

Proyecto Final

Taller de Sistemas Embebidos.
Primer Cuatrimestre de 2025.

MÁQUINA EXPENDEDORA DE BEBIDAS.

Grupo 4:

- | | |
|----------------------------|--------|
| • Aguilar Espinoza, Isabel | 106539 |
| • Cazas Telias, Lucas | 105776 |
| • Piñeiro, Oriana | 105489 |
| • Zambrano, Amddy | 106540 |



Introducción al Proyecto

Este proyecto se centra en el diseño y la construcción de un prototipo para una máquina expendedora de bebidas. El sistema está gobernado por una placa de desarrollo NUCLEO-F103RB y cuenta con una interfaz de usuario que incluye una pantalla LCD y botones para la selección e interacción.

Características Principales:

- **Controlador:** Placa NUCLEO-F103RB
- **Interfaz de Usuario:** Pantalla LCD y botones.
- **Funcionalidades:**
 - Modo Cliente: Selección de 4 bebidas.
 - Modo Configuración: Gestión de precios y stock.
 - Detección de monedas (simulada mediante un botón).
 - Indicador de estado a través de un LED RGB.

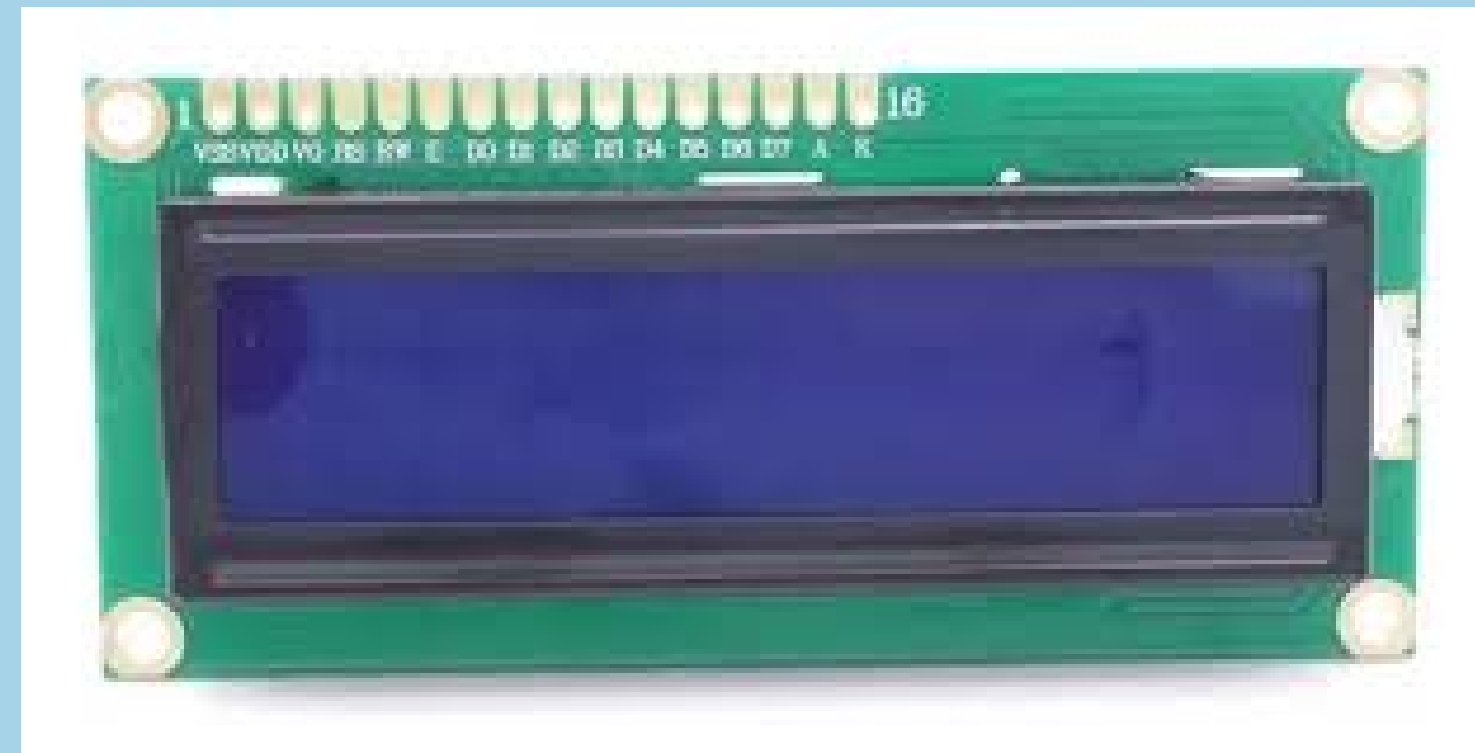
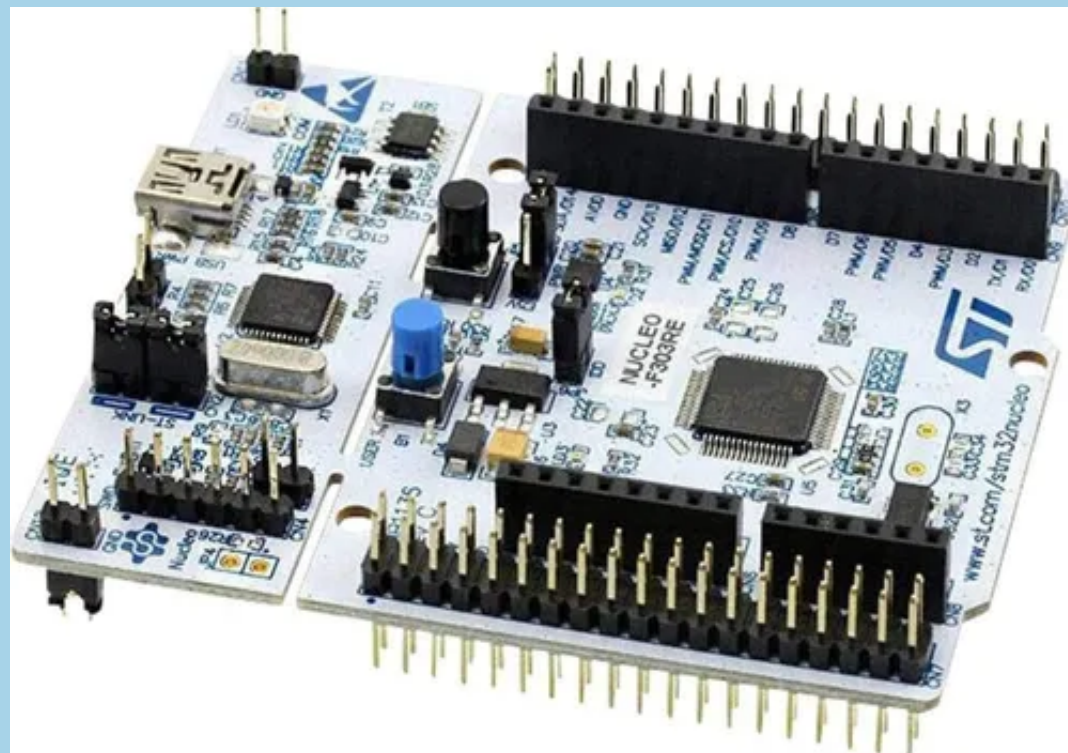


Arquitectura del Hardware

La placa NUCLEO-F103RB funciona como la unidad de control principal del sistema, gestionando la interconexión con los diversos componentes periféricos para la operación de la máquina expendedora.

Componentes Claves:

- **Placa NUCLEO-F103RB:** Encargado de procesar la lógica de la máquina.
- **Display LCD (16x2):** Proporciona información al usuario, como la bebida seleccionada, el precio y los estados de la máquina.



- **LED RGB:** Ofrece una indicación visual del estado del sistema (Entrega de la bebida seleccionada).
- **Botones:** Permiten la interacción del usuario (cliente y técnico) para navegar por los menús, seleccionar bebidas y confirmar acciones.
- **Sensor de Temperatura (LM35):** Monitorea la temperatura de refrigeración de las bebidas. En caso de que la temperatura no sea la adecuada, el sistema entra en modo error.
- **Sensor de Monedas:** Se simula a través de un botón (COIN) para registrar el ingreso de crédito.

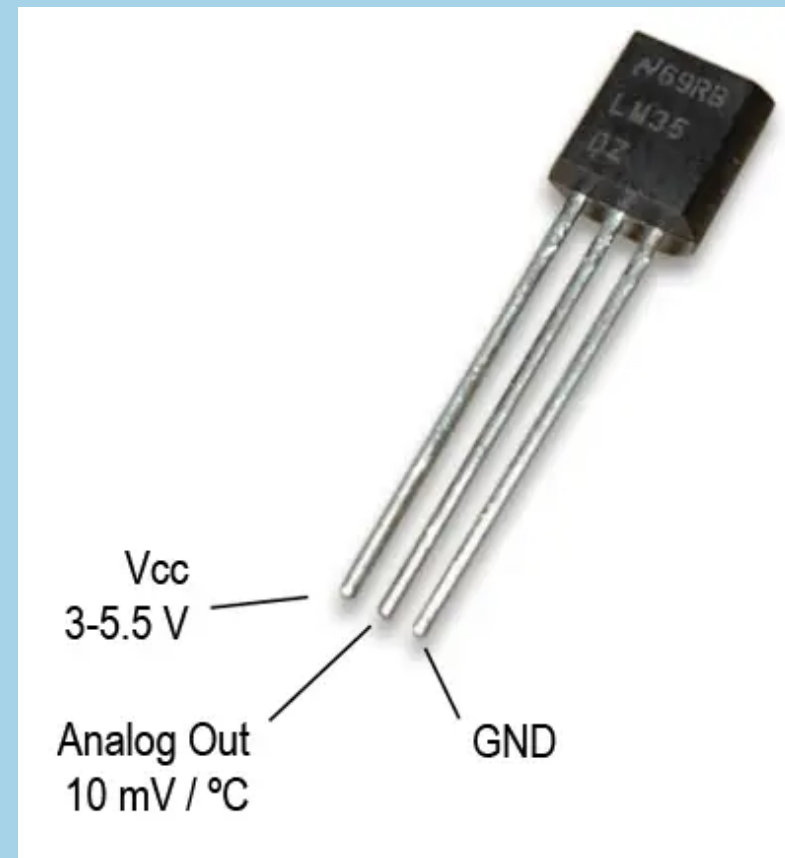
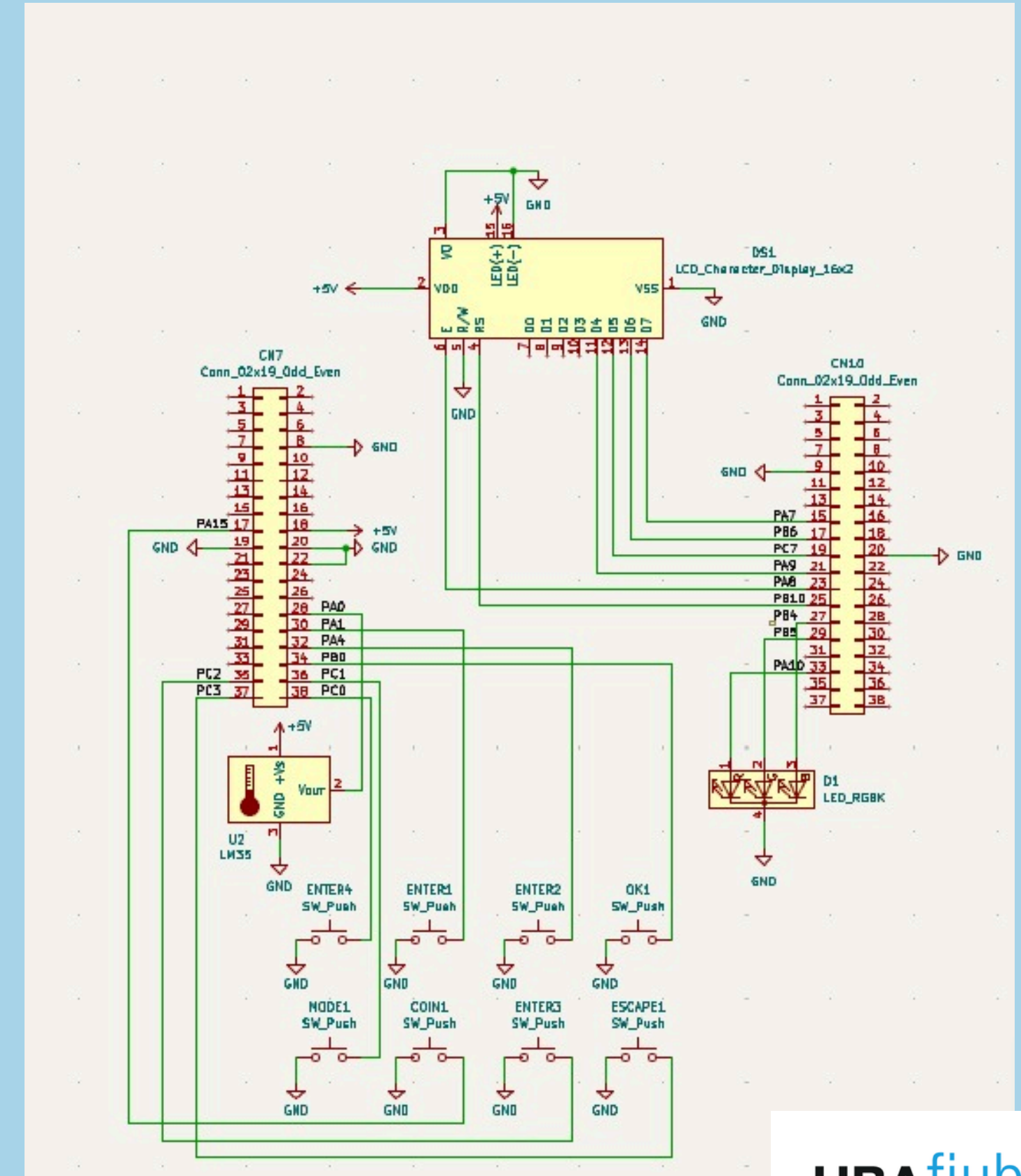


Diagrama de Conexiones (Pinout)

La siguiente tabla detalla las conexiones entre la placa NUCLEO y los periféricos.

El diagrama adjunto a la derecha ilustra cómo se interconectan los componentes

Placa NUCLEO	Pin Arduino	Pin del componente	Componente
GND		GND	LED
PA10	D2	B	LED
PB4	D5	G	LED
PB5	D4	R	LED
GND		VSS	Display
5V		VDD	Display
GND		V0	Display
PB10	D6	RS	Display
GND		RW	Display
PA8	D7	E	Display
PA9	D8	D4	Display
PC7	D9	D5	Display
PB6	D10	D6	Display
PA7	D11	D7	Display
5V		A	Display
GND		K	Display
PC0	A5	ENTER4	Botones
PC1	A4	MODE	Botones
PB0	A3	OK	Botones
PA1	A1	ENTER1	Botones
PA4	A2	ENTER2	Botones
PC3		ESCAPE	Botones
PA15		COIN	Botones
PC2		ENTER3	Botones
GND		GND	Botones
GND		GND	Sensor
5V		Vs	Sensor
PA0	A0	Vout	Sensor



Diseño del Software: Máquina de Estados

El firmware de la máquina expendedora se ha estructurado en torno a un modelo de máquina de estados para gestionar la interacción con el usuario de una manera ordenada y predecible. Se utilizó **Itemis CREATE** para simular el Diagrama de Estados.

Menú de Usuario

User Menu - State Transition Table				
Current State	Event	[Guard]	Next State	Actions
MAIN_USER	EV_ENTER1		DRINK	drink_number = 0
	EV_ENTER2		DRINK	drink_number = 1
	EV_ENTER3		DRINK	drink_number = 2
	EV_ENTER4		DRINK	drink_number = 3
	EV_COIN		MAIN_USER	coins++
DRINK	EV_ESCAPE		MAIN_USER	drink_number = 0
	EV_OK	[coins >= price[drink_number] && stock[drink_number] > 0]	MAIN_USER	stock[drink_number]-- ; coins = coins - price[drink_number] ; drink_number = 0
	EV_COIN		DRINK	coins++
	EV_MODE		MAIN_CONFIG	drink_number = 0
	EV_TEMP_ERROR		MAIN_ERROR	drink_number = 0

Menú de Error

Error Menu - State Transition Table				
Current State	Event	[Guard]	Next State	Actions
MAIN_ERROR	EV_TEMP_OK		MAIN_USER	

Menú de configuración

Configuration Menu - State Transition Table				
Current State	Event	[Guard]	Next State	Actions
MAIN_CONFIG	EV_ENTER1		STOCK	drink_number = 0
	EV_ENTER2		STOCK	drink_number = 1
	EV_ENTER3		STOCK	drink_number = 2
	EV_ENTER4		STOCK	drink_number = 3
STOCK	EV_ESCAPE		MAIN_CONFIG	drink_number = 0
	EV_ENTER1		PRICE	
	EV_OK		SET_STOCK	
SET_STOCK	EV_ESCAPE		STOCK	stock_value = 0
	EV_ENTER1	[stock_value < 99]	SET_STOCK	stock_value++
	EV_ENTER2	[stock_value > 0]	SET_STOCK	stock_value--
	EV_OK		MAIN_CONFIG	stock[drink_number] = stock_value ; stock_value = 0 ; drink_number = 0
PRICE	EV_ESCAPE		MAIN_CONFIG	drink_number = 0
	EV_ENTER1		STOCK	
	EV_OK		SET_PRICE	
SET_PRICE	EV_ESCAPE		PRICE	price_value = 0
	EV_ENTER1	[price_value < 9]	SET_PRICE	price_value++
	EV_ENTER2	[price_value > 0]	SET_PRICE	price_value--
	EV_OK	[price_value > 0]	MAIN_CONFIG	price[drink_number] = price_value ; price_value = 0 ; drink_number = 0
	EV_MODE		MAIN_CONFIG	drink_number = 0
	EV_TEMP_ERROR		MAIN_ERROR	drink_number = 0

Diagrama de Estados Menú del usuario

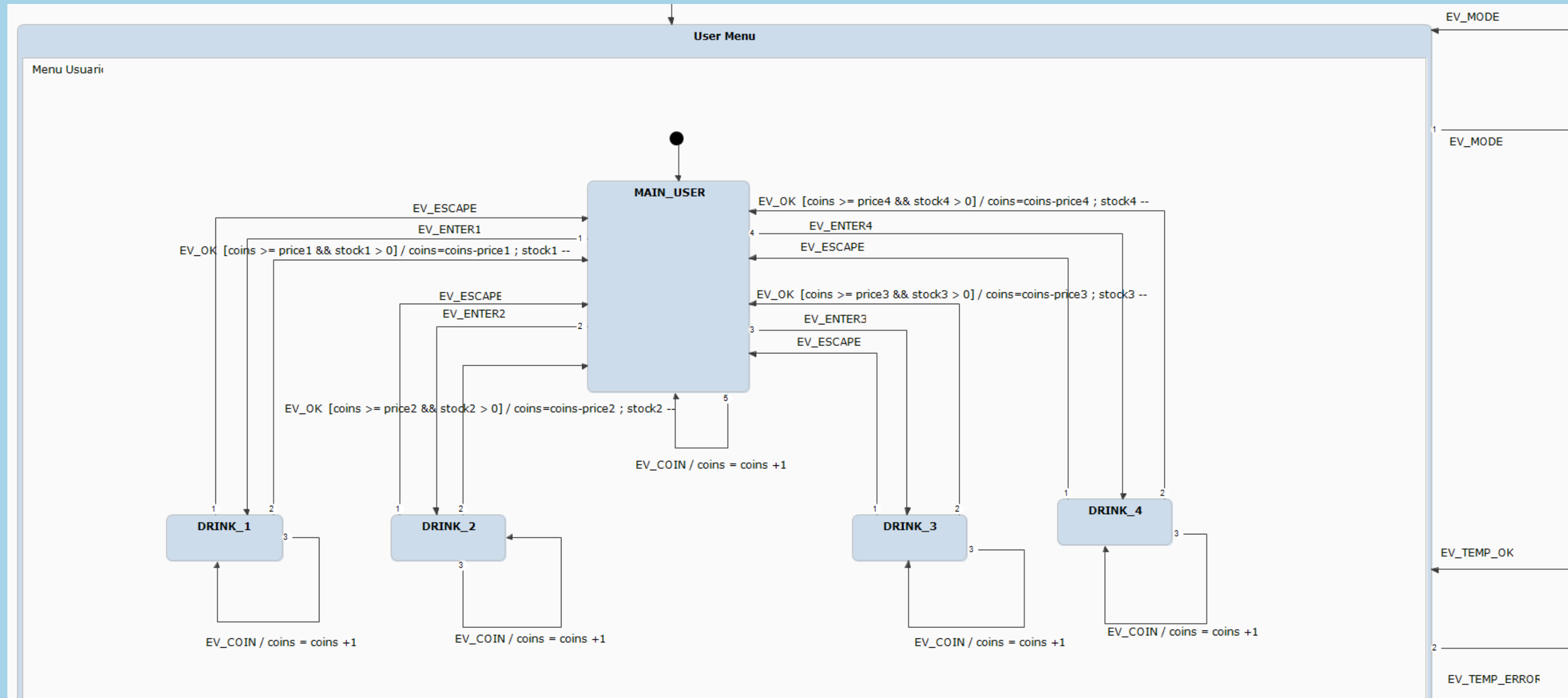


Diagrama de Estados Menú del Técnico

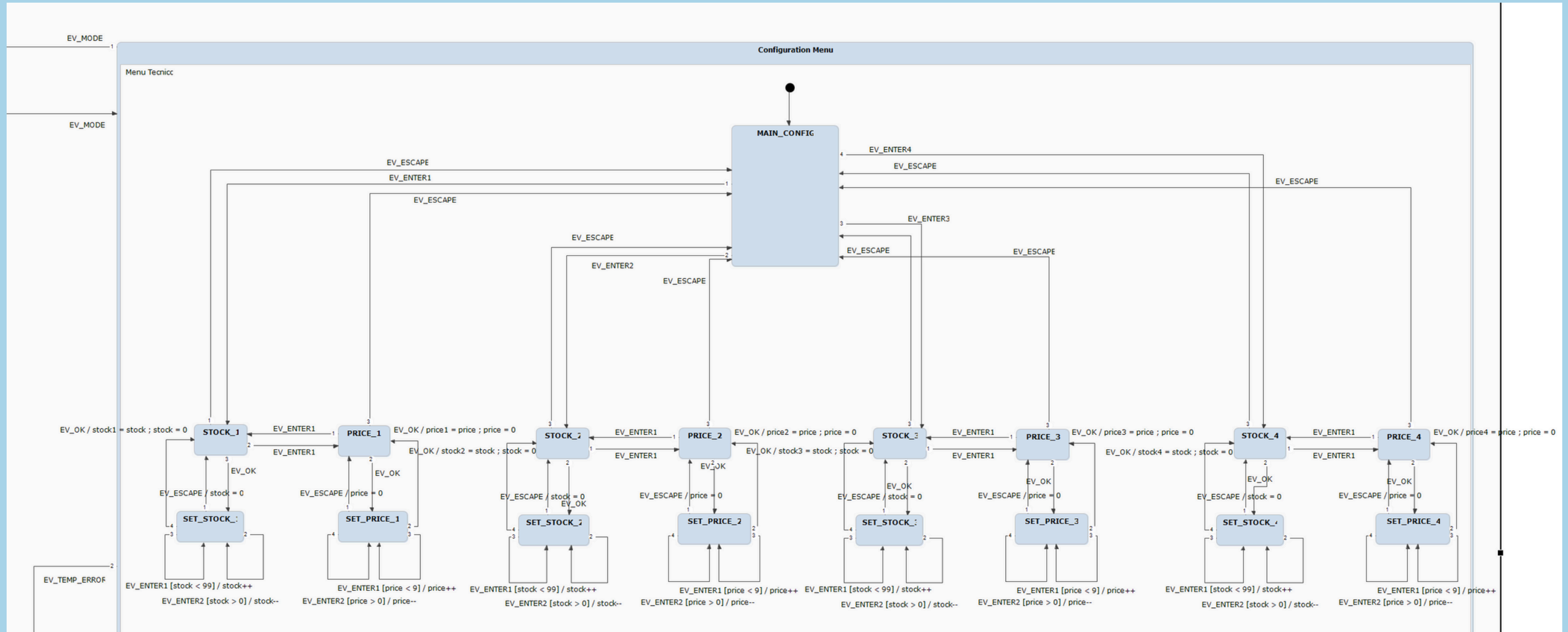
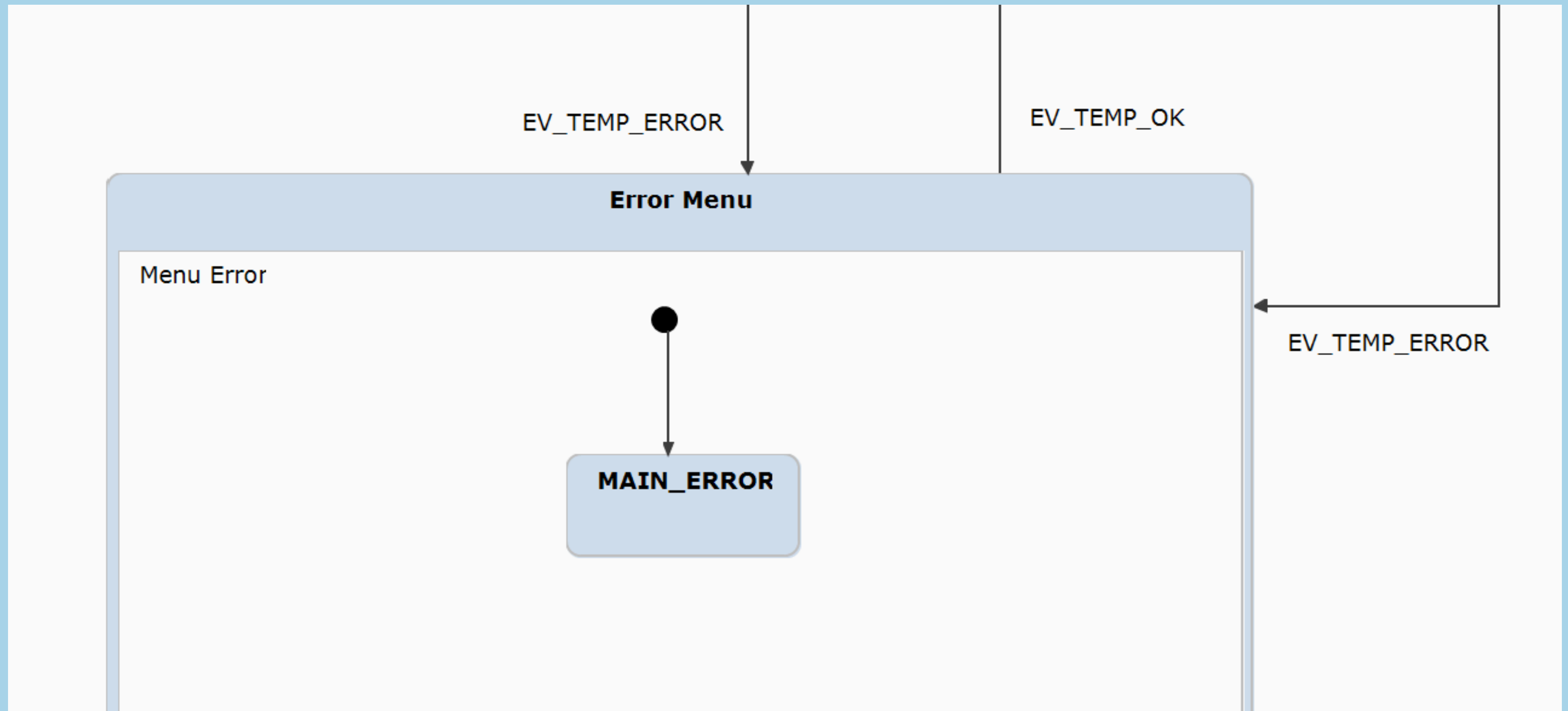


Diagrama del Estado de error



Estados Principales del Menú de Usuario:

- **MAIN_USER:** Estado de reposo donde el usuario puede iniciar una selección o insertar monedas.
- **DRINK:** Estado que se activa tras la selección de una bebida. Aquí el usuario puede confirmar la compra o añadir más crédito.
- **MAIN_CONFIG:** Un estado previsto para la configuración de la máquina, como ajustar precios y el stock inicial.
- **MAIN_ERROR:** Un estado de seguridad para gestionar posibles fallos, como un error de temperatura.

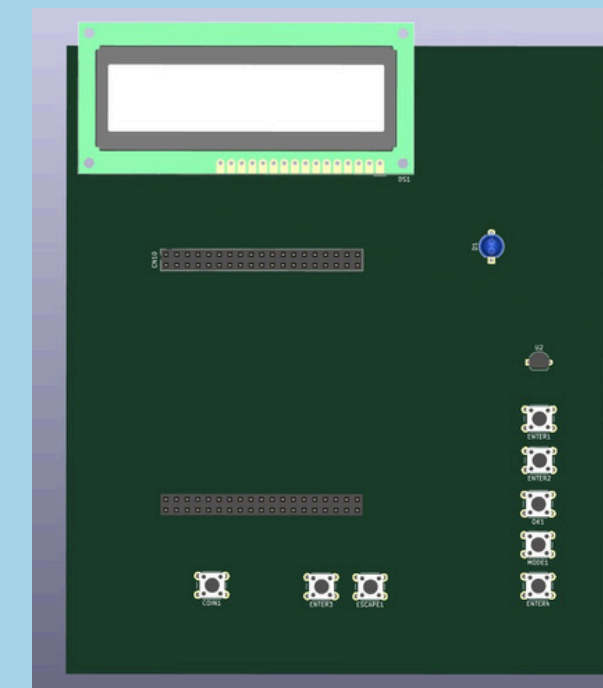
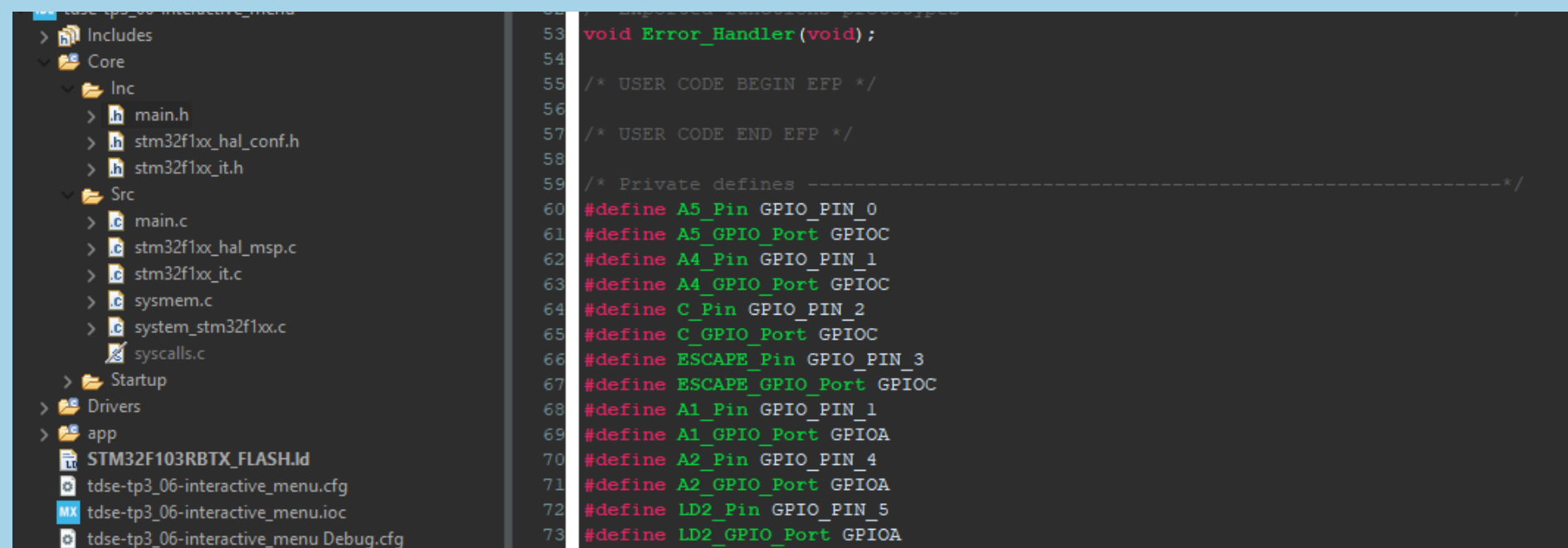
Las funciones de incrementar y decrementar el stock y los precios se consolidaron en los botones ENTER1 y ENTER2 respectivamente.

Flujo de Interacción del Usuario

- **Selección de Bebida:** Desde el estado MAIN_USER, al presionar uno de los botones ENTER (1-4), el sistema transita al estado DRINK, asignando el número de bebida correspondiente.
- **Ingreso de Crédito:** En cualquier momento, el evento EV_COIN incrementa el contador de monedas del usuario.
- **Confirmación de Compra:** En el estado DRINK, al presionar OK (EV_OK), el sistema ejecuta una guarda de validación. Si el crédito es suficiente ($\text{coins} \geq \text{price}$) y hay existencias ($\text{stock} > 0$), la compra se procesa.
- **Despacho y Actualización:** Tras una compra exitosa, el stock de la bebida se decrementa, el crédito se descuenta y el sistema regresa a MAIN_USER. La luz LED se enciende de un color diferente para la representación de una bebida (bebida 1: verde, bebida 2: rojo, bebida 3, azul, bebida 4: celeste).
- **Cancelación:** El evento EV_ESCAPE permite al usuario cancelar la operación en cualquier momento y volver al menú principal.

Entorno de Desarrollo y Diseño de PCB

- **Entorno de Desarrollo:** El software embebido fue desarrollado en el STM32CubeIDE.
- **Diseño de PCB con KiCad:** El diseño del circuito impreso (PCB) se realizó con el software KiCad. Se definieron reglas de diseño específicas, como un ancho de pista mínimo de 1.0 mm para asegurar la integridad y la calidad de la placa de circuito impreso. En base a este diseño se soldaron las pistas y los pines para conectar los componentes en la placa de desarrollo experimental.



Análisis de Consumo Energético

Se realizó un análisis del consumo de corriente de los componentes clave para evaluar la demanda energética del sistema.

Componente	Consumo
Microcontrolador	7.8mA
Led Azul	3.3mA
Led Verde	4.0mA
Led Rojo	7.6mA
LM35	0.05mA
Botón	0.09mA
Display (backlight)	20.6mA
Display	0.15mA

Conclusión

La culminación de este proyecto es un prototipo funcional de una máquina expendedora de bebidas que demuestra una exitosa integración entre el hardware y el software.

La lógica de control se implementó a través de una máquina de estados que gestiona los modos de usuario, la selección de bebidas y la validación de compra. A nivel de hardware, se realizó un diseño de circuito impreso en KiCad con parámetros definidos, como un ancho de pista mínimo de 1.0 mm, lo que asegura la fiabilidad de las conexiones. Este enfoque integral, que abarca desde el consumo energético de cada componente hasta las conexiones específicas del microcontrolador, establece una base de ingeniería sólida para el futuro desarrollo y la producción de una versión comercial.

