# ESPECIFICACIÓN Y DISEÑO

Low Level Design. Horno de incubación de bacterias

## Arquitectura basada en eventos

Se ha realizado el diseño detallado mediante la construcción de una máquina de estados, que se especificará y comentará a continuación. Se ha extendido la arquitectura cliente-servidor adoptando una arquitectura basada en eventos, en la que HW1 actúa como cliente, y HW2 actúa como servidor. En esta arquitectura, la captura, la comunicación, el procesamiento y la permanencia de los eventos son la estructura central de la solución. Esto difiere del modelo tradicional Cliente-Servidor, basado en solicitudes. (“¿Qué es la arquitectura basada en eventos?,” 2021).

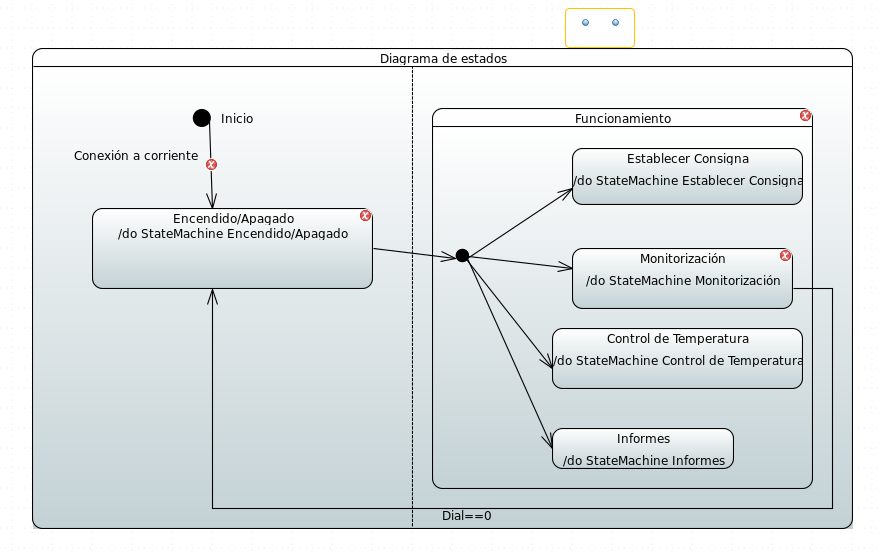
Se ha tomado como referencia el capítulo 11 del libro “A practical guide to sysml”, en el que se explica como modelar una arquitectura basada en eventos mediante el uso de máquinas de estados. La descripción que se ofrece en esta guía, correspondiente a una maquina de estados basada en eventos, cumple los requisitos de nuestro sistema y parece ajustarse de manera óptima a la metodología escogida para describir dicho sistema

“*State machines are typically used in SysML to describe the state-dependent behaviour of a block throughout its lifecycle, which is defined in terms of its states and the transitions between them. A state machine for a block may start, for example, when it initiates power up, then transition through multiple states in response to different stimuli, and terminate when it completes power down. The state machine defines how the block’s behaviour changes as it transitions between different states and while the block is in different states. “*

(Friedenthal, S., Moore, A., & Steiner, R., 2014)

En este modelo se especifican las distintas máquinas de estados mediante el uso de diagramas en los que se incluyen regiones, que son zonas en las que se agrupan varios estados de los que solo puede estar uno activo al mismo tiempo, estados, que representan una condición sobre la ejecución que está llevando a cabo el bloque, y transiciones, que especifican cuando se realizará un cambio de estado. También existen pseudoestados, usados para especificar acciones a realizar en la transición (junction pseudostate), y para decidir que transición realizar cuando hay varias alternativas (choice pseudostate).

## Comentario general

Se procede a realizar un comentario sobre el modelado del sistema usando la arquitectura antes explicada. Se ha seguido el ejemplo del libro “A practical guide to sysml…” (Friedenthal, S., Moore, A., & Steiner, R., 2014) ubicado en la página 286, correspondiente a la figura 11.0.

En esta máquina de estados, se muestran las distintas submáquinas que describirán el comportamiento del programa, y como está estructurado su flujo de ejecución.

Podemos observar cómo después del estado inicial, se llega en primera instancia al estado Encendido/Apagado, descrito por la máquina de estados homónima. Tras haber realizado las operaciones correspondientes a dicha máquina de estados, se transita a una segunda región en la que se ejecutarán distintos estados de forma concurrente.

Cada uno de los estados tiene su periodo de activación determinado, que, por motivos de claridad de imagen, no han sido incluidos en la captura adjunta del diagrama. Estos estados describirán el funcionamiento del sistema en su régimen general, mediante lo que la ya citada guía SysML llama “submachine state”.

## Encendido/Apagado

En este diagrama se muestran las distintas acciones que realizarán en el sistema de encendido/apagado.

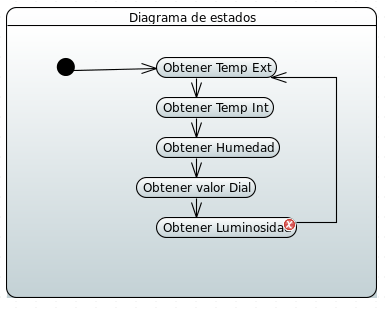
Podemos observar que tras el estado inicial hay una submáquina de estados que realizará dos acciones. La primera será mostrar un mensaje de bienvenida por el display, y la segunda realizará una espera de 3 segundos. Tras esto, se llega a la inicialización de periféricos, otra submáquina de estados que se encargará de inicializar todos los periféricos necesarios para que el sistema funcione correctamente. Primero se inicializará la USART, seguido por el timer, el PWM, el LED, la célula peltier y por ultimo el conversor analógico digital.

Una vez inicializados los periféricos, se comprobará si el dial está a 0 por si se debe apagar el sistema.

## Establecer Consigna

Este diagrama muestra las acciones que se realizarán en el sistema de establecer consigna. Se puede ver que el sistema es un bucle, por lo que estará funcionando mientras el sistema se encuentre activo.

## Monitorización



En este diagrama se muestran las distintas adquisiciones de datos que sea realizarán mediante el uso del conversor analógico digital. Este periférico ya se encuentra inicializado por lo que solo será necesario acceder a las distintas entradas analógicas mediante la multiplexión, respetando el tiempo de carga del condensador, ya que de otra manera correríamos el riesgo de contaminar los valores leídos con los antiguos valores.

Estos valores serán introducidos en variables globales, que permitan el acceso desde otros procesos como el de control de temperatura o establecimiento de consigna.

Antes de ser introducidos en las variables globales, los valores adquiridos, en milivoltios, serán convertidos a los valores que representan en la unidad correspondiente a la medida tomada. Se usarán las fórmulas indicadas en el diagrama en lenguaje natural como comentario.

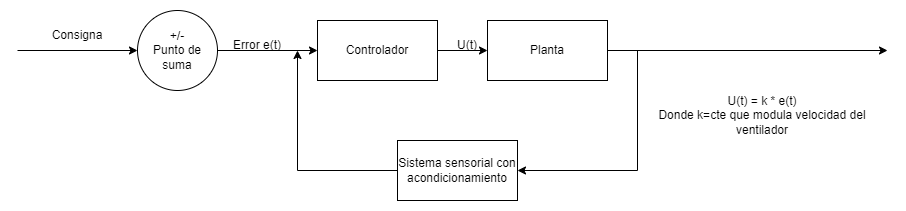
Así, por ejemplo, para monitorizar el valor de temperatura exterior, se tomará el valor devuelto por el sensor en mV, mediante el uso del CAD y luego se aplicará la fórmula

(SensorTemp)

Será el valor Temp el que quedará guardado en la variable global TemperaturaExterior

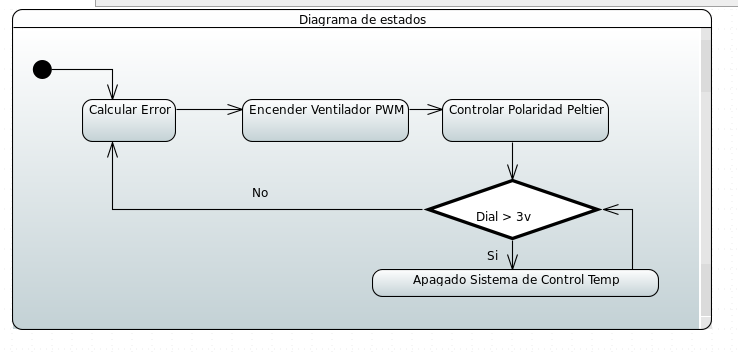
## Control Temperatura

Se hará un control de temperatura siguiendo un modelo de control de lazo cerrado, (con retroalimentación) en el que el error se calculará mediante la diferencia del valor “Temperatura Interior” y el valor “Consigna”.



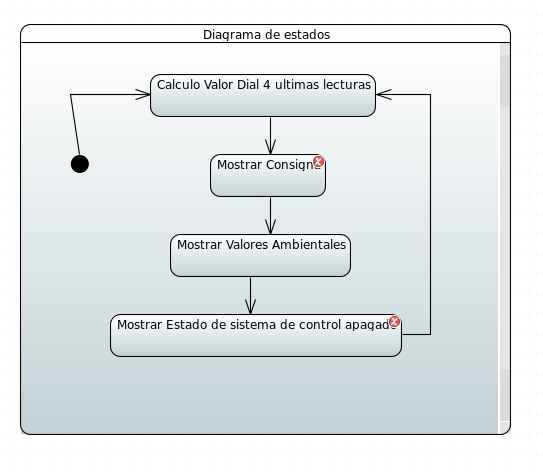
Esquema de sistema de control de lazo cerrado a implementar. Fuente propia

Tal y como podemos ver en el diagrama superior, en el punto de suma se calcula el error calculando la diferencia entre la consigna establecida, que es un valor de entrada, y la temperatura adquirida en el sistema sensorial. El controlador, determinará las acciones a tomar para reducir el error. En nuestro caso estas acciones serán; 1) Controlar la polaridad de la célula Peltier. 2) Controlar la intensidad con la que se activará el ventilador PWM. En Planta, se llevarán a cabo las acciones determinadas por Controlador. Después de haber realizado dichas acciones se volverá a evaluar la temperatura y calcular el error.



En la imagen superior podemos ver la máquina de estados correspondiente al diagrama antes explicado. Como comprobación adicional, se comprobará el estado del dial. Siguiendo las especificaciones, si el voltaje devuelto por el dial es mayor a 3V, el sistema de control de temperatura quedará apagado. Esta comprobación será realizada en el pseudoestado indicado, que cerrará el lazo en caso de que el voltaje sea menor a 3V y el sistema tenga que seguir funcionando.

## Informes



## Bibliografía

¿Qué es la arquitectura basada en eventos? (2021). Retrieved November 28, 2021, from Redhat.com website: <https://www.redhat.com/es/topics/integration/what-is-event-driven-architecture>.

Friedenthal, S., Moore, A., & Steiner, R. (2014). A practical guide to sysml : The systems modeling language. Elsevier Science & Technology.