**✅ M1 模組設計（只產參數組合）**

**1️⃣ M1 功能說明（改版後）**

|  |  |
| --- | --- |
| **功能** | **說明** |
| 🎯 單一目的 | **僅產出策略參數組合**，不產出 signal 或進行回測 |
| ⚙ 可自選策略類型 | RSI、CROSS、MACD 等 |
| 📁 輸出格式 | 每支股票與策略產出一組 param\_log（含 param\_id 對照）與 param\_map |
| 🧊 自動限流控制 | max\_workers + download\_delay 控制對多股票產生策略的速度 |
| ✅ 後續由 M2 使用這些參數對不同時間區間進行回測 |  |

**2️⃣ 使用者選單輸入（main\_controller）**

shell

複製編輯

【策略參數產生 M1】

請輸入股票代碼（例如 AAPL,TSLA）：  
> AAPL,TSLA

請選擇策略類型：  
1. RSI  
2. CROSS  
> 1

請輸入欲產生幾組策略參數（預設=100）：  
>

請輸入同時產生上限 max\_workers（預設=3）：  
>

請輸入每支股票產生後延遲秒數（預設=1）：  
>

**3️⃣ 使用者執行過程與結束畫面**

**⏳ 執行中**

csharp

複製編輯

[INFO] 為 AAPL 產生 RSI 參數組合...  
 ⮑ 產出 100 組 param\_id + param\_dict  
⏳ 暫停 1 秒...

**✅ 結束畫面**

pgsql

複製編輯

📁 產出完成：  
✔️ param\_log\_RSI\_AAPL.json  
✔️ signal\_param\_map\_RSI\_AAPL.json

**4️⃣ M1 讀取與輸出檔案格式**

| 來源 | SQLite：database/stock\_price.db（僅載入股票代碼確認） |

| 輸出 | signals/ 下建立以下檔案 |

* param\_log\_RSI\_AAPL.json
* signal\_param\_map\_RSI\_AAPL.json

欄位範例如：

json

複製編輯

"0001": { "period": 14, "overbought": 70, "oversold": 30 }

**5️⃣ 程式碼模組結構**

bash

複製編輯

modules/  
├── m1\_param\_generator.py # 主模組（控制流程）  
utils/  
├── param\_generator.py # 各策略的參數組合規則  
├── file\_saver.py # 寫入 param\_log 與 param\_map  
signals/  
└── param\_log\_\*.json, signal\_param\_map\_\*.json