

# Sistema de monitoreo de temperatura

Especificaciones

V1.0 — Enero 2026

Erika Garcia

Diseño Digital II

# Índice

<b>Indice</b>	<b>2</b>
<b>1. Introducción</b>	<b>3</b>
<b>2. Especificaciones del Sistema</b>	<b>3</b>
2.1. Parámetros de Operación . . . . .	3
2.1.1. Rango de Temperatura y Escalado . . . . .	3
2.1.2. Resolución y Precisión . . . . .	3
2.2. Lógica de Control . . . . .	4
2.2.1. Estados del Sistema . . . . .	4
2.2.2. Condiciones de Transición . . . . .	4
2.2.3. Lógica de persistencia . . . . .	4
2.2.4. Lógica de alerta automática . . . . .	4
<b>3. Arquitectura general</b>	<b>5</b>
3.1. Microarquitectura . . . . .	5
3.2. Jerarquía de diseño . . . . .	5
3.3. Puertos del sistema . . . . .	6
3.4. Descripción de Módulos . . . . .	6
3.4.1. Módulo: comparador_temp . . . . .	6
3.4.2. Módulo: persistencia_ctr . . . . .	6
3.4.3. Módulo: estado_temp (FSM) . . . . .	7
<b>4. Verificación</b>	<b>8</b>
4.1. Estrategia de verificación . . . . .	8
4.1.1. Cobertura de estados . . . . .	8
4.1.2. Estados verificable . . . . .	8
4.2. Testbench . . . . .	8
4.3. SystemVerilog Assertions (SVA) . . . . .	9
4.4. Plan de Pruebas . . . . .	9
4.5. Cobertura Funcional (Covergroups) . . . . .	9
4.5.1. Cobertura de Rangos de Temperatura . . . . .	9
4.5.2. Cobertura de Estados de la FSM . . . . .	9
4.5.3. Cobertura de Contadores . . . . .	10
4.5.4. Cobertura de Actuadores . . . . .	10

4.5.5. Cobertura de Transiciones de la FSM . . . . .	10
4.5.6. Cover Properties . . . . .	10
4.6. Resultados de Simulación . . . . .	10
4.6.1. Métricas de Cobertura . . . . .	10
4.6.2. Análisis de Resultados . . . . .	11
<b>5. Análisis y discusión</b>	<b>11</b>
<b>6. Conclusiones</b>	<b>11</b>
<b>7. Apéndices</b>	<b>11</b>
7.1. comparador_temp . . . . .	11

## Índice de figuras

1. Microarquitectura del sistema de monitoreo de temperatura . . . . .	5
2. Cobertura del sistema . . . . .	11

## Índice de cuadros

1. Rangos operativos del sistema . . . . .	3
2. Estados de operación del sistema . . . . .	4
3. Puertos del sistema . . . . .	6
4. Cobertura de estados . . . . .	8
5. Plan de pruebas implementado . . . . .	9

# 1. Introducción

Este documento presenta las especificaciones de un sistema digital de monitoreo de temperatura con arquitectura jerárquica de FSM, implementado en SystemVerilog para aplicaciones de control térmico en invernaderos y sistemas embebidos. EL propósito es:

- Proporcionar los requisitos del sistema.
- Describir la arquitectura implementada (procesamiento, temporización, control).
- Especificar los estados de la máquina de estados finitos (FSM) y condiciones de transición.
- Documentar el mecanismo de contador de persistencia para filtrado de transitorios.
- Proporcionar la estructura de RTL y metodología de verificación.

## 2. Especificaciones del Sistema

### 2.1. Parámetros de Operación

#### 2.1.1. Rango de Temperatura y Escalado

El sistema opera en el rango típico de sensores comerciales de temperatura de propósito general (-40.0°C a 85.0°C), con una resolución de 0.1°C, utilizando escalado  $\times 10$  para convertir a valores enteros (-400 a 850).

Los umbrales implementados en el diseño están específicamente enfocados para monitoreo ambiental en invernaderos, donde temperaturas inferiores a 18.0°C o superiores a 25.9°C pueden afectar el crecimiento vegetal.

$$\text{Valor procesado} = \text{Temperatura (°C)} \times 10$$

Condición	Rango Real (°C)	Rango Escalado
BAJO	$T < 18,0$	$T < 180$
NORMAL	$18,0 \leq T \leq 25,9$	$180 \leq T \leq 259$
ALTO	$T > 25,9$	$T > 259$

Cuadro 1: Rangos operativos del sistema

#### 2.1.2. Resolución y Precisión

- **Resolución:** 0.1°C
- **Representación:** 11 bits con signo en complemento a 2
- **Ancho de entrada:** 11 bits (rango -1024 a 1023)
- **Rango de interés:** -400 a 850

## 2.2. Lógica de Control

### 2.2.1. Estados del Sistema

Estado	Código	Calefactor	Ventilador	Alerta
NORMAL	00	0	0	0
BAJA	01	0	0	0
ALTO	10	0	0	0
ALERTA	11	Dependiente de $T$	Dependiente de $T$	1

Cuadro 2: Estados de operación del sistema

Nota: En el estado ALERTA es donde se toma la decisión: si  $T < 180$  se activa Calefactor=1, si  $T > 259$  se activa Ventilador=1.

### 2.2.2. Condiciones de Transición

- **NORMAL a BAJA**: Cuando temperatura  $< 180$  ( $18.0^{\circ}\text{C}$ )
- **NORMAL a ALTO**: Cuando temperatura  $> 259$  ( $25.9^{\circ}\text{C}$ )
- **BAJA a NORMAL**: Cuando temperatura retorna a  $[180, 259]$
- **ALTO a NORMAL**: Cuando temperatura retorna a  $[180, 259]$
- **BAJO a ALERTA**: Después de N ciclos en estado BAJA
- **ALTO a ALERTA**: Después de N ciclos en estado ALTO
- **ALERTA a NORMAL**: Cuando temperatura retorna a rango normal

### 2.2.3. Lógica de persistencia

- **Propósito**: Filtrar transitorios breves del sensor
- **Valor por defecto**:  $N = 5$  ciclos de reloj
- **Contador**: Se implementa dos contadores independientes, con el objetivo de separar la persistencia en temperatura baja y alta. Incrementa mientras la temperatura está fuera de rango normal
- **Reset**: Cuando temperatura vuelve a rango normal

### 2.2.4. Lógica de alerta automática

#### Características:

- Alerta se activa cuando la condición de temperatura baja o alto se mantiene durante N ciclos consecutivos, descartando eventos transitorios aislados.
- Se desactiva automáticamente al volver a temperatura normal.
- La activación de actuadores es específica: el calefactor solo opera bajo persistencia de frío y el ventilador bajo persistencia de calor.

### 3. Arquitectura general

#### 3.1. Microarquitectura

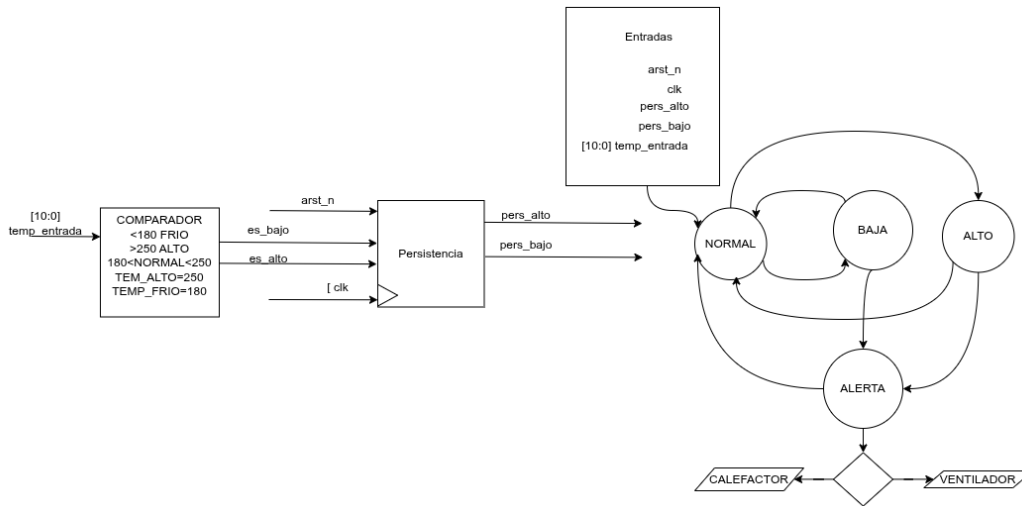


Figura 1: Microarquitectura del sistema de monitoreo de temperatura

#### 3.2. Jerarquía de diseño

El proyecto se divide en arquitectura RTL y el entorno de verificación.

##### Arquitectura RTL (Register Transfer Level)

Compuesta por módulos en SystemVerilog que definen el comportamiento lógico del sistema de monitoreo de temperatura:

- **monitoreo\_top.sv:** Módulo que integra todos los componentes RTL.
- **comparador\_temp.sv:** Evalúa la temperatura y genera las señales `es_bajo` y `es_alto`.
- **persistencia\_ctr.sv:** Implementa la lógica de dos contadores para persistencia de frío y calor.
- **estado\_temp.sv:** Máquina de estados finitos (FSM) que gestiona la lógica de control.

##### Entorno de Verificación

Compuesto por módulos dedicados a la validación y cobertura del diseño:

- **monitoreo\_pkg.sv:** Package que contiene las definiciones de clases para generación de estímulos y parámetros de configuración, asegurando la consistencia de datos.
- **interface\_monitoreo.sv:** Encapsula las señales del DUT (Device Under Test), facilitando la conexión con el testbench y permitiendo el monitoreo de señales internas.

- **monitoreo.tb.sv**: Testbench principal que implementa múltiples casos de prueba, incluyendo pruebas de persistencia, transiciones, límites y valores extremos.
- **fv\_monitoreo.sv**: Módulo de verificación formal que contiene aserciones (SVA) para validar propiedades críticas del sistema como exclusión mutua, persistencia y seguridad de actuadores.
- **cov\_monitoreo.sv**: Implementa covergroups para medir la cobertura funcional, incluyendo estados de la FSM, transiciones, rangos de temperatura y valores de contadores.

### 3.3. Puertos del sistema

Puerto	Dirección	Ancho	Descripción
clk	Entrada	1 bit	Reloj del sistema
arst_n	Entrada	1 bit	Reset asíncrono activo en bajo
temp_entrada	Entrada	11 bits	Temperatura escalada del sensor ( $T \times 10$ )
alerta	Salida	1 bit	Indicador de condición de alerta
calefactor	Salida	1 bit	Activación del sistema de calefacción
ventilador	Salida	1 bit	Activación del sistema de ventilación
estado_actual	Salida	2 bits	Estado actual de la FSM
cont_bajo	Salida	3 bits	Contador de persistencia de frío (0-5)
cont_alto	Salida	3 bits	Contador de persistencia de calor (0-5)
per_bajo	Salida	1 bit	Señal de persistencia de frío alcanzada
per_alto	Salida	1 bit	Señal de persistencia de calor alcanzada

Cuadro 3: Puertos del sistema

### 3.4. Descripción de Módulos

#### 3.4.1. Módulo: comparador\_temp

Se encarga de determinar si la temperatura de entrada se encuentra por debajo del umbral de frío o por encima del umbral de calor. Implementa lógica combinacional. Se tienen dos parámetros internos escalados: TEMP\_BAJO fijado en 180 y TEMP\_ALTO en 259. El módulo genera dos señales de salida de un solo bit, es\_bajo y es\_alto, que actúan como indicadores de zona de peligro. La señal es\_bajo se activa de forma asíncrona siempre que la entrada sea estrictamente inferior al límite de 18.0 °C, mientras que es\_alto cuando la temperatura supera el umbral de 25.9 °C.

```
assign es_bajo = (temp_entrada < TEMP_BAJO);
assign es_alto = (temp_entrada > TEMP_ALTO);
```

#### 3.4.2. Módulo: persistencia\_ctr

El módulo persistencia\_ctr actúa como un filtro para eliminar la sensibilidad del sistema ante ruidos eléctricos o fluctuaciones térmicas transitorias del sensor. Su arquitectura se fundamenta en el uso de dos contadores síncronos independientes, cont\_bajo y cont\_alto, ambos de 3 bits, controlados por un parámetro configurable  $N = 5$ . El funcionamiento del módulo se rige por el flanco de subida del reloj (clk) y un reset asíncrono (arst\_n) que garantiza un estado inicial nulo. Mientras la señal es\_bajo se mantenga activa, el contador respectivo incrementa su valor secuencialmente hasta alcanzar el límite  $N$ . Si es\_bajo se desactiva (la temperatura sale de la zona de frío), el contador se resetea inmediatamente a cero. El comportamiento del contador de calor es similar al anterior.

Cuando cualquiera de ellos alcanza este valor  $N$ , se activa la señal de persistencia correspondiente (`per_bajo` o `per_alto`). Estas señales se generan de forma combinacional, lo que permite que la FSM reaccione en el mismo ciclo en que se cumple la condición de persistencia.

```

if (es_bajo) begin
    if (cont_bajo < N) cont_bajo <= cont_bajo + 1;
end else begin
    cont_bajo <= 3'd0;
end

assign per_bajo = (cont_bajo >= N-1) && es_bajo;

```

### 3.4.3. Módulo: estado\_temp (FSM)

El módulo `estado_temp` constituye la unidad de control central del sistema, implementando una Máquina de Estados Finitos (FSM) de tipo Moore-Mealy que gestiona la lógica de alerta y la activación de actuadores. La FSM está definida con cuatro estados operativos: NORMAL (00), BAJO (01), ALTO (10) y ALERTA (11). El diseño utiliza un enfoque síncrono para las transiciones de estado, regido por el flanco de subida de `clk`, y cuenta con un reset asíncrono (`arst_n`) que fuerza el retorno inmediato al estado NORMAL, garantizando un punto de partida seguro. La FSM permite transiciones directas entre BAJO y ALTO, si la temperatura cruza los umbrales antes de completar el ciclo de persistencia, de modo que, si en estado ALERTA se pierde la persistencia pero la temperatura sigue fuera de rango, el sistema retroceda a los estados intermedios (BAJO o ALTO) para re-evaluar la condición.

La lógica de salida de los actuadores se implementa de forma combinacional para asegurar una respuesta inmediata:

- **alerta:** Se activa mediante asignación continua cuando la FSM está en estado ALERTA.
- **calefactor:** Se activa solo en estado ALERTA cuando `temp_registrado < 180` y haya alcanzado la persistencia en bajo.
- **ventilador:** Se activa solo en estado ALERTA cuando `temp_registrado > 259` y haya alcanzado la persistencia en alto.

```

always_comb begin
    calefactor = 1'b0;
    ventilador = 1'b0;

    if (estado == ALERTA) begin
        if (temp_registrado < TEMP_BAJO)
            calefactor = 1'b1;
        else if (temp_registrado > TEMP_ALTO)
            ventilador = 1'b1;
    end
end

assign alerta = (estado == ALERTA);

```



## 4. Verificación

### 4.1. Estrategia de verificación

#### 4.1.1. Cobertura de estados

Estado	Entradas	Salidas	Ciclos
NORMAL	Temperatura [180,259]	calef=0, vent=0, alerta=0	por definir
BAJO	Temperatura < 180	transición a ALERTA	por definir
ALTO	Temperatura > 259	transición a ALERTA	por definir
ALERTA	Contador $\geq$ N	alerta=1: si $T < 180$ calefactor =1, si $T > 259$ ventilador =1	por definir

Cuadro 4: Cobertura de estados

#### 4.1.2. Estados verificable

1. NORMAL  $\rightarrow$  BAJO ( $T < 180$ )
2. NORMAL  $\rightarrow$  ALTO ( $T > 259$ )
3. BAJO  $\rightarrow$  NORMAL ( $T$  vuelve a [180,259])
4. ALTO  $\rightarrow$  NORMAL ( $T$  vuelve a [180,259])
5. BAJO  $\rightarrow$  ALERTA (contador  $\geq$  N)
6. ALTO  $\rightarrow$  ALERTA (contador  $\geq$  N)
7. ALERTA  $\rightarrow$  NORMAL ( $T$  vuelve a rango)
8. BAJO  $\rightarrow$  ALTO (cambio directo  $T > 259$ )
9. ALTO  $\rightarrow$  BAJO (cambio directo  $T < 180$ )

### 4.2. Testbench

- Arquitectura del testbench - Secuencias de prueba

### 4.3. SystemVerilog Assertions (SVA)

### 4.4. Plan de Pruebas

Caso	Nombre	Descripción	Estímulos	Resultado
1	T_NORMAL_B	Normal básico	4 valores normales	Verificado
2	T_BAJO_INM	Frío inmediato	4 valores bajos	Verificado
3	T_ALTO_INM	Calor inmediato	4 valores altos	Verificado
4	T_BAJO_PERS	Frío persistente	5 ciclos ¡180	Verificado
5	T_ALTO_PERS	Calor persistente	5 ciclos ¡259	Verificado
6	T_RECU_AUTO	Recuperación BAJO→NORMAL	5 frío + 5 normal	PASS
7	T_BAJO_ALTO	Transición BAJO→ALTO	5 frío + calor	Verificado
8	T_ALTO_BAJO	Transición ALTO→BAJO	5 calor + frío	Verificado
9	T_TRANS_ALTO	Transitorio calor	3 calor + normal	Verificado
10	T_TRANS_BAJO	Transitorio frío	3 frío + normal	Verificado
11	T_LIM_BAJO	Límite 179→180	179, 180	Verificado
12	T_LIM_ALTO	Límite 259→260	259, 260	Verificado
13	LIMITE_INFERIOR	Valores extremos negativos	-1024 a -401	PASS
14	LIMITE_SUPERIOR	Valores extremos positivos	851 a 1023	PASS
15	RESET_SISTEMA	Reset en diferentes estados	Reset en NORMAL y ALERTA	PASS

Cuadro 5: Plan de pruebas implementado

### 4.5. Cobertura Funcional (Covergroups)

#### 4.5.1. Cobertura de Rangos de Temperatura

```
cp_temp: coverpoint temp_entrada {  
    bins min_sensor      = { MIN_SENSOR };  
    bins rango_bajo      = { [MIN_SENSOR+1 : TEMP_BAJO-2] };  
    bins limite_bajo     = { TEMP_BAJO - 1 };  
    bins limite_normal_inf = { TEMP_BAJO };  
    bins rango_normal    = { [TEMP_BAJO+1 : TEMP_ALTO-1] };  
    bins limite_normal_sup = { TEMP_ALTO };  
    bins limite_alto     = { TEMP_ALTO + 1 };  
    bins rango_alto      = { [TEMP_ALTO+2 : MAX_SENSOR-1] };  
    bins max_sensor      = { MAX_SENSOR };  
}
```

#### 4.5.2. Cobertura de Estados de la FSM

```
cp_estado: coverpoint estado_actual {  
    bins NORMAL = {2'b00};  
    bins BAJO   = {2'b01};  
    bins ALTO   = {2'b10};  
    bins ALERTA = {2'b11};  
    illegal_bins otros = default;  
}
```

#### 4.5.3. Cobertura de Contadores

```
cp_contador_bajo: coverpoint cont_bajo {  
    bins cero      = {0};  
    bins uno       = {1};  
    bins dos       = {2};  
    bins tres      = {3};  
    bins cuatro    = {4};  
    bins cinco     = {5};  
    bins otros     = {[6:7]};  
}
```

#### 4.5.4. Cobertura de Actuadores

```
cp_actuadores: coverpoint {calefactor, ventilador} {  
    bins ambos_off = {2'b00};  
    bins solo_calefactor = {2'b10};  
    bins solo_ventilador = {2'b01};  
    illegal_bins ambos_on = {2'b11};  
}
```

#### 4.5.5. Cobertura de Transiciones de la FSM

```
cp_fsm_trans: coverpoint estado_actual {  
    bins normal_a_bajo = (2'b00 => 2'b01);  
    bins normal_a_alto = (2'b00 => 2'b10);  
    bins bajo_a_normal = (2'b01 => 2'b00);  
    bins alto_a_normal = (2'b10 => 2'b00);  
    bins bajo_a_alerta = (2'b01 => 2'b11);  
    bins alto_a_alerta = (2'b10 => 2'b11);  
    bins recu_auto      = (2'b11 => 2'b00);  
    bins transitorio_bajo = (2'b00 => 2'b01 => 2'b00);  
    bins transitorio_alto = (2'b00 => 2'b10 => 2'b00);  
}
```

#### 4.5.6. Cover Properties

```
'COV(mon, exclusion_mutua_visto, 1'b1 |->, (!(ventilador && calefactor)))  
'COV(mon, alerta_activada, 1'b1 |->, (alerta == 1))  
'COV(mon, cont_bajo_cinco_alcanzado, 1'b1 |->, (cont_bajo == 5))  
'COV(mon, cont_alto_cinco_alcanzado, 1'b1 |->, (cont_alto == 5))  
'COV(mon, estado_alerta_alcanzado, 1'b1 |->, (estado_actual == 2'b11))
```

### 4.6. Resultados de Simulación

#### 4.6.1. Métricas de Cobertura

.....

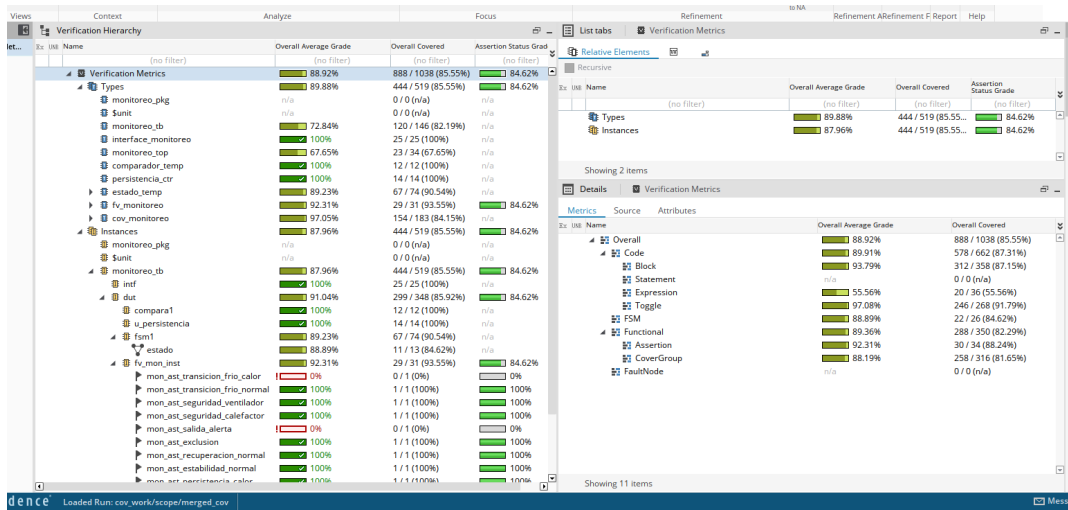


Figura 2: Cobertura del sistema

#### 4.6.2. Análisis de Resultados

[Texto explicando los resultados obtenidos y áreas de mejora]

## 5. Análisis y discusión

## 6. Conclusiones

Logros del Proyecto Posibles Extensiones

## 7. Apéndices

### Apéndice A

#### 7.1. comparador\_temp

Listing 1: Unidad de Control