be applied to the DS18B20 and are available at www.maximintegrated.com.

Application Note 27: Understanding and Using Cyclic Redundancy Checks with Maxim iButton Products

Application Note 122: Using Dallas' 1-Wire ICs in 1-Cell Li-Ion Battery Packs with Low-Side N-Channel Safety FETs Master

Application Note 126: 1-Wire Communication Through Software

Application Note 162: Interfacing the DS18B20/DS1822 1-Wire Temperature Sensor in a Microcontroller Environment

Application Note 208: Curve Fitting the Error of a Bandgap-Based Digital Temperature Sensor

Application Note 2420: 1-Wire Communication with a Microchip PICmicro Microcontroller

Application Note 3754: Single-Wire Serial Bus Carries Isolated Power and Data

Sample 1-Wire subroutines that can be used in conjunction with Application Note 74: Reading and Writing iButtons via Serial Interfaces can be downloaded from the Maxim website.

Revision History

REVISION DATE	DESCRIPTION	PAGES CHANGED
030107	In the Absolute Maximum Ratings section, removed the reflow oven temperature value of +220°C. Reference to JEDEC specification for reflow remains.	19
101207	In the Operation—Alarm Signaling section, added "or equal to" in the description for a TH alarm condition	5
	In the Memory section, removed incorrect text describing memory.	7
	In the Configuration Register section, removed incorrect text describing configuration register.	8
042208	In the Ordering Information table, added TO-92 straight-lead packages and included a note that the TO-92 package in tape and reel can be ordered with either formed or straight leads.	2
1/15	Updated Benefits and Features section	1

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE	TOP MARK
DS18B20	-55°C to +125°C	3 TO-92	18B20
DS18B20+	-55°C to +125°C	3 TO-92	18B20
DS18B20/T&R	-55°C to +125°C	3 TO-92 (2000 Piece)	18B20
DS18B20+T&R	-55°C to +125°C	3 TO-92 (2000 Piece)	18B20
DS18B20-SL/T&R	-55°C to +125°C	3 TO-92 (2000 Piece)*	18B20
DS18B20-SL+T&R	-55°C to +125°C	3 TO-92 (2000 Piece)*	18B20
DS18B20U	-55°C to +125°C	8 FSOP	18B20
DS18B20U+	-55°C to +125°C	8 FSOP	18B20
DS18B20U/T&R	-55°C to +125°C	8 FSOP (3000 Piece)	18B20
DS18B20U+T&R	-55°C to +125°C	8 FSOP (3000 Piece)	18B20
DS18B20Z	-55°C to +125°C	8 SO	DS18B20
DS18B20Z+	-55°C to +125°C	8 SO	DS18B20
DS18B20Z/T&R	-55°C to +125°C	8 SO (2500 Piece)	DS18B20
DS18B20Z+T&R	-55°C to +125°C	8 SO (2500 Piece)	DS18B20

*Denotes a lead-free package. A "+" will appear on the top mark of lead-free packages.

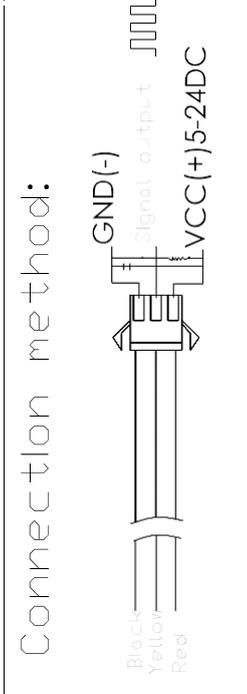
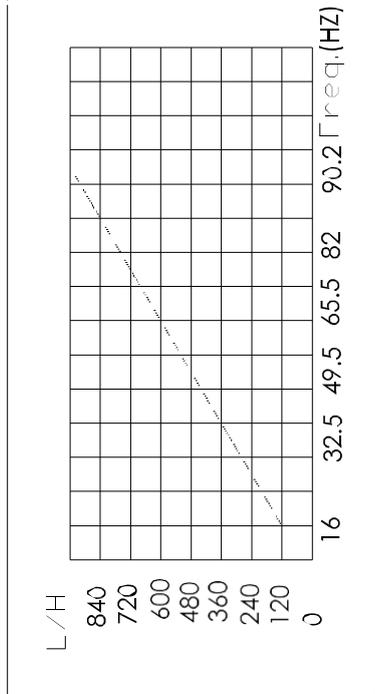
T&R = Tape and Reel.

*TO-92 packages in tape and reel can be ordered with straight or formed leads. Choose "SL" for straight leads. Bulk TO-92 orders are straight leads only.

Flow-Pulse	Flow-Curve	Connection method
------------	------------	-------------------

Flow Range: 100L/H-1800H-L/H

Flow (L/H)	Freqzq.(HZ)	Error range
120L/H	16	±10
240L/H	32.5	
360L/H	49.3	
480L/H	65.5	
600L/H	82	
720L/H	90.2	

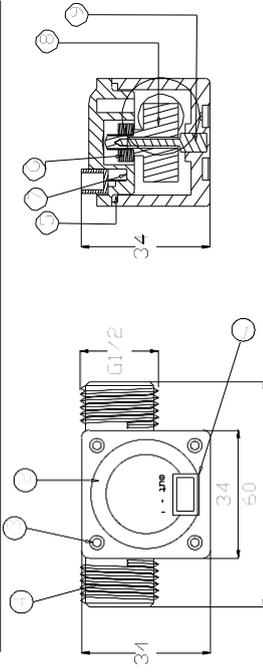


YIFA the plastics Ltd Product Introduction

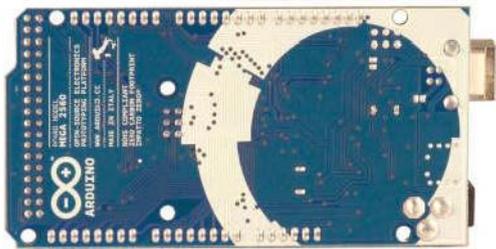
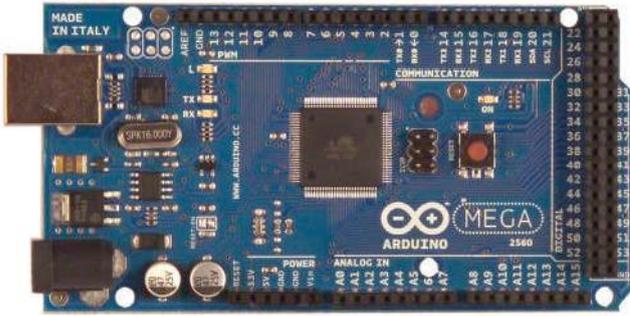
1. Model: YF-21
 2. Product Name: cell sensor
 3. Flow Range: 1-30L/MIN
 4. (1) Connection Method
-
- (2) Voltage Range: 3.5-24VDC, Pulse Characteristic: $F=7Q(L/MIN)$.
 (3) Extent of error: ±5%.
 (4) Flow-Pulse
 2L/MIN=16HZ 4L/MIN=32.5HZ 6L/MIN=49.3HZ
 8L/MIN=65.5HZ 10L/MIN=82HZ

5. BOM

No.	Item	Material	Qty.
1	Connection wire		1
2	Bonnet	PA	1
3	Screw		4
4	Valve body	PA	1
5	Leak press valve		1
6	Magnet		1
7	Hall		1
8	Impeller	POM	1
9	Rustless steel axis	SUS304	1
10			
11			



Arduino Mega 2560



Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Schematic & Reference Design

EAGLE files: [arduino-mega2560-reference-design.zip](#)

Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted into the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- ✦ **VIN**. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- ✦ **5V**. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- ✦ **3V3**. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- ✦ **GND**. Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- ✦ **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX)**. Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- ✦ **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2)**. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the `attachInterrupt()` function for details.
- ✦ **PWM: 0 to 13**. Provide 8-bit PWM output with the `analogWrite()` function.
- ✦ **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS)**. These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- ✦ **LED: 13**. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- ✦ **I²C: 20 (SDA) and 21 (SCL)**. Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and `analogReference()` function.

There are a couple of other pins on the board:

- ✦ **AREF**. Reference voltage for the analog inputs. Used with `analogReference()`.
- ✦ **Reset**. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I²C (TWI) and SPI communication. The Arduino software includes a [Wire library](#) to simplify use of the I²C bus; see the [documentation on the Wiring website](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available in the [Arduino repository](#). The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can

have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

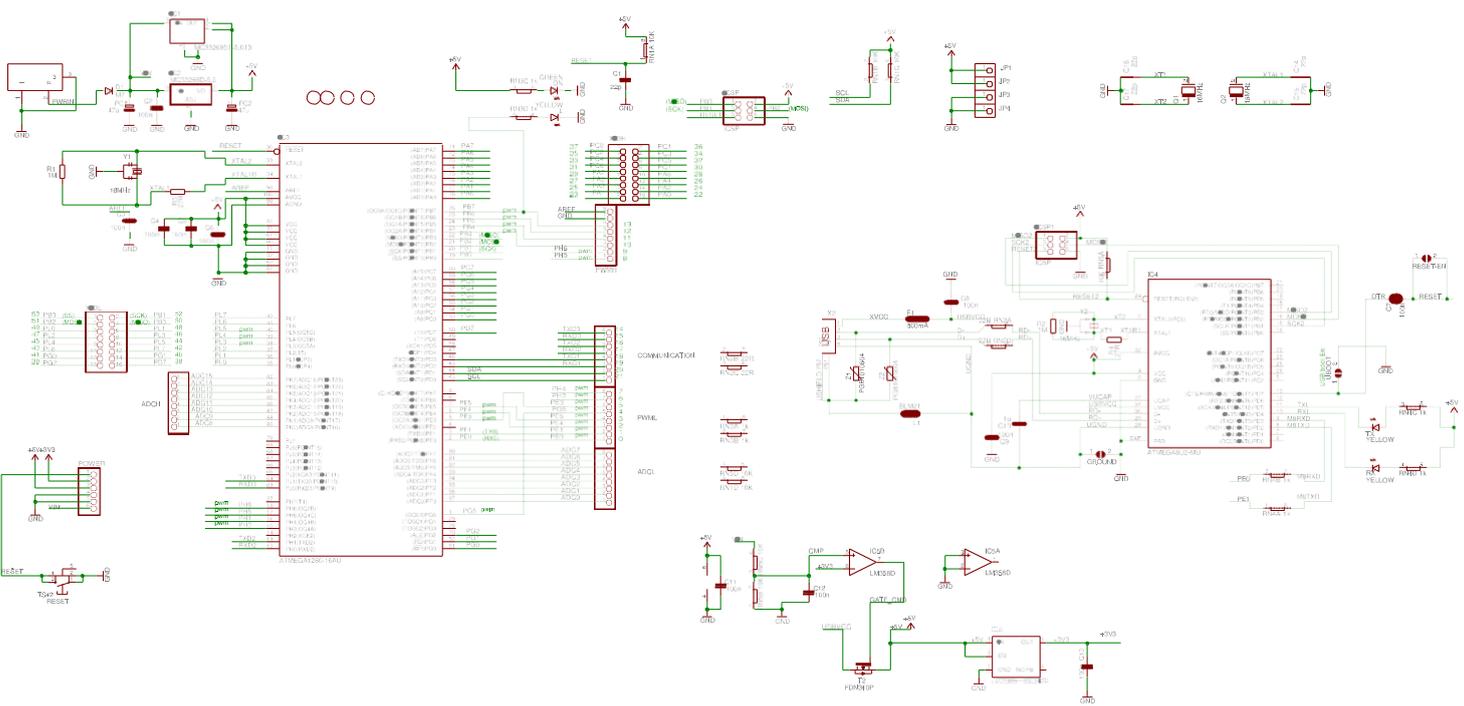
Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega2560 and Duemilanove / Diecimila. *Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).*

Arduino™ Mega 2560 Reference Design

Reference Design ART-PR-0017 © 2010 Autodesk, Inc. ALL RIGHTS RESERVED. ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, ARE DISCLAIMED. THIS DOCUMENT IS PROVIDED AS-IS. THE USER ASSUMES ALL LIABILITY FOR ANY DAMAGE TO PERSONS OR PROPERTY. THE USER ASSUMES ALL LIABILITY FOR ANY DAMAGE TO PERSONS OR PROPERTY. THE USER ASSUMES ALL LIABILITY FOR ANY DAMAGE TO PERSONS OR PROPERTY.





Product Overview

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.0) or ATmega168 (Arduino Nano 2.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one. The Nano was designed and is being produced by Gravitech.

Index

Technical Specifications

Page 2

How to use Arduino Programming Environment, Basic Tutorials

Page 6

Terms & Conditions

Page 7



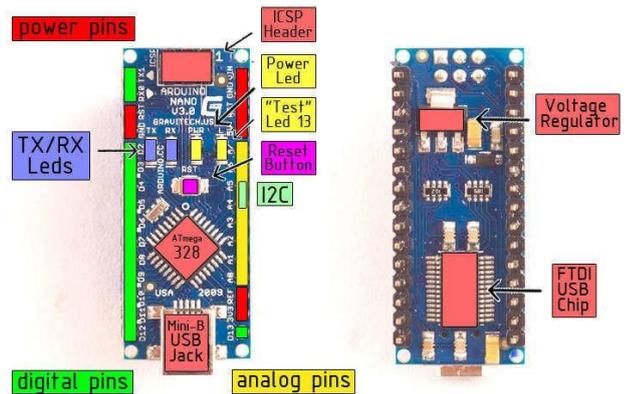
Arduino Nano 3.0 (ATmega328): [schematic](#), [Eagle files](#).

Arduino Nano 2.3 (ATmega168): [manual](#) (pdf), [Eagle files](#). Note: since the free version of Eagle does not handle more than 2 layers, and this version of the Nano is 4 layers, it is published here unrouted, so users can open and use it in the free version of Eagle.

Summary

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"

the board



Power

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

The FTDI FT232RL chip on the Nano is only powered if the board is being powered over USB. As a result, when running on external (non-USB) power, the 3.3V output (which is supplied by the FTDI chip) is not available and the RX and TX LEDs will flicker if digital pins 0 or 1 are high.

Memory

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader); the ATmega328 has 32 KB, (also with 2 KB used for the bootloader). The ATmega168 has 1 KB of SRAM and 512 bytes of EEPROM (which can be read and written with the [EEPROM library](#)); the ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

Input and Output

Each of the 14 digital pins on the Nano can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.
- External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e., 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- I²C: 4 (SDA) and 5 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the [Wiring website](#)).

There are a couple of other pins on the board:

- AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and ATmega168 ports](#).



Communication

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 and ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the [FTDI drivers](#) (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Nano's digital pins.

The ATmega168 and ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a [Wire library](#) to simplify use of the I2C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega168 or ATmega328 datasheet.

Programming

The Arduino Nano can be programmed with the Arduino software ([download](#)). Select "Arduino Duemilanove, or Nano w/ ATmega168" or "Arduino Duemilanove or Nano w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega168 or ATmega328 on the Arduino Nano comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Nano is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega168 or ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Nano is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Nano. While it is programmed to ignore malformed data (i.e., anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.





How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you'll need to install the FTDI Drivers to let your PC talk to the board. First **Plug the Arduino to your PC via USB cable.**

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

File>Sketchbook> Arduino-0017>Examples> Digital>Blink

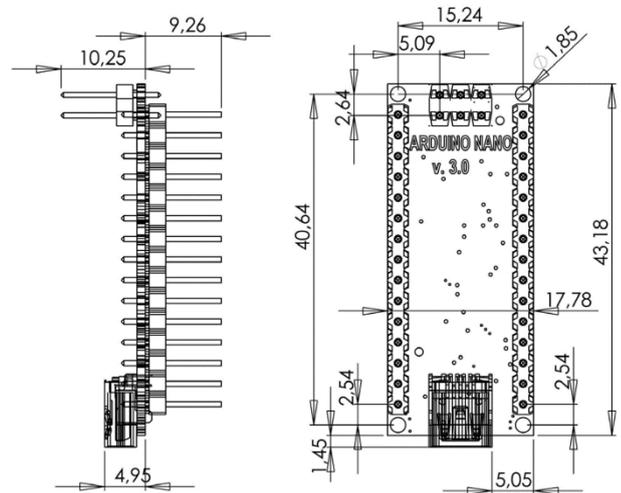
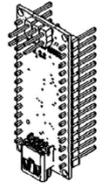
Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select Arduino NANO and with the AtMEGA you're using (probably 328)

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.



Dimensioned Drawing



Terms & Conditions



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, the producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

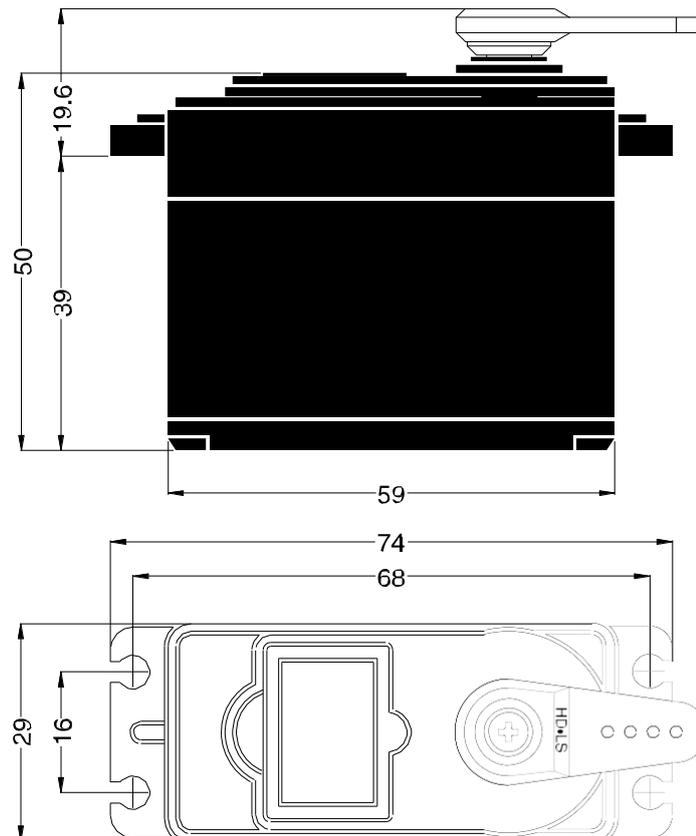
4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

GENERAL SPECIFICATION OF HS-755HB QUARTER SCALE SERVO

1. TECHNICAL VALUE

CONTROL SYSTEM	:+PULSE WIDTH CONTROL 1500usec NEUTRAL	
OPERATING VOLTAGE RANGE	:4.8V TO 6.0V	
OPERATING TEMPERATURE RANGE	:-20°C TO +60°C(-46°F TO +86°F)	
TEST VOLTAGE	:AT 4.8V	AT 6.0V
OPERATING SPEED	:0.28sec/60°IAT NO LOAD	0.23sec/60°I NO LOAD
STALL TORQUE	:11kg.cm(152.75oz.in)	13.2kg.cm(183.31oz.in)
STANDING TORQUE	:8.8kg.cm(122.20oz.in)/5°I HOLD OUT	10.5kg.cm(145.81oz.in)/5°I HOLD OUT
IDLE CURRENT	:8mA AT STOPPED	8mA AT STOPPED
RUNNING CURRENT	:230mA/60°I AT NO LOAD RUNNING	250mA/60°I AT NO LOAD RUNNING
STALL CURRENT	:1500mA	1800mA
DEAD BAND WIDTH	:5usec	5usec
OPERATING TRAVEL	:40°I ONE SIDE PULSE TRAVELING 400usec	
DIRECTION	:CLOCK WISE/PULSE TRAVELING 1500 TO 1900usec	
MOTOR TYPE	:CORED/METAL BRUSH	
POTENTIOMETER TYPE	:6 SLIDER/INDIRECT DRIVE	
AMPLIFIER TYPE	:ANALOG AMPLIFIER	
DIMENSIONS	:59x29x50mm(2.32x1.14x1.96in)	
WEIGHT	:110g(3.88oz)	
BALL BEARING	:DUAL/MR106	
GEAR MATERIAL	:HEAVY DUTY RESIN	
HORN GEAR SPLINE	:24 SEGMENTS/Ø5.76	
SPLINED HORNS	:HEAVY DUTY/HD-IS, HD-O, HD-LS, HD-IL, HD-LL	
CONNECTOR WIRE LENGTH	:300mm(11.81in)	
CONNECTOR WIRE STRAND COUNTER	:60EA	
CONNECTOR WIRE GAUGE	:22AWG	



2. FEATURES

- DURABLE HEAVY DUTY RESIN GEARS WITH BALL BEARINGS
- WATER & DUST TIGHT

3. APPLICATIONS

- FOR LARGE MODELS



16 x 4 Character LCD

- FEATURES**
- Type: Character
 - Display format: 16 x 4 characters
 - Built-in controller: ST 7026 (or equivalent)
 - Duty cycle: 1/16
 - 5 x 8 dot matrix character
 - +5 V power supply (also available for +3 V)
 - EL to be driven by pin 1, pin 2, pin 15, pin 16 or A and K
 - UV operation: -3 V power supply
 - Material categorization: For definitions of compliance please see www.vishay.com/doc/85512

MECHANICAL DATA

ITEM	STANDARD VALUE	UNIT
Module Dimension	70.0 x 92.0	
Viewing Area	60.0 x 70.0	
Dot Size	0.55 x 0.35	mm
Dot Pitch	0.60 x 0.60	
Mounting Hole	65.0 x 50.0	
Character Size	2.55 x 4.75	

ABSOLUTE MAXIMUM RATINGS

ITEM	SYMBOL	MIN.	TYP.	MAX.	UNIT
Power Supply	V _{DD} to V _{SS}	-0.3	-	7.0	V
Input Voltage	V _i	-0.3	-	V _{DD}	V

Note:

- V_{DD} = 0 V, V_{SS} = 5.1 V

DISPLAY CHARACTER ADDRESS CODE

Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
DD RAM Address	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61
DD RAM Address	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77
DD RAM Address	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93
DD RAM Address	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109
DD RAM Address	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125
DD RAM Address	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141
DD RAM Address	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157
DD RAM Address	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173
DD RAM Address	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189
DD RAM Address	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205
DD RAM Address	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221
DD RAM Address	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237
DD RAM Address	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253
DD RAM Address	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269
DD RAM Address	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285
DD RAM Address	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301
DD RAM Address	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317
DD RAM Address	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333
DD RAM Address	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349
DD RAM Address	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365
DD RAM Address	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381
DD RAM Address	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397
DD RAM Address	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413
DD RAM Address	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429
DD RAM Address	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445
DD RAM Address	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461
DD RAM Address	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477
DD RAM Address	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493
DD RAM Address	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509
DD RAM Address	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525
DD RAM Address	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541
DD RAM Address	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557
DD RAM Address	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573
DD RAM Address	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589
DD RAM Address	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605
DD RAM Address	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621
DD RAM Address	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637
DD RAM Address	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653
DD RAM Address	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669
DD RAM Address	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685
DD RAM Address	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701
DD RAM Address	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717
DD RAM Address	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733
DD RAM Address	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749
DD RAM Address	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765
DD RAM Address	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781
DD RAM Address	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797
DD RAM Address	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813
DD RAM Address	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829
DD RAM Address	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845
DD RAM Address	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861
DD RAM Address	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877
DD RAM Address	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893
DD RAM Address	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909
DD RAM Address	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925
DD RAM Address	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941
DD RAM Address	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957
DD RAM Address	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973
DD RAM Address	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989
DD RAM Address	990	991	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005
DD RAM Address	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021
DD RAM Address	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037
DD RAM Address	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053
DD RAM Address	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069
DD RAM Address	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085
DD RAM Address	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101
DD RAM Address	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117
DD RAM Address	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130			

TERMÓMETRO DIGITAL FLEXIBLE

INSTRUCCIONES DE USO.

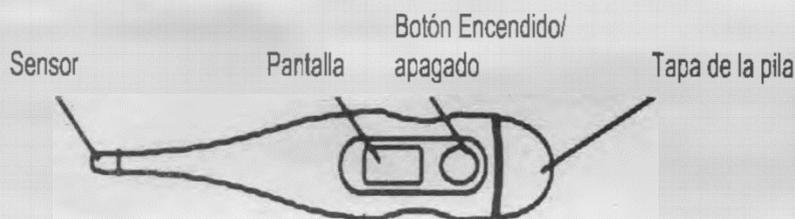
¿QUÉ ES UNA TEMPERATURA "NORMAL"?

Aunque generalmente se acepta como normal si la temperatura es 37.0 °C, la temperatura puede variar desde 36.1 °C a 37.2 °C y todavía puede ser considerada "normal". La variación de la temperatura puede ser causada por actividades tales como hacer ejercicio, fumar, comer e incluso beber. Se puede tener, también, una variación dependiendo en donde se tome la temperatura, por ejemplo la temperatura rectal es más alta que la oral. También hay que considerar que la temperatura del cuerpo es más baja por la mañana que por la tarde.

ESPECIFICACIONES

- Rango de medición: 32.0°C a 42.9°C.
- Resolución: 0.1°C.
- Precisión: ± 0.1°C (de 35.0°C a 39.0°C).
± 0.2°C en otro rango de medición.
- Pantalla temperatura baja: Si la temperatura es < 32.0 °C aparece L °C.
- Pantalla temperatura alta: Si la temperatura es ≥ 43.0 °C aparece H °C.
- Pila (incluida): 1.5 V. AG3 ó LR41
- Duración de la pila: 2 años.
- Auto apagado: Después de 8 minutos sin utilizar.
- Dimensiones: 128 mm x 19 mm x 10 mm
- Peso: Aproximadamente 9.8 gr (incluyendo la batería)
- Alarma de fiebre: Sonaran 30 pitidos cortos durante 10 segundos si la temperatura es > 37.8 °C.
- Alarma de fin de medición: Sonaran 10 pitidos largos durante 10 segundos si la temperatura es < 37.8 °C.
- Condiciones de medición: Temperatura entre 10 °C y 35 °C.
Humedad: entre el 30 % y el 85 % HR
- Condiciones de almacenamiento: Temperatura entre -10 °C y 60 °C.
Humedad: entre 25 % y el 90 % HR.

COMO UTILIZAR EL TERMÓMETRO



- Desinfectar el sensor antes del uso.
- Presione el botón de encendido; en la pantalla aparecerá 188.8
- Suelta el botón de encendido, se mostrará la temperatura tomada la última vez con una "M" en la esquina derecha de la pantalla.
- La pantalla mostrará L₀ con °C parpadeando, entonces el termómetro está listo para la medición. Si la habitación tiene una temperatura > 32.0 °C, se mostrará dicha temperatura en vez de L₀.
- Coloque el sensor en el lugar deseado para tomar la temperatura (oral, axilar o rectal)
- Cuando la medición haya finalizado sonará un pitido.
- El termómetro se apagará automáticamente pasados 8 minutos. Sin embargo es aconsejable apagar el termómetro después del uso para prolongar la duración de la batería.

USC ORAL

Coloque el sensor del termómetro debajo de la lengua en contacto con el tejido de la boca; mantenga la boca cerrada mientras dure la medición.

Tiempo aproximado de medición: 30 segundos.

Sugerencia: mantenga la boca cerrada 2 minutos antes de la medición para una toma de temperatura más segura.

FUENTE DE PODER ATX

DEFINICIÓN DE FUENTE ATX

ATX son las siglas de ("Advanced Technology eXtended") a tecnología avanzada extendida, que es una segunda generación de fuentes de alimentación introducidas al mercado para computadoras con microprocesador intel® Pentium MMX, y a partir de ese momento, se extiende su uso.

La fuente ATX es un dispositivo que se acopla internamente en el gabinete de la computadora, el cual se encarga básicamente de transformar la corriente alterna de la línea eléctrica comercial en corriente directa, así como reducir su voltaje. Esta corriente es utilizada por los elementos electrónicos y eléctricos de la computadora. Otras funciones son las de suministrar la cantidad de corriente y voltaje que los dispositivos requieren así como protegerlos de problemas en el suministro eléctrico como subidas de voltaje. A la fuente ATX se le puede llamar fuente de poder ATX, fuente de alimentación ATX, fuente digital, fuente de encendido digital, fuentes de pulsador, entre otros nombres.

ATX es el estándar actual de fuentes que sustituyeron a las [fuentes de alimentación AT](#).

CARACTERÍSTICAS GENERALES DE LA FUENTE ATX

- Es de encendido digital, es decir, tiene un pulsador en lugar de un interruptor mecánico como sus antecesoras.
- Algunos modelos integran un interruptor mecánico trasero para evitar consumo innecesario de energía eléctrica, evitando el estado de reposo "Stand By" durante el cual consumen cantidades mínimas de electricidad.
- Este tipo de fuentes se integran desde los equipos con microprocesador intel®, Pentium, MMX hasta los equipos con los más modernos microprocesadores.
- El apagado de este tipo de fuentes puede ser manipulado con [software](#).

PARTES QUE COMPONEN LA FUENTE ATX

Internamente cuenta con una serie de circuitos encargados de transformar la electricidad para que esta sea suministrada de manera correcta a los dispositivos. Externamente consta de los siguientes elementos:

- 1.- Ventilador: ayuda al aire caliente del interior de la fuente y del gabinete, para mantener frescos los circuitos.
- 2.- Interruptor de seguridad: permite encender la fuente de manera mecánica.
- 3.- Conector de alimentación: recibe el cable de corriente desde el outlet de pared.
- 4.- Selector de voltaje: permite seleccionar el voltaje de 127V o 240V.
- 5.- Conector SATA: utilizado para alimentar los discos duros y las unidades ópticas tipo SATA.
- 6.- Conector de 4 terminales: utilizado para alimentar de manera directa al microprocesador.
- 7.- Conector ATX: alimenta de electricidad a la tarjeta principal.
- 8.- Conector de 4 terminales MOLEX: utilizado para alimentar las unidades ópticas.
- 9.- Conector de 4 terminales IDE: alimenta las disquetes.
- 10.- Conector de 4 terminales IDE: alimenta las disquetes.

CONECTORES DE LA FUENTE ATX PINOUT

Para alimentarse, tiene un conector de 3 contactos, esto a su vez recibe alimentación desde la red eléctrica doméstica.

Para alimentar cuenta con básicamente 4 tipos de conectores:

- Para unidades de 3.5" ([disquetes](#)) y [unidades para discos ZIP](#)
- Para unidades de 5.25" ([unidades lectoras de CD](#), [unidades para DVD](#)).
- Para alimentar la tarjeta principal.
- Para alimentar unidades SATA/SATA 2 ([discos duros SATA](#) y [unidades para DVD SATA](#)).

Conector ATX versión 2 (20 terminales)	Interconecta la fuente ATX con la tarjeta principal (MotherBoard)		1. Negro (+3.3V) 2. Negro (+3.3V) 3. Negro (Tierra) 4. Rojo (+5 Volts) 5. Negro (Tierra) 6. Rojo (+5 Volts) 7. Negro (Tierra) 8. Gris (Power Good) 9. Púrpura (+5VSB) 10. Amarillo (+12V) 11. Amarillo (+12V) 12. Naranja (+3.3V)	13. Naranja (+3.3V) 14. Azul (+12 V) 15. Negro (Tierra) 16. Verde (Power On) 17. Negro (Tierra) 18. Negro (Tierra) 19. Negro (Tierra) 20. Blanco (-5V) 21. Rojo (+5 Volts) 22. Rojo (+5 Volts) 23. Rojo (+5 Volts) 24. Negro (Tierra)
Conector Data procesador de 4 terminales	Alimenta a los procesadores equios		1. Negro (Tierra) 2. Negro (Tierra)	3. Amarillo (+12V) 4. Amarillo (+12V)
Conector Pata (6 y 8 terminales)	Alimenta directamente los tarjetas de video tipo PCI		1. Negro (Tierra) 2. Negro (Tierra) 3. Negro (Tierra) 4. Negro (Tierra)	5. Amarillo (+12V) 6. Amarillo (+12V) 7. Amarillo (+12V) 8. Amarillo (+12V)

FUENTES SLI / XFire

Las tecnologías SLI/XFire implementadas en las [tarjetas de video](#), requieren un alto consumo de energía eléctrica, por lo que la MotherBoard ya no es un medio efectivo para transmitir la electricidad necesaria para alimentarlas, por ello se han integrado conexiones directas entre la fuente ATX y las tarjetas de video tipo PCI-E. Cabe mencionar que no es necesario el uso de fuentes especiales con estos conectores, ya que se han introducido al mercado adaptadores que permiten a cualquier fuente ATX, pueda alimentar las tarjetas de video mencionadas.

FUENTES ATX EXTERNAS

Algunos equipos debido a su reducido tamaño, tienen la necesidad de recibir la alimentación por medio de fuentes externas tipo de adaptador AC/DC, diseñadas en específico para la marca y el modelo. Estas tienen un conector especial hacia el gabinete y por lo regular son muy resistentes, cable rebobado y selladas, ya que están expuestas a líquidos, movimientos bruscos, caídas etc. Un ejemplo de ello es la fuente que alimenta los modelos de computadora de la serie GX620 / SX620 de la alimentación para equipos de la marca DELL® Optiplex GX620.

POTENCIA DE LA FUENTE ATX

Las fuentes ATX comerciales manejan potencia eléctrica de 300 Watts (W), 350 W, 400 W, 480 W, 500 W, 530 W, 1200 W y hasta 1350 W. Repasando algunos términos de electricidad, recordemos que la electricidad no es otra cosa más que electrones

circulando a través de un medio conductor. La potencia eléctrica de una fuente ATX se mide en Watts (W) y esta variable está en función de otros dos factores:

<ul style="list-style-type: none"> El voltaje es el fuerza con que son impulsados los electrones a través de un medio conductor. El voltaje se mide en Volts (V) en nuestro caso es de 127 V. La corriente es la cantidad de electrones que circular por un punto en específico cada segundo. Su unidad de medida es Amperes (A). 	<p>Ejemplo: si una fuente ATX indica que es de 400 W entonces:</p> <p>Potencia eléctrica = Voltaje X Corriente , $W = V \times A$</p> <p>Sabemos que el voltaje es de 127 V y tenemos los Watts, así que despejamos la corriente:</p> $A = W / V \quad , \quad A = 400 W / 127 V \quad , \quad A = 3.4$ <p>Entonces, lo que interesa es la cantidad de corriente que puede suministrar la fuente, porque a mayor cantidad de corriente, habrá mayor potencia y podrá alimentar una mayor cantidad de dispositivos. En este caso es de 3.4 Amperes.</p>
--	---

FUNCIONAMIENTO DE UNA FUENTE ATX

En el siguiente esta se muestran diferentes etapas por las que la electricidad es transformada para alimentar los dispositivos de una computadora. Si gustas conocer más sobre electricidad, consulta nuestra sección: [Electricidad básica](#).

CONECTOR	DISPOSITIVOS	IMAGEN DE CONECTOR	ESQUEMA	LÍNEAS DE ALIMENTACIÓN
Tipo MOLEX	Dispositivos de 5.25" Unidades ópticas de 5.25" ATAPI y discos duros de 3.5" IDE			1.- Red +5V (Alimentación +5 Volts) 2.- Black GND (Tierra) 3.- Black GND (Tierra) 4.- Yellow +12V (Alimentación +12Volts)
Tipo IDE	Dispositivos de 3.5"			1.- Red +5V (Alimentación +5 Volts) 2.- Black GND (Tierra) 3.- Black GND (Tierra) 4.- Yellow +12V (Alimentación +12Volts)
Tipo SATA / SATA 2	Discos duros 3.5" SATA / SATA 2			1.- V11 (+3.3 Volts) 2.- V25 (+3.3 Volts) 3.- V31 (+3.3 Volts) 4.- GND (Tierra) 5.- GND (Tierra) 6.- GND (Tierra) 7.- V5 (+5 Volts) 8.- V5 (+5 Volts) 9.- V5 (+5 Volts) 10.- GND (Tierra) 11.- Invertido 12.- GND (Tierra) 13.- V12 (+12 Volts) 14.- V12 (+12 Volts) 15.- V12 (+12 Volts)

- 1.- Transformación: el voltaje de la línea eléctrica comercial se reduce como ejemplo de 127 Volts a aproximadamente 12 Volts y 5 V. Utiliza un elemento electrónico llamado transformador.
- 2.- Rectificación: se transforma el voltaje de corriente alterna en voltaje de corriente directa, esto lo hace dejando pasar solo los valores positivos de la onda (se genera corriente continua), por medio de elementos electrónicos llamados diodos.
- 3.- Filtrado: esta le da calidad a la corriente continua y suaviza el voltaje, por medio de elementos electrónicos llamados capacitores.
- 4.- Estabilización: el voltaje ya suavizado se le da la forma lineal que utilizan los dispositivos. Se usa un elemento electrónico especial llamado circuito integrado. Esta fase es la que entrega la energía necesaria a la computadora.

USOS ESPECÍFICOS

Se utilizan para suministrar energía eléctrica necesaria para el correcto funcionamiento de los dispositivos, encontrándose en gabinetes, horcones, gabinetes, monitores y torres de computadoras. Dependiendo de la cantidad de dispositivos a alimentar, deberá ser mayor la capacidad de la fuente. Actualmente todos los equipos modernos tienen una fuente de alimentación ATX, de igual modo los sistemas operativos son capaces de controlar las fuentes ATX internamente: apagar el sistema desde el

Conector ATX versión 1 (20 terminales + 4)	Interconecta la fuente ATX con la tarjeta principal (MotherBoard)		1. Naranja (+3.3V) 2. Naranja (+3.3V) 3. Negro (Tierra) 4. Rojo (+5 Volts) 5. Negro (Tierra) 6. Rojo (+5 Volts) 7. Negro (Tierra) 8. Gris (Power Good) 9. Púrpura (+5VSB) 10. Amarillo (+12V) 11. Amarillo (+12V) 12. Naranja (+3.3V) 13. Negro (Tierra) 14. Rojo (+5V)	11. Naranja (+3.3V) 12. Azul (+12 V) 13. Negro (Tierra) 14. Negro (Tierra) 15. Verde (Power On) 16. Negro (Tierra) 17. Negro (Tierra) 18. Blanco (-5V) 19. Rojo (+5 Volts) 20. Rojo (+5 Volts)
--	---	--	--	---

botón "Inicio" de Microsoft® Windows, se cerraba el sistema y se quedaba en pantalla un mensaje de apagar el sistema desde el botón mecánico del gabinete.

ANEXO B Manual de instalación grifo monomando E108.05/B1



E108.05/B1

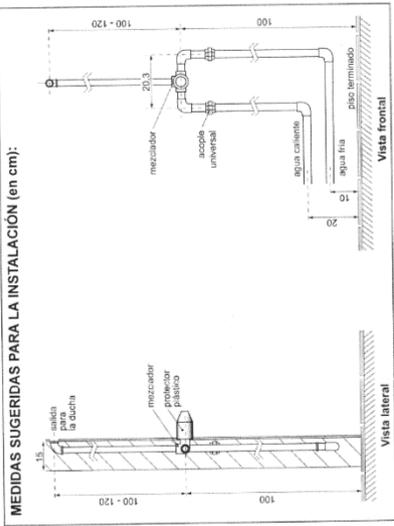
LEA Y CONSERVE ESTA INFORMACIÓN:

Antes de instalar la grifería, deseje muy bien la tubería para evitar fugas, goteos, ruidos o eliminar impurezas o residuos de construcción.

RECOMENDACIONES:

Asegúrese que la tubería está bien vista e instalarla con un profesional en la manija, para prevenir errores que afecten el funcionamiento de la grifería. Preferiblemente, ajuste las partes visibles de la grifería SIN UTILIZAR herramientas demarcadas en el manual de producto, jalen la superficie y al instalarla.

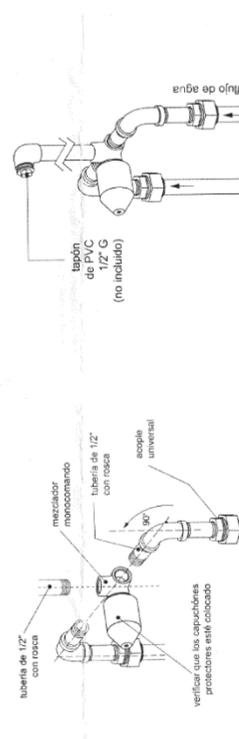
En caso de que sea necesario, utilizar herramientas propias al acabado de la grifería con un recorte de caucho (de tubo de barra) colocado entre los dientes de la herramienta y la grifería para ajustar sin dañarlos. Nuestras Manijas Plásticas AL PULGÓN están diseñadas para ser instaladas en el agua en este hoyo.



MEDIDAS SUGERIDAS PARA LA INSTALACIÓN (en cm):

1. Instalación del mezclador:

Instalación de tuberías

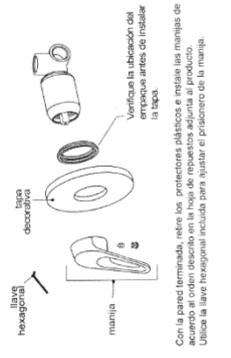
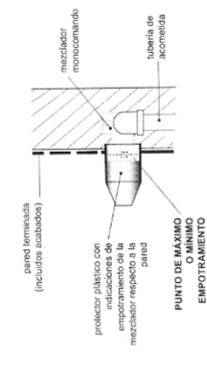


En las uniones, utilice sellantes de fibra, evite los que se cristalizan. En la salida de agua de la ducha coloque un tapón de PVC de 1/2" G (no incluido). No permita que el agua se acumule en la tubería. Si el agua no está en su lugar (a su sellante líquido espere 24 horas para que adquiere su máxima adherencia antes de realizar la prueba). Mantenga el tapón para evitar que residuos de construcción ingresen en la tubería, confirme el paralelismo vertical de la tubería y la horizontalidad de mezclador, realice el sellado de la pared y coloque el revestimiento.

Para las uniones entre el mezclador y la tubería utilizar un sellante apropiado para evitar fugas.

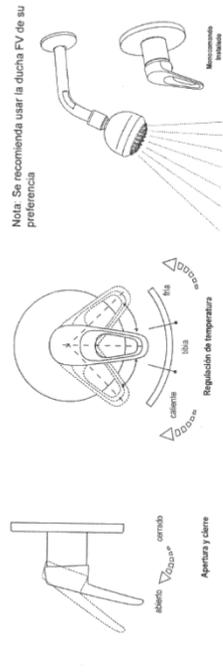
Las tuberías de agua fría y caliente deben estar verticales, paralelas y quedar al mismo nivel para que el mezclador se instale horizontalmente.

3. Máximo y mínimo empotramiento:



Con la pared terminada, retire los protectores plásticos e instale las manijas de la manija. Verifique la ubicación del tapón decorativo antes de instalar la manija. Utilice la llave hexagonal incluida para ajustar el prisionero de la manija.

FUNCIONAMIENTO:



F.V. le ofrece una diversa gama de brazos y cabezas de ducha para complementar este producto.

SUGERENCIAS DE MANTENIMIENTO:

CARTUCHOS CERÁMICOS:
La precisión básica es la limpieza de las partes de alimentación para evitar la acumulación de residuos que lleguen a dañar a los cartuchos cerámicos. No presione más de la cuenta.
El cartucho cerámico debe ser revisado únicamente por personas capacitadas, caso contrario la F. Viegener se responsabilizará por daños ocasionados.

AIREDADORES - CORTACORROS - FILTROS DE DUCHA:
Con el uso normal se acumulan impurezas en el interior de estos componentes. Para evitar esto, revise los aeradores (localizados del lado o debajo de la salida de agua) y limpie el producto según las siguientes sugerencias:
• Nunca use materiales de limpieza que contengan ácidos clorhídrico, sulfúrico, fosfórico o sulfónico.
• Nunca mezcle varios tipos de sustancias para realizar la limpieza.
• Nunca use materiales de limpieza o aparatos con efecto abrasivo, como FV-Magic (ref.: EL-002.0 DH), esponjas, estropajo, ampolletas, cepillos de la estropajo o estropajo metálicos.
• Nunca use disolventes como el thinner sobre piezas acrílicas porque disminuye su resistencia y provoca su rotura.
• Si usa limpiadores en spray, rocíe primero hacia una tela suave o esponja y después rocíe directamente hacia la grifería.
• Después de limpiar, enjuague completamente con agua limpia para quitar cualquier residuo.
• Séquela inmediatamente utilizando un paño de algodón suave y limpio que no raye la superficie.
• Frote la superficie de la grifería con cera FV-Magic (ref.: EL-002.0 DH), y así se evitará la acumulación de residuos y reflete usando un paño suave de algodón. No deje la cera en la grifería por tiempo prolongado.
• La limpieza regular con FV-Magic previene las manchas y calcificación.
LA GARANTÍA NO CUBRE DAÑOS CAUSADOS POR EL TRATAMIENTO INADECUADO DE LA SUPERFICIE DE LA GRIFERÍA.