**Carrera: Ingeniería Electrónica**

**Asignatura: Técnicas Digitales II**

**Año: 2025**

**Grupo N° 7**

**Listado de integrantes del grupo.**

**Alumnos:** Acosta Marcelo

Carrasco Sebastián

Soria Aldana

Vallejo Walter Manuel

**Profesor:** Enrique Mancilla

# **Informe de proyecto Final**

**Título del Proyecto Final:**

Sistema de medición de humedad y temperatura con sensor infrarrojo

* 1. **Consideraciones sobre el hardware del proyecto**

Objetivos: La idea general se basa en la lectura de la humedad y temperatura que se mostrará en el display 16X2 I2C cuando exista la presencia de una persona u objeto, que detectará el sensor infrarrojo MZ80 y nos dará la información completa captada por el sensor DH11. De la misma forma en un futuro se dará información atreves de un modulo bluetooth a un celular.

**Gráfico, Gráfico de cajas y bigotes

El contenido generado por IA puede ser incorrecto.**

**Funcionamiento:**

**1.- Inicialización:** Entradas: No

Salidas: # LD1,LD2,LD3 se encienden 3 seg

# Mensaje 1: “PROYECTO FINAL”

GRUPO 7

Tiempo del mesaje : 3 seg de retraso

**Descripción:** En este estado, el sistema realiza la configuración inicial de los periféricos y módulos necesarios. Primero, se inicializan el reloj del sistema, los pines GPIO y el bus I2C para la comunicación con la pantalla LCD. Luego, se ejecuta la función **HAL\_Init()** para configurar la capa de abstracción de hardware. También se inicializan los módulos de comunicación serie (UART y USB) y el protocolo Ethernet.

Posteriormente, se configura e inicializa la pantalla LCD I2C mediante la función **HD44780\_Init(2),**se limpia la pantalla con **HD44780\_Clear()** y se muestran mensajes de prueba para verificar el correcto funcionamiento del LCD.

**2.- Lectura de Datos:** Entradas: # Tiempo 1 = 3 seg

# Dato 1 = Temperatura

# Dato 2 = Humedad

Salidas: # LD1,LD 2: se encienden cuando esta haciendo la lectura de datos.

# Mensaje 2: lectura de los valores.(“Esperando…”)

**Descripción:** El sistema intenta obtener los valores de temperatura y humedad del sensor DHT11. Primero, se llama a la función **DHT11\_ok()** para verificar si el sensor está listo para proporcionar datos. Si el sensor responde correctamente, se ejecuta la función **DHT11\_Read(),** que actualiza las variables **DHT11\_TEMP y DHT11\_HUM** con los valores de temperatura y humedad actuales. Los datos leídos se formatean en cadenas de texto utilizando **sprintf()** y se almacenan en los arreglos texto3 y texto2.

Si el sensor no responde correctamente, se muestra un mensaje de error en la pantalla LCD.

**3.- Presentación de Datos:** Entradas: # Tiempo 2 = 2 seg

# Dato 1 = Temperatura

# Dato 2 = Humedad

Salidas: # LD1, LD3: se enciende cuando da la presentación de los datos

# Mensaje 3: “Temp= [Dato1] C

Hum= [Dato2] %”

Si la lectura falla:

# Mensaje de error:” Error DTH11”

**Descripción:** En este estado, los valores de temperatura y humedad obtenidos se muestran en la pantalla LCD. Primero, se limpia la pantalla **(HD44780\_Clear()),** luego se posiciona el cursor en la primera línea **(HD44780\_SetCursor(0,0))** y se muestra la temperatura en el formato **"Temp=XXC".** Posteriormente, en la segunda línea **(HD44780\_SetCursor(0,1))** se muestra la humedad en el formato **"Hum=XX%".**

Si la lectura anterior falló, en lugar de mostrar los valores, se presenta un mensaje de error en la pantalla LCD.

Finalmente, se introduce un retraso de 2 segundos **(HAL\_Delay(2000))** antes de regresar al estado de Lectura de Datos para actualizar la información.

**1.2 Circuito del proyecto**

**Diagrama, Esquemático

El contenido generado por IA puede ser incorrecto.**

* 1. **Listado de componentes**

|  |  |  |
| --- | --- | --- |
| **PERIFERICOS** | **CANTIDAD** | **FOTO** |
| STM32F429ZI | 1 |  |
| SENSOR DHT11 | 1 |  |
| SENSOR INFRARROJO MZ80 | 1 |  |
| MODULO BLUETOOTH HC-06 | 1 |  |
| DISPLAY I2C | 1 |  |
| LED | 3 |  |
| RESISTENCIAS | 1. de 220 ohm   1 de 100 ohm |  |

**2.1 Link al repositorio grupal donde puede encontrarse el proyecto:**

[**https://github.com/MarceloAcosta27/Grupo\_7\_TDII\_2024/tree/master**](https://github.com/MarceloAcosta27/Grupo_7_TDII_2024/tree/master)

**Descripción de funcionamiento de la aplicación del proyecto**.

. El proyecto consiste en un **sistema embebido** basado en un **STM32**, diseñado para medir y mostrar en tiempo real la **temperatura y humedad** mediante el **sensor DHT11**. Los datos obtenidos se visualizan en un **LCD** y se transmiten a un dispositivo externo mediante **Bluetooth**.

Para indicar el estado del sistema, se utilizan **LEDs** que cambian según las condiciones detectadas. Si el sensor funciona correctamente, la lectura se muestra en la pantalla y se envía por Bluetooth. En caso de error, se encienden todos los **LEDs de alerta**, se muestra un mensaje en el **LCD**, y se envía una notificación de falla.

El sistema opera en un **bucle continuo**, verificando periódicamente el estado del sensor y actualizando los datos cada segundo.

Diagrama

El contenido generado por IA puede ser incorrecto.

**2.3. Listado de los módulos de software desarrollados en el proyecto.**

Los módulos de software para el manejo de los periféricos de la aplicación se ubicarán en sus correspondientes carpetas, ***Inc*** y ***Src*** ubicados en el siguiente path de la carpeta de proyecto del IDE STM Cube: ***Drivers\API***

• Módulo de software para el manejo del sensor DHT11, se ubicará en las carpetas: **API\_DHT11.c** y **API\_DHT11.h**

• Módulo de software para el manejo del módulo adaptador I2C a Display LCD16x2 se ubicará en las carpetas: **API\_Iiquidcrystal\_i2c.c** y **API\_Iiquidcrystal\_i2c.h**

• Módulo de software para el manejo del módulo GPIO se ubicará en las carpetas **API\_GPIO.c** y **API\_GPIO.h**; incluirá las funciones para la implementación del método antirebote en caso de incluir el uso del pulsador de usuario u otro pulsador, asi como las funciones de encendido de leds.

•Modulo software para el manejo de modulo bluetooth HC05, se ubica en las carpetas **API\_HC05.c y API\_HC05.h**

**2.4 Utilizo en la carpeta API los siguientes módulos:**

**API\_GPIO**: Este módulo es de desarrollo propio en su totalidad y “wrappea” las funciones de la HAL de encendido, apagado y “toggleado” de los pines GPIO usándolas en funciones implementadas para un uso sencillo del manejo de los 3 Leds de usuario de la placa STM alas que solo se les pasa como parámetro el LEDx que quiero utilizar.

**API\_Liquidcrystal\_i2c**: Este modulo proporciona una implementación completa para controlar un LCD I2C en una **STM32F4**, permitiendo escribir texto, manipular el cursor, configurar la pantalla y gestionar la retroiluminación de manera eficiente.

**API\_DHT11\_LIB:** Este Modulo proporciona una implementación para leer la temperatura y la humedad desde un DHT11 en una STM32, asegurando una comunicación correcta y verificando la validez de los datos.

**API\_puertos:** Este puerto proporciona una interfaz optimizada para configurar y manejar pines GPIO en STM32, permitiendo:

Configuración rápida y eficiente de los pines como entrada o salida.

Compatibilidad con diferentes microcontroladores STM32, incluyendo modelos STM32F1 y otros.

**API\_delays:** Manejo preciso de temporización con temporizadores y registros de ciclo de reloj.

Dos enfoques para configurar los pines: uno optimizado mediante registros y otro más simple con HAL.

**API\_uart.h:** El uart proporciona una API para enviar y recibir datos mediante UART en un STM32, facilitando la comunicación con otros dispositivos, como una PC o sensores.

**API\_HC-05:** Este módulo es un dispositivo de comunicación inalámbrica que permite establecer conexiones serie (UART) entre microcontroladores y otros dispositivos Bluetooth. Funciona en modo maestro o esclavo y se configura mediante comandos AT. Su alimentación es de 3.3V en la lógica de comunicación, aunque puede recibir 5V en VCC. Se utiliza comúnmente en proyectos de robótica, domótica y comunicación entre dispositivos electrónicos.

**Nota**:En cada módulo API se ubica la función de inicialización y configuración del módulo correspondiente utilizado por este.

**2.5 Listado de los periféricos que utiliza en el proyecto**

**Modulo lcd I2C:** El módulo LCD I2C es una pantalla de cristal líquido equipada con un adaptador que utiliza el protocolo I2C para la comunicación. Se emplea en proyectos electrónicos para mostrar información de forma sencilla y eficiente.

**Sensor DHT11:** Es un sensor de temperatura y humedad digital de bajo costo que envía datos a un microcontrolador a través de una señal digital de un solo cable. Tiene una precisión básica y es ideal para aplicaciones domésticas.

**Sensor Infrarrojo MZ80:** Es un sensor óptico que detecta objetos cercanos mediante la emisión y recepción de luz infrarroja. Se usa en robótica para evitar obstáculos o detectar bordes.

**Modulo Bluetooth HC-05:** Es un módulo de comunicación inalámbrica que permite la transmisión de datos vía Bluetooth mediante una conexión serie (UART). Se puede configurar en modo maestro o esclavo y se usa en aplicaciones de control y comunicación remota.

**GPIO:** Son pines de propósito general en un microcontrolador o microprocesador que pueden configurarse como entradas o salidas digitales para interactuar con otros dispositivos electrónicos.

1. **Buenas prácticas de programación**

**3.1** Ubicamos el desarrollo de la MEF en main.c directamente por lo que a continuación se deja el link al inicio de la misma, en el lazo infinito.  
<https://github.com/MarceloAcosta27/Grupo_7_TDII_2024/blob/master/Proyecto/ProyectoFinal_Grupo7_v2/Core/Src/main.c#L141>

**3.2 Restringir las funciones a una página impresa simple.**

A continuación, se muestra la función mas extensa utilizada en todo el proyecto:  
<https://github.com/MarceloAcosta27/Grupo_7_TDII_2024/blob/master/Proyecto/ProyectoFinal_Grupo7_v2/Drivers/API/Src/API_puertos.c#L16>

**3.3** El código compila sin warnings, No se utilizan recursiones ni control de flujo del tipo goto, los lazos usados tienen límites fijos, no hay lazos que lleven a una recursividad infinita, salvo el lazo infinito de main.c.

**4. Datasheets de componentes utilizados.**

**PLACA STM32F429ZI**

<https://www.alldatasheet.com/html-pdf/882039/STMICROELECTRONICS/STM32F429ZI/1948/1/STM32F429ZI.html>

**SENSOR DHT11**

<https://agelectronica.lat/pdfs/textos/D/DHT11-SENSOR-MODULE.PDF>

**SENSOR INFRARROJO MZ80**

<https://datasheet4u.com/share_search.php?sWord=e18-d80nk&gad_source=1&gclid=Cj0KCQiA8fW9BhC8ARIsACwHqYouGcxRMVnfNr3FkuSbNYHKy7oaXa1r5M8rAan3xBe--V22jLO7d3QaApt9EALw_wcB>

**MODULO BLUETOOTH HC-06**

<https://www.alldatasheet.es/html-pdf/1179032/ETC1/HC-06/109/1/HC-06.html>

**DISPLAY 12C**

<https://www.nxp.com/docs/en/data-sheet/PCF8574_PCF8574A.pdf>

<https://www.alldatasheet.com/datasheet-pdf/view/1358894/SPARKFUN/LCD-09067.html>

**LED**

<https://www.alldatasheet.es/datasheet-pdf/view/219258/BRIGHT/BIR-BM13E4G-2.html>

**MODULO BLUETOOTH HC-05:**

<https://www.alldatasheet.es/html-pdf/1492299/ETC/HC-05/60/1/HC-05.html>

**RESISTENCIAS**