Name:

Email:

Github:

——————————

2024 東華大學資訊工程系畢

——————————

嗨，我是XXX，我的專長是c/c++資料結構，java物件導向和sql語法，對於mcu韌體產業有高度興趣，因此自學了stm32，做了時鐘日歷的小型專案。

此外，曾於 Java 後端開發團隊中，參與 RESTful API 設計與 JWT 認證流程，具備基本 Web 後端開發與 SQL 操作能力。未來希望專注於 C/C++ 韌體開發領域，持續累積硬體控制經驗，並貢獻於穩定可靠的嵌入式產品開發。

—————————— 工作經歷

A 公司 職位java後端工程師 工作日期起訖2025/01~2025/02

Git版本控制/restful api/jwt/sql語法 前後端分離

————————— 技能專長

**嵌入式開發（STM32 / HAL）**

* STM32CubeIDE / HAL Library
* GPIO 控制 / 外部中斷 EXTI / UART / DMA / I2C / RTC / Timer / Rotary Encoder
* 實作 callback-based 模組、模組化驅動封裝

**◼ 程式語言與資料結構**

* C / C++（熟悉指標、動態記憶體、二元樹、stack/queue/double linked list 等）
* Java（物件導向設計、Spring Boot 框架應用）
* SQL（基本查詢、資料表設計）

**◼ Web 基礎**

* Java Spring Boot + Vue.js 前後端分離架構
* RESTful API 設計、JWT 登入驗證流程

**◼ 工具 & 版本控制**

* STM32CubeIDE、IntelliJ IDEA、Git、Postman

**◼ 外語能力**

* JLPT N1（具備良好的日文閱讀能力）

自傳

我於 2024 年 7 月畢業於國立東華大學資訊工程學系。大一時，我接觸到第一門程式語言 C，建立了我使用指標對底層記憶體操作、流程控制的初步理解；大二則學習了 C++ 的物件導向設計，進一步應用於資料結構實作，強化了我對指標、陣列、動態記憶體管理的掌握，這些經驗也開啟了我對韌體的興趣。

因此在 2025 年 3 月開始自學 STM32，並購買實體開發板，搭配 Key's King 的 YouTube 教學影片，實際動手撰寫 C 語言韌體，開始理解 MCU 運作流程（如上電啟動流程、中斷NVIC、寄存器控制等）。這段期間，我也學會如何使用 STM32CubeIDE 與 HAL 驅動庫開發周邊功能（GPIO、UART、I2C、Timer、Encoder、RTC 等），並逐步養成以模組化架構撰寫可重用驅動的習慣。

除了 C/C++ 外，我也具備 SQL 和 Java 的基礎知識，能夠快速理解資料庫設計與簡單後端邏輯，具備良好的學習力與除錯能力，相信未來無論接觸何種語言或技術，都能迅速上手、完成任務。

在未來的職場生涯中，只求自己的表現能盡善盡美，達成客戶的要求，能夠完成自己被付予的任務，對公司帶來更多的貢獻，並與公司共同成長。

## 專案成就

STM32 時鐘日曆系統 with Rotary Encoder (C語言)

開發平台：STM32CubeIDE、HAL 驅動庫

開發語言：C

顯示模組：OLED（使用自定義字型與圖形介面）

使用者介面：旋鈕編碼器（支援方向旋轉與按壓）

專案期間：2025/05 – 2025/05

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

專案簡介：

使用 STM32 做一個時鐘日曆系統，透過旋鈕進行時間設定，結合 OLED 螢幕顯示年/月/日、時/分/秒與星期。

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

旋鈕編碼器驅動（kk\_knob）模組化封裝：

利用 TIM1 的 Encoder 模式偵測旋鈕旋轉方向與按鍵偵測（含消抖機制）。

自定義callback有旋鈕順時針callback旋鈕逆時針callback按下旋鈕callback，觸發callback就透過串口發送告知旋鈕是做了順時針還是逆時針還是按下按鈕

時間顯示與設定邏輯（task\_main）：

使用者可透過按下旋鈕個別設定「年/月/日/時/分/秒」。

設定完成後即寫入 RTC 模組，回復為即時顯示模式。

設計閃爍光標動畫標示目前正在設定的時間項目

模組化架構：

kk\_knob.h/.c：獨立封裝旋鈕邏輯，易於重用。

task\_main.h/.c：主任務控制流程、畫面更新與使用者互動。

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

設計與實作 Rotary Encoder 驅動模組。

撰寫 OLED 顯示與畫面更新邏輯（含字串排版、閃爍光標）。

撰寫主任務流程，包括模式切換邏輯、RTC 時間同步與使用者操作流程。

使用 UART 傳輸除錯資訊以驗證旋鈕操作是否正確呼叫對應回調函數。

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

熟悉 STM32 HAL TIM 編碼器模式與 GPIO 中斷處理。

學會如何設計 callback-based 模組接口。

### 一個會員登入登出發文留言前後端

2024/11

採用java spring boot + vue.js + mysql + redis 靜像 + nginx管理圖片jpg靜態資源

後端有1.註冊功能 2.登入功能 3.發文功能 4.留言功能 5.登出功能

1.註冊功能service層用repository.save這個jpa

2.登入功能先看看電話號碼和email是否有重複，無重複就把電話號碼交給jwtUtil.generateToken(user.getPhoneNumber());，所以JWT裡面是有唯一識別ID:phoneNumber，生成一個 HttpOnly Cookie 儲存 JWT

3.發文功能用戶可以創建文章也可以使用crud查詢/修改/刪除自己發過的文

4.留言功能可以對其他用戶發過的文章做留言

5.登出功能讓後端生成一個過期的JWT放到HttpOnly Cookie並且使用redis service-cacheService封存之前的jwt

前端和後端一樣有分層1.model層2.components層3.service層

1.model層 資料格式

2.components層有html 還有把資料傳給service層

3.service層做axios.get/axios.put/axios.delete/axios.get.post發請求到後端controll層

docker靜像:使用nginx管理後端static靜態jpg資源/使用redis靜像

使用Nginx掛載到後端圖片資源

使用 WebSocket 當用戶登入後前端收到用戶登入成功的消息後會向後端 WebSocket發送連接請求socket = new WebSocket('wss://localhost:8443/notifications'); ，進行WebSocket 连接和存储 WebSocket 会话，WebSocket連接時有attributes.put("jwtToken", jwtToken); attributes.put("phoneNumber", phoneNumber); 有塞入唯一識別ID phoneNumber和確認jwtToken是否有效

實現當兩個用戶都處於登入狀態時，一個用戶向另一個用戶的貼文留言後，貼文的主人可以及時在前端收到提示