Reporte trabajo práctico final PdM

1. Federico Leonardo Alderisi

2. Link al repositorio con el código fuente del TP:

   https://github.com/fedealde/PdM_workspace_FA.git

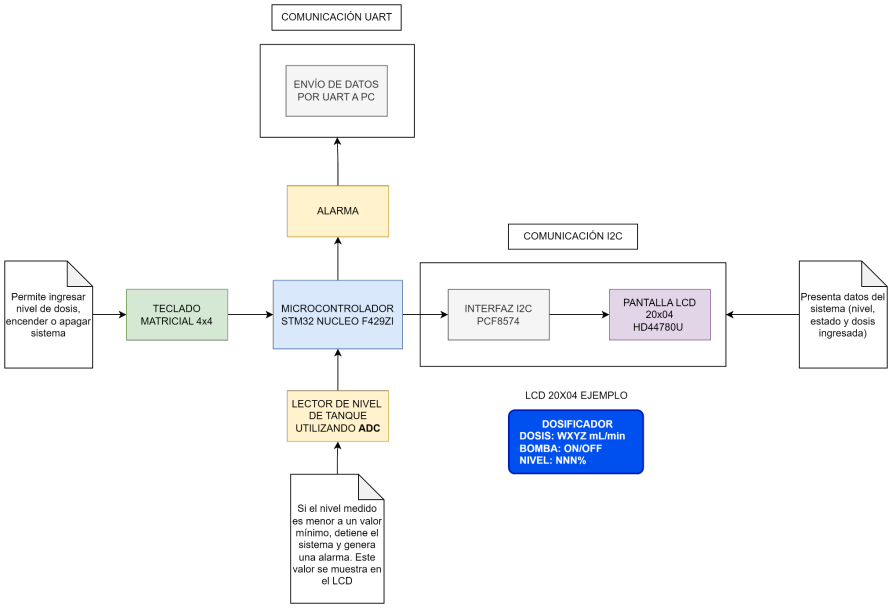
   VIDEO EXPLICATIVO:
   https://drive.google.com/file/d/1ic4IIOrRR_fZeu6891zYJVNOeKKGztza/view?usp=sharing

3. 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 dosis deseada, sensamos nivel del tanque y activamos o desactivamos una bomba. El ingreso de datos es a través de un teclado matricial de 4x4, estos se verifican para luego mostrarlos a través de un display LCD 20x04, con el cuál se comunica a través de un módulo I2C. Además, un sensor de nivel simulado con un potenciómetro dispara una alarma por bajo nivel de líquido, activando un buzzer, apagando la bomba de dosificación y enviando mensajes periódicos por UART respecto al estado de alarma (consta de una alarma de nivel bajo y extremadamente bajo).

   El proyecto consta de los módulos API correspondientes a cada configuración de periféricos GPIO, I2C, UART, Keypad y ADC. Estos tienen su archivo port que permiten la abstracción del hardware.

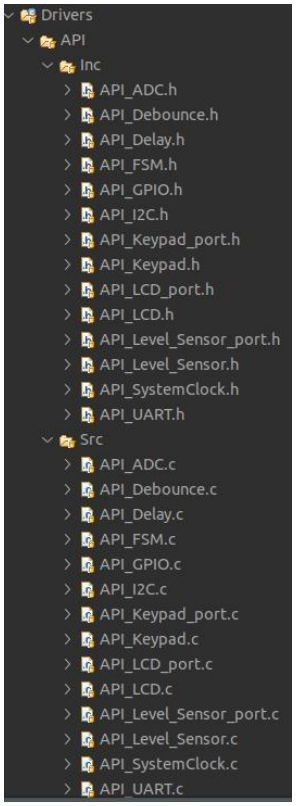
   Consta de una FSM que permite la funcionalidad de ingreso de datos, y sub FSM para el manejo del teclado. Además consta de la verificación de datos, consulta de ingreso de datos, sensado automático, envío de mensaje de alarma y accionamiento de una bomba.

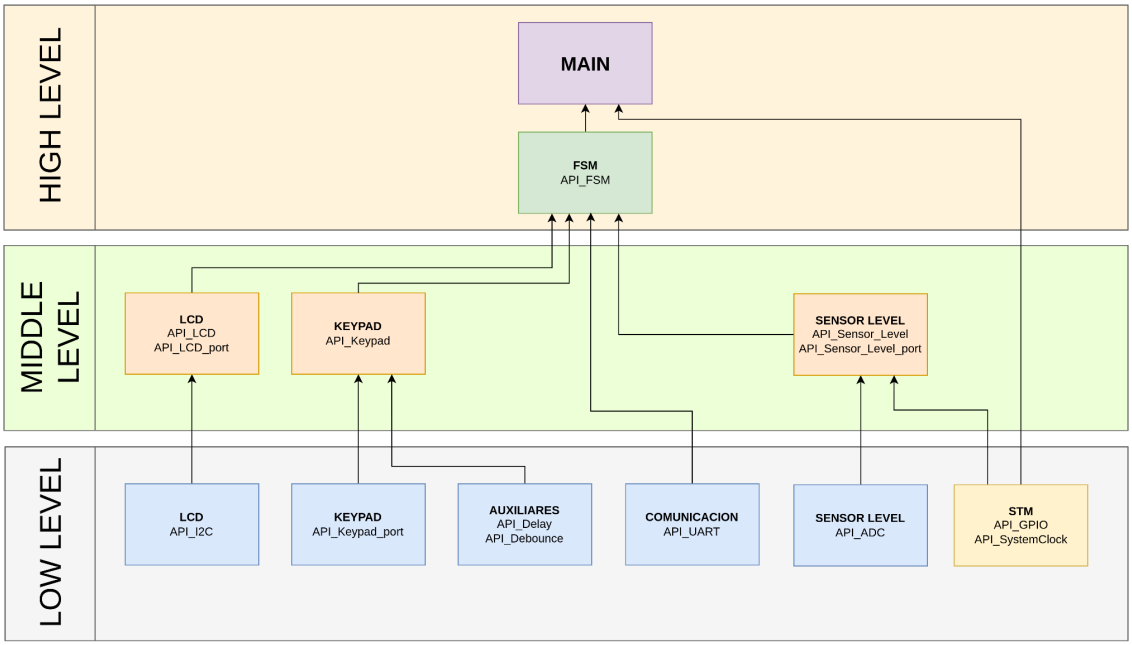
   Se desarrolló a través de una NUCLEO 429zi de STM32, utilizando STM32 CubeIDE. El proyecto se desarrolló utilizando MX solo para el código de ADC por lo que se debe desactivar la autogeneración de código, en caso de ser modificado.



4. Liste el o los módulos de software que desarrolló en su TP:

A continuación se muestra los archivos generados para el manejo de los periféricos y la FSM principal:





5. Liste los periféricos que utiliza en su TP:

   - GPIOs para alarma, bomba y 8 para el control del keypad. (10 en total)
   - I2C para LCD 20x04.
   - UART para envío de mensaje de error. ADC para medición de nivel de tanque.

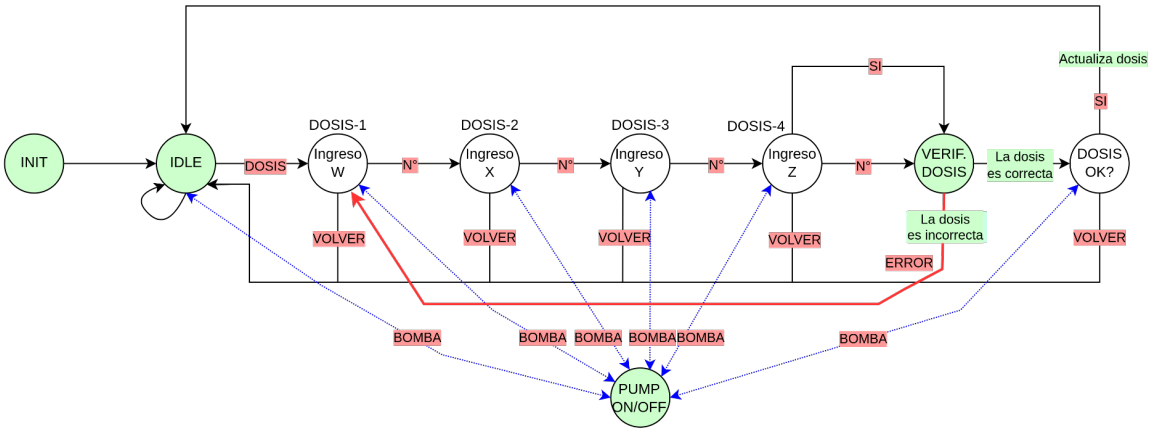
6. Dé un ejemplo de variable global privada con un link al código fuente.
   https://github.com/fedealde/PdM_workspace_FA/blob/26d8592c24b1708160b27ba6ba5c3a3965e1b94f/TP%20Final%20Alderisi/Drivers/API/Src/API_Level_Sensor_port.c#L10

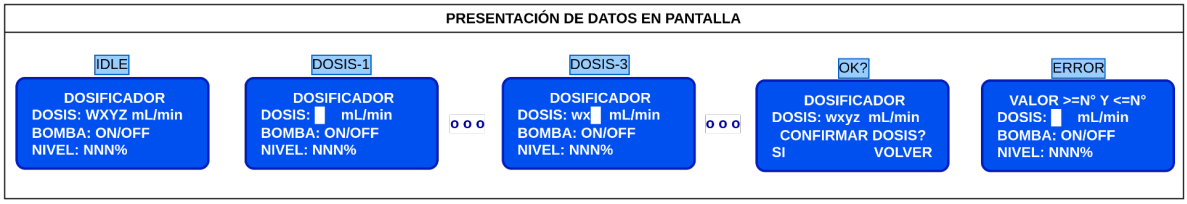
   Este muestra: **static ADC_HandleTypeDef *handler = NULL;**

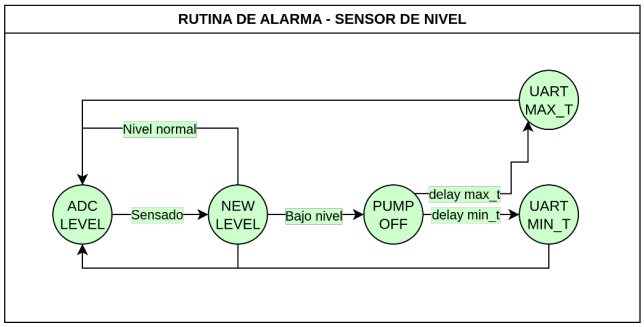
   Es el handler que utilizamos para utilizar el ADC

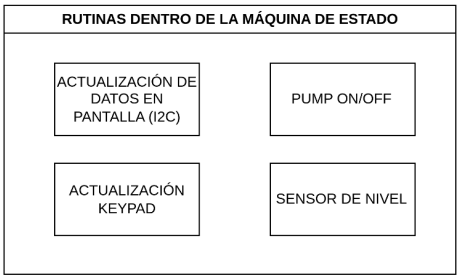
7. Ponga el link a la primera línea de la función de actualización de la MEF de elaboración propia:

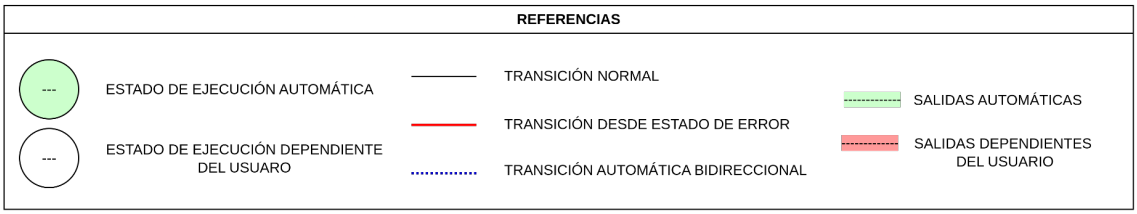
   https://github.com/fedealde/PdM_workspace_FA/blob/20fc9b9799463d7788126480394e6ece8232a5f1/TP%20Final%20Alderisi/Drivers/API/Src/API_FSM.c#L214











8. Si utilizó alguna de las reglas de the power of ten rules, mencionar cuál y poner el link al código fuente donde se aplica.

   1) Evitar control de flujo complejo como goto y recursiones. **SI**
   2) Todos los bucles deben tener límites fijos. **SI (excepto en main)**
   3) Evitar el uso de asignación dinámica de memoria. **SI**
   4) Restringir las funciones a una página impresa simple. **SI**
   5) Usar un mínimo de dos chequeos en tiempo de ejecución por función (assertions). En general: pre y post condiciones. **SI (utilizo funciones de Error y retorno valores en caso de falla)**

      Ejemplo: Verifico si se pudo enviar el dato o no por I2C

https://github.com/fedealde/PdM_workspace_FA/blob/20fc9b9799463d7788126480394e6ece8232a5f1/TP%20Final%20Alderisi/Drivers/API/Src/API_LCD_port.c#L16

6) Restringir la visibilidad de la información al mínimo posible. **SI (a través de static o limitando el acceso a los .h)**
7) Chequear el valor de retorno de todas las funciones non-void. **NO**
8) Usar el preprocesador moderadamente. **SI**
9) Limitar el uso de punteros a una sola desreferencia. No usar punteros a funciones. **SI**
10) Compilar con todos los warnings activos. **SI**

9. Si utilizó alguna otra regla/técnica de buenas prácticas de programación, explicar cuál y poner el link al código fuente donde se aplica:

   A través de un wrapper, inicializo una función de más bajo nivel y retorno si fue exitoso o no, limitando el acceso al mismo.

   https://github.com/fedealde/PdM_workspace_FA/blob/20fc9b9799463d7788126480394e6ece8232a5f1/TP%20Final%20Alderisi/Drivers/API/Src/API_LCD_port.c#L12

10. Ponga el link a un comentario en el código que explique el por qué:

    https://github.com/fedealde/PdM_workspace_FA/blob/20fc9b9799463d7788126480394e6ece8232a5f1/TP%20Final%20Alderisi/Drivers/API/Src/API_FSM.c#L429