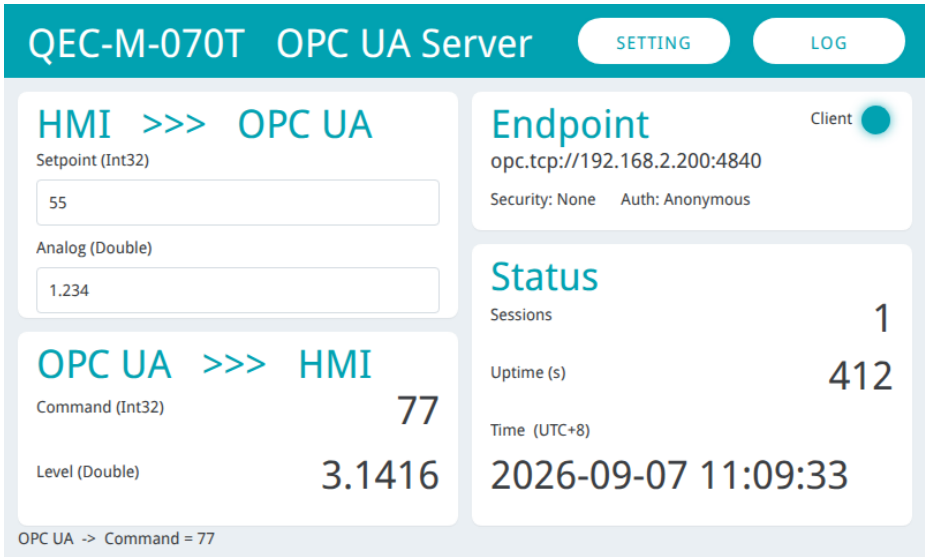


QEC OPC UA Server 測試工具 — 使用手冊 v 1.0

用 86Duino 讓 QEC 對外提供 OPC UA 資料，並從上位系統驗證整條連線

開發指南 | 版本 1.0 | September 2026

ICOP Technology Inc.



封面 Main 頁，已有一個 Client 連線。

修訂紀錄

版本	日期	說明
1.0	September 2026	首版。對應 QEC_OPCTUA_Server_HMI 釋出版：面板 REBOOT、Active now 對照欄、節點方向保護。

目錄

修訂紀錄	1
目錄	2
1. 概觀	3
1.1 系統架構	3
1.1.1 硬體需求	3
1.1.2 通訊架構	4
2. 建置 QEC MDevice	5
2.1 PC 端安裝	5
2.1.1 86Duino Coding IDE 502	5
2.1.2 open62541 library	5
2.1.3 UaExpert (用來驗證 Server)	5
2.2 程式架構說明	8
2.2.1 網路設定	9
2.2.2 HMI	9
2.2.3 應用邏輯	10
2.2.4 排程與執行	10
2.2.5 OPC UA Server	10
2.3 上傳程式	11
3. OPC UA 位址空間與節點契約	13
3.1 節點表	13
3.2 存取方向	14
3.3 連上之後會看到什麼	14
4. HMI 頁面功能說明	16
4.1 Main — 雙向資料流	16
4.2 Log — 面板上的即時 Console	17
4.3 Settings — 改 IP / Port 不必重新燒錄	18
CONFIRM 的驗證規則	20
二次確認	21
5. 實際跑一次：第一筆資料交換	22
5.1 這一章要做什麼	22
5.2 開始之前	22
5.3 步驟	22
5.4 連上之後	24
5.5 雙向各寫一次	24
過程中會一併看到的現象	25
5.6 這些訊息是正常的	25
6. 網路設定的持久化	27
6.1 為什麼要重開機才生效	27
6.2 Port 為什麼可以在面板上改	27
7. 疑難排解	28
8. 完整程式與下載	30
9. 後續延伸	31
9.1 接上你自己的上位系統	31
9.2 換成你自己的資料	31
9.3 讓 QEC 反過來當 Client	31
9.4 與 EtherCAT 共存	31
9.5 上線之前要處理的安全性	32
10. 聯絡我們	33

1. 概觀

QEC OPCUA_Server_HMI 是跑在 QEC-M 系列自帶螢幕的主站上的 OPC UA Server 測試工具。燒錄後開機即在 `opc.tcp://<IP>:4840` 上線，讓 PC 上的 UaExpert、現場的 SCADA 或 MES 直接連進來讀寫資料。

整套工具由三個部分組成，全部跑在同一顆 Vortex86EX2 上：

組成	實作	說明
OPC UA Server	open62541 v1.1.2 (86Duino / DMP 移植版)	Objects/QEC 底下 10 顆節點，Security None / Anonymous
HMI 操作面板	觸控面板，LVGL86 + 86HMI 設計工具	三頁：Main / Log / Settings
設定持久化	EEPROM bank 0	IP / Mask / Gateway / Port 可在面板上改，重開機生效

不需要外部 OPC UA Gateway、不需要在 PC 端另外跑 Server、不需要額外的 HMI 主機。

這支工具的用途有三個：

1. 驗證 QEC 與現場上位系統之間的 OPC UA 連線真的通
2. 確認雙向資料流成立 (面板輸入上位讀得到、上位寫入面板顯示得出來)
3. 在導入自己的應用之前，先確認網路設定與資料型別都對

1.1 系統架構

1.1.1 硬體需求

元件	建議規格	說明
QEC MDevice	QEC-M 系列自帶螢幕的主站	本手冊的截圖以 QEC-M-070T (Vortex86EX2，7" 800×480 電阻式觸控面板) 為例，系列中其他自帶螢幕的型號同樣適用
網路	1× Gigabit RJ45	OPC UA 走這個孔。另外 2× 10/100 RJ45 是 EtherCAT 專用，插錯不會有連線
上位系統 PC	Windows 10/11 或 Linux x86-64，與 QEC 同網段	跑 UaExpert 或現場既有的 SCADA / MES
電源	依機型規格	現場若無法斷電重開，面板的 REBOOT 鈕可以軟體重開機

軟體需求

元件	版本	說明
86Duino Coding IDE	502	QEC 官網公告版本
open62541 library	v1.1.2	由 IDE 的 Library Manager 安裝 (搜尋 OPC)

元件	版本	說明
86HMI 設計工具	隨 IDE 附帶	產生 <code>myhmi.h / myhmi.cpp</code>
UaExpert	2.x	Unified Automation 的免費 OPC UA Client，用來驗證

換到 QEC-M 系列的其他機型：程式本身不用改。HMI 版面是依 800×480 排的，若你的機型解析度不同，用 86HMI 設計工具重新排一次版即可（見 §8）。另外，`QEC.ModelName` 節點與開機 banner 裡的 "QEC-M-070T" 都是寫死的字串，換機型時記得一起改。

1.1.2 通訊架構

OPC UA 的連線方向永遠是 **Client 主動**、**Server 被動**，兩個 Client 之間無法直接連線。所以「誰是 Server」決定了整個系統的連線架構。



圖 1-1 系統架構與資料流：上位系統以 Client 身分連進 QEC，而 HMI 面板、OPC UA Server 與設定區都跑在同一顆 Vortex86EX2 上。

本工具中 QEC 扮演 **Server**。判斷方式很單純，看 sketch 用哪一組 API：

```

UA_Server_new() / UA_Server_run_startup() / UA_Server_run_iterate() → QEC 是 Server（本手冊）
UA_Client_new() / UA_Client_connect() / UA_Client_readValue...() → QEC 是 Client（另一份手冊）

```

`QEC.Sessions` 就是目前連進這台 QEC 的 Client 數量——只有 Server 端算得出來。

兩種角色可以在同一支 sketch 裡並存（QEC 對上位當 Server、對下游當 Client），library 兩邊都支援。

非阻塞架構：`loop()` 每一圈都呼叫 `UA_Server_run_iterate(server, false)`，不會卡住。這是日後要與 EtherCAT 共存的前提（見 §9）。

2. 建置 QEC MDevice

2.1 PC 端安裝

2.1.1 86Duino Coding IDE 502

1. 從 QEC 官網下載並安裝 **86Duino Coding IDE 502**
2. 用 USB 線連接 QEC MDevice 與 PC
3. 開啟 Windows **裝置管理員**，確認「連接埠 (COM port)」底下出現 QEC 的 COM port（範例中是 **COM12**）
4. IDE 的 **Tools** → **Board** 選你手上的 QEC-M 型號（本手冊是 **QEC-M-070T**）；**Tools** → **Port** 選上一步看到的 COM port

2.1.2 open62541 library

1. **Sketch** → **Include Library** → **Manage Libraries...**
2. 搜尋欄輸入 **OPC** → 找到 **open62541** → **Install**
3. 安裝完成後，**Sketch** → **Include Library** 的清單裡會出現 **open62541**，版本 **1.1.2**

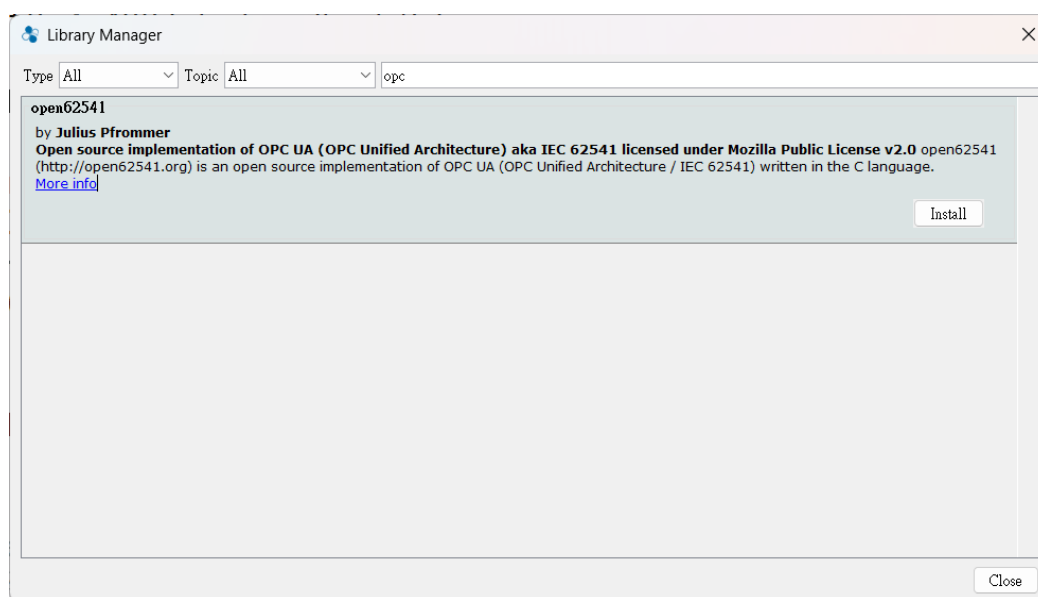


圖 2-1 Library Manager 搜尋opc 之後，清單裡的open62541 就是要裝的library；點Install，裝完在Sketch → Include Library 會看到版本 1.1.2。

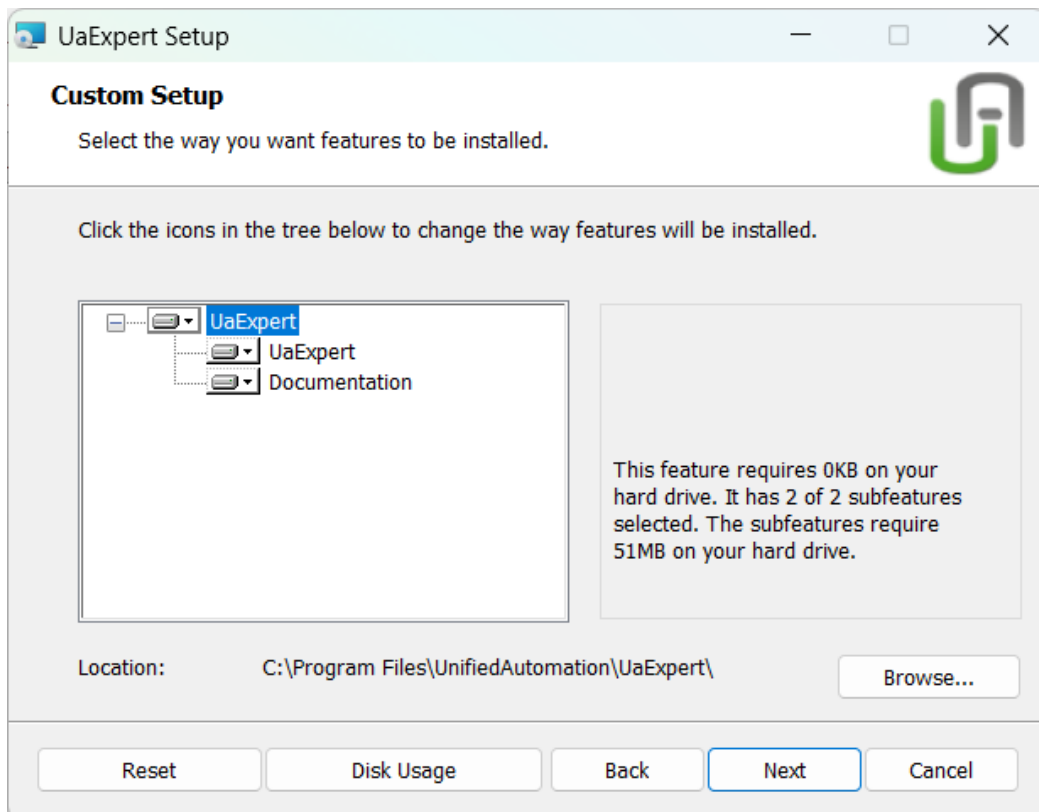
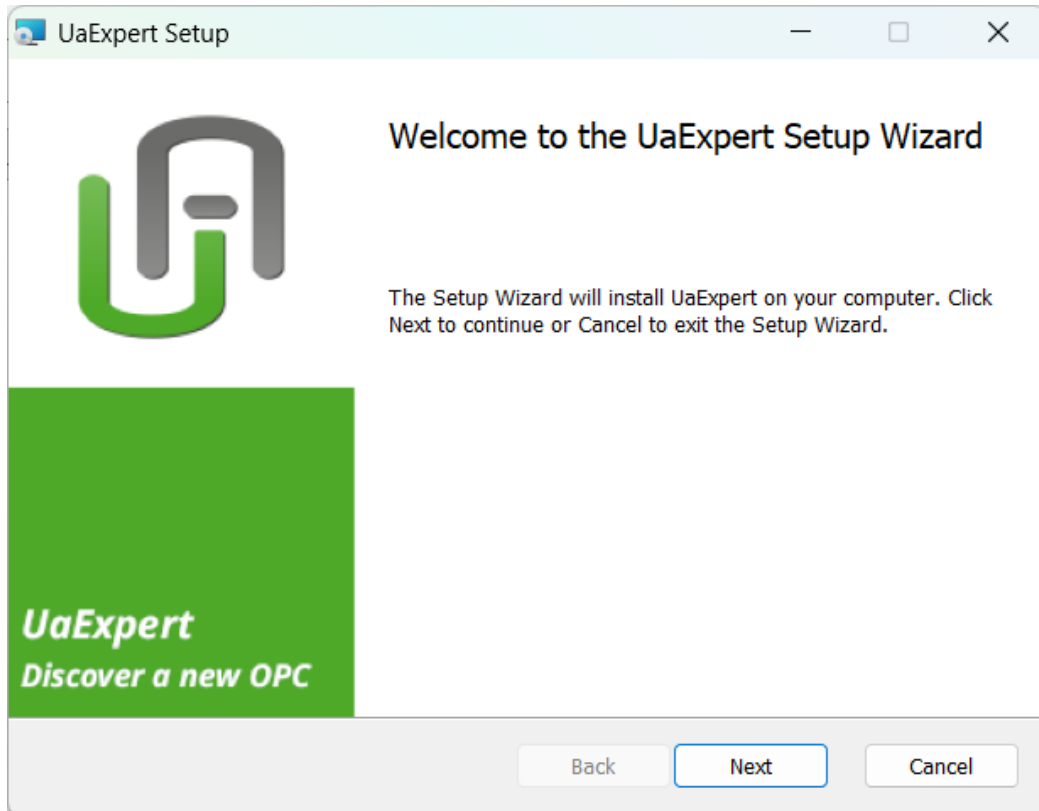
離線安裝：把 **open62541** 資料夾直接複製到 IDE 的 **libraries** 底下，重開 IDE。

升級注意：機器上若還留著更早的版本（資料夾名為大寫 **Open62541**），請先刪掉。新版 **library.properties** 的 **name** 是小寫 **open62541**，兩份並存時 IDE 可能挑到舊的。

2.1.3 UaExpert（用來驗證 Server）

UaExpert 是 Unified Automation 的免費 OPC UA Client，§5 就是用它連進 QEC 的 Server。

1. 到 Unified Automation 網站註冊帳號 (免費版就夠)
2. Downloads → OPC UA Clients → UaExpert → 下載並安裝
3. 安裝精靈一路 Next : Custom Setup 保持預設 (UaExpert + Documentation · 約 51 MB) → Install → Finish



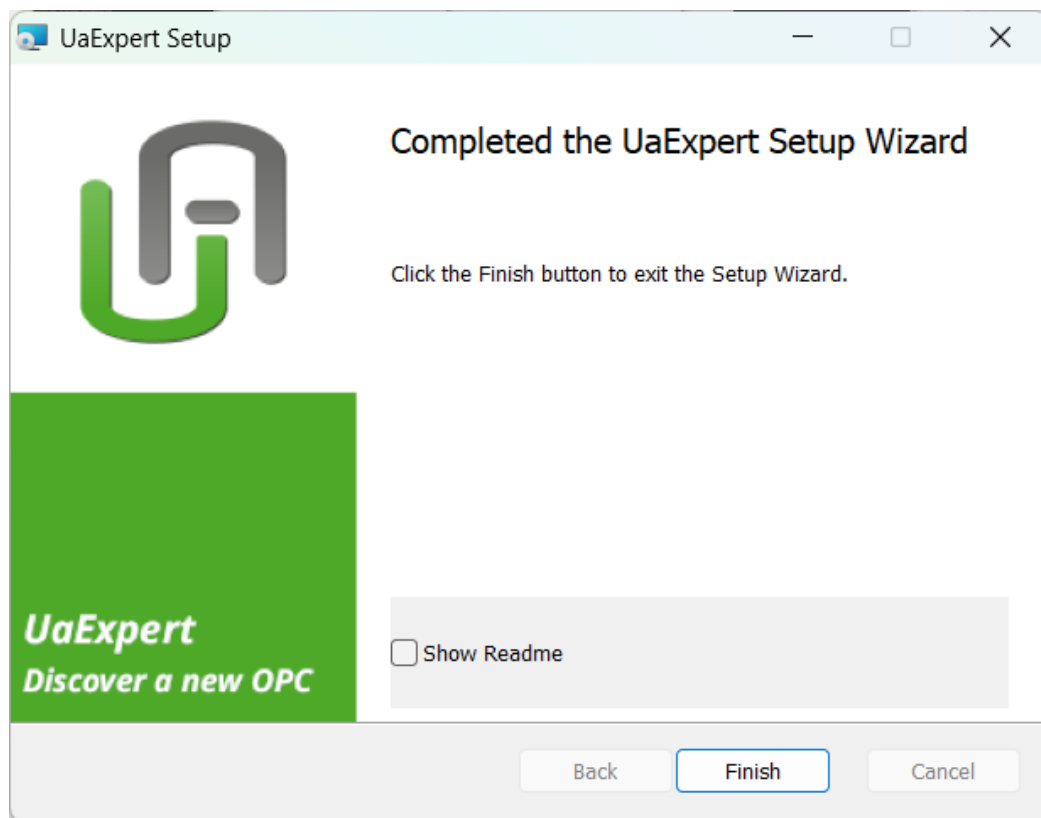


圖 2-2 / 2-3 / 2-4 安裝精靈的三個關鍵頁面。Custom Setup 不用動，預設就會把 UaExpert 與 Documentation 都裝進 C:\Program Files\UnifiedAutomation\UaExpert\。

1. 第一次啟動會跳出 **New Application Instance Certificate**，填好 Common Name 按 OK

New Application Instance Certificate

SUBJECT:

Common Name: UaExpert@ [redacted] ✓

Organization: [redacted] ✗

Organization Unit: [redacted] ✗

Locality: [redacted] ✗

State: [redacted] ✗

Country: [redacted] ✗

(Two letter code, e.g. DE, US, ...)

OPC UA INFORMATION

Application URI: urn:[redacted]:UnifiedAutomation:UaExpert ✓

Domain Names: [redacted] + ✓

IP Addresses: [redacted] + ✗

CERTIFICATE SETTINGS

RSA Key Strength: 2048 bits ▼ Signature Algorithm: Sha256 ▼ Certificate Validity: 5 Years ▼

☐ Password protect private key

Password: [redacted] ✗

Password (repeat): [redacted] ✗

OK CANCEL

圖 2-5 第一次啟動才會出現這個對話框。Common Name 預設是 UaExpert@<電腦名稱>，直接用預設值就好——這張憑證只是 UaExpert 用來表明自己身分的，與 QEC 這一端無關（Server 是 Security None，不驗憑證）。Organization 欄位旁的紅色標記不影響使用，留空即可。圖中的電腦名稱已遮蔽。

這一步做過一次就不會再出現。憑證存在 %APPDATA%\unifiedautomation\uaexpert\PKI\，換機器或砍掉重裝才需要重做。

2.2 程式架構說明

QEC_OPCTUA_Server_HMI.ino 約 720 行，可以分成五層來看：

分層	代表檔案 / 函式	功能
Network Settings	<code>cfgLoad() / cfgSave() / cfgDefaults() / parseIp() / maskValid() / Ethernet.begin()</code>	EEPROM 讀寫與驗證、靜態 IP 套用
HMI	<code>myhmi.h / myhmi.cpp</code> 、 <code>Hmi.*</code> 、 <code>qlog()</code> / <code>setStatus()</code> / <code>setCfgStatus()</code>	三頁版面、觸控鍵盤、面板 log、狀態列
Application Logic	<code>handleHmiEvents() / syncOpcToHmi() / tick1s() / doConfirm() / doDefault() / doReboot()</code>	按鍵處理、雙向資料同步、設定驗證與重開機
Task / Runtime	<code>setup() / loop() / helperYield() / BEGIN_HMI_EVENT_PROC</code>	單執行緒非阻塞排程：OPC UA 每圈跑、200 ms 同步、1 s tick
OPC UA Server	<code>UA_Server_new() / UA_ServerConfig_setMinimal() / buildAddressSpace() / addVar() / writeInt32() / readInt32()</code>	建立 server、節點樹、節點讀寫

2.2.1 網路設定

網路設定放在 **EEPROM bank 0 (EEPROM_16K)** 的前 18 bytes：

addr	內容
0	magic 'Q'
1	magic 'C'
2	version (目前 1)
3–6	IP
7–10	Subnet Mask
11–14	Gateway
15–16	Port (high, low)
17	checksum = XOR(0..16)

開機時 `cfgLoad()` 驗證 magic、version 與 checksum，任一項不符就自動回到 sketch 的 **DEF_*** 預設值，並在 log 標明 `no valid EEPROM config - using defaults`。所以第一次燒錄、或 EEPROM 被清空，機器一樣開得起來。

```
#define DEF_IP      "192.168.2.200"
#define DEF_MASK    "255.255.255.0"
#define DEF_GW      "192.168.2.1"
#define DEF_PORT     4840
```

出廠預設值只在 EEPROM 沒有有效資料時採用。建議把它們設成現場最常用的網段。

2.2.2 HMI

所有 log 同時輸出到 **Serial Monitor (115200)** 與 **面板 Log 頁**，時間戳是本地時間：

```
static void qlog(const char *s) {
    Serial.println(s);
    if (!hmiReady) return;
    /* ...timestamp, write to the TextList... */
    if (logNext > LOG_LINES) { /* full: clear but leave a marker */
        /* ... */
    }
}
```

Log 頁寫滿 99 行時會清空，並在第 1 行留下 ----- log full - cleared and restarted -----，避免「回來只剩幾行、還不知道被清過」。

2.2.3 應用邏輯

週期	做什麼
每圈	UA_Server_run_iterate(server, false)
200 ms	syncOpcToHmi()：讀 Command / Level 更新面板、更新 Sessions / Uptime / LED
1 s	tick1s()：Counter +1、寫 Uptime / Sessions、更新面板時鐘
事件	handleHmiEvents()：換頁、輸入套用、CONFIRM / DEFAULT / REBOOT

Setpoint / Analog 只在使用者按下確認時寫入 (Hmi.numberInputApplied())，不是每圈輪詢，所以打字過程不會一路寫進節點。

2.2.4 排程與執行

```
void loop() {
    if (server && serverOk) UA_Server_run_iterate(server, false);

    unsigned long now = millis();
    if (now - lastFast >= 200) { lastFast = now; syncOpcToHmi(); }
    if (now - lastSlow >= 1000) { lastSlow = now; tick1s(); }

    BEGIN_HMI_EVENT_PROC
    {
        handleHmiEvents();
    }
    END_HMI_EVENT_PROC
}
```

BEGIN_HMI_EVENT_PROC / END_HMI_EVENT_PROC 展開後是一個最多跑 128 次的 Hmi.eventPop() 迴圈，結尾呼叫 helperYield()——畫面刷新是 helperYield() 在做的。這一點在寫任何「等待」邏輯時都要記住：delay() 期間面板不會更新，要讓畫面動就得跑 helperYield()。

2.2.5 OPC UA Server

setup() 的順序很關鍵，任一步顛倒都會出問題：

```
putenv("TZ=CST-8"); tzset(); /* [0] time zone, before any UA_DateTime_now() */
Hmi.begin(); /* [1] HMI */
cfgLoad(); /* [2] settings */
```

```

Ethernet.begin(mac, ip, dns, gw, mask);          /* [3] static IP; shares the SWS stack with open62541 */
server = UA_Server_new();
UA_ServerConfig_setMinimal(cfg, cfgPort, NULL); /* [4] setMinimal() rather than setDefault() so the
configured port takes effect */
cfg->customHostname = UA_STRING_ALLOC(ipstr);    /*      publish the discovery URL as an IP, not the library
default host name */
cfg->accessControl.activateSession = ...;        /* ... session-counting hooks ... */
buildAddressSpace();                            /* ... [5] address space ... */
UA_Server_run_startup(server);

```

`addVar()` 每顆節點都補上 `attr.dataType` :

```

attr.dataType = type->typeId; /* without this UaExpert shows BaseDataType */

```

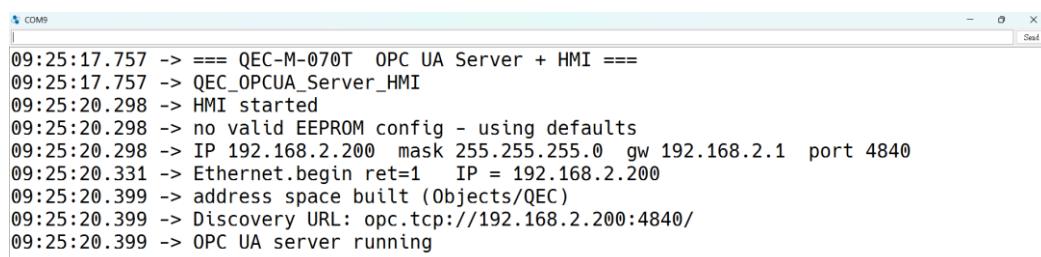
2.3 上傳程式

1. 確認 **DEF_IP** 等出廠預設值與現場網段相符
2. 按 IDE 的 **Upload** (→ 箭頭)
3. 編譯 + 上傳約 **30–60 秒**
4. 開啟 Serial Monitor (**115200**) · 應該在數秒內看到 :

```

=== QEC-M-070T OPC UA Server + HMI ===
QEC_OPCUA_Server_HMI
HMI started
config loaded from EEPROM
IP 192.168.2.200 mask 255.255.255.0 gw 192.168.2.1 port 4840
Ethernet.begin ret=1 IP = 192.168.2.200
address space built (Objects/QEC)
Discovery URL: opc.tcp://192.168.2.200:4840/
OPC UA server running

```



```

09:25:17.757 -> === QEC-M-070T OPC UA Server + HMI ===
09:25:17.757 -> QEC_OPCUA_Server_HMI
09:25:20.298 -> HMI started
09:25:20.298 -> no valid EEPROM config - using defaults
09:25:20.298 -> IP 192.168.2.200 mask 255.255.255.0 gw 192.168.2.1 port 4840
09:25:20.331 -> Ethernet.begin ret=1 IP = 192.168.2.200
09:25:20.399 -> address space built (Objects/QEC)
09:25:20.399 -> Discovery URL: opc.tcp://192.168.2.200:4840/
09:25:20.399 -> OPC UA server running

```

圖 2-6 Serial Monitor 的開機訊息。必須看得到 `Ethernet.begin ret=1` (不是 0) 與 `Discovery URL: opc.tcp://<IP>:4840/` (是 IP · 不是 86Duino) · 最後一行是 `OPC UA server running`。這台機器的設定從來沒有寫入過 · 所以第 4 行是 `no valid EEPROM config - using defaults` —— 見下方說明。前兩行是 `reset` 之後約 2 秒才印出來的 (`delay(2000)`) ; 它們跟後面幾行之間的空檔則是 `Hmi.begin()` 。

檢查點	必須是
<code>Ethernet.begin ret=</code>	1 (0 代表網路沒起來 · 見 §7)
<code>IP =</code>	你設定的 IP · 不是 0.0.0.0
<code>Discovery URL:</code>	<code>opc.tcp://<IP>:<port>/</code> · 不是 <code>opc.tcp://86Duino:4840/</code>
最後一行	<code>OPC UA server running</code>

第一次燒錄時，第 4 行會是 **no valid EEPROM config - using defaults**，圖 2-6 就是這個狀態。這不是故障：EEPROM 從來沒有被寫入過，所以 `cfgLoad()` 退回 sketch 的 `DEF_*` 值 (§2.2.1)，機器就用那組開起來。在 Settings 頁按一次 **CONFIRM**，之後每次開機都會顯示 **config loaded from EEPROM**。如果按過 **CONFIRM** 之後還是一直顯示 **no valid EEPROM config**，那才是真的有問題——見 §7。

上傳前請先在 UaExpert 按 **Disconnect**。上傳會 reset 板子，UaExpert 的 watchdog 重試幾次 timeout 後就不再自動重連，板子回來也不會自己接上 (見 §7)。

3. OPC UA 位址空間與節點契約

3.1 節點表

Root				
└ Objects				
└ Server			(OPC UA 標準節點, NS0)	
└ QEC			(Object, ns=1;s=QEC)	
└ Setpoint	Int32	RO	← 面板出, Client 只讀	
└ Analog	Double	RO	← 面板出, Client 只讀	
└ Command	Int32	RW	← Client 寫進來	
└ Level	Double	RW	← Client 寫進來	
└ Counter	Int32	RO		
└ Uptime	Int32	RO		
└ Sessions	Int32	RO		
└ ModelName	String	RO		
└ SketchVersion	String	RO		
└ IPAddress	String	RO		

NodeId	型別	存取	說明
ns=1;s=QEC.Setpoint	Int32	RO	面板 nin1 輸入 → 上位系統讀取
ns=1;s=QEC.Analog	Double	RO	面板 nin2 輸入 (浮點) → 上位系統讀取
ns=1;s=QEC.Command	Int32	RW	上位系統寫入 → 面板 num1 顯示
ns=1;s=QEC.Level	Double	RW	上位系統寫入 (浮點) → 面板 num2 顯示, 4 位小數
ns=1;s=QEC.Counter	Int32	RO	每秒 +1。訂閱它就看得到 DataChange 一直推過來
ns=1;s=QEC.Uptime	Int32	RO	開機秒數 (millis()) 約 49.7 天繞回一次, 屆時歸零)
ns=1;s=QEC.Sessions	Int32	RO	目前連線的 Client session 數
ns=1;s=QEC.ModelName	String	RO	"QEC-M-070T"
ns=1;s=QEC.SketchVersion	String	RO	"QEC_OPCTUA_Server_HMI" —— 即 QEC_SKETCH_VERSION 的值, 不帶版號
ns=1;s=QEC.IPAddress	String	RO	實際 bind 到的 IP。與 Settings 頁想設的值不同 → 代表還沒重開機

Server 規格

項目	內容
Endpoint	opc.tcp://<QEC_IP>:<port> (port 可在面板設定, 預設 4840)
SecurityPolicy	None (UA_ENABLE_ENCRYPTION 未編譯)
MessageSecurityMode	None
UserTokenType	Anonymous
ApplicationUri	urn:open62541.server.application
Subscription / DataChange	支援

項目	內容
Method Call	支援
Discovery (ServersOnNetwork)	未編譯 —— 必須手動輸入 Endpoint URL
Event / ModelChangeEvents	未編譯 —— 改節點後要手動 Refresh

3.2 存取方向

方向是節點契約的一部分。 **Setpoint** / **Analog** 對 Client 唯讀， **Command** / **Level** 才是給 Client 寫的：

方向	節點	誰是資料來源
HMI → OPC UA	Setpoint (Int32) 、 Analog (Double)	面板。Client 只能讀
OPC UA → HMI	Command (Int32) 、 Level (Double)	Client。面板只顯示
系統狀態	Counter / Uptime / Sessions / ModelName / SketchVersion / IpAddress	程式。Client 只能讀

為什麼 **Setpoint** / **Analog** 是唯讀？ 面板不會回讀這兩顆節點 (`syncOpcToHmi()` 只讀 **Command** / **Level**)。若開放 Client 寫，上位系統把 **Setpoint** 寫成 999 之後，節點是 999、面板還顯示舊值，兩邊不一致而且沒有任何提示。改成唯讀之後，寫錯方向會直接收到 **BadNotWritable**，而不是「看起來成功但面板沒反應」。

兩種型別 (Int32 / Double) 都還有可寫的節點 (**Command** / **Level**)，寫入測試不受影響。

註：`UA_Server_writeValue()` 是 server 端的本地寫入，不受 **accessLevel** 限制，所以程式照樣寫得進 RO 節點——**QEC.Counter** 一直是 RO 也照樣每秒 +1。

3.3 連上之後會看到什麼

任何一支標準 OPC UA Client 連進來，展開 **Objects/QEC** 就是上面那十顆節點。有三件事值得多看一眼，因為它們是「這條路真的通了」的直接證據：

- **DataType** 是具體型別。點 **Setpoint** 看 Attributes，會顯示 **Int32** 而不是 **BaseDataType** —— 代表上位系統不需要猜型別就能正確解讀。
- **Counter** 會自己遞增。拖進 Data Access View，Value 每秒 +1、Timestamp 跟著更新，這表示 Subscription 的 DataChange 推播是通的。
- 面板同時有反應。**Client** LED 全亮、**Sessions** 變成 1 —— 連線狀態在裝置端 就看得見，不必回上位系統查。

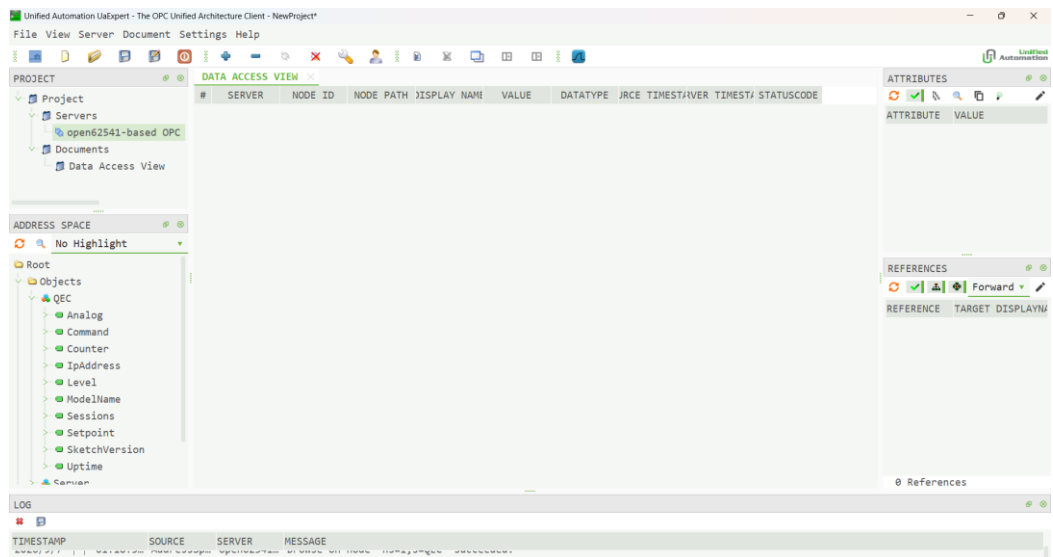


圖 3-1 UaExpert 的 Address Space · Root / Objects / QEC 展開、10 類節點都看得到。

4. HMI 頁面功能說明

三頁，Material / Light 亮色主題，版面依 800×480 排。

4.1 Main — 雙向資料流

左半邊是資料流的兩個方向，右半邊是連線資訊：endpoint、Client LED、Sessions 與 Uptime。

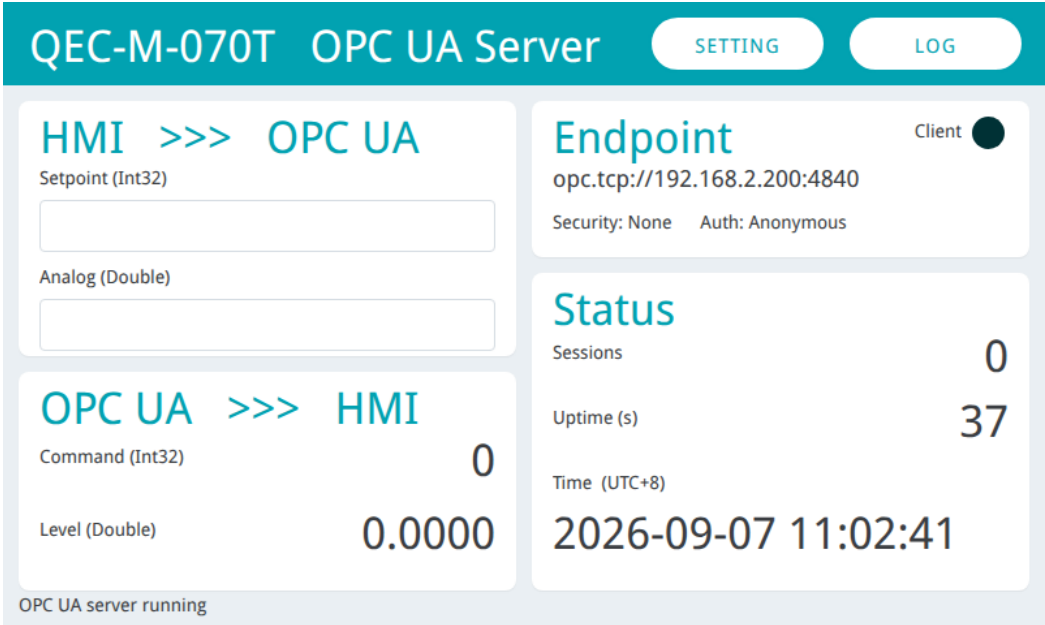


圖 4-1 Main 頁未連線：Client LED 微亮、Sessions = 0。

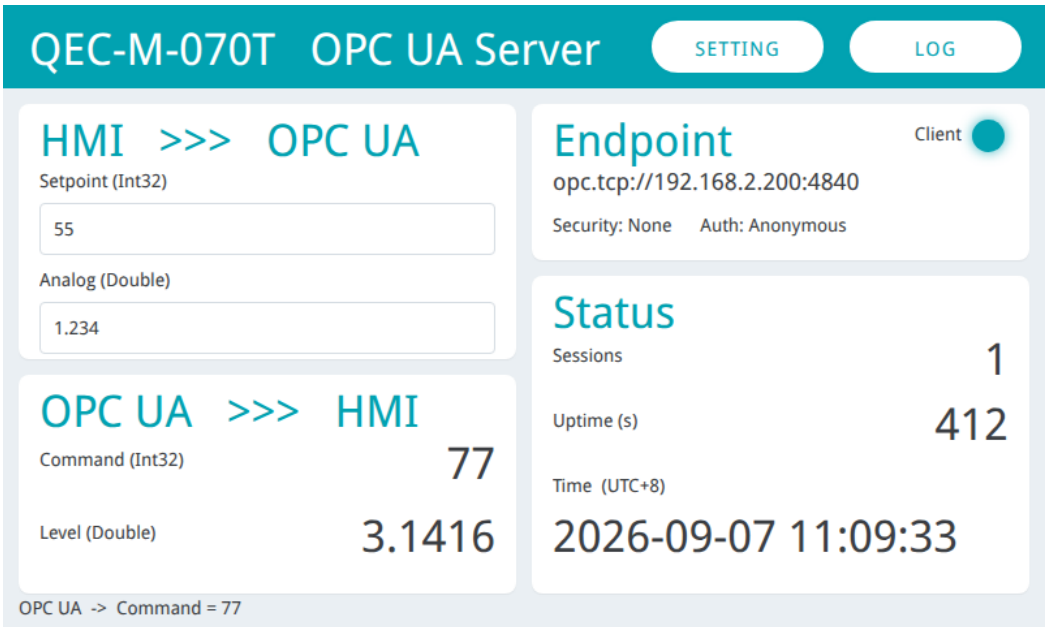


圖 4-2 Main 頁已連線：LED 全亮、Sessions = 1、Setpoint = 55、Analog = 1.234、Command = 77、Level = 3.1416，狀態列顯示 OPC UA -> Command = 77。

欄位 / 按鈕	功能
Setpoint (Int32) 輸入框	點一下叫出觸控數字鍵盤 → 輸入 → 確認，立刻寫進 QEC.Setpoint

欄位 / 按鈕	功能
Analog (Double) 輸入框	同上，浮點，寫進 QEC.Analog
Command (Int32) 顯示	QEC.Command 的現值，每 200 ms 更新
Level (Double) 顯示	QEC.Level 的現值，4 位小數
Endpoint	開機時由程式填入實際生效的 IP 與 port，不是寫死的字串
Client + LED	有 Client 連線時全亮、無連線時微亮
Sessions	目前連線的 Client 數
Uptime (s)	開機秒數
Time (UTC+8)	本地時間，時區標籤由程式依 TZ 計算
狀態列 (底部)	最新一筆事件，與 Log 頁最後一行相同。寬 769 px，長訊息不會截字
SETTING	切到 Settings 頁 (同時把欄位重新填成已儲存的值，沒按 CONFIRM 的編輯會被丟掉)
LOG	切到 Log 頁

事件訊息範例

```
HMI -> Setpoint = 55
HMI -> Analog = 1.2340
OPC UA -> Command = 77
OPC UA -> Level = 3.1416
client connected (sessions = 1)
client disconnected (sessions = 0)
```

寫入失敗時訊息尾端會多一個 **FAILED**，同時在 log 留下 `write QEC.Setpoint failed: <StatusCode>`。

4.2 Log — 面板上的即時 Console

機器輸出的每一則訊息都在這裡，與 Serial Monitor 同步。

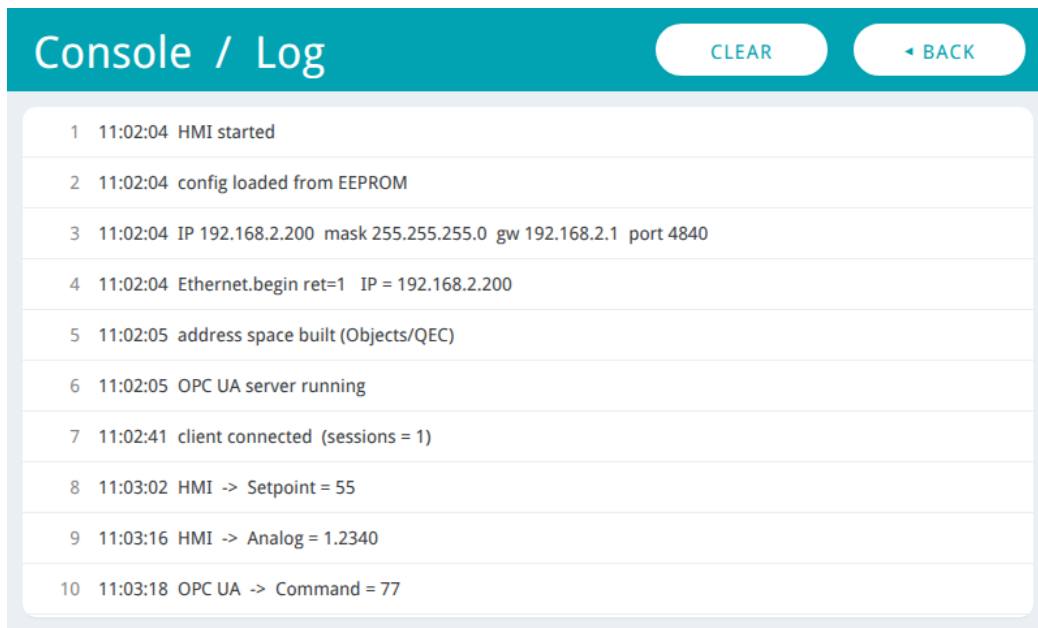


圖 4-3 Log 頁，含開機那幾行 + client connected (sessions = 1) + 一筆 HMI -> Setpoint。

欄位 / 按鈕	功能
TextList (p2t11)	99 行、帶行號、自動捲到最新一行
寫滿時	清空並在第 1 行寫 ----- log full - cleared and restarted -----
CLEAR	清空並從第 1 行重新開始
BACK	回 Main 頁

內容與 Serial Monitor 一致 —— 只有開機那兩行 (=== ... === 與 sketch 名稱) 例外，它們在 `Hmi.begin()` 之前就印出去了，只會出現在 Serial。現場只有面板、沒有筆電時，這一頁就是唯一的除錯視窗。

4.3 Settings — 改 IP / Port 不必重新燒錄

左欄是存進 EEPROM 的值，右欄 **Active now** 是現在真正在跑的值，兩欄並排就看得出改了沒生效。

Settings ◀ BACK

Network Configuration

Active now

IP Address

192.168.2.200

IP

192.168.2.200

Subnet Mask

255.255.255.0

Mask

255.255.255.0

Gateway

192.168.2.1

GW

192.168.2.1

OPC UA Port

4840

Port

4840

Stored in EEPROM - press REBOOT (or power-cycle) to apply

DEFAULT

REBOOT

✓ CONFIRM

圖 4-4 Settings 頁剛進入時：左欄與右欄 Active now 相同。

Settings ◀ BACK

Network Configuration

Active now

IP Address

192.168.2.210

IP

192.168.2.200

Subnet Mask

255.255.255.0

Mask

255.255.255.0

Gateway

192.168.2.1

GW

192.168.2.1

OPC UA Port

4840

Port

4840

Stored in EEPROM - press REBOOT (or power-cycle) to apply

DEFAULT

REBOOT

✓ CONFIRM

Saved: 192.168.2.210:4840 - press REBOOT to apply

圖 4-5 左欄已改成新 IP、右欄 Active now 還是舊值，狀態列顯示 Saved: ... - press REBOOT to apply。這張最能說明「存了但還沒生效」。

欄位 / 按鈕	功能
IP Address / Subnet Mask / Gateway	TextInput，點一下叫出觸控鍵盤
OPC UA Port	NumberInput，1–65535
Active now (右欄)	開機當下真正生效的值。與左欄不同 → 代表改過設定但還沒重開機
CONFIRM	驗證後寫入 EEPROM
DEFAULT	回到 sketch 的 DEF_* 出廠預設值並寫入。需 3 秒內按兩次
REBOOT	軟體重開機讓設定生效。需 3 秒內按兩次
狀態訊息 (底部)	驗證失敗的原因、存檔結果、二次確認提示

欄位 / 按鈕	功能
BACK	回 Main 頁

CONFIRM 的驗證規則

任一項不通過就顯示原因、不寫入 EEPROM：

檢查	失敗訊息
IP 為四段 0-255	IP address format error
Subnet 為四段 0-255	Subnet mask format error
Gateway 為四段 0-255	Gateway format error
Port 介於 1-65535	Port must be 1 - 65535
IP 第一段不得為 0	IP address must not start with 0
遮罩必須連續	Subnet mask must be contiguous, e.g. 255.255.255.0
IP 不能是網路位址	IP is the network address of this subnet
IP 不能是廣播位址	IP is the broadcast address of this subnet

Settings

Network Configuration

Active now

IP Address: 192.168.2.200

Subnet Mask: 255.0.255.0

Gateway: 192.168.2.1

OPC UA Port: 4840

IP: 192.168.2.200

Mask: 255.255.255.0

GW: 192.168.2.1

Port: 4840

Stored in EEPROM - press REBOOT (or power-cycle) to apply

DEFAULT REBOOT CONFIRM

Subnet mask must be contiguous, e.g. 255.255.255.0

圖 4-6 Subnet 填 255.0.255.0 後按 CONFIRM，狀態列顯示 Subnet mask must be contiguous, e.g. 255.255.255.0。

遮罩連續性用位元判斷（255.0.255.0 這種會被擋掉，存進去重開機就整台連不上）；網路 / 廣播位址是依遮罩判斷的，所以 192.168.5.0 在 /16 底下仍然合法。

Gateway 不在同網段只提示不阻擋（內網測試常常不需要 gateway），存檔訊息會變成 Saved: ... - WARNING gateway not in this subnet。

二次確認

DEFAULT 與 REBOOT 都會覆寫現場設定或中斷連線，所以都要在 **3 秒內按兩次**：

第一次按 → 狀態列：Press REBOOT again within 3 s to restart the device
第二次按 → 執行

Settings

Network Configuration

Active now

IP Address	192.168.2.200	IP	192.168.2.200
Subnet Mask	255.255.255.0	Mask	255.255.255.0
Gateway	192.168.2.1	GW	192.168.2.1
OPC UA Port	4840	Port	4840

Stored in EEPROM - press REBOOT (or power-cycle) to apply

DEFAULT REBOOT ✓ CONFIRM

Press REBOOT again within 3 s to restart the device

圖 4-7 按一次 REBOOT 後，狀態列顯示 Press REBOOT again within 3 s to restart the device。

兩顆鈕互斥：按 REBOOT 會取消已經按過一次的 DEFAULT，反過來也一樣；按 CONFIRM 則是兩個都取消。進入 Settings 頁時也會全部歸零。

5. 實際跑一次：第一筆資料交換

5.1 這一章要做什麼

用 PC 上的 **UaExpert** 當上位系統，走一次最短的路：連上 → 看到節點 → 雙向各寫一次值。全部做完大約十分鐘，中間不需要寫任何程式。

跑完這一遍，你就確認了三件事：QEC 在你的網路上以標準 OPC UA 的方式被找得到、它公開的資料型別是正確的、而且資料可以雙向流動。接下來要接的是 SCADA、MES 還是你自己寫的程式，對 QEC 來說沒有差別——它們用的是同一套標準介面。

UaExpert 只是這裡最方便的示範對象。手邊已經有 SCADA / MES，或是你自己用 Python `opcua`、C# OPC Foundation SDK 寫的 client，都可以直接拿來連，端點與節點名稱完全一樣。

5.2 開始之前

1. QEC MDevice 已燒錄 **QEC OPCUA_Server_HMI**，Serial 出現 **OPC UA server running**
2. 網路線插在 **Gigabit RJ45** 孔（不是那兩個 10/100 EtherCAT 孔）
3. PC 與 QEC 同網段（PC 上 `ipconfig` 確認）
4. PC 已安裝 UaExpert（安裝步驟見 §2.1.3）

5.3 步驟

#	步驟	操作
1	安裝 UaExpert	Unified Automation 網站註冊 → Downloads → OPC UA Clients → UaExpert → 下載安裝
2	建立憑證	首次啟動會要求建立 Application Instance Certificate ，填 Common Name 後產生
3	開啟 Add Server	Server → Add...
4	輸入 Endpoint	展開 Custom Discovery → 雙擊 < Double click to Add Server... > → 輸入 <code>opc.tcp://192.168.2.200:4840</code> → Enter
5	展開並選安全設定	點那一列左邊的 > 展開，選底下的 None - None (uatcp-uasc-uabinary)
6	選驗證方式	Authentication Settings 選 Anonymous → OK
7	連線	在 Project 樹的該 Server 上按右鍵 → Connect
8	確認連上	UaExpert Log 出現 changed to 'Connected' ；QEC 面板 LED 全亮、 Sessions = 1
9	瀏覽節點	Address Space 展開 Root / Objects / QEC ，確認 10 顆節點

#	步驟	操作
10	拖進 DA View	把 Setpoint / Analog / Command / Level / Counter 拖進 Data Access View

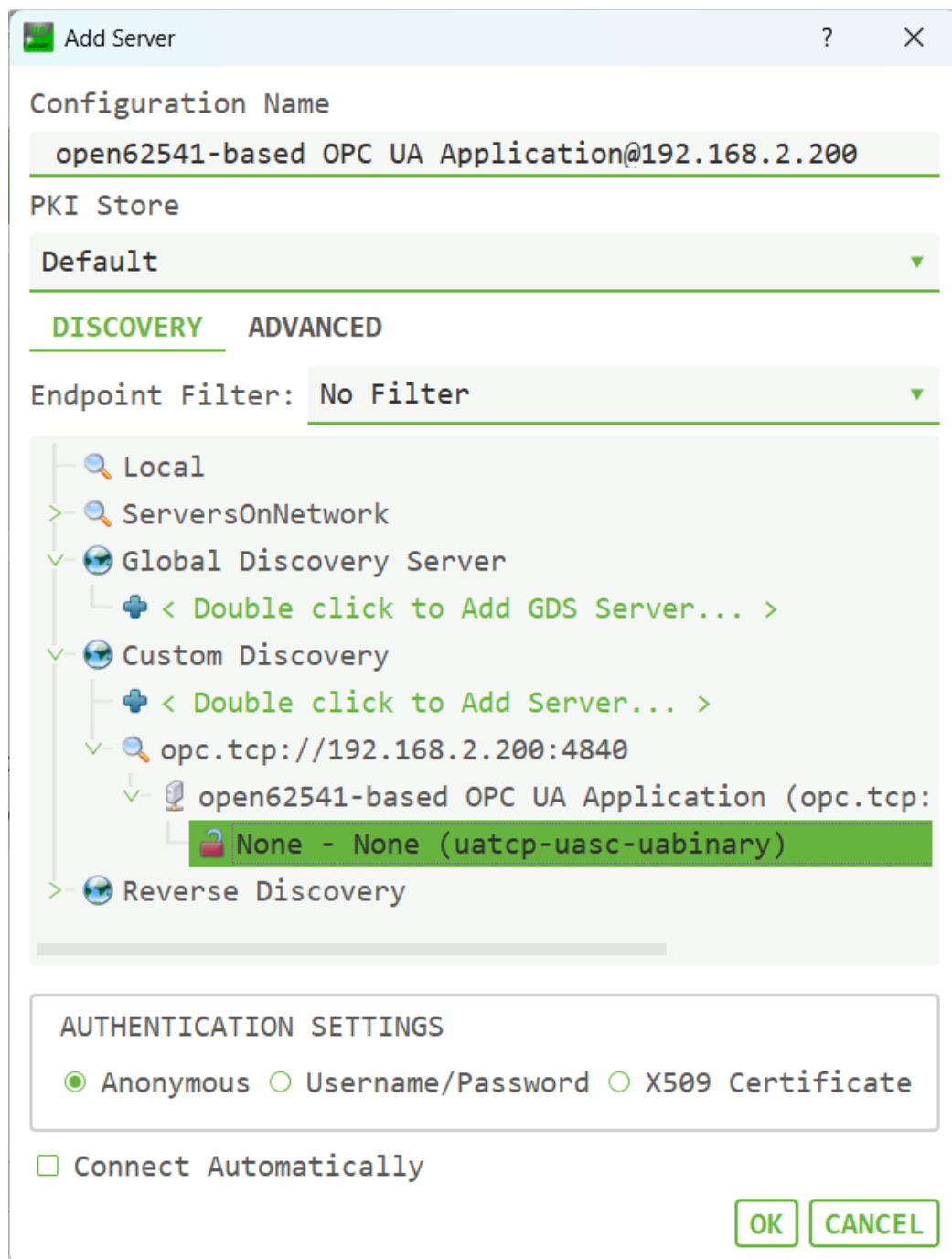


圖 5-1 Add Server 對話框：Endpoint 已展開、None - None (uatcp-uasc-uabinary) 被選中、Authentication 是 Anonymous。注意 endpoint 是用 IP 輸入的。

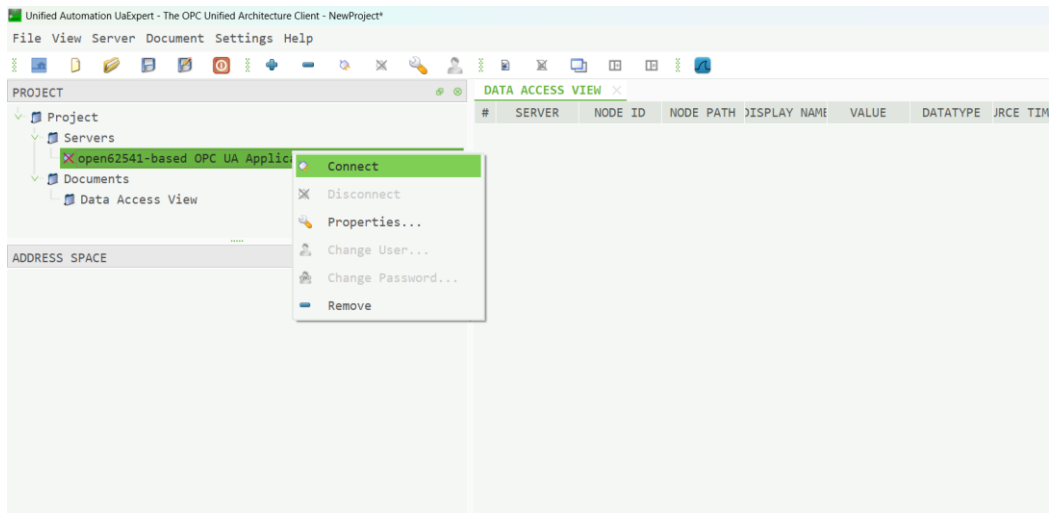


圖 5-2 在 Server 上按右鍵 → Connect 的功能表。

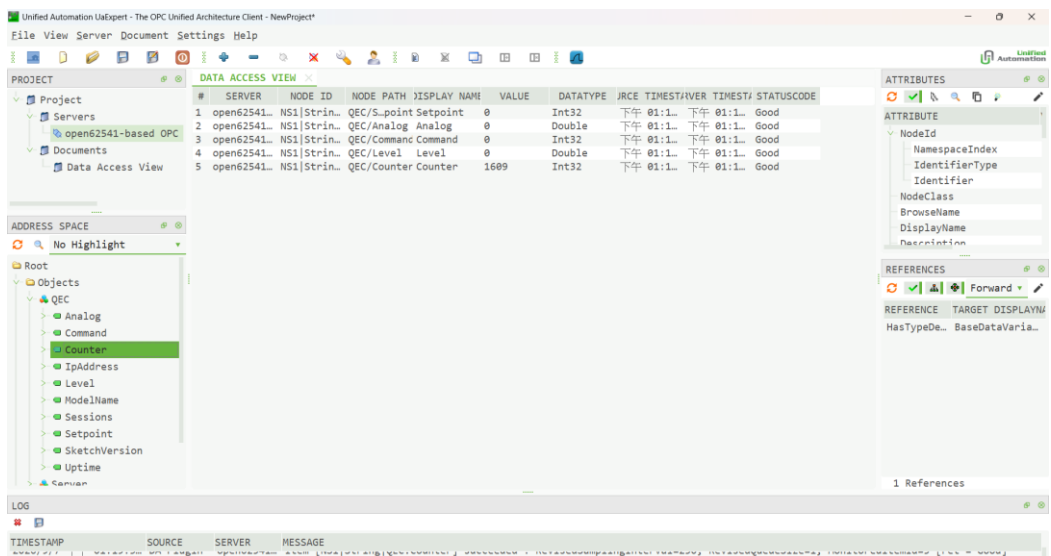


圖 5-3 已連線，DA View 有 5 列（含持續遞增的 Counter），DataType 顯示 Int32 / Double，StatusCode 全 Good。圖中 Counter 是 1609，每秒 +1。

第 5 步是最容易卡住的地方。只選到 URL 那一列時 OK 按鈕是灰的，必須展開後選底下的 **None - None (uatcp-uasc-uabinary)**。另外 **Local Network** 掃描永遠掃不到這台機器（Discovery 服務未編譯），一定要手動輸入 URL。

5.4 連上之後

UaExpert 的 Log 會出現 **changed to 'Connected'**，Address Space 長出 **Objects/QEC**；同一時間 QEC 面板的 **Client LED** 全亮、Sessions 變成 1、Log 頁多一行 **client connected (sessions = 1)**。兩端同時有反應，就是連線成立的意思。

5.5 雙向各寫一次

這一節只有兩個動作，一個方向一次。做完就代表整條路是通的。

面板 → 上位系統

在 Main 頁的 **Setpoint** 欄位輸入 **55** 並確認，UaExpert 的 DA View 裡 **QEC/Setpoint** 立刻變成 55，面板狀態列顯示 **HMI -> Setpoint = 55**。浮點的話用 **Analog** 輸入 **1.234**。

上位系統 → 面板

在 UaExpert 對 **QEC/Command** 寫入 **77**，面板的 **Command** 在 200 ms 內顯示 **77**，狀態列顯示 **OPC UA -> Command = 77**。浮點的話寫 **QEC/Level = 3.1416**，面板會顯示到小數點後 4 位。

過程中會一併看到的現象

你會看到	意思
Counter 每秒 +1、Timestamp 跟著跳	Subscription 的 DataChange 推播正常
Source Timestamp 與面板 Time 一致、不差 8 小時	時區設定正確 (TZ=CST-8)
寫 QEC/Setpoint 會被拒、回 BadNotWritable	方向保護生效 (§3.2)。這是預期行為，不是故障
UaExpert 按 Disconnect 後面板 Sessions 回到 0	連線狀態在裝置端就看得到的
斷電再開，log 出現 config loaded from EEPROM	網路設定存在 EEPROM 裡，不會掉 (§6)

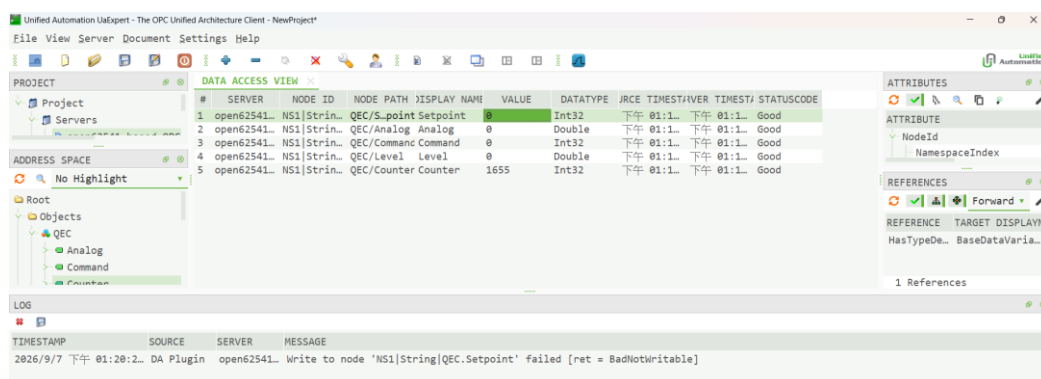


圖 5-4 對唯讀的 **QEC/Setpoint** 寫入時，Server 回 **BadNotWritable**。寫錯方向會當場被擋下來，而不是「看起來成功、面板卻沒反應」。

到這裡，**QEC** 的 **OPC UA** 介面就算確認可用了。節點名稱、型別、方向都在 §3，要加自己的節點只是在 **buildAddressSpace()** 多一行 (§9)。

5.6 這些訊息是正常的

第一次接觸這支工具的人最常把下面幾則當成故障，其實都不是：

現象	說明
Add Server 時 OK 按鈕是灰的	只選到 discovery URL 那一列。點 > 展開，選底下的 None - None (uatcp-uasc-uabinary)
Local Network 掃不到 QEC	正常，Discovery 未編譯。必須手動輸入 Endpoint URL
The server returned no certificate	正常，SecurityPolicy 是 None，忽略就好

現象	説明
Register for ModelChangeEvents failed: BadNotSupported	正常・Event 訂閱未編譯。副作用只有一個：改節點後要手動 Rebrowse

6. 網路設定的持久化

改網路設定不必重新編譯、不必重新燒錄，也不必斷電重開。現場換網段時，這是 QEC 與「IP 寫死在韌體裡」的裝置最主要的差異。

機制	目的	作法
EEPROM 設定區	讓 IP / Mask / GW / Port 在斷電後保留	bank 0 前 18 bytes · magic + version + XOR checksum；任一項不符自動回到 DEF_*
Active now 欄	分清「已存但還沒生效」與「正在跑」	Settings 頁右欄，顯示開機當下生效的那組值；每次進 Settings 頁由 activeToUi() 重新填入
REBOOT 鈕	讓「存檔 → 生效」在面板上就能走完	3 秒內按兩次

6.1 為什麼要重開機才生效

OPC UA Server 在 UA_Server_run_startup() 時就已經 bind 到當時的 IP，customHostname 與 discoveryUrl 也在那一刻定案。要換 IP 必須把 SWS stack 重新初始化（SwsSock::init() 內部會先 Sws_SockCleanup()）再整個重建 server——在單執行緒的 loop() 裡做這件事風險過高。

因此設計為 存檔 → 重開機 → 套用，這是對既有連線影響最小的作法。

6.2 Port 為什麼可以在面板上改

原廠 UA_ServerConfig_setDefault() 把 port 寫死 4840，本專案直接呼叫底層的 setMinimal() 並帶入設定值：

```
UA_ServerConfig_setMinimal(cfg, cfgPort, NULL);
```

7. 疑難排解

問題	可能原因	解法
編譯錯誤 'p3b4' was not declared in this scope	改過 <code>data_project.hpj</code> 的版面之後沒有用 86HMI 設計工具存檔 —— <code>myhmi.h/cpp</code> 還是舊的，缺少新元件	開設計工具載入 <code>data_project.hpj</code> 存檔一次，重新產生 <code>myhmi.h/cpp</code> ，再編譯。沒有改版面的話不會遇到（這兩個檔隨專案附上）
Serial 印 IP = 0.0.0.0、Ethernet.begin ret=0	網路線插錯孔 / IP 與他人衝突 / EEPROM 值異常	確認插 Gigabit 孔；換一個沒被占用的 IP；面板 SETTING → DEFAULT (×2) → REBOOT (×2)
Discovery URL: 顯示 86Duino 而不是 IP	customHostname 沒設到	確認 .ino 在 run_startup() 之前有 <code>cfg->customHostname = UA_STRING_ALLOC(ipstr)</code>
開機停在 Log 頁、顯示 run_startup FAILED	IP 衝突 / port 被占用 / 網路線沒插 Gigabit 孔	面板按鍵仍然可用： BACK → SETTING 改設定或按 DEFAULT → REBOOT 。不必重新燒錄
UaExpert 加入時 OK 按鈕是灰的	只選到 discovery URL 那一列	點 > 展開，選底下的 None - None (uatcp-uasc-uabinary)
UaExpert 連上但 Browse 空白 / 一直重連	Discovery URL 不是 IP	見上面第 3 列
BadNodeIdUnknown	UaExpert 還在用前一次連線快取下來的節點清單	Disconnect → Remove server → 重新 Add；DA View 裡舊的那幾列也一起刪掉重拖
新增節點後 UaExpert 看不到	Event / ModelChangeEvents 未編譯	手動 Rebrowse
Source Timestamp 不會更新	正常，Subscription 是 report-by-exception	訂閱 QEC.Counter 或 Server/ServerStatus/CurrentTime
UaExpert 的 Timestamp 差 8 小時	TZ 沒設或設錯	.ino 開頭 <code>#define QEC_TZ "TZ=CST-8"</code> （台北 UTC+8，POSIX 寫 -8）；RTC 存 UTC 時改 <code>"TZ=UTC0"</code>
重新上傳後 UaExpert 連不回來，一串 BadTimeout	正常——上傳會 reset 板子，UaExpert 的 watchdog 放棄重連	右鍵 Connect；或 Disconnect → 等 3 秒 → Connect；或 Remove 後重新 Add。養成上傳前 Disconnect 的習慣
Settings 改了但沒生效	沒重開機	按 REBOOT （3 秒內兩次），或手動斷電重開。對照 Settings 頁右欄 Active now
改了 Port 之後連不上	UaExpert 還在用舊 port	Endpoint 改成 <code>opc.tcp://<IP>:<新 port></code>
REBOOT 按了沒反應	這個 IDE 版本的 port IO 行為不同	改成手動斷電重開就好；其他功能都不受影響
每次開機都顯示 no valid EEPROM config	EEPROM 沒寫入成功或 checksum 不符	在 Settings 頁按一次 CONFIRM ；仍失敗則 EEPROM 可能異常
狀態列訊息被截字	那個 Label 的寬度被設計工具依文字縮小了	見 §8 的警告；確認 p1116 寬度仍在 700 px 以上

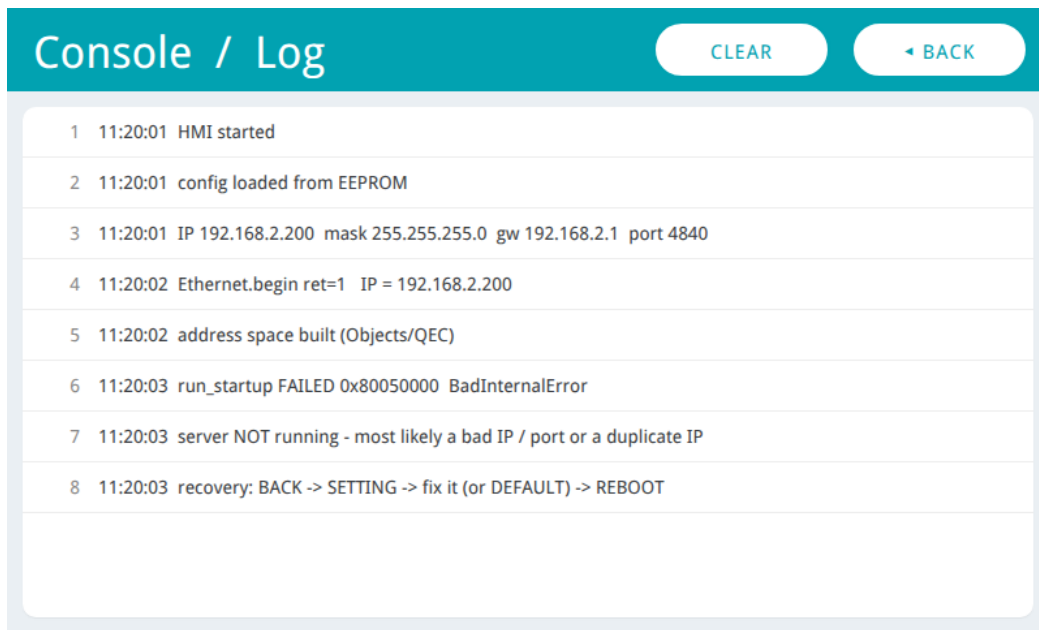


圖 7-1 `run_startup()` 失敗時的樣子：面板跳到 Log 頁、印出狀態碼與救援路徑。按鍵仍然有反應，所以一個設錯的 IP 不會讓面板失去操作能力、只能重新燒錄。

8. 完整程式與下載

QEC_OPCUA_Server_HMI\ ├─ QEC_OPCUA_Server_HMI.ino ├─ README.md ├─ CHANGELOG.md ├─ myhmi.h ├─ myhmi.cpp ├─ data\ └─ _project.hpj	主程式（約 720 行，手改這個） 開發者技術文件（節點表、五條 library 限制、繞法） 程式的設計決策與實測記錄 86HMI 設計工具產生（勿手改） 86HMI 設計工具產生（勿手改） HMI 版面專案（3 頁，用設計工具改）
--	--

檔案	說明
QEC_OPCUA_Server_HMI.ino	全部的應用邏輯。五層結構見 §2.2
data_project.hpj	HMI 版面。改版面請用設計工具，不要直接編輯 myhmi.*
README.md	含 86Duino 版 open62541 的五條限制與本專案的繞法（固定 IP、Discovery URL、時區、 attr.dataType 、session 計數）。library 原始碼一行都沒有修改
CHANGELOG.md	含設計工具會覆寫 Label 寬度等已知問題與實作注意事項

⚠ 要改 HMI 版面的話先讀這段：Label 寬度會被設計工具覆寫。在 .hpj 的 JSON 裡設 Label 的 **width** 沒有用——用 86HMI 設計工具存檔時，工具會依「設計期文字」重新 auto-fit 每個 Label 的寬度（實測約 8.15 px/字）。所以本專案的佔位文字都刻意填成「最長的現實情況」，例如狀態列 **p1116** 的佔位字是 94 個字元，存檔後 auto-fit 到 **769 px**。改動這些 Label 的文字之後，請回頭確認寬度沒有被縮小，否則執行時的長訊息會被截字。改完版面記得存檔，**myhmi.h / myhmi.cpp** 才會跟著更新。

範例程式 ZIP 下載：https://www.qec.tw/wp-content/uploads/2026/09/QEC_OPCUA_Server.zip

9. 後續延伸

確認 QEC 的 OPC UA 介面可用之後，接下來就是把它接進你自己的系統。這一節列的是常見的下一步，以及每一步大概要動到哪裡。

9.1 接上你自己的上位系統

這一步通常不用改 QEC 這邊的任何東西。手上的 SCADA / MES / historian 只要是標準 OPC UA Client，填入同一個 endpoint 就會連上；自己寫程式的話，Python 的 `opcua`、C# 的 OPC Foundation SDK、Node-RED 的 OPC UA 節點都可以。

需要給對方的資訊只有三項：

項目	值
Endpoint	<code>opc.tcp://<QEC 的 IP>:<port></code> (面板 Main 頁那一行照抄)
安全性	SecurityPolicy None 、MessageSecurityMode None 、 Anonymous
節點	<code>ns=1;s=QEC.<名稱></code> ，清單見 §3.1

9.2 換成你自己的資料

目前那十顆節點是示範用的。要換成現場真正的資料，在 `buildAddressSpace()` 加一行就有一顆新節點：

```
addVar("QEC.MyValue", "MyValue", &UA_TYPES[UA_TYPES_FLOAT], &f0, RW);
```

三件事建議照著現有節點的作法做：型別明確寫進 `attr.dataType` (否則上位系統看到的是 `BaseDataType`)、方向想清楚再決定 **RO** 還是 **RW** (§3.2 的理由同樣適用)、節點命名沿用 **QEC.** 前綴，之後分群比較好整理。

節點一多就值得分層，例如 `Objects/QEC` 底下再分 `DeviceInfo` / `Motor1` / `IO`。

9.3 讓 QEC 反過來當 Client

如果要讀的是別人的 OPC UA Server (他牌 PLC、上位系統、儀器)，角色就反過來了。那一端的重點完全不同——連線是非同步的、節點由對方定義，所以工具需要 **在面板上直接 Browse 對方節點樹** 的能力。見 [《QEC OPC UA Client 測試工具 — 使用手冊》](#)。

兩台 QEC 對接是最單純的組合：一台跑 Server、一台跑 Client，中間不需要 PC。

9.4 與 EtherCAT 共存

這正是 `loop()` 從一開始就採非阻塞寫法的原因。位置已經留好了：每一圈 `UA_Server_run_iterate()` 之後更新 EtherCAT，把 PDO 值寫進節點、把節點值寫成 `TargetPosition`。

建議第一版先把馬達相關節點開成 Read Only，確認資料流與週期都穩定之後，再開放寫入。

9.5 上線之前要處理的安全性

這一版未編入 `UA_ENABLE_ENCRYPTION`，SecurityPolicy 只有 None、驗證只有 Anonymous —— 任何連得到這個 IP 的人都能讀寫節點。用於測試與封閉的產線網段沒有問題，但要跨到辦公網路或對外之前，請先用防火牆或獨立網段把它隔開。

10. 聯絡我們

業務諮詢	info@qec.tw
技術支援	info@icop.com.tw
官方網站	https://www.qec.tw

QEC OPC UA Server Test Tool - User Manual v 1.0 ICOP Technology Inc.