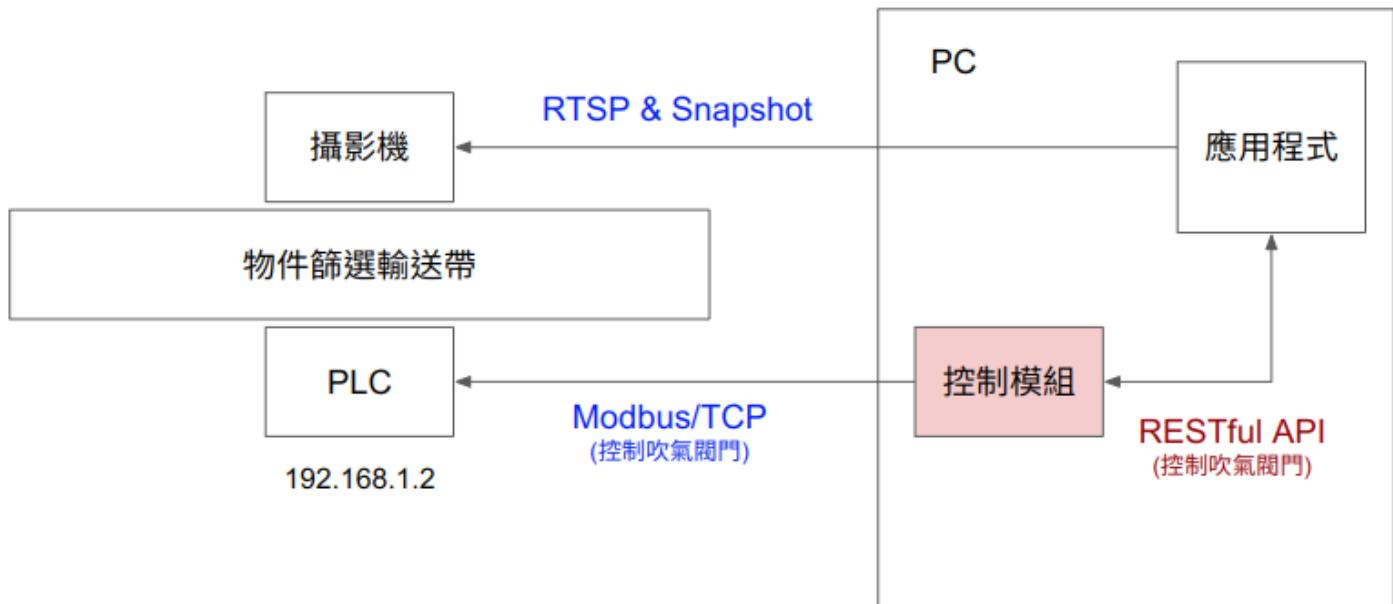


機台控制程式安裝與使用說明



- 『控制模組』提供應用程式使用 RESTful API，轉換 Modbus/TCP 控制指令到『物件篩選輸送帶』控制兩個噴頭的吹氣動作。
- 主程式安裝於 D:\app 目錄中，可以複製打包此目錄到其他主機執行

› 本機 › DATA (D:) › app ›

名稱	修改日期	類型	大小
jdk-17.0.6	2023/2/10 下午 02:48	檔案資料夾	
logs	2023/2/10 下午 02:55	檔案資料夾	
application.yaml	2023/2/10 下午 02:54	YAML 檔案	1 KB
modbus-gateway.jar	2023/2/10 下午 02:52	JAR 檔案	18,219 KB
Postman-win64-Setup.exe	2023/2/10 下午 02:57	應用程式	163,506 KB
run.bat	2023/2/10 下午 02:55	Windows 批次檔案	1 KB

- jdk-17.0.6 為 java 執行環境。
- logs 目錄為主程式執行產生的紀錄檔，用於排除障礙。
- application.yaml 檔案為設定檔，主要設定『物件篩選輸送帶』的 IP 位置，以及 Modbus 控制點位的暫存器位置等資訊。
- modbus-gateway.jar 為主程式。
- run.bat 為使用 java 指令執行 modbus-gateway.jar 的執行腳本。
- Postman-win64-Setup.exe 為模擬『應用程式』向『控制模組』下達 RESTful API 的工具，可選用安裝。

- 設定步驟一，用記事本開啟 application.yaml 確定『物件篩選輸送帶』的 IP 位置正確，預設為 192.168.1.2



```
application.yaml - 記事本

檔案 編輯 檢視

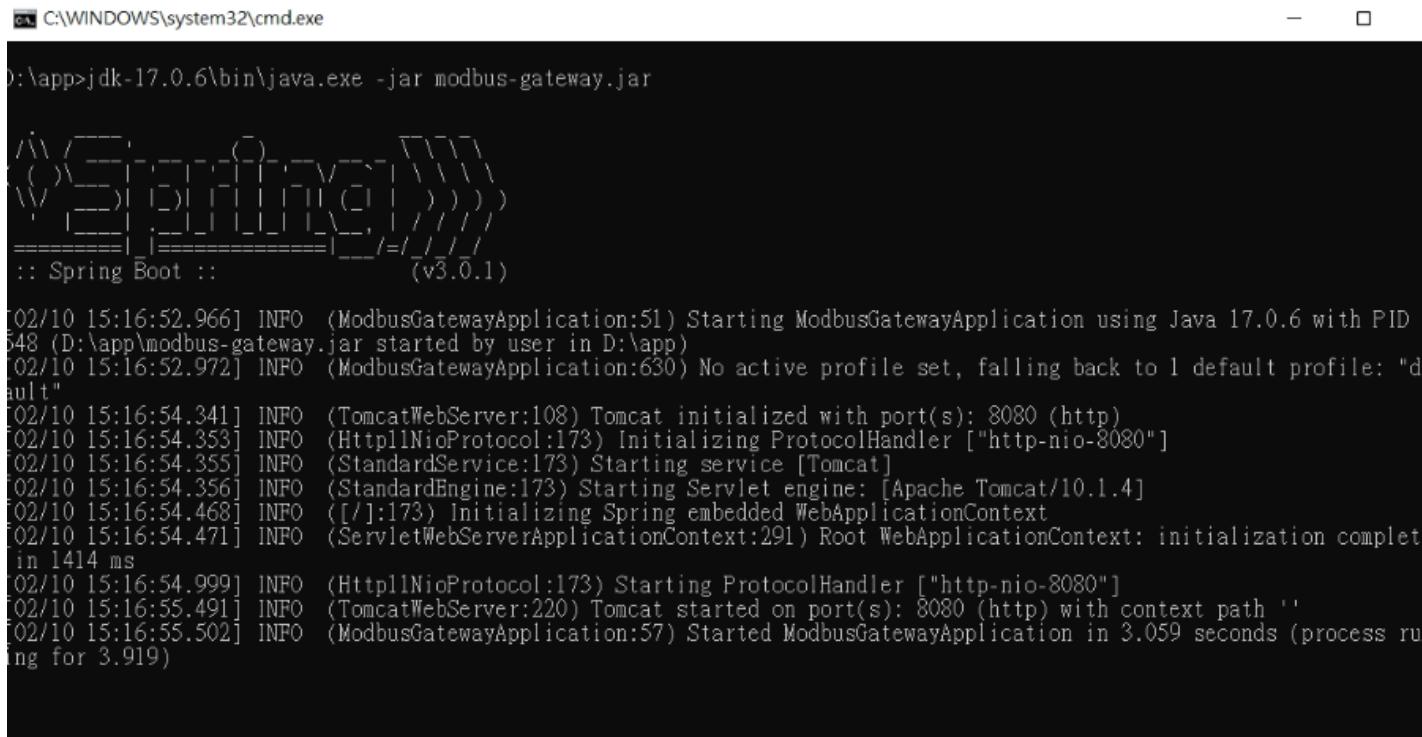
settings:
  slave:
    host: 192.168.1.2 ←
    port: 502
    slave-id: 1

  relays:
    - name: Relay1
      address: 1033
      quantity: 1

    - name: Relay2
      address: 1043
      quantity: 1
```

- host 為『物件篩選輸送帶』的 IP 位置。
- port 為『物件篩選輸送帶』的 Modbus/TCP 預設 Port Number，預設為 502。
- slave-id 為『物件篩選輸送帶』的 Modbus Device ID，預設為 1。
- relays 內設定不同噴嘴所對應的 Modbus Register Address，保持預設值不用變動。

4. 雙擊 run.bat 啟動主程式



```
C:\WINDOWS\system32\cmd.exe

D:\app>jdk-17.0.6\bin\java.exe -jar modbus-gateway.jar

:: Spring Boot :: (v3.0.1)

[02/10 15:16:52.966] INFO  (ModbusGatewayApplication:51) Starting ModbusGatewayApplication using Java 17.0.6 with PID 548 (D:\app\modbus-gateway.jar started by user in D:\app)
[02/10 15:16:52.972] INFO  (ModbusGatewayApplication:630) No active profile set, falling back to 1 default profile: "default"
[02/10 15:16:54.341] INFO  (TomcatWebServer:108) Tomcat initialized with port(s): 8080 (http)
[02/10 15:16:54.353] INFO  (Http1NioProtocol:173) Initializing ProtocolHandler ["http-nio-8080"]
[02/10 15:16:54.355] INFO  (StandardService:173) Starting service [Tomcat]
[02/10 15:16:54.356] INFO  (StandardEngine:173) Starting Servlet engine: [Apache Tomcat/10.1.4]
[02/10 15:16:54.468] INFO  ([/]:173) Initializing Spring embedded WebApplicationContext
[02/10 15:16:54.471] INFO  (ServletWebServerApplicationContext:291) Root WebApplicationContext: initialization completed in 1414 ms
[02/10 15:16:54.999] INFO  (Http1NioProtocol:173) Starting ProtocolHandler ["http-nio-8080"]
[02/10 15:16:55.491] INFO  (TomcatWebServer:220) Tomcat started on port(s): 8080 (http) with context path ''
[02/10 15:16:55.502] INFO  (ModbusGatewayApplication:57) Started ModbusGatewayApplication in 3.059 seconds (process running for 3.919)
```

『控制模組』使用 Spring Boot 框架開發，預設使用 TCP/8080 做為 RESTful API 的進入位置

5. RESTful API 控制方式

讀取第一組噴嘴狀態：

```
GET http://localhost:8080/api/relay/Relay1
```

回傳結果：1 代表噴氣中、0 代表待命中

指令範例：

```
curl -X GET http://localhost:8080/api/relay/Relay1
```

讀取第二組噴嘴狀態：

```
GET http://localhost:8080/api/relay/Relay2
```

回傳結果：1 代表噴氣中、0 代表待命中

指令範例：

```
curl -X GET http://localhost:8080/api/relay/Relay2
```

要求第一組噴嘴噴氣：

```
POST http://localhost:8080/api/relay/Relay1?value=1
```

指令範例：

```
curl -X POST "http://localhost:8080/api/relay/Relay1?value=1"
```

要求第二組噴嘴噴氣：

```
POST http://localhost:8080/api/relay/Relay2?value=1
```

指令範例

```
curl -X POST "http://localhost:8080/api/relay/Relay2?value=1"
```
