

Carrera de Especialización en Sistemas Embebidos

Sistema de Dosificación Trabajo Práctico Final

Trabajo Práctico Final - PCSE



Autor: Ing. Federico Leonardo Alderisi
Docente: Ing. Israel Pavelek

19/04/2024

Página 1 / 8

Tabla de Contenidos

1	INTRODUCCIÓN.....	3
1.1	Propuesta.....	3
1.2	Periféricos utilizados.....	3
1.3	Diagrama Ilustrativo.....	4
2	DESARROLLO.....	5
2.1	Finite State Machines.....	5
2.2	Diagrama de la FSM.....	5
2.3	Presentación por LCD de los estados.....	5
2.4	Rutinas y referencias.....	6
2.5	Prototipado.....	6
2.6	Modularización.....	7
2.7	Modularización dentro del proyecto CubeIDE.....	8
2.8	Mensaje de error a través de la UART.....	8
2.9	Links.....	8

1 INTRODUCCIÓN

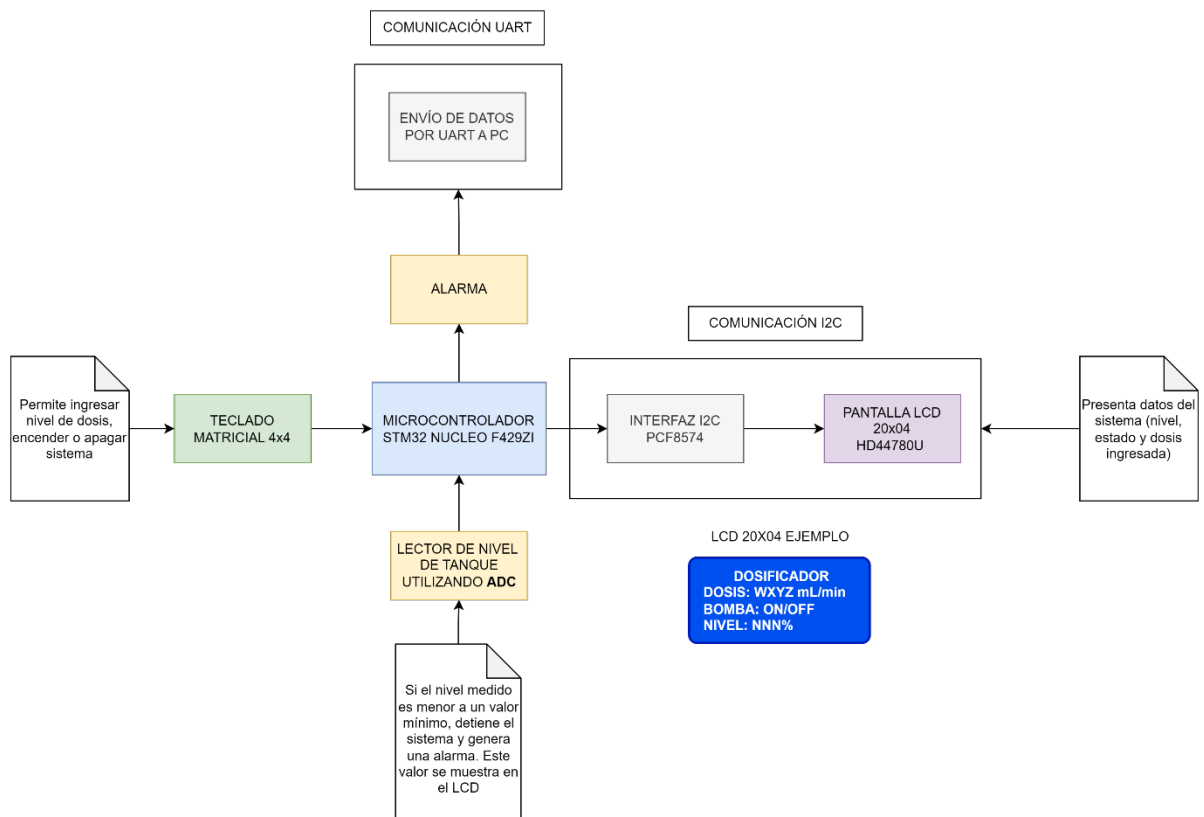
1.1 PROPUESTA

1. Este proyecto se basa en la propuesta de proyecto final de la especialidad CESE. Este primer desarrollo se enfoca en un dosificador controlado, el cual ingresamos la dosis deseada, medimos nivel del tanque, generando una alarma según su estado, y activamos o desactivamos una bomba.
2. La interacción del usuario con el sistema es a través de un teclado matricial de 4x4. Estos datos se verifican, para luego mostrarlos a través de un display LCD 20x04, el cual se comunica a través de un módulo I2C.
3. Además, un sensor de nivel simulado con un potenciómetro dispara una alarma por bajo nivel de líquido, activando un buzzer, y envía mensajes periódicos por UART respecto al estado de alarma. Esta alarma tiene dos estados, una de nivel bajo y otra de extremadamente bajo. En este último, se apaga la bomba para evitar dañarla.
4. El proyecto consta de los siguientes módulos APIs correspondientes a cada configuración de periféricos GPIO, I2C, UART, Keypad, ADC y SystemClock. Estos tienen su archivo port que permiten la abstracción del hardware. Además, contamos con otras APIs utilizadas como herramientas de las APIs principales, como el caso de Debounce y Delay.
5. El sistema consta de una FSM que permite la funcionalidad de ingreso de datos y el cambio de estado; y una sub FSM para el manejo del teclado. Además, consta de la verificación de datos, consulta de ingreso de datos, sensado automático, envío de mensaje de alarma y accionamiento de una bomba.
6. Se desarrolló a través de una SMT32 NUCLEO F429ZI, utilizando STM32 CubeIDE. El proyecto se desarrolló utilizando MX solo para generar el código de ADC, por lo que se debe desactivar la autogeneración de código, en caso de que se modifique el código, para evitar corromper el proyecto.

1.2 PERIFÉRICOS UTILIZADOS

1. GPIO: 1 para alarma, 1 para bomba, 8 para teclado (4 lectura y 4 escritura).
2. ADC: 1 entrada para el sensor de nivel.
3. UART: Se utiliza el ST-LINK propietario de SMT32 NUCLEO F429ZI
4. I2C: Utilizado como adaptador para el LCD 20x04

1.3 DIAGRAMA ILUSTRATIVO

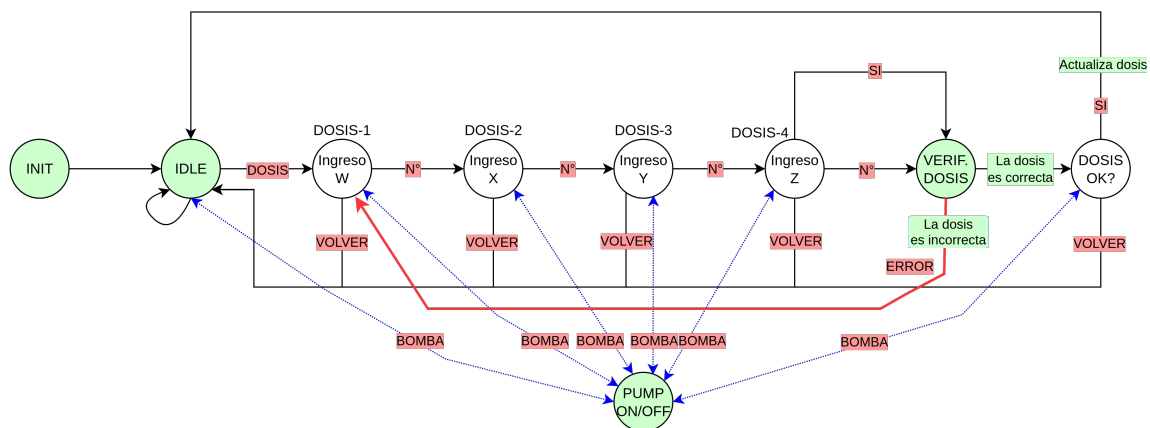


2 DESARROLLO

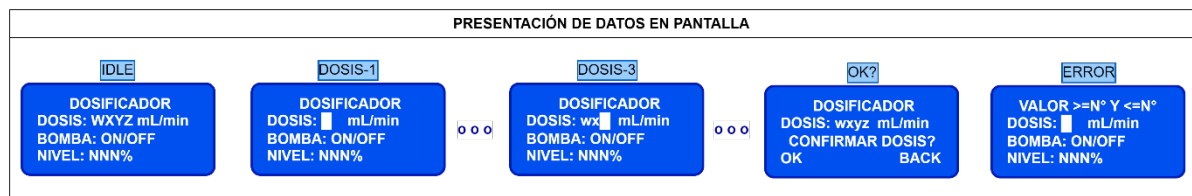
2.1 FINITE STATE MACHINES

1. Dentro de la FSM se encuentran distintos estados correspondientes al ingreso de datos, verificación, confirmación y encendido o apagado de la bomba.
2. Este último estado puede ser accedido desde cualquier parte de la FSM, permitiendo al usuario accionar la bomba en cualquier momento. Es importante ya que, en caso de emergencia, no puede estar limitado su control.
3. En el estado idle se verifican los cambios de las variables y se actualizan sus valores por pantalla.
4. Finalmente tenemos una función que se ejecuta luego de la FSM, donde se actualiza el nivel del tanque y se verifica si es bajo o extremadamente bajo. Esto permite avisar a través del buzzer y UART si está bajo (sonido intermitente) o extremadamente bajo (sonido continuo).
5. Si este está extremadamente bajo, la bomba se desactiva y no se puede activar desde el teclado, hasta recuperar un nivel de líquido superior.

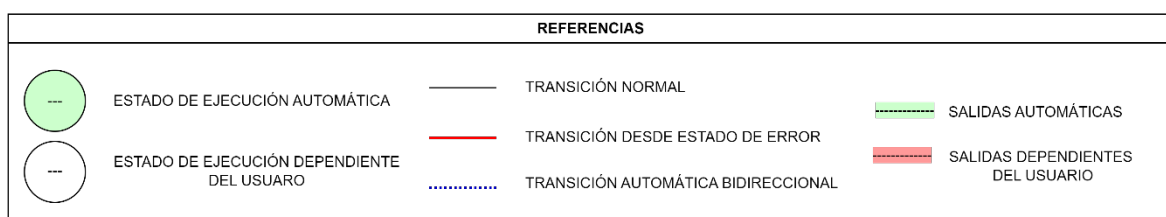
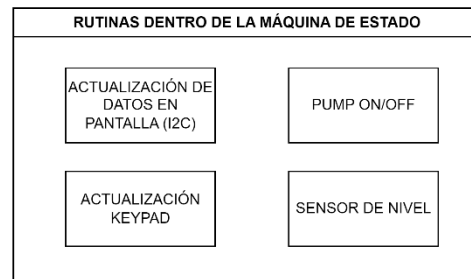
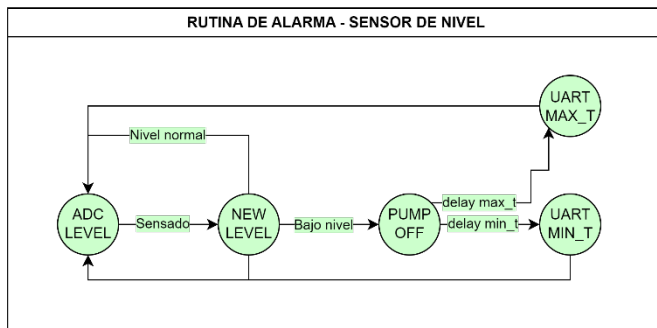
2.2 DIAGRAMA DE LA FSM



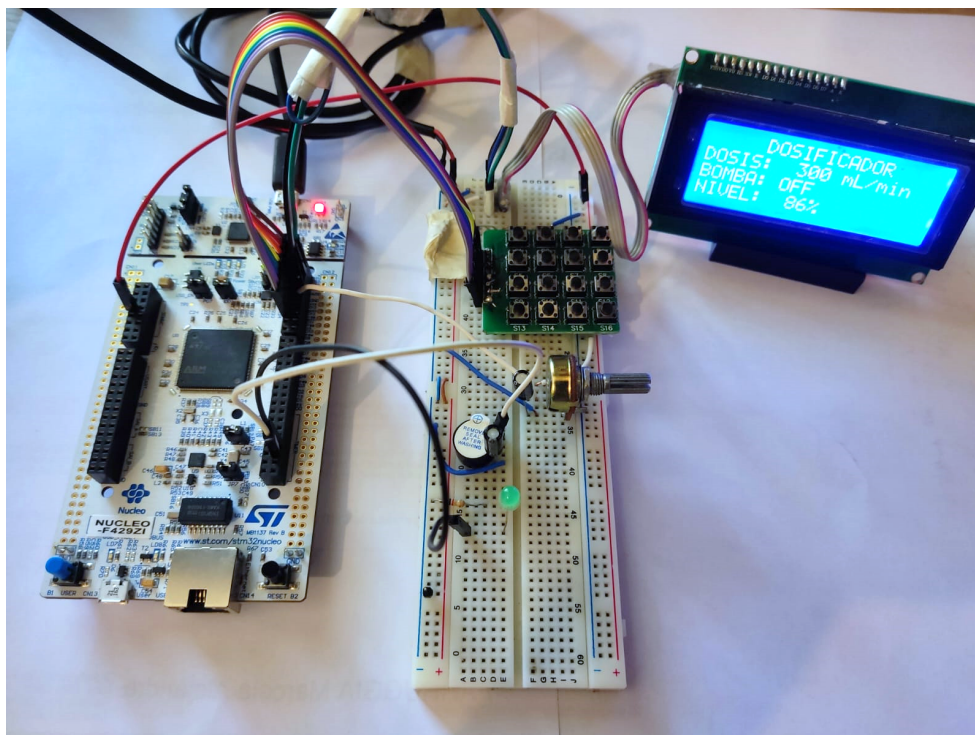
2.3 PRESENTACIÓN POR LCD DE LOS ESTADOS



2.4 RUTINAS Y REFERENCIAS

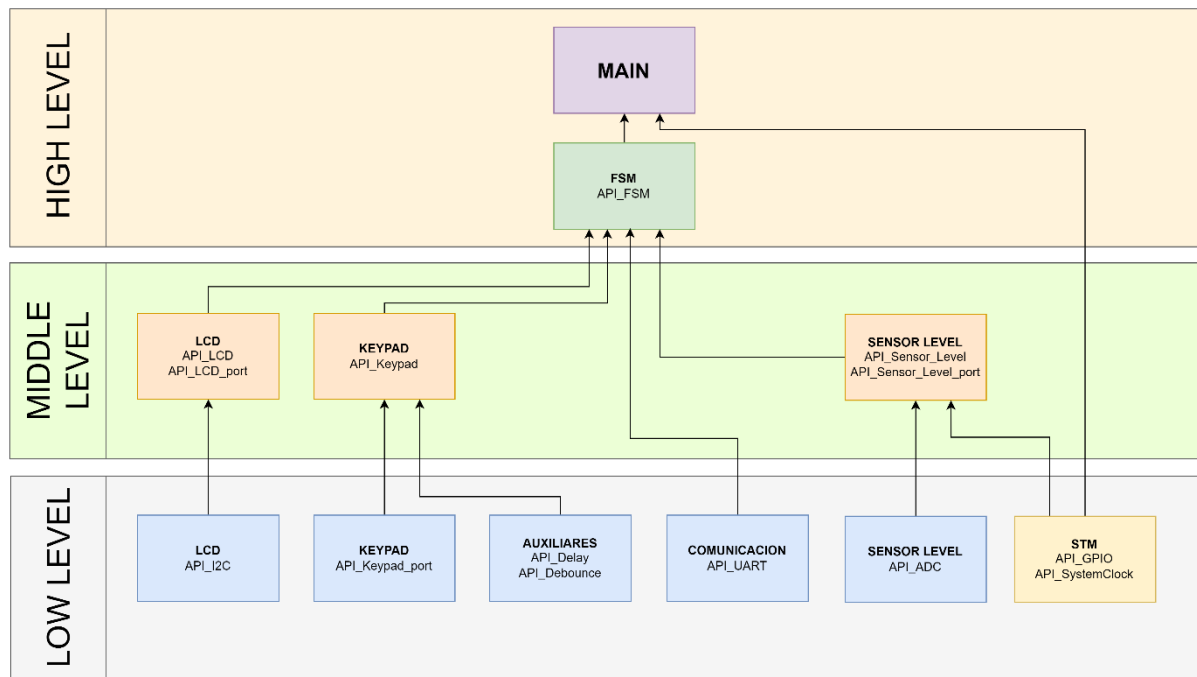


2.5 PROTOTIPADO

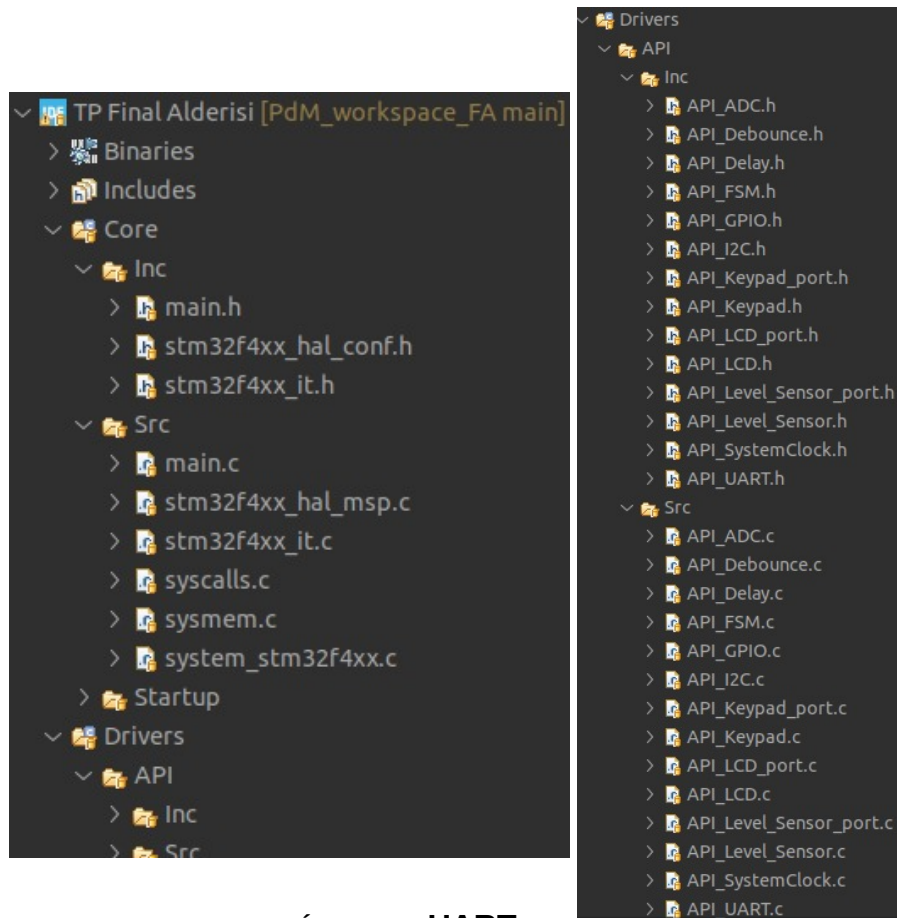


2.6 MODULARIZACIÓN

1. En la imagen se puede observar los distintos módulos y como se relacionan entre ellos. En low level podemos observar las configuraciones de los periféricos y funciones que dependen directamente del hardware, por ejemplo: las funciones HAL del módulo utilizado.
2. En middle level podemos observar los módulos que interaccionan con low level y otros que interaccionan con high level. Esto permite que a través de la configuración previa de los archivos port, podamos utilizar los módulos en distintos microcontroladores.
3. En high level nos encontramos con el main y la FSM.



2.7 MODULARIZACIÓN DENTRO DEL PROYECTO CUBEIDE



2.8 MENSAJE DE ERROR A TRAVÉS DE LA UART

```
[17:52:10:245]
[17:52:10:245] UART CONFIGURATION:
[17:52:10:299] #BaudRate: 9600
[17:52:10:299] #WordLength: 0
[17:52:10:299] #StopBits: 0
[17:52:10:317] #Parity: 1536
[17:52:10:326] #HwFlowCtl: 0
[17:52:10:340] #Mode: 12
[17:52:10:355] #OverSampling: 0
[17:52:10:373]
[17:52:17:731] WARNING: ALARM_MAX_TH ACTIVATED - LOW LIQUID LEVEL
[17:52:22:932] WARNING: ALARM_MIN_TH ACTIVATED - EXTREMELY LOW LIQUID LEVEL - PUMP OFF
```

2.9 LINKS

1. Repositorio: https://github.com/fedealde/PdM_workspace_FA/tree/main/TP%20Final%20Alderisi
2. Video explicativo: https://drive.google.com/file/d/1ic4IIOrRR_fZeu6891zYJVNOeKKGztza/view?usp=sharing