

入門指南

QEC-M-090T x AZD-KRED x OVR6048K1-V

EtherCAT 控制 + HMI 整合

ICOP x **Orientalmotor**
Intelligent control on processor

日期：2025.11.17

(版本：v1r2)



QEC-M-090T
(HMI + EtherCAT 主站)

EtherCAT®



OVR6048K1-V
(6 軸機器手臂, AZ 系列)

修訂紀錄

日期	版本	描述
2025/10/10	Version1.0	首次發行。
2025/10/30	Version1.1	新增 Kinematics Viewer 參數於 API 中的使用說明。
2025/11/17	Version1.2	<ul style="list-style-type: none">名詞更新：「Master/Slave」改為「MDevice / SubDevice」修正第 59 頁錯字 (down → done)於第 5 頁新增 86Duino IDE 501+ 介紹

1. 專案介紹

物料清單與需求

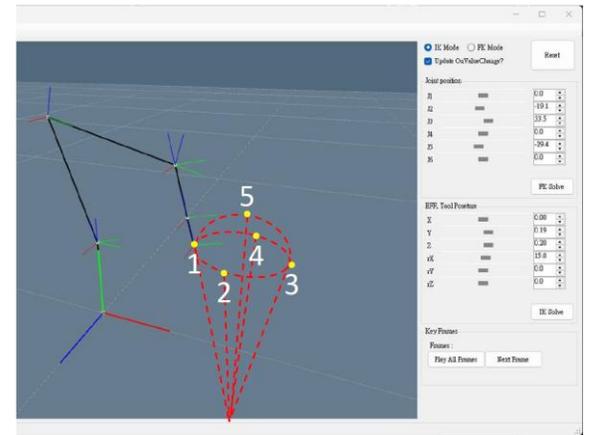


專案介紹

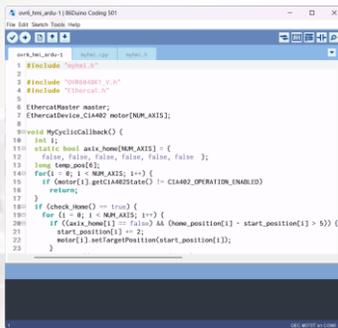
我們使用 QEC-M-090T (EtherCAT 主站 + HMI) 與 6 台 AZD-KRED 驅動器，為 OVR6048K1-V 建立一套完整的開發流程。

系統運行於 ECAT_SYNC 與 DC 同步模式，涵蓋配線、初始參數範本、原點回歸與安全機制、初學者可用的關節空間 **Kinematics Test**，以及任務空間的「**Sphere Around**」展示。

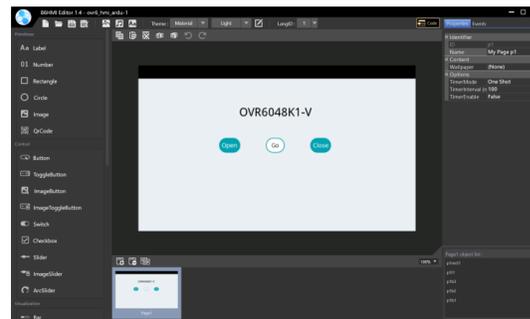
此外，我們也提供低程式碼開發路徑：透過與 **86HMI** 結合的 **ArduBlock** 積木，讓使用者可以快速進行運動原型開發、監看 PDO，並在 86Duino 上快速驗證結果。



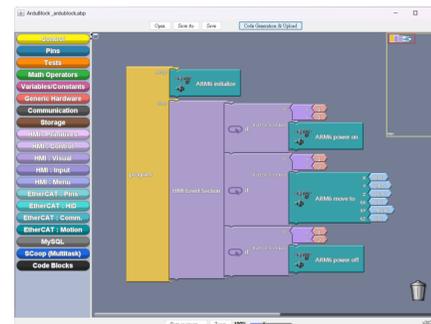
Kinematics Viewer



86Duino



86HMI



ArduBlock

關於軟體：86Duino IDE 501+

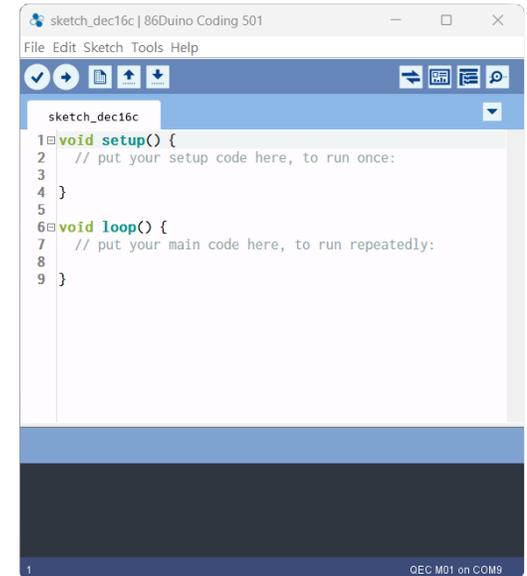
86Duino 整合開發環境 (IDE) 可讓使用者輕鬆撰寫程式並上傳至 86Duino 板卡。它支援 Windows、macOS 與 Linux。此開發環境以 Java 撰寫，並建立於 Arduino IDE、Processing、DJGPP 及其他開源軟體之上。



86Duino IDE 501+ Logo



QEC.TW 網站



86Duino IDE 501+

501+ 有哪些新功能？

86Duino IDE 501+ 這個新版速度更快、功能更強。

它現在支援更多第三方 EtherCAT SubDevice，並新增更多 EthercatDevice 類別，包括一個通用的 CiA-402 SubDevice 類別，可控制任何符合 CiA-402 標準的 EtherCAT 伺服驅動器。

* 註：針對 Oriental Motor 六軸 OVR6048K1-V 手臂，我們提供了特製版 86Duino IDE (內含模型輔助與範例)。

物料清單與需求

硬體

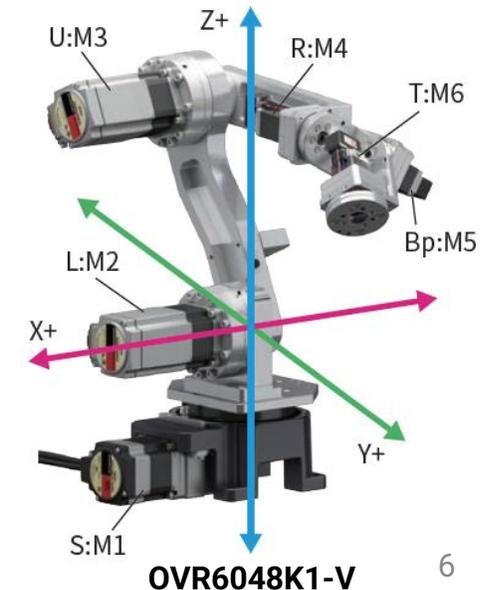
- **EtherCAT 主站** : QEC-M-090T
- **EtherCAT 驅動器** : AZD-KRED x6
- **機械手臂與機構** : OVR6048K1-V (馬達如下)
 - S: M1: DGB130R36-AZAKL
 - L: M2: AZM66MKH + CSG-20-100-2UH-LW-SP-B
 - U: M3: AZM66MKH + CSG-20-100-2UH-LW-SP-B
 - R: M4: AZM26AK + CSF-8-50-2UP-SP-A
 - Bp: M5: AZM24AK + CSF-8-50-2UP-SP-A
 - T: M6: AZM24AK + CSF-8-50-2UP-SP-A



QEC-M-090T



AZD-KRED

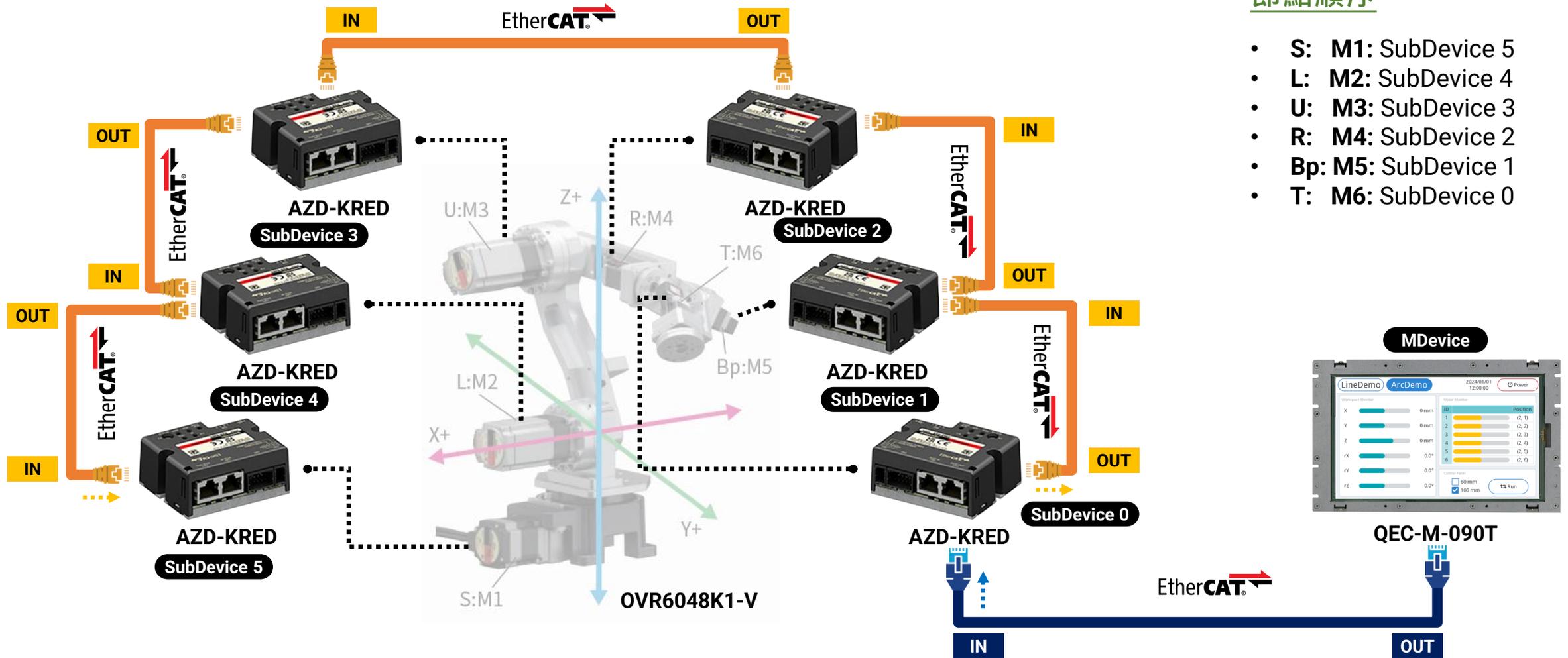


OVR6048K1-V

軟體

- **86Duino IDE 501 Preview (Oriental Motor Robotics Arm 版本)**

配線圖



節點順序

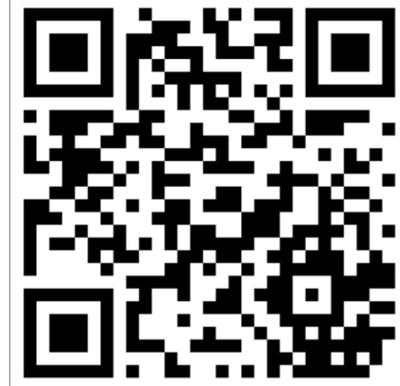
- S: M1: SubDevice 5
- L: M2: SubDevice 4
- U: M3: SubDevice 3
- R: M4: SubDevice 2
- Bp: M5: SubDevice 1
- T: M6: SubDevice 0



* 註：採固定位置式定址（由線材順序固定），請勿交換。

主站參考 | QEC-M-090T

- 有關 **QEC-M-090T** 的規格、配線、功能與安全事項，請一律以 **QEC** 官方手冊為準。
<https://www.qec.tw/product/qec-m-090t/>
- 本專案使用 **QEC-M-090T** (EtherCAT 主站開放式 (Open frame) 9" LCD)。
- 產品優勢：
 - 即時的獨立雙核心 **CPU** 整合 **EtherCAT** 功能。
 - 86Duino 整合開發環境 (IDE)。
 - 9 吋 TFT LCD 電阻式觸控螢幕 (800×480)。
 - 高可靠度 SLC eMMC 儲存，並支援 HDMI 輸出作為 HMI。
 - 工業級操作溫度 -20°C 至 +70°C。
 - 完整 **EtherCAT** 堆疊功能，支援 PDO/SDO、CiA-402 (PP/CSP/CSV)、診斷與紀錄
 - 內建電壓、溫度與系統狀態監控。



機械手臂參考 | OVR6048K1-V

- 有關手臂規格（尺寸、負載、工作半徑、限制），請一律依據 **Oriental Motor** 官方文件。
- 請參閱產品頁與手冊（CAD、安裝、安全注意事項）。

OVR 六軸關節機器人

<https://www.orientalmotor.com.tw/tw-zh/products/robots-controllers/ovr-6-axis-articulated-robot>

- 整合前請確認以下項目：
 - 負載 / 工作半徑。
 - 各關節限制（J1-J6，機械與軟體限制）。
 - **Home / Zero** 定義。
 - 減速比與單位縮放（deg pulse、速度/加速度單位）。
 - 末端工具介面（法蘭、可用 I/O / 配線）。
 - 安全項目（急停、禁入區域、走線）。



驅動器參考 | AZD-KRED

- 有關 AZ-mini 驅動器的規格、配線、功能與安全，請一律依據 **Oriental Motor** 官方手冊。

AZ-mini-Driver:

<https://www.orientalmotor.com.tw/tw-zh/products/alpha-step/az-mini-driver>

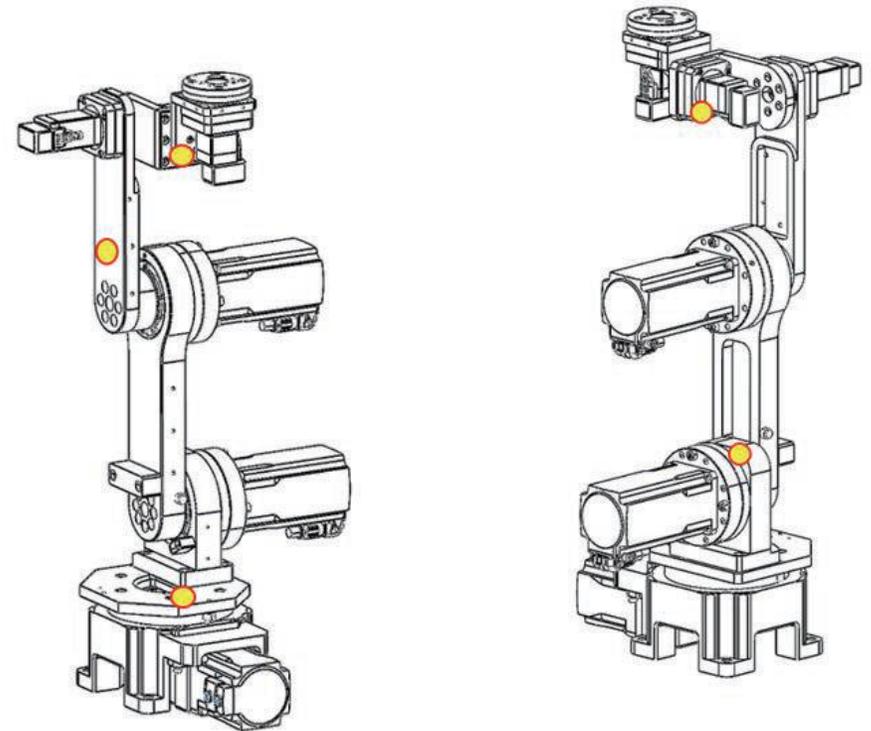
- 本專案使用 **AZD-KRED (EtherCAT)**。
- 整合前請確認以下項目：
 - 型號與介面 (AZ-mini AZD-KRED EtherCAT)。
 - 電源額定與接地/遮蔽。
 - 馬達/煞車相容性與接頭。
 - **ABZO 絕對型編碼器**歸零與備份方式。
 - 單位與解析度 (電子齒輪、pulse/rev、速度/加速度單位與限制)。
 - 回原點參數 (方式、速度、偏移、到位判定)。
 - 環境與安裝 (溫度/振動、線長、散熱)。



* 註：本文件建議使用的驅動器為 **AZD-KRED (EtherCAT)**。

零點姿態 (Zero Pose)

- 目的：建立所有關節在座標、路徑與 API 上的標準參考姿態。
- 依據：依 OEM 手冊，此機種提供 5 個定位銷孔 ($\varnothing 4h7$) 用於機械式設定零點姿態。
- 執行時機：首次使用前，以及保養或更換驅動器後。
- 建議流程：
 1. 切換到 Manual/Teach 模式，低速接近該姿態。
 2. 在 5 個位置插入定位銷，確認連桿對齊。
 3. 允許 $\pm 0.1^\circ$ 公差 (可設定) 。
 4. 安全重點：停止或運動中，請勿輸入驅動器 FREE 訊號。
此動作會切斷馬達電流與保持力，機器手臂可能因重力失去姿態而掉落。



OVR6048K1 – Zero Pose

* 註：精確位置與限制請依 OEM 手冊。

零點姿態 與 座標慣例

零點姿態 (Zero Pose)

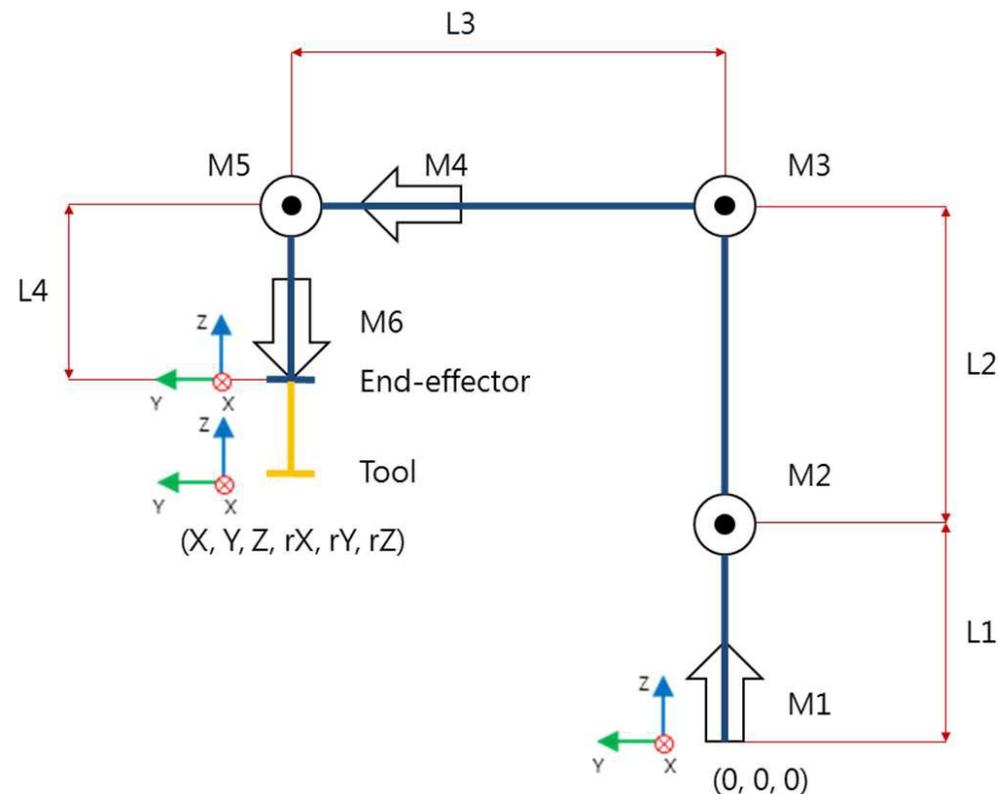
- 基座座標原點位於關節 **M1** 中心的 $(0,0,0)$ 。
- $J1-J6 = 0^\circ$ 的標準姿態作為全域參考。
- 姿態旋轉順序 (外旋)：先繞世界座標 X ，再 Y ，再 Z 。
- 合成矩陣：
 $R_{\text{world}} = R_z(rZ) \cdot R_y(rY) \cdot R_x(rX)$
(依 $X \rightarrow Y \rightarrow Z$ 順序套用)



OVR6048K1
Zero-pose front
view 1



OVR6048K1
Zero-pose front
view 2



各軸解析度

- AZD-KRED (驅動器) 的預設解析度為 **10,000** 。
- OVR6048K1-V 各軸型號如下：

Axis	Motor/Reducer	Total ratio	Steps / rev	Steps / deg	Deg / step
M1	DGB130R36-AZAKL	36	360,000	1,000.0	0.0010°
M2	AZM66MKH + CSG20-100	100 ¹	1,000,000	2,777.78	0.00036°
M3	AZM66MKH + CSG20-100	100 ¹	1,000,000	2,777.78	0.00036°
M4	AZM26AK + CSF8-50	50	500,000	1,388.89	0.00072°
M5	AZM24AK + CSF8-50	50	500,000	1,388.89	0.00072°
M6	AZM24AK + CSF8-50	50	500,000	1,388.89	0.00072°



OVR6048K1-V

DGB130R36-AZAKL

AZM66MKH-CSG20-100

AZM66MKH-CSG20-100

AZM26AK-CSF8-50

AZM24AK-CSF8-50

AZM24AK-CSF8-50

2. 運動學檢視器 (Kinematics Viewer)

OVR6048K1 版本



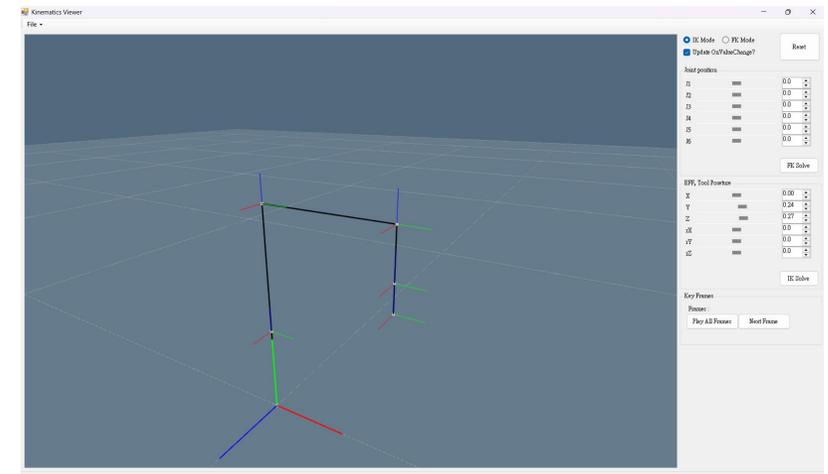
運動學檢視器 介紹 - 1

這是什麼？

- 一個輕量化 3D 模擬器，已預先設定 **OVR6048K1** 的連桿長度、零點姿態與各軸資訊。
- 支援 **FK** (正向運動學) 與 **IK** (逆向運動學) 並可即時視覺化。

為什麼要使用？

- 在實機運行前驗證姿態、路徑與奇異點。
- 產生 waypoint / timing，並交給 QEC 程式 / HMI 使用。



Kinematics viewer

運動學檢視器 介紹 - 2

主要功能

- 模式：IK / FK 切換，可選擇 **Update OnValueChange** 即時重算。
- 姿態輸入/輸出：可編輯 **J1-J6** 或 **TCP (X, Y, Z, rX, rY, rZ)**，並雙向求解。
- 關鍵影格：可用 **Next Frame** 或 **Play All Frames** 播放。
- 匯入路徑：載入 `testPath.txt` 模擬完整運動時間軸。

慣例

- 外旋順序為 **X → Y → Z** (世界座標)。
- 零點姿態與 OVR6048K1 參考一致；位置單位為 **m/mm**，角度為 **deg**。

快速操作

- 滑鼠：左鍵拖曳旋轉、中鍵拖曳平移、滾輪縮放。
- 動作：**FK Solve / IK Solve / Reset**。

運動學檢視器使用者介面功能

視角操作

- 滑鼠中鍵：旋轉
- 滑鼠右鍵：平移

模式切換 (目前未啟用)

- IK Mode FK Mode
- Update OnValueChange?
- Reset

全部重設

可用來觀察IK計算結果跳動程度

Joint position	Value
J1	0.0
J2	0.0
J3	0.0
J4	0.0
J5	0.0
J6	0.0

FK 單次計算按鈕

可拖曳滑桿或直接輸入數值

EFF, Tool Poseure	Value
X	0.00
Y	0.24
Z	0.27
rX	0.0
rY	0.0
rZ	0.0

IK 單次計算按鈕

路徑點操作功能 · 播放下一步

路徑點操作功能 · 播放全部

備註

A. 路徑檔格式：每行一個 waypoint：{ms, X, Y, Z, rX, rY, rZ}載入文字檔後，可用 Play All Frames / Next Frame 測試。若格式或數值錯誤，會顯示錯誤對話框。

B. 定義運動路徑終點 waypoint 時，請先利用 UI 檢查關節角是否跳動。現行 IK 不會限制關節範圍，且 $\pm 180^\circ$ 附近翻轉在 3D 視圖中可能不明顯，請查看右上角滑桿確認關節是否接近極限。

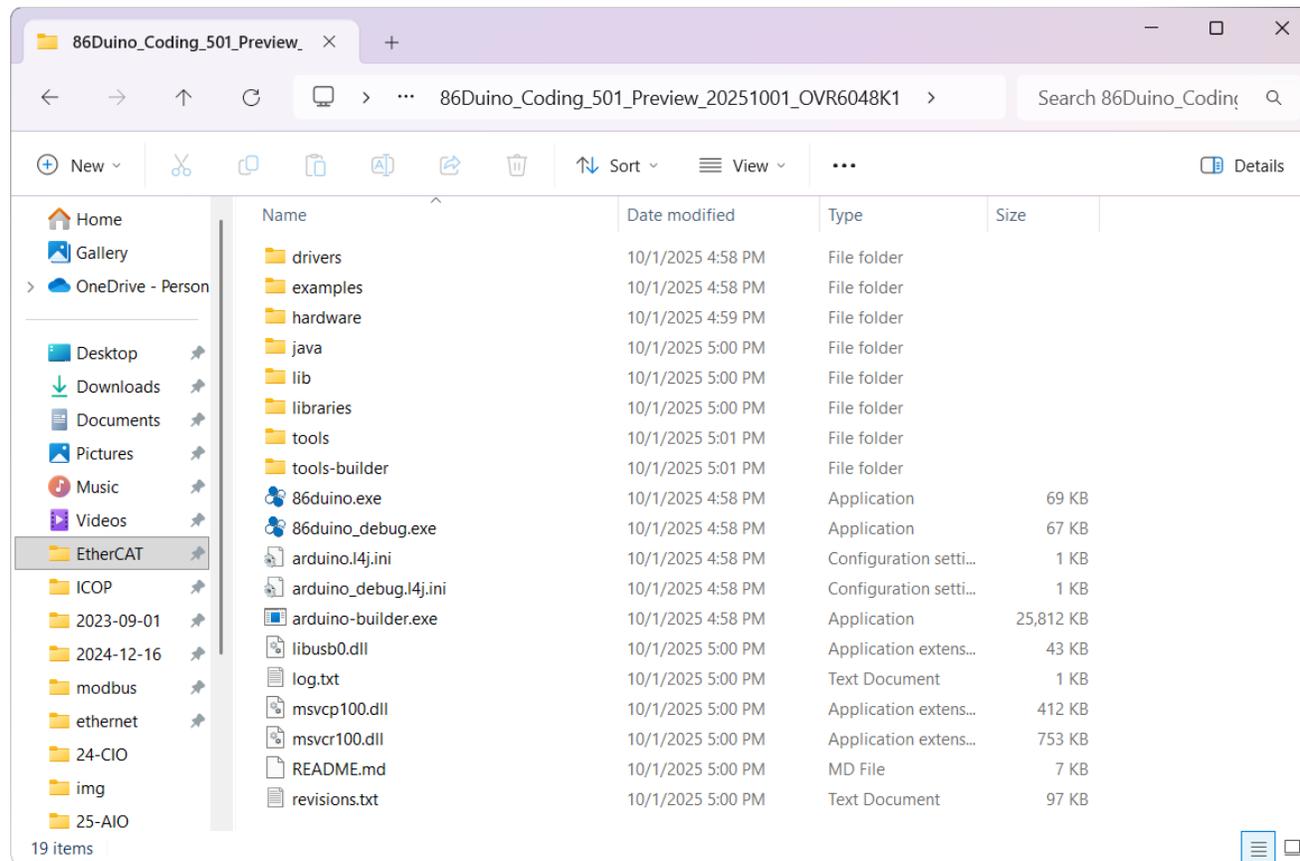
IK Result ← IK計算結果訊息 · 0 = error

Kinematics viewer

啟動 運動學檢視器 - 1

步驟 1：開啟 IDE 資料夾

- 前往你的 86Duino IDE 目錄（例如：...\86Duino_Coding_501_Preview_20251001_OVR6048K1）



Kinematics viewer file path

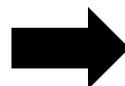
啟動 運動學檢視器 - 2

步驟 2 : 前往工具位置

- 開啟 : **tools** → **KinematicsViewer** 。

Name	Date modified	Type	Size
drivers	10/1/2025 4:58 PM	File folder	
examples	10/1/2025 4:58 PM	File folder	
hardware	10/1/2025 4:59 PM	File folder	
java	10/1/2025 5:00 PM	File folder	
lib	10/1/2025 5:00 PM	File folder	
libraries	10/1/2025 5:00 PM	File folder	
tools	10/1/2025 5:01 PM	File folder	
tools-builder	10/1/2025 5:01 PM	File folder	
86duino.exe	10/1/2025 4:58 PM	Application	69 KB
86duino_debug.exe	10/1/2025 4:58 PM	Application	67 KB
arduino.14j.ini	10/1/2025 4:58 PM	Configuration setti...	1 KB
arduino_debug.14j.ini	10/1/2025 4:58 PM	Configuration setti...	1 KB
arduino-builder.exe	10/1/2025 4:58 PM	Application	25,812 KB
libusb0.dll	10/1/2025 5:00 PM	Application extens...	43 KB
log.txt	10/1/2025 5:00 PM	Text Document	1 KB
msvcp100.dll	10/1/2025 5:00 PM	Application extens...	412 KB
msvcr100.dll	10/1/2025 5:00 PM	Application extens...	753 KB
README.md	10/1/2025 5:00 PM	MD File	7 KB
revisions.txt	10/1/2025 5:00 PM	Text Document	97 KB

Kinematics viewer file path 2



Name	Date modified	Type	Size
ArduBlockTool	10/1/2025 5:00 PM	File folder	
BootMenu	10/1/2025 5:00 PM	File folder	
BurnAllRes	10/1/2025 5:00 PM	File folder	
DownloadAllRes	10/1/2025 5:00 PM	File folder	
EcConfig	10/1/2025 5:00 PM	File folder	
Extra	10/1/2025 5:01 PM	File folder	
HMI	10/1/2025 5:01 PM	File folder	
KinematicsViewer	10/1/2025 5:01 PM	File folder	
LDmicro	10/1/2025 5:01 PM	File folder	
Mangler	10/1/2025 5:01 PM	File folder	
ResetECMaster	10/1/2025 5:01 PM	File folder	
howto.txt	10/1/2025 5:00 PM	Text Document	6 KB

Kinematics viewer file path 3

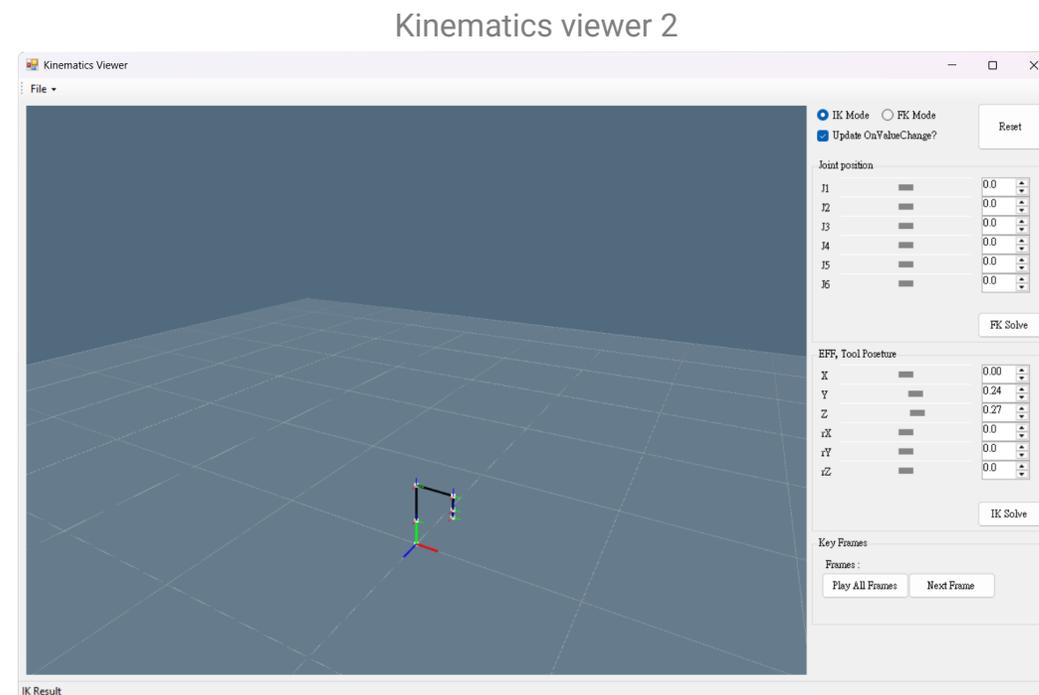
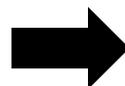
啟動 運動學檢視器 - 3

步驟 3：執行檢視器

- 雙擊 **KinematicsViewer.exe**。

Name	Date modified	Type	Size
all.txt	10/1/2025 5:01 PM	Text Document	12 KB
all2.txt	10/1/2025 5:01 PM	Text Document	12 KB
BulletSharp.dll	10/1/2025 5:01 PM	Application extens...	2,101 KB
glew32.dll	10/1/2025 5:01 PM	Application extens...	383 KB
glut32.dll	10/1/2025 5:01 PM	Application extens...	232 KB
KinematicsViewer.exe	10/1/2025 5:01 PM	Application	2,082 KB
step1_CW_xz_x-ry.txt	10/1/2025 5:01 PM	Text Document	3 KB
step2_CW_xy_x-ry.txt	10/1/2025 5:01 PM	Text Document	2 KB
step3_CW_xy_y_r-x-y.txt	10/1/2025 5:01 PM	Text Document	2 KB
step4_CW_xy_x_r-xy.txt	10/1/2025 5:01 PM	Text Document	2 KB
step5_CCW_yz_y-rx.txt	10/1/2025 5:01 PM	Text Document	3 KB
step6_CW_xy_y_rxy.txt	10/1/2025 5:01 PM	Text Document	2 KB
testPath.txt	10/1/2025 5:01 PM	Text Document	1 KB
testPath_withError.txt	10/1/2025 5:01 PM	Text Document	1 KB

Kinematics viewer file path 4

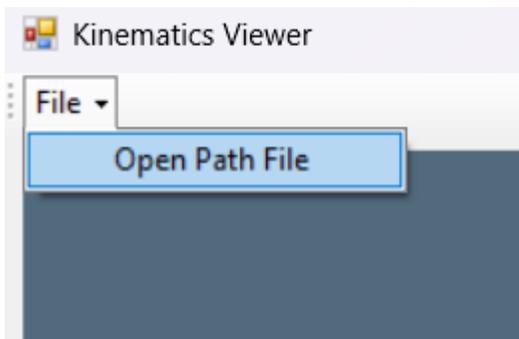


你會看到該執行檔與數個範例 .txt 檔案放在一起。

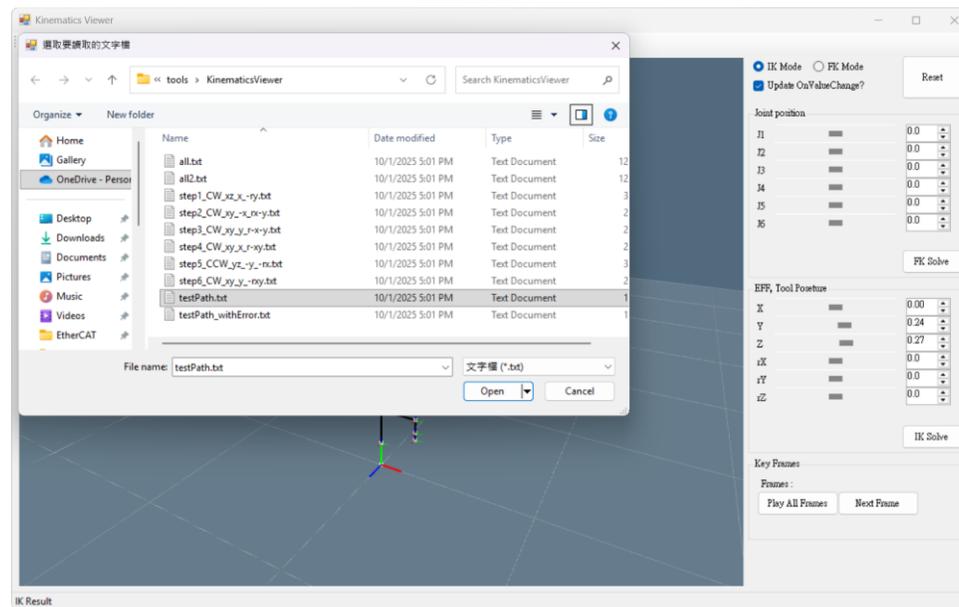
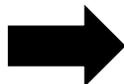
啟動 運動學檢視器 - 4

步驟 4：載入運動路徑

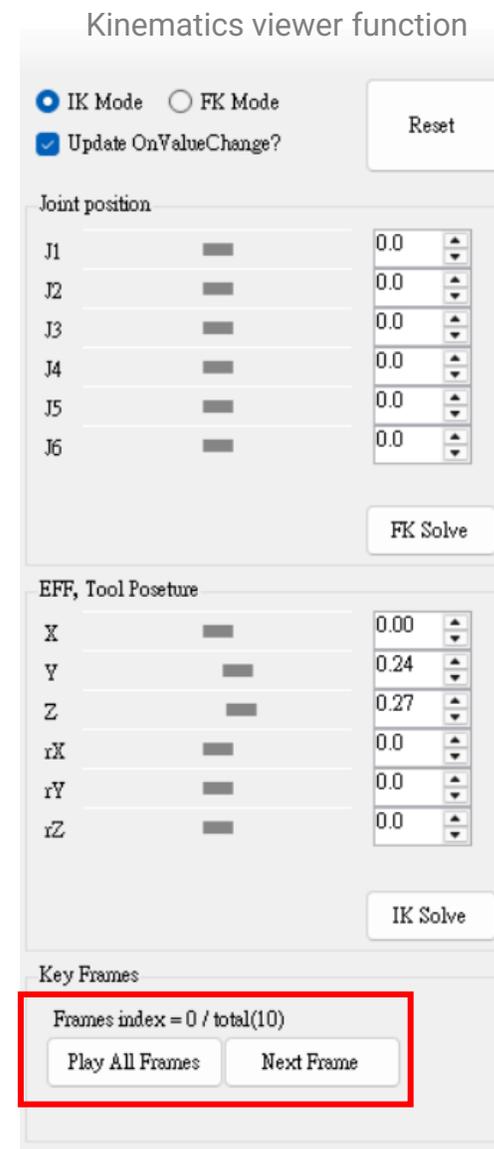
- 在檢視器中：**File** → **Import/Open Path...** → 選擇 **testPath.txt** (或其他範例檔，如 all.txt、all2.txt、step1_...txt 等)
- 使用 **Play All Frames / Next Frame** 模擬軌跡



Kinematics viewer – file open 1



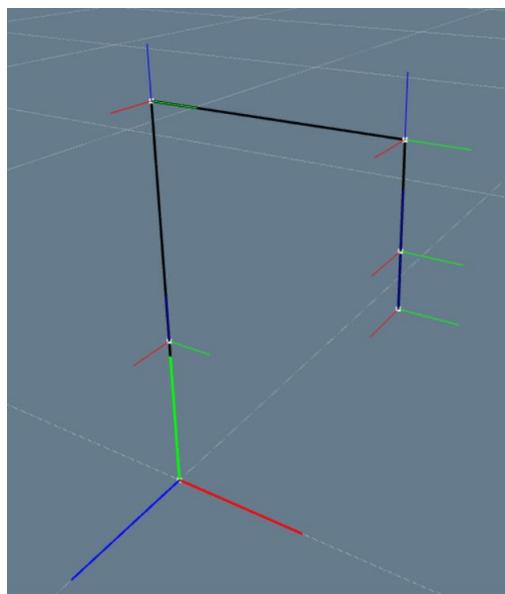
Kinematics viewer – file open 2



啟動 運動學檢視器 - 5

步驟 5：變更數值

在檢視器中，使用者可於「**Joint position**」或「**EFF, Tool Posture**」區域拖曳滑桿或輸入數值，以在視窗中模擬機械手臂動作。完成後，使用者可將「**EFF, Tool Posture**」中的數值輸入對應的 **API 函式** 或 **積木**，讓實體手臂移動到與 Kinematics Viewer 相同的位置。



Viewer Windows

Joint position		
J1	<input type="range"/>	0.0
J2	<input type="range"/>	0.0
J3	<input type="range"/>	0.0
J4	<input type="range"/>	0.0
J5	<input type="range"/>	0.0
J6	<input type="range"/>	0.0

FK Solve

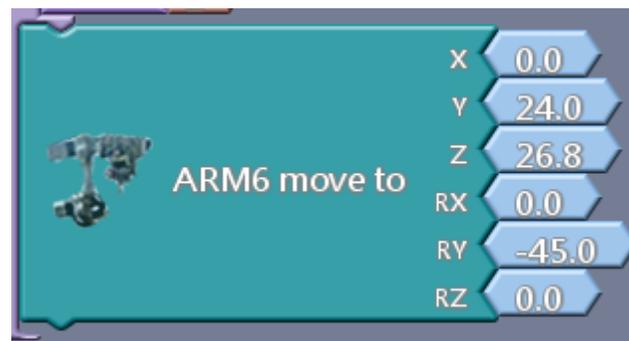
EFF, Tool Posture		
X	<input type="range"/>	0.00
Y	<input type="range"/>	0.24
Z	<input type="range"/>	0.27
rX	<input type="range"/>	0.0
rY	<input type="range"/>	0.0
rZ	<input type="range"/>	0.0

IK Solve

Kinematics viewer function

```
if (Hmi.buttonClicked(p1b4)) {  
  Set_Des_Position(0.0, 24.0, 26.8, 0.0, -45.0, 0.0);  
  prepare_move();  
}  
if (Hmi.buttonClicked(p1b5)) {  
  Set_Des_Position(0.0, 24.0, 26.8, 0.0, -20.0, 0.0);  
  prepare_move();  
}  
if (Hmi.buttonClicked(p1b6)) {  
  Set_Des_Position(0.0, 24.0, 26.8, 0.0, 20.0, 0.0);  
  prepare_move();  
}  
if (Hmi.buttonClicked(p1b7)) {  
  Set_Des_Position(0.0, 24.0, 26.8, 0.0, 45.0, 0.0);  
  prepare_move();  
}
```

Kinematics Example Code

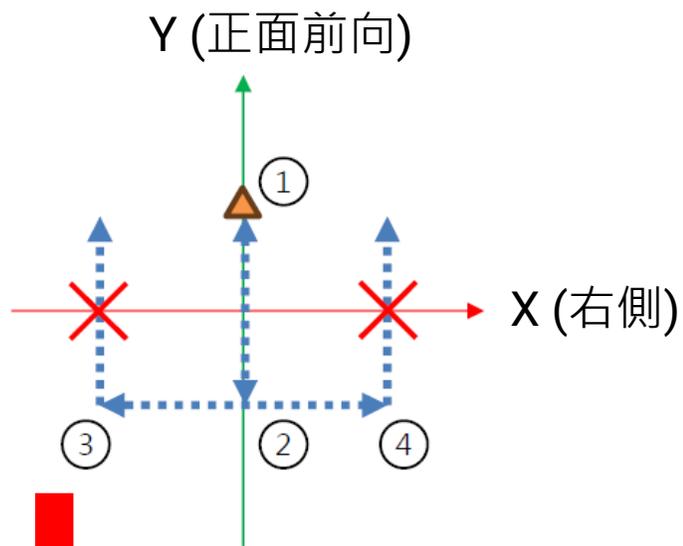


ARM6 move to

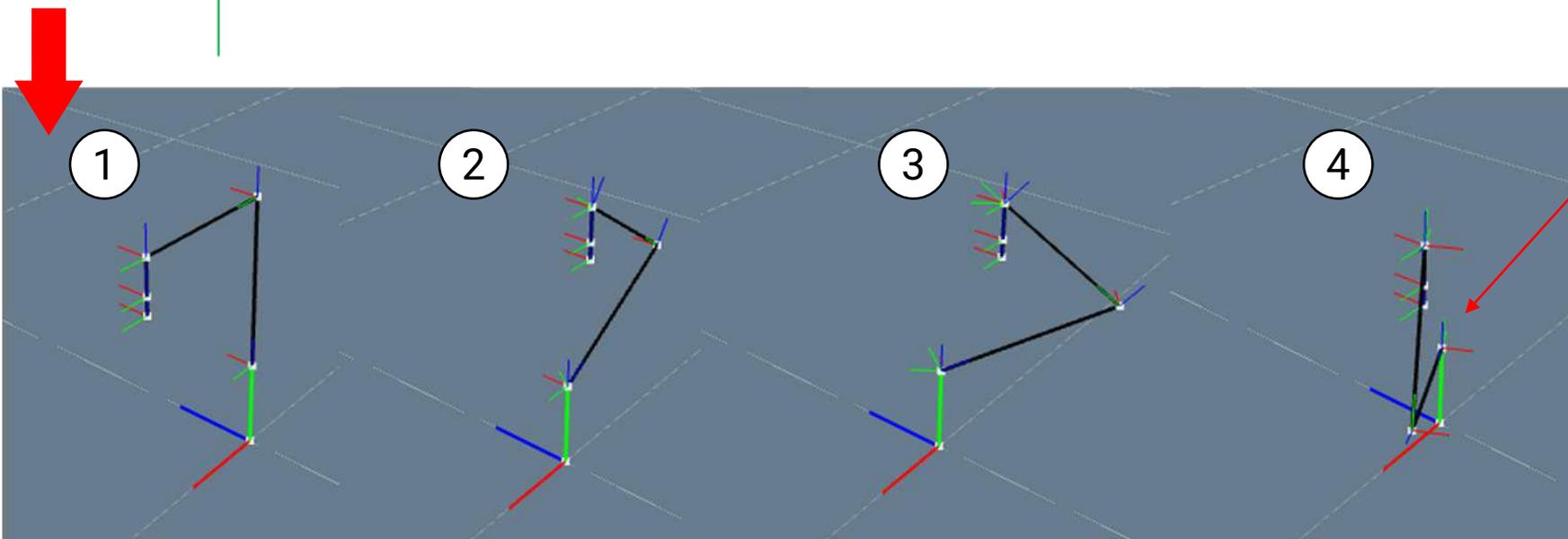
X	0.0
Y	24.0
Z	26.8
RX	0.0
RY	-45.0
RZ	0.0

ArduBlock Moving the Robotics Arm Block

路徑規畫需避免 - 1



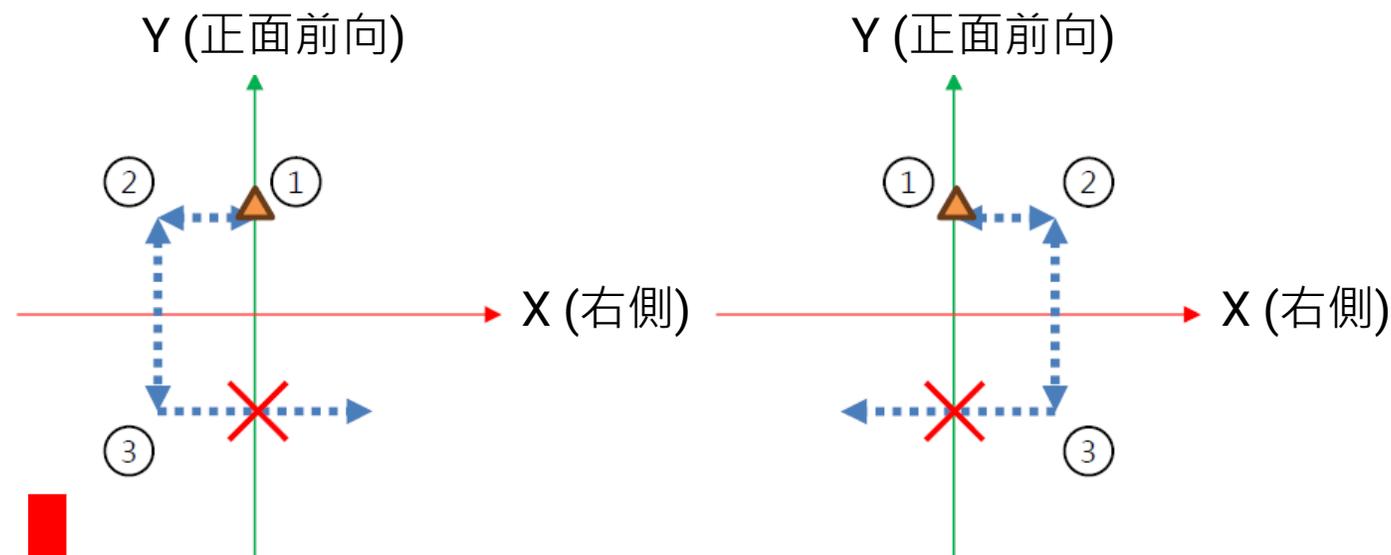
* 註：某些運動路徑會使基座接近極限位置旋轉。
圖中以 X 標記的跨象限路徑必須避免，應返回
原本路徑所在象限。



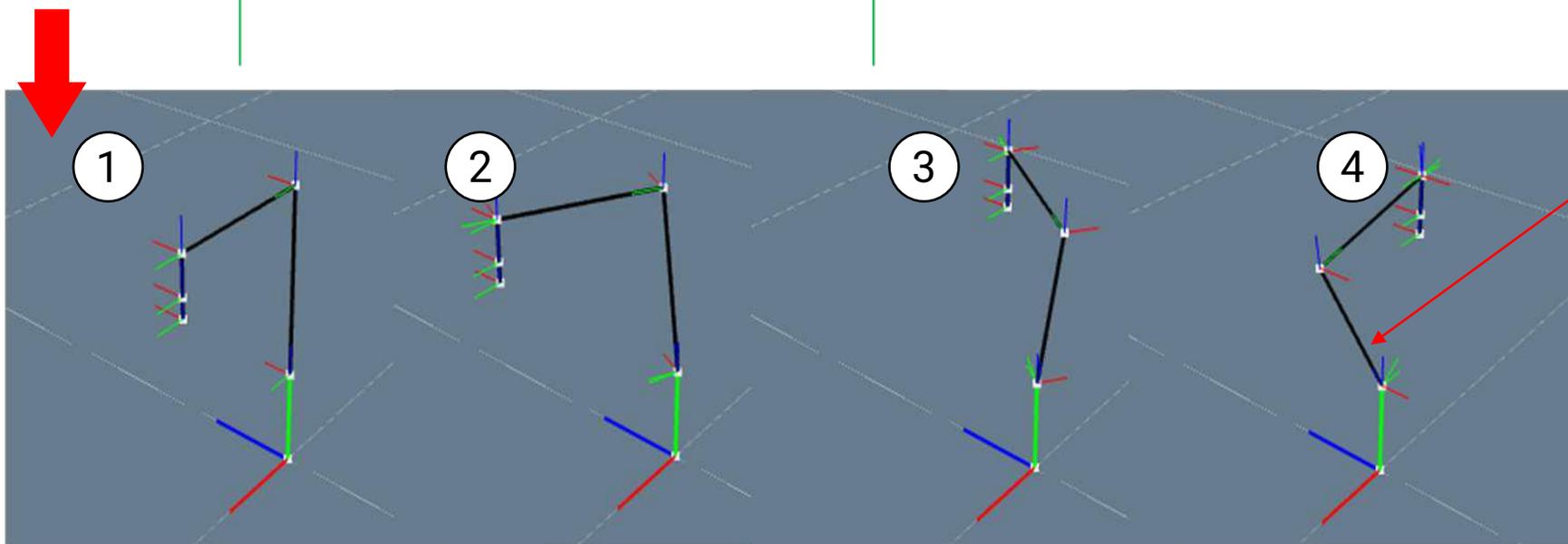
Base theta1 接近 $\pm 180^\circ$

Kinematics avoid path -1

路徑規畫需避免 - 2

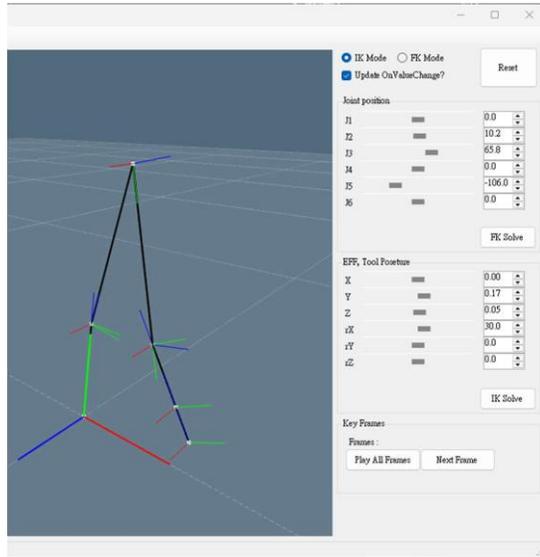


* 註：某些運動路徑會使基座接近極限位置旋轉。
圖中以 X 標記的跨象限路徑必須避免，應返回
原本路徑所在象限。

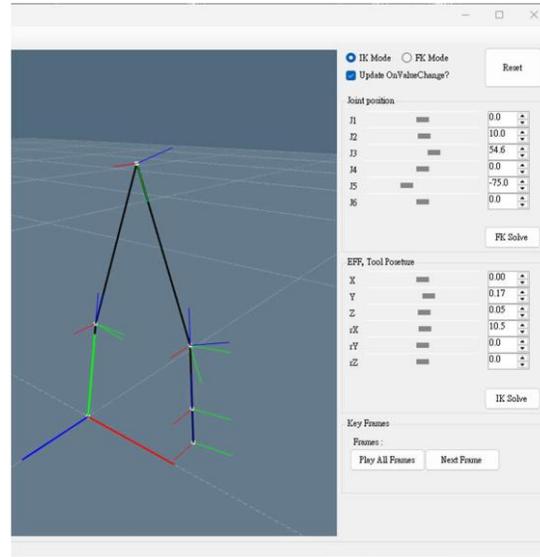


Base theta1 接近 $\pm 180^\circ$

路徑規劃需避免 - 3



Kinematics avoid path - 3



Kinematics avoid path - 4

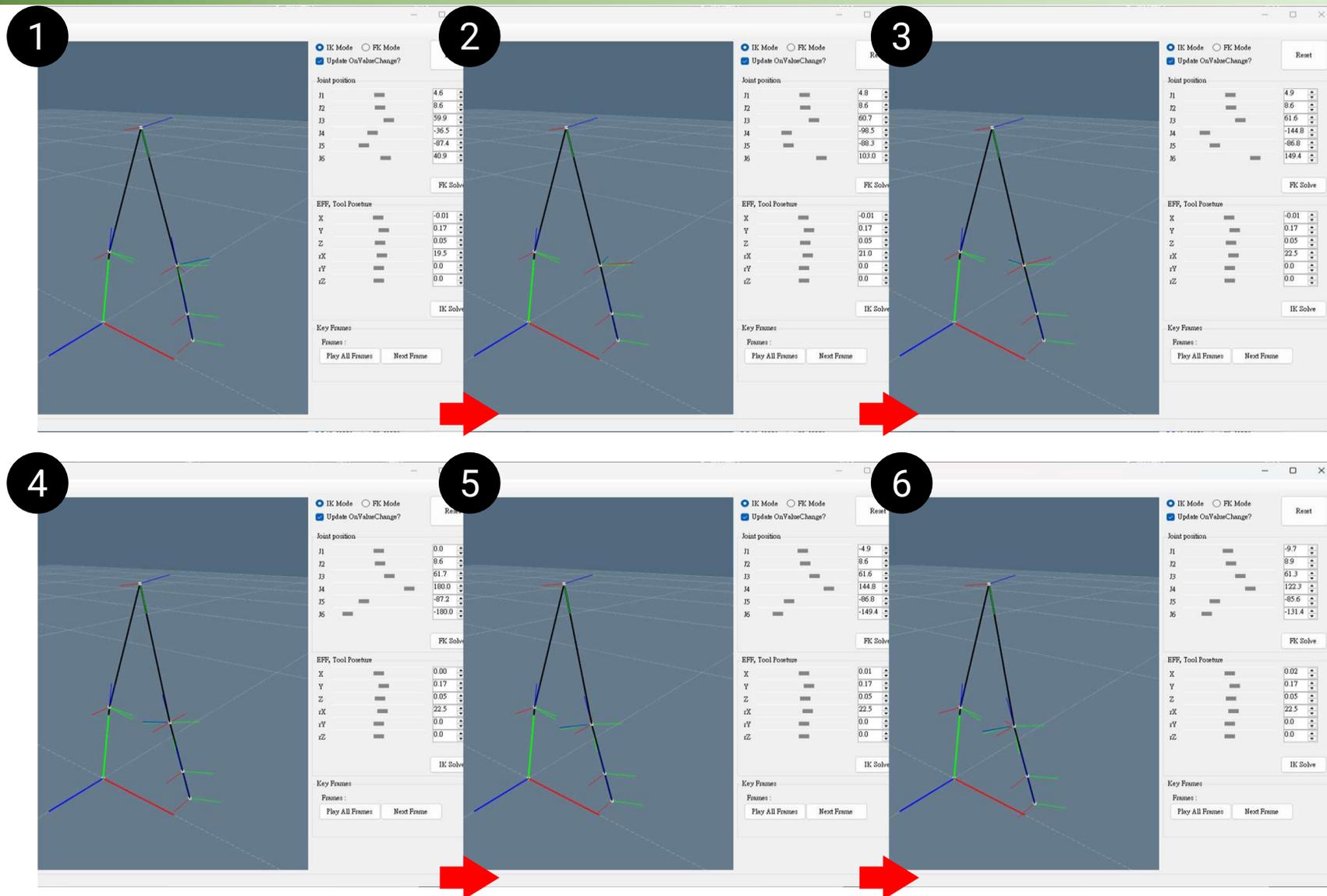
請注意手腕旋轉。當位於 $X=0$ 平面時，**Joint 5** 接近 -90° 是可接受的。

然而，若 X 稍微偏離 $X=0$ 平面（例如 $X=-0.01$ ），當 $J5$ 更接近 $\pm 90^\circ$ 時， $\text{Theta } 4 / 6$ 會被推向邊界，因為 $\text{Theta } 4$ 會最小化變化量以減少抖動。

此時若終點位置 X 跨越象限（例如 $X=-0.01 \rightarrow 0.01$ ）， $\text{Theta } 4$ 與 $\text{Theta } 6$ 會跨越 $\pm 180^\circ$ 。因此，當 $J5$ 接近 $\pm 90^\circ$ 時，應避免此類動作。

（下一頁續）

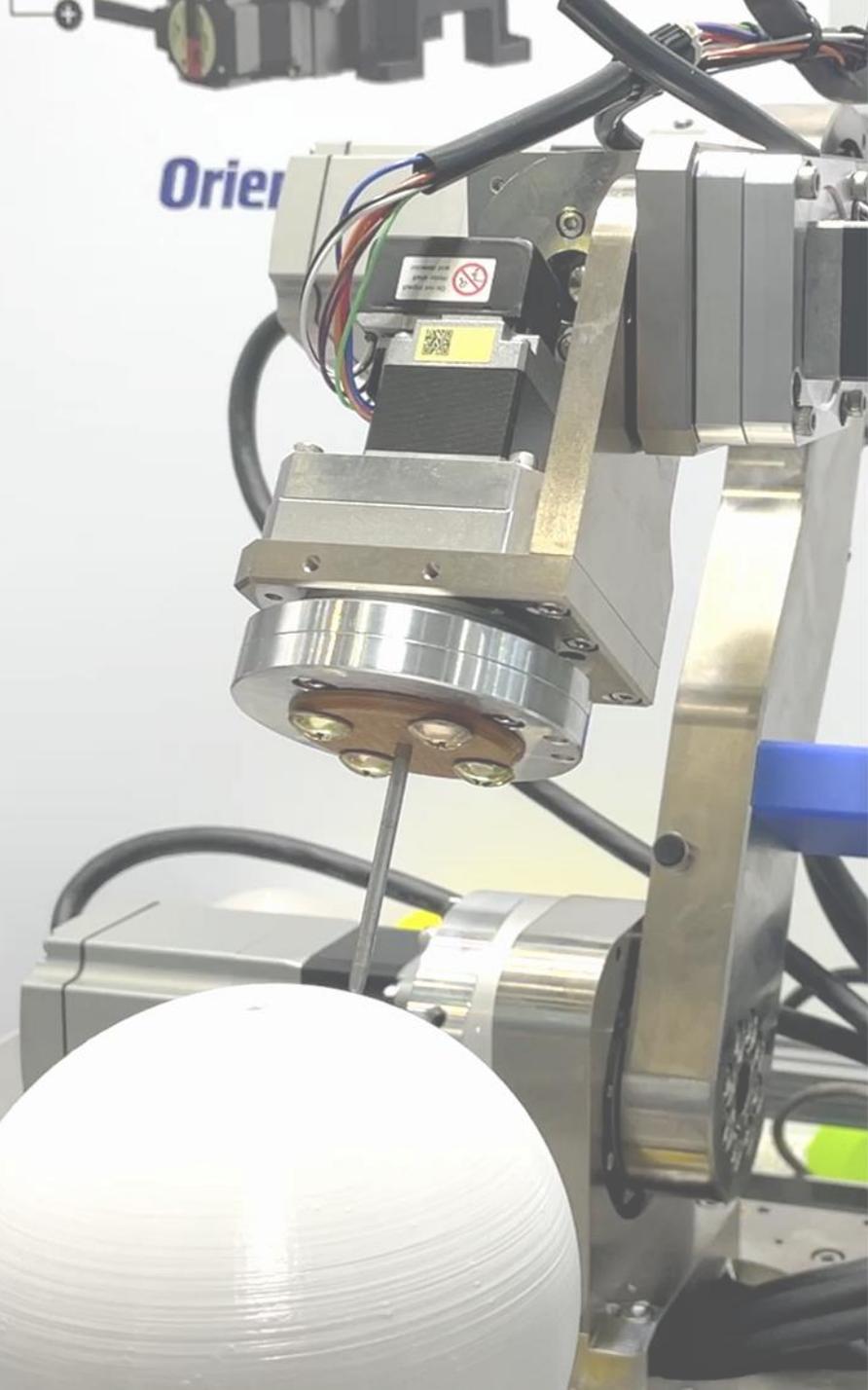
路徑規畫需避免 - 4



Kinematics avoid path - 5

3. 如何使用？

API 介紹、程式撰寫方式與圖形化工具

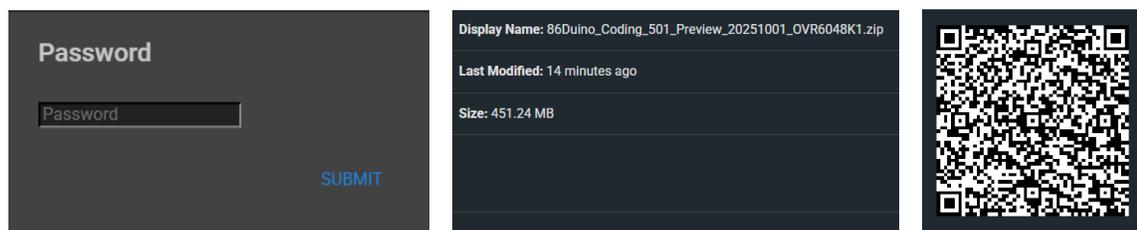


軟體概述 & 下載

針對 **Oriental Motor** 六軸機械手臂，我們釋出特製版 **86Duino IDE**，支援 **OVR6048K1-V** 專用的運動、運動學與檢視器功能。

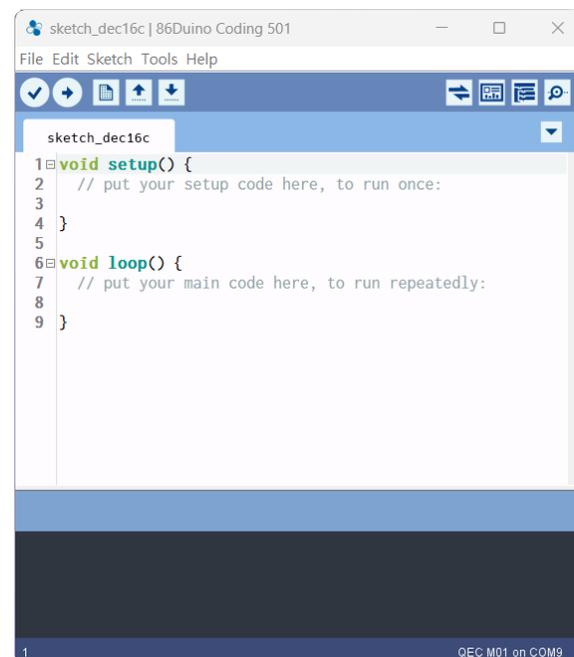
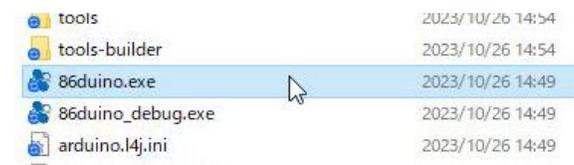
下載位置：https://ftp.dmp.com.tw/share/Ch_rBlbF

- 密碼：icop
- 檔名：86Duino_Coding_501_Preview_20251001_OVR6048K1.zip



86Duino IDE 501 主要新版特色是速度更快、功能更強，支援更多第三方 EtherCAT SubDevice，並新增 CiA 402 通用 EtherCAT SubDevice 類別。

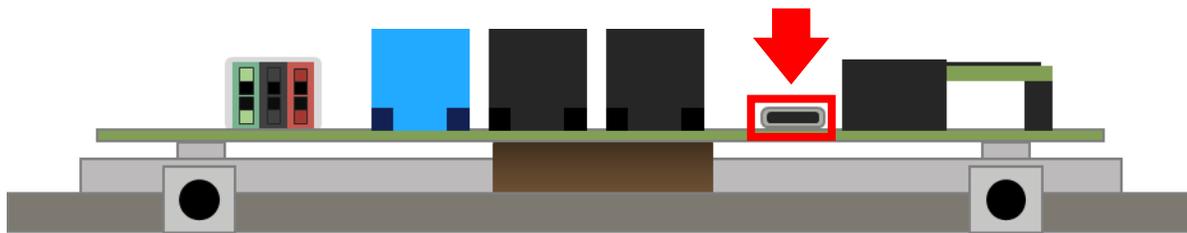
- **EtherCAT Library API 使用手冊：**
<https://www.gec.tw/ethercat/api/ethercat-library-api-user-manual/>



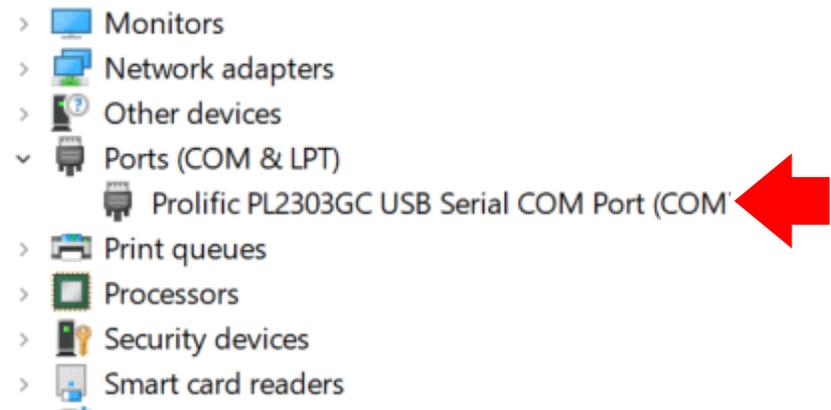
開發環境設定 - 1

請依以下步驟設定環境：

1. 使用 USB Type-C to USB 線將 QEC-M-090T 連接至電腦 (已安裝 86Duino IDE) 。
2. 開啟 QEC 電源 。
3. 開啟「裝置管理員」 (Win+X 後選單) → 「 **Ports (COM & LPT)** 」 中，你應該會看到「 **Prolific PL2303GC USB Serial COM Port (COMx)** 」；若未出現，需安裝驅動程式。
(Windows 的 PL2303 驅動程式可由此[下載](#))

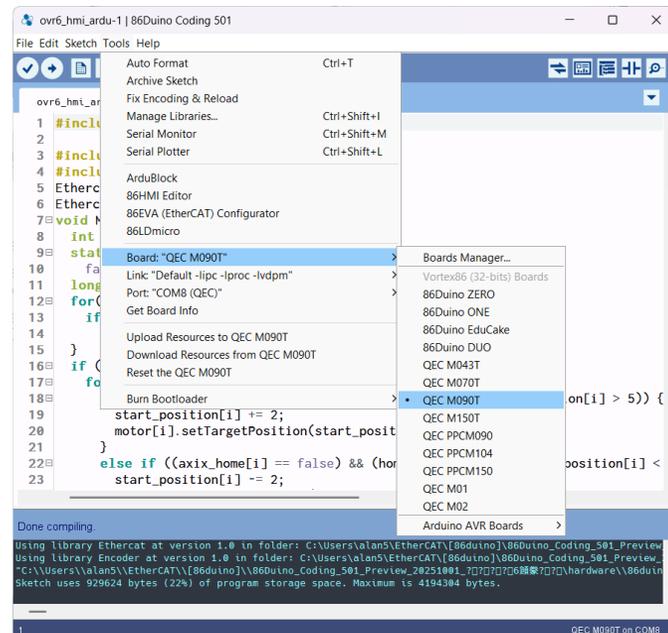


QEC-M Open frame board side view

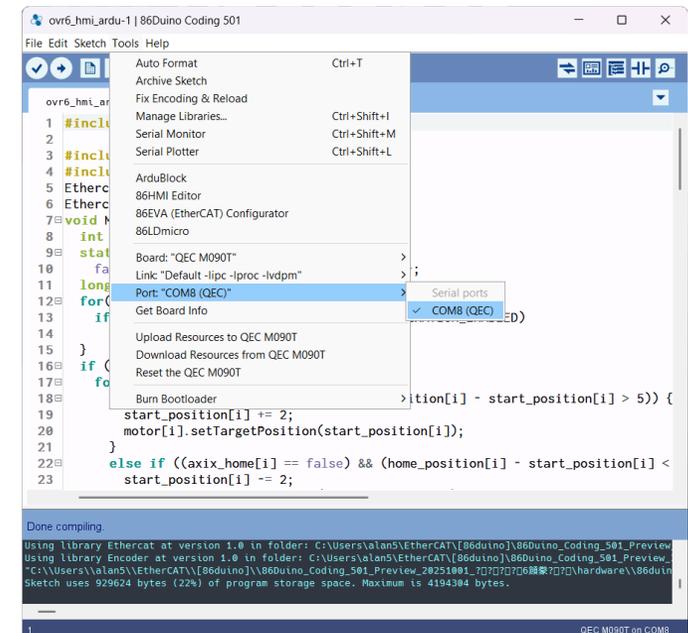


開發環境設定 - 2

4. 開啟 86Duino IDE。
5. 選擇正確板卡：**Tools** → **Board** → **QEC-M090T** (或你使用的 QEC MDevice 型號)。
6. 選擇 Port：**Tools** → **Port** → 選擇對應 **USB 連線埠** (此例為 **COM8**)。



Tool - Board

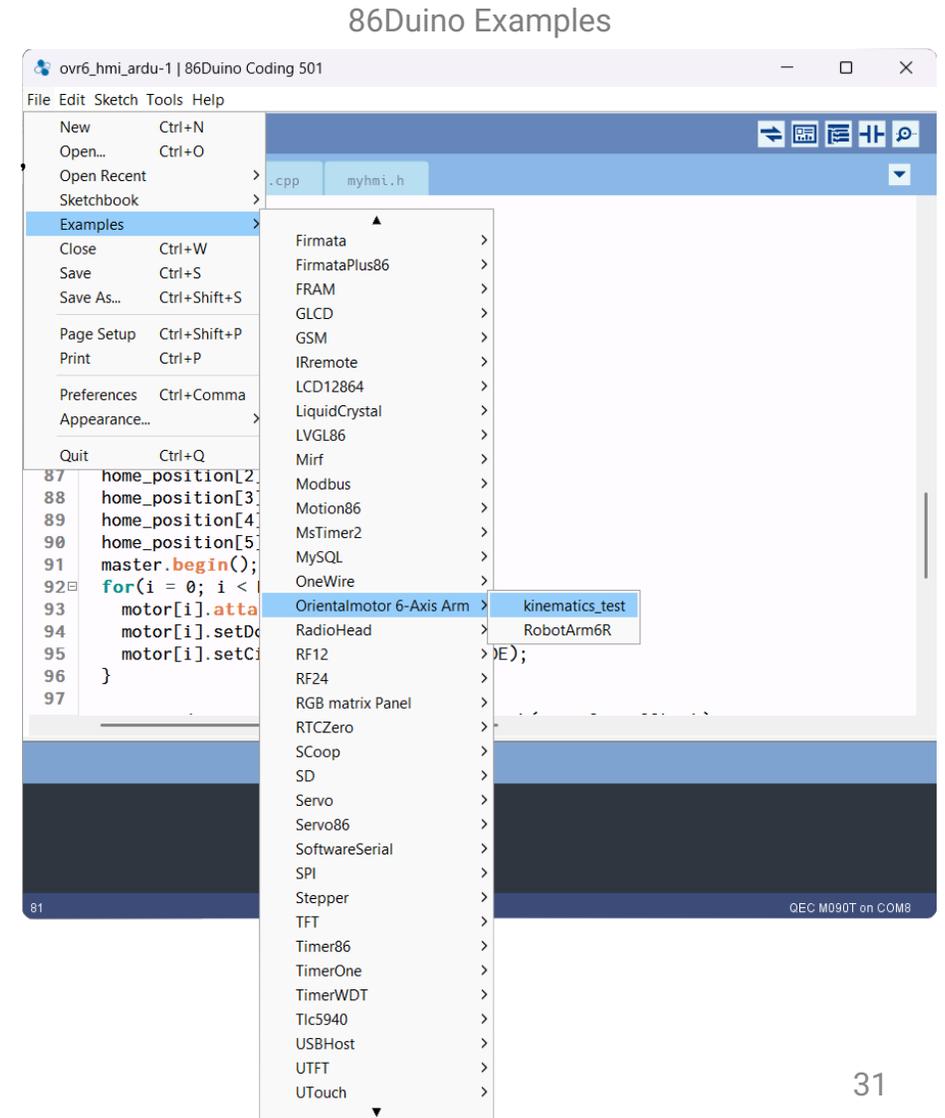


Tool - Port

範例：Kinematics Test - 1

步驟 1：開啟範例

- 在 **Tools** → **Board** 選擇 **QEC-M090T**，並設定正確 Port 後即可找到範例集。
- 開啟：**File** → **Examples** → **Orientalmotor 6-Axis Arm** → **kinematics_test**。



範例：Kinematics Test - 2

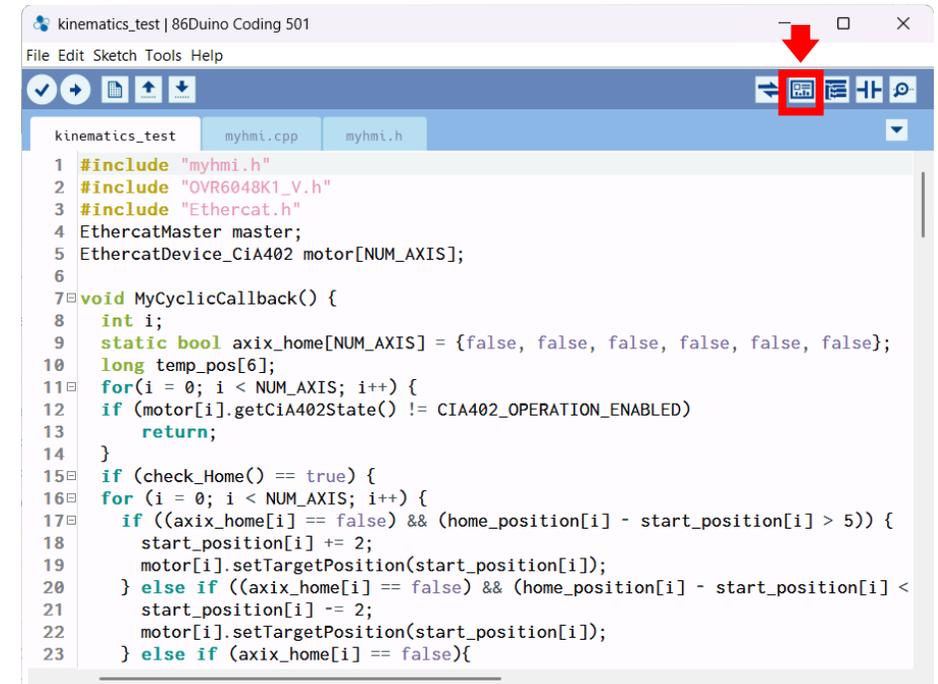
步驟 2：查看程式並開啟 HMI

- 開啟 kinematics_test 後會載入對應 sketch。
此範例使用 3 個函式庫：
 - myhmi.h** – HMI helpers
 - OVR6048K1_V.h** – robot motion helpers
 - Ethercat.h** – EtherCAT protocol
- 本範例設定如下：
 - EtherCAT 週期：**250 microseconds (µs)**.
 - EtherCAT 模式：**ECAT_SYNC**.
 - Distributed Clock (DC): **Enable (250 µs)**.
 - CiA-402 模式：**CSP (Cyclic Synchronous Position)**.

```
91 master.begin();
92 for(i = 0; i < NUM_AXIS; i++) {
93     motor[i].attach(i, master);
94     motor[i].setDc(cycle_time*10);
95     motor[i].setCiA402Mode(CIA402_CSP_MODE);
96 }
97
98 error_code = master.attachRTCCyclicCallback(MyCyclicCallback);
99 if (error_code != 0) {
100     printf("error : %d\n", error_code);
101 }
102 master.start(cycle_time*10, ECAT_SYNC); // cycle_time = 25000; // nanosecond
```

Kinematics_test.ino Code about EtherCAT

- 點擊右上角 86HMI 圖示以啟動 HMI Editor。



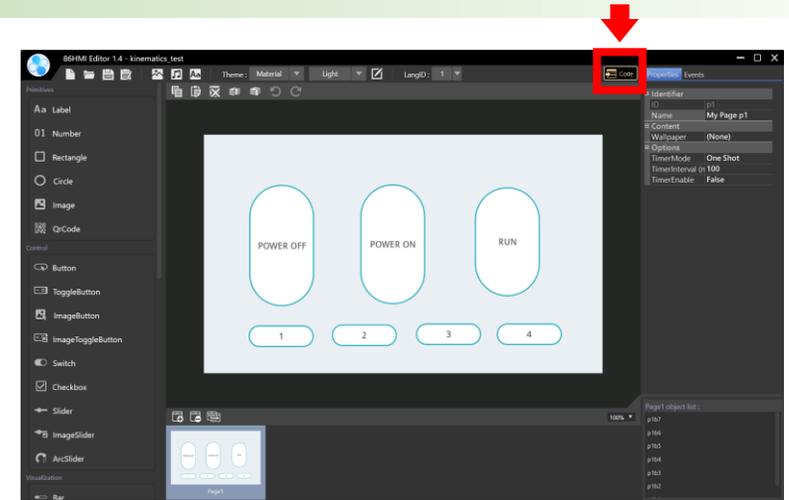
```
kinematics_test | 86Duino Coding 501
File Edit Sketch Tools Help
kinematics_test myhmi.cpp myhmi.h
1 #include "myhmi.h"
2 #include "OVR6048K1_V.h"
3 #include "Ethercat.h"
4 EthercatMaster master;
5 EthercatDevice_CiA402 motor[NUM_AXIS];
6
7 void MyCyclicCallback() {
8     int i;
9     static bool axix_home[NUM_AXIS] = {false, false, false, false, false, false};
10    long temp_pos[6];
11    for(i = 0; i < NUM_AXIS; i++) {
12        if (motor[i].getCiA402State() != CIA402_OPERATION_ENABLED)
13            return;
14    }
15    if (check_Home() == true) {
16        for (i = 0; i < NUM_AXIS; i++) {
17            if ((axix_home[i] == false) && (home_position[i] - start_position[i] > 5)) {
18                start_position[i] += 2;
19                motor[i].setTargetPosition(start_position[i]);
20            } else if ((axix_home[i] == false) && (home_position[i] - start_position[i] <
21                start_position[i] - 2;
22                motor[i].setTargetPosition(start_position[i]);
23            } else if (axix_home[i] == false){
```

Kinematics_test.ino

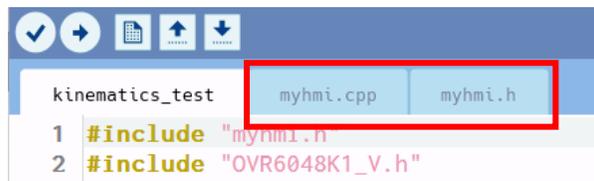
範例：Kinematics Test - 3

步驟 3：86HMI 版面與程式碼產生

- 此 HMI 範例包含 7 個按鈕：power_on、power_off、run、move1 ~ move4。我們將在下一頁中將每個按鈕對應到對應的程式碼。
- 點擊右上角「Code」按鈕可產生事件骨架與 include 敘述，若系統提示，請貼到主程式中。
- 產生後會看到 myhmi.h 與 myhmi.cpp，代表 LVGL86 函式庫的 helper 與 executor。

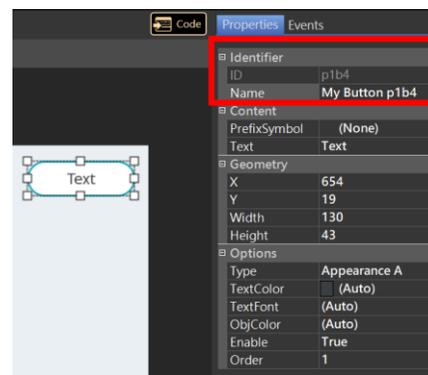


Kinematics_test HMI example



86HMI generation files

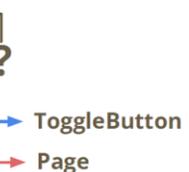
86HMI ID Naming rule



Naming rules:

Serial number

p?tb?



* 補充：在 86HMI 中，每個物件都有唯一 ID。命名規則為頁碼 + 類別序號。

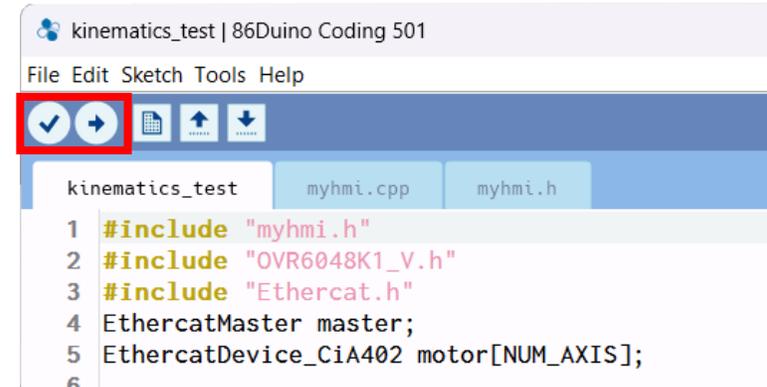
範例：Kinematics Test - 4

步驟 4：編譯與上傳

程式完成後，點選工具列上的編譯  按鈕進行編譯。

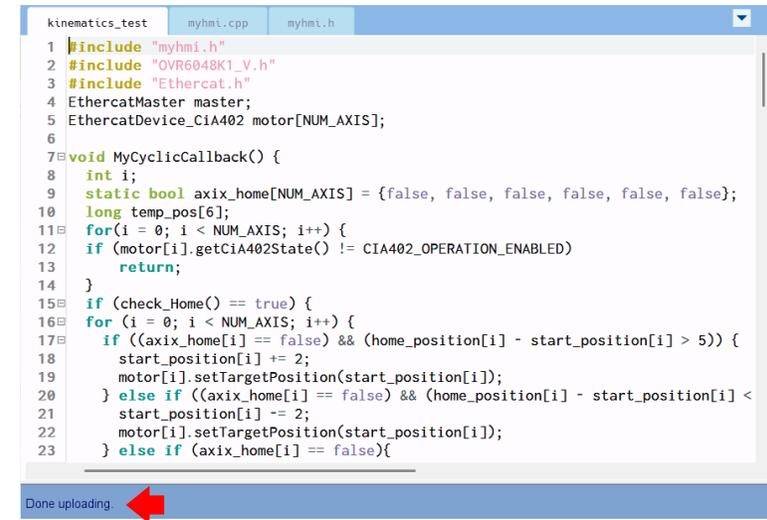
確認沒有錯誤後，再按上傳  按鈕。

上傳完成後，下方視窗會顯示「Done uploading」。



```
kinematics_test | 86Duino Coding 501
File Edit Sketch Tools Help
[Compile] [Upload]
kinematics_test myhmi.cpp myhmi.h
1 #include "myhmi.h"
2 #include "OVR6048K1_V.h"
3 #include "Ethercat.h"
4 EthercatMaster master;
5 EthercatDevice_CiA402 motor[NUM_AXIS];
6
```

86Duino Upload & Verify icon



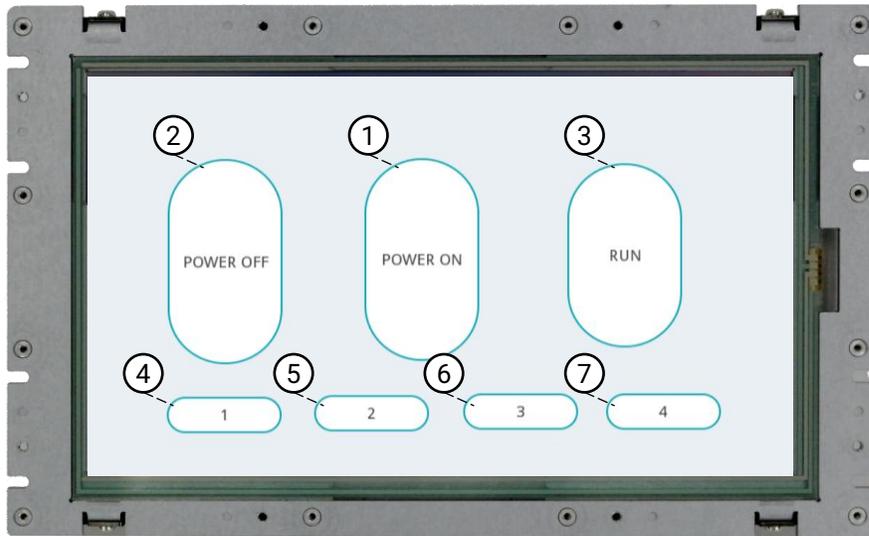
```
kinematics_test myhmi.cpp myhmi.h
1 #include "myhmi.h"
2 #include "OVR6048K1_V.h"
3 #include "Ethercat.h"
4 EthercatMaster master;
5 EthercatDevice_CiA402 motor[NUM_AXIS];
6
7 void MyCyclicCallback() {
8   int i;
9   static bool axix_home[NUM_AXIS] = {false, false, false, false, false, false};
10  long temp_pos[6];
11  for(i = 0; i < NUM_AXIS; i++) {
12    if (motor[i].getCiA402State() != CIA402_OPERATION_ENABLED)
13      return;
14  }
15  if (check_Home() == true) {
16    for (i = 0; i < NUM_AXIS; i++) {
17      if ((axix_home[i] == false) && (home_position[i] - start_position[i] > 5)) {
18        start_position[i] += 2;
19        motor[i].setTargetPosition(start_position[i]);
20      } else if ((axix_home[i] == false) && (home_position[i] - start_position[i] <
21        start_position[i] - 2;
22        motor[i].setTargetPosition(start_position[i]);
23      } else if (axix_home[i] == false){
```

86Duino Down uploading

範例：Kinematics Test - 5

結論

成功將程式上傳至 QEC-M-090T 後，即可在裝置上看到對應 HMI 畫面。



QEC-M-090T with kinematics_test example

各按鈕功能

- ① **POWER ON (p1b1)**
`power_on();`
- ② **POWER OFF (p1b2)**
`power_off();`
- ③ **RUN (p1b3)**
`run_demo3(5.02);`
- ④ **1 (p1b4)**
`Set_Des_Position(0.0, 24.0, 26.8, 0.0, -45.0, 0.0);`
`prepare_move();`
- ⑤ **2 (p1b5)**
`Set_Des_Position(0.0, 24.0, 26.8, 0.0, -20.0, 0.0);`
`prepare_move();`
- ⑥ **3 (p1b6)**
`Set_Des_Position(0.0, 24.0, 26.8, 0.0, 20.0, 0.0);`
`prepare_move();`
- ⑦ **4 (p1b7)**
`Set_Des_Position(0.0, 24.0, 26.8, 0.0, 45.0, 0.0);`
`prepare_move();`

* 註：下一頁將說明各函式的功能與動作執行方式。

使用流程 & API 說明 - 1

使用流程

- **setup** → `begin()` ▶ `attach()` ▶ `setDc()` ▶ `setCiA402Mode()` ▶ `attachRTCyclicCallback()` ▶ `start()`
- **power_on** → `enable()` ▶ `poweron_init()` ▶ `home()`
- **move** → `Set_Des_Position()` ▶ `prepare_move()` / `prepare_arc_move()` (targets stream via callback)
- **loop** → HMI event.

API 說明

OVR6048K1_V.h

API	Where	Purpose	Typical use
<code>OVR6048K1_V_begin()</code>	setup	Init internal buffers/params	Call once before EtherCAT start
<code>OVR6048K1_V_loop()</code>	loop	Internal housekeeping/state	Call every loop iteration
<code>OVR6048K1_V_callback(long* pos)</code>	cyclic callback	Produce next target (relative/abs)	If returns true, write targets to PDO
<code>Set_Des_Position(x,y,z,rx,ry,rz)</code>	loop/HMI before motion	Set goal pose (required before line/arc)	Then call <code>prepare_move()</code> or arc API
<code>prepare_move()</code>	loop/HMI	Execute linear move to goal	Runs via the cyclic callback
<code>prepare_arc_move()</code>	loop/HMI	Execute arc move (uses plane/center/dir)	Set plane/center/dir first
<code>set_arc_plane(XY/XZ/YZ)</code>	loop/HMI	Select arc plane	Pre-req for arc
<code>set_arc_dir(ARC_CW/ARC_CCW)</code>	loop/HMI	Arc direction	Optional if default ok
<code>Set_center_offset(cx,cy,cz)</code>	loop/HMI	Arc center (relative)	Used before <code>prepare_arc_move()</code>
<code>poweron_init()</code>	power_on	Initialize internal parameters after enable	Call once when enabling
<code>emergency_stop()</code>	error/safety	Immediate stop; re-init required to resume	Use for stop

使用流程 & API 說明 - 2

API 說明

Ethercat.h

API	Where	Purpose	Typical use
master.begin()	setup	Scan SubDevice, enter Pre-Op	First EtherCAT call
motor[i].attach(i, master)	setup	Bind SubDevice i to MDevice	For each axis
motor[i].setDc(cycle_ns)	setup	Enable Distributed Clock (DC) & set cycle time	e.g., 250'000 ns (250 µs) or 1'000'000 ns
master.attachRTCyclicCallback(cb)	setup	Register real-time cyclic callback	Your PDO loop
master.start(cycle_ns, ECAT_SYNC)	setup	Configure SM/FMMU/DC; enter OP	Start cyclic op
pdoWrite()/pdoRead()	callback	Write/Read process data	Typically wrapped by motor APIs

Ethercat.h > CiA402

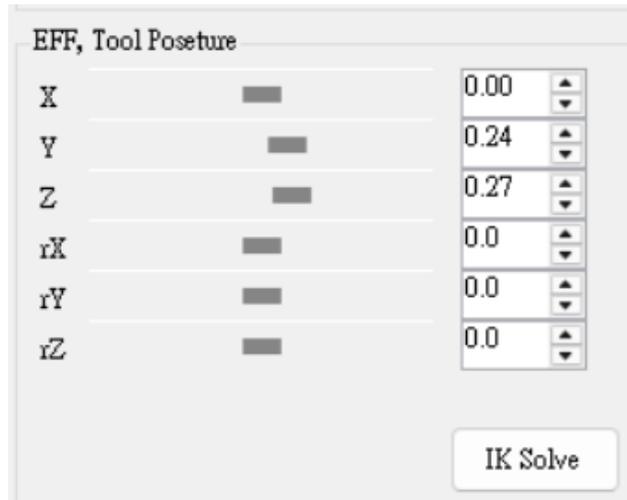
API	Where	Purpose	Typical use
motor[i].setCiA402Mode(CIA402_CSP_MODE)	setup/loop	Select CSP mode	Before start
motor[i].enable()	power_on/loop	Enter Operation Enabled	After OP & ready
motor[i].getCiA402State()	callback/loop	Query drive state	Gate motion logic
motor[i].setTargetPosition(pulse)	callback/loop	Write target (PDO)	From OVR callback/logic
motor[i].getPositionActualValue()	callback/loop	Read actual position	Homing/monitoring

直接將 Viewer 數值用於 API

使用者可將 **Kinematics Viewer** 中的姿態數值直接套用到程式碼（或 **ArduBlock**）中，以驅動機械手臂。在 Viewer 中，拖曳滑桿或輸入數值來模擬姿態，然後將這些值複製到 API。

「**EFF, Tool Posture**」中的欄位（**X, Y, Z, rX, rY, rZ**）會 1:1 對應到：

```
Set_Des_Position(0.0, 24.0, 26.8, 0.0, -45.0, 0.0);  
prepare_move();
```



Kinematics viewer : EFF, Tool Posture

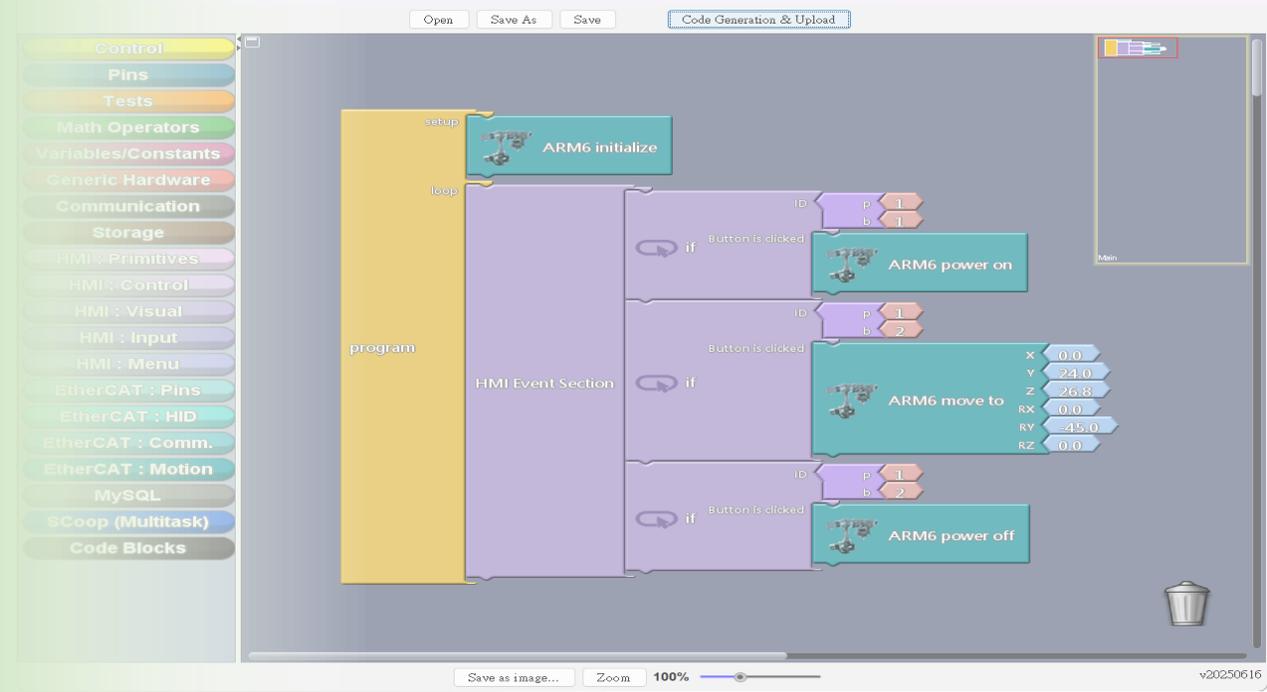
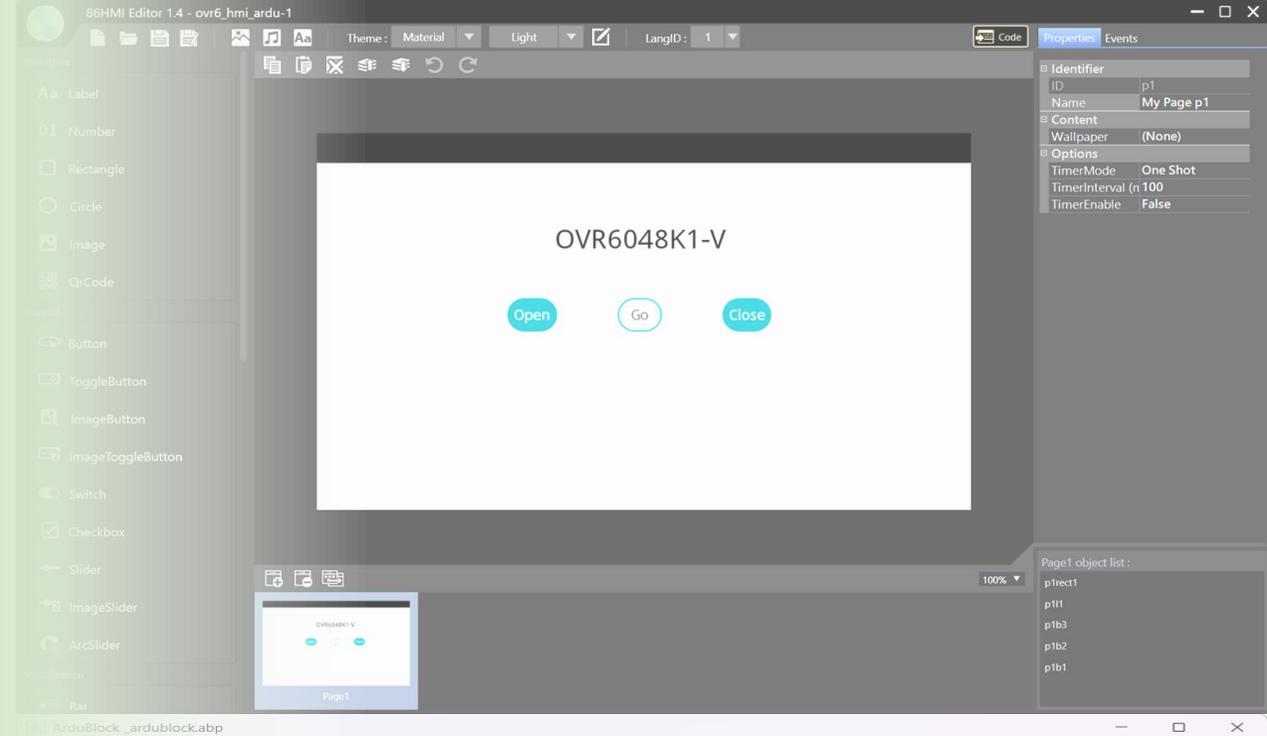
步驟：

1. 在 EFF 工具姿態中，設定/確認 X、Y、Z、rX、rY、rZ 座標。
2. 將它們 1:1 複製到 `Set_Des_Position(x, y, z, rX, rY, rZ);`。
3. 呼叫 `prepare_move();` 執行。

然後，您可以控制機械手臂移動到與運動學檢視器中相同的位置。

4. 入門指南範例 - 1

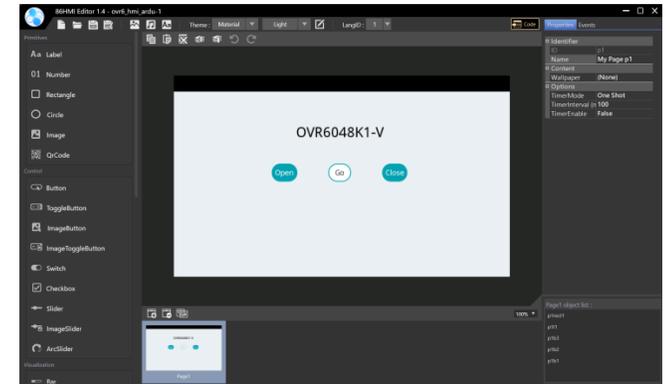
低程式碼範例：86HMI + ArduBlock



86HMI + ArduBlock - 介紹 1

86HMI – UI 設計工具 (HMI)

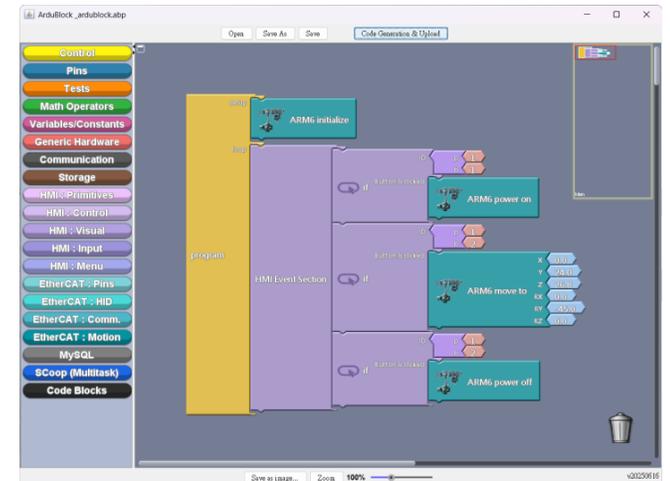
- 是什麼：基於 LVGL86 Library 的拖拉式畫面編輯器。
- 可建立什麼：按鈕、標籤等，並綁定唯一 ID (例如 p1b1、p1b2)。
- 會產生什麼：C/C++ UI 程式碼骨架，可貼入 sketch；由自動產生的 **HMI Event Section** 處理。
- 為何重要：讓操作員可直接從面板觸發動作 (開關電源、預設移動、執行 demo)。



86HMI Editor

ArduBlock – 控制邏輯 (積木)

- 是什麼：建構在 86Duino IDE 上的積木式編輯器。
- 可建立什麼：HMI、EtherCAT 與 Motion 的高階邏輯。
- 會產生什麼：可執行的 86Duino 程式碼，呼叫對應 API (像是 `power_on()`, `Set_Des_Position()`, `prepare_move()`)。



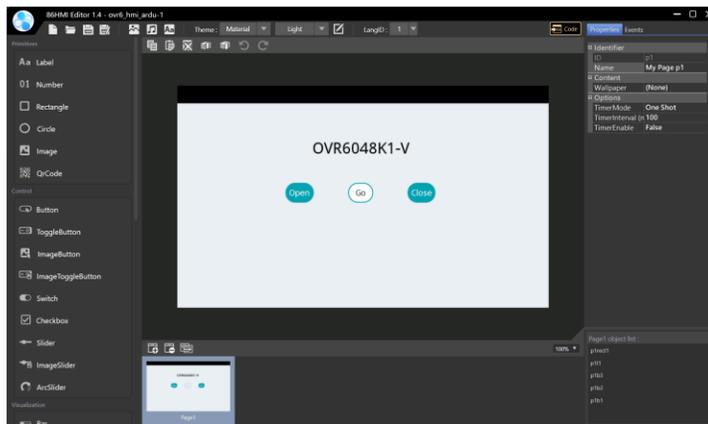
ArduBlock

* 註：ARM6 類別積木僅適用於 OVR6048K1-V 專案。

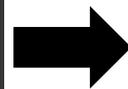
86HMI + ArduBlock - 介紹 2

為什麼兩者都要用？

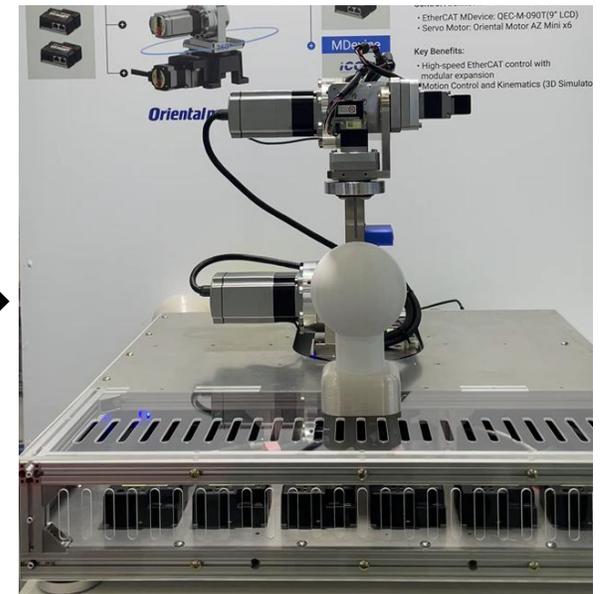
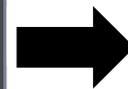
- **86HMI** 負責設計操作介面。
- **ArduBlock** 負責把 UI 事件連接到機器人動作。
- 結合後即可形成從 **按鈕** → **API 呼叫** → **EtherCAT 運動** 的低程式碼開發流程。



86HMI Editor



ArduBlock



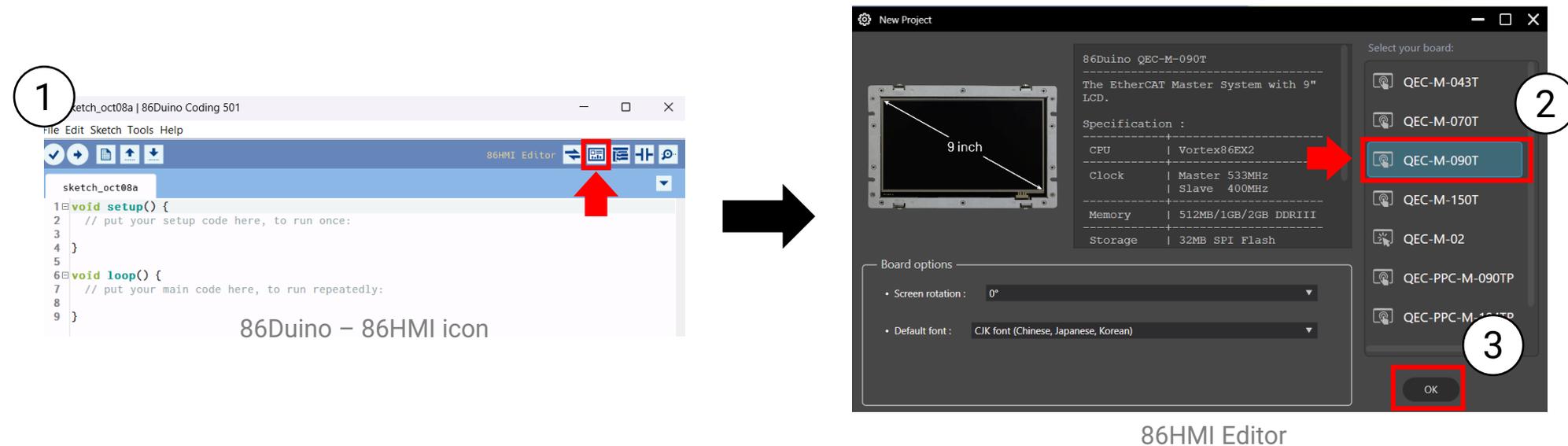
OVR6048K1

範例：Low code Kinematics - 1

步驟 1：開新專案並設計 HMI (86HMI)

請先參考 [範例：Kinematics test](#) 完成環境設定。

- ① 開啟 86HMI Editor。
- ② 選擇 QEC-M-090T。
- ③ 點擊「OK」後進入編輯器。

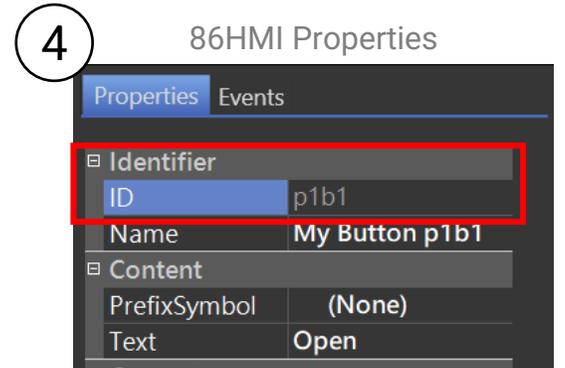
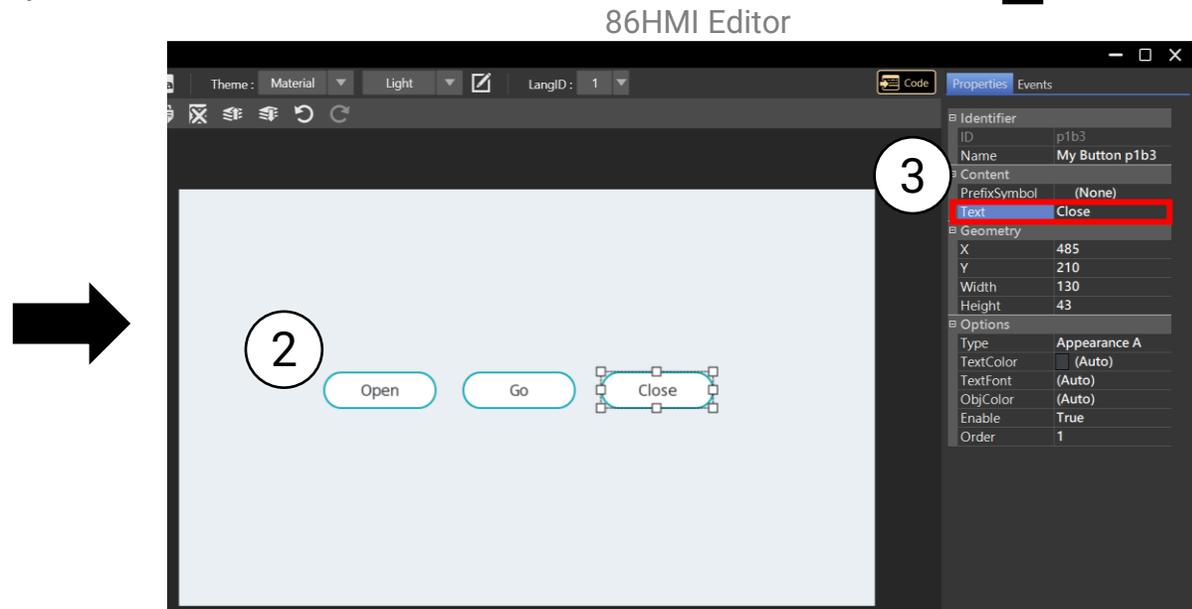
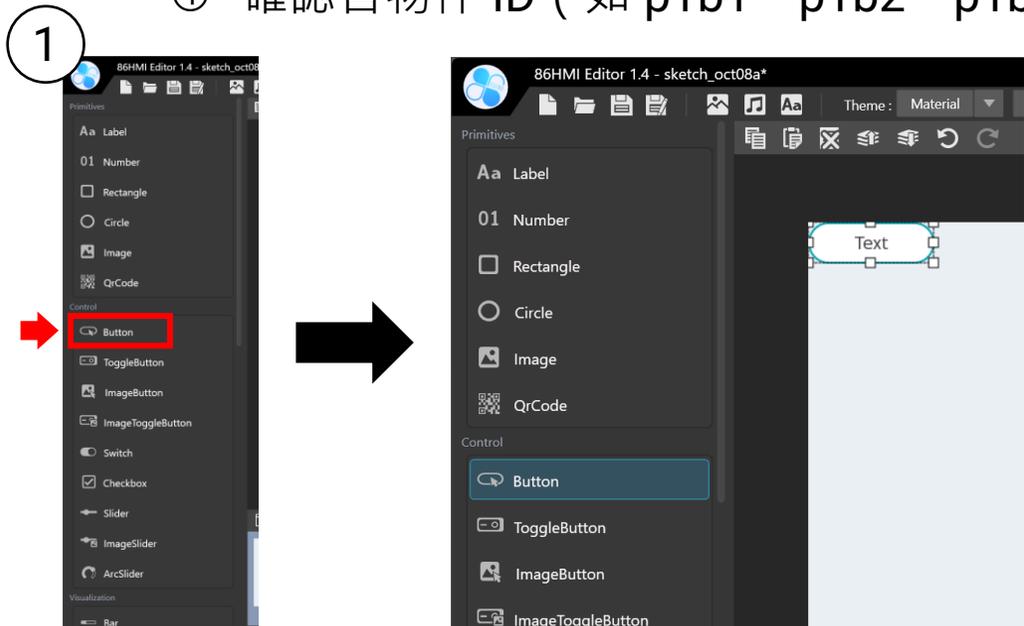


範例：Low code Kinematics - 2

步驟 2：新增 3 個按鈕

在首頁建立 3 個按鈕元件：**Open**、**Go**、**Close**

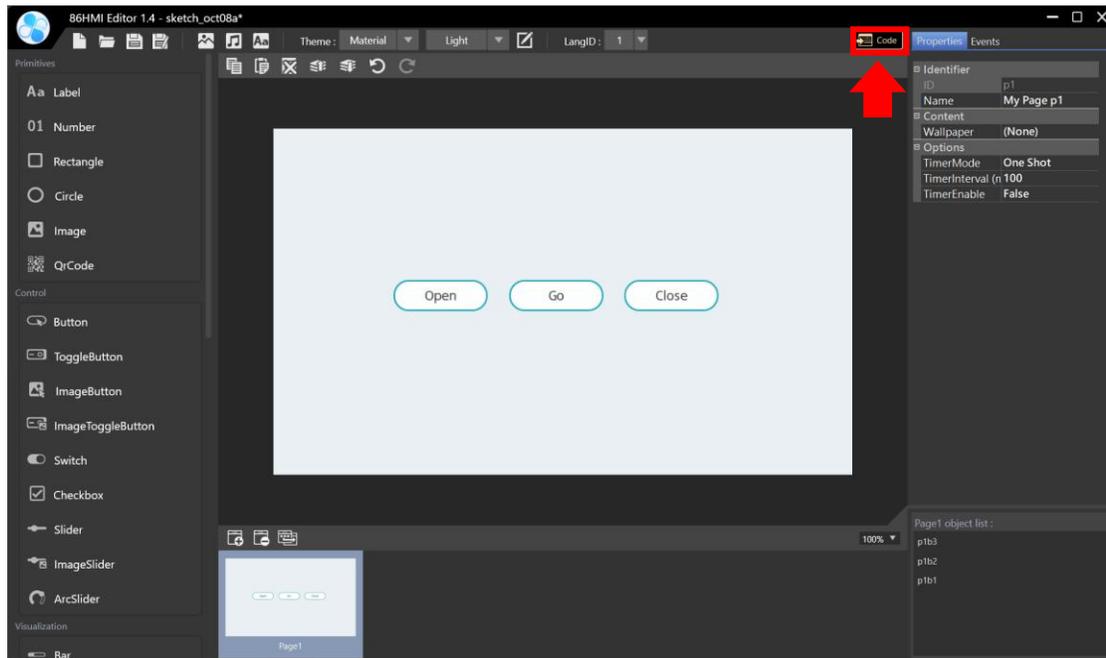
- ① 點選左側選單中的「**Button**」
- ② 複製出 **3 個按鈕**
- ③ 於右側屬性選單將 Text 內容改為 **Open**、**Go**、**Close**
- ④ 確認各物件 ID (如 p1b1、p1b2、p1b3)



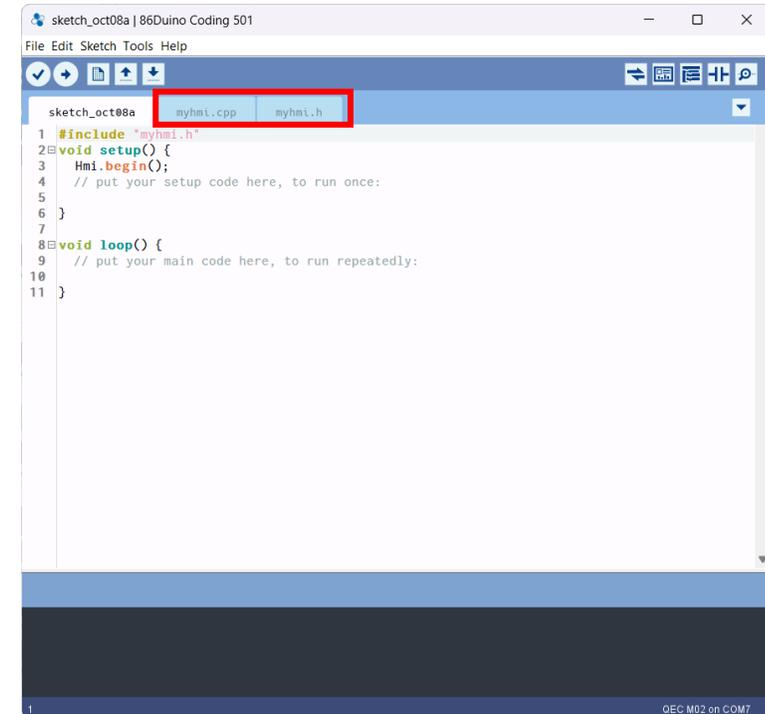
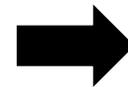
範例：Low code Kinematics - 3

步驟 3：86HMI 產生程式碼

點擊右上角「Code」產生 UI 程式碼到你的 sketch 中。



86HMI Editor

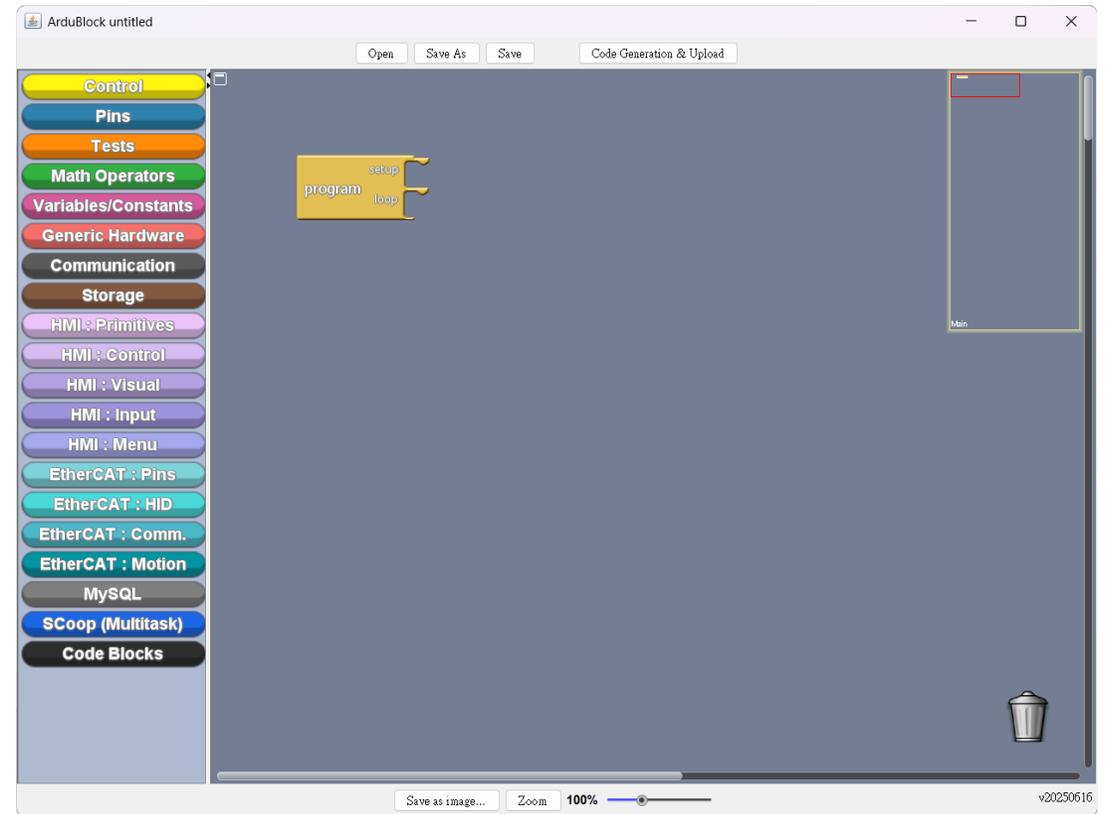
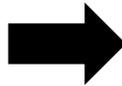


86Duino – 86HMI files

範例：Low code Kinematics - 4

步驟 4：開啟 ArduBlock

點擊右上角圖示開啟 ArduBlock。



ArduBlock

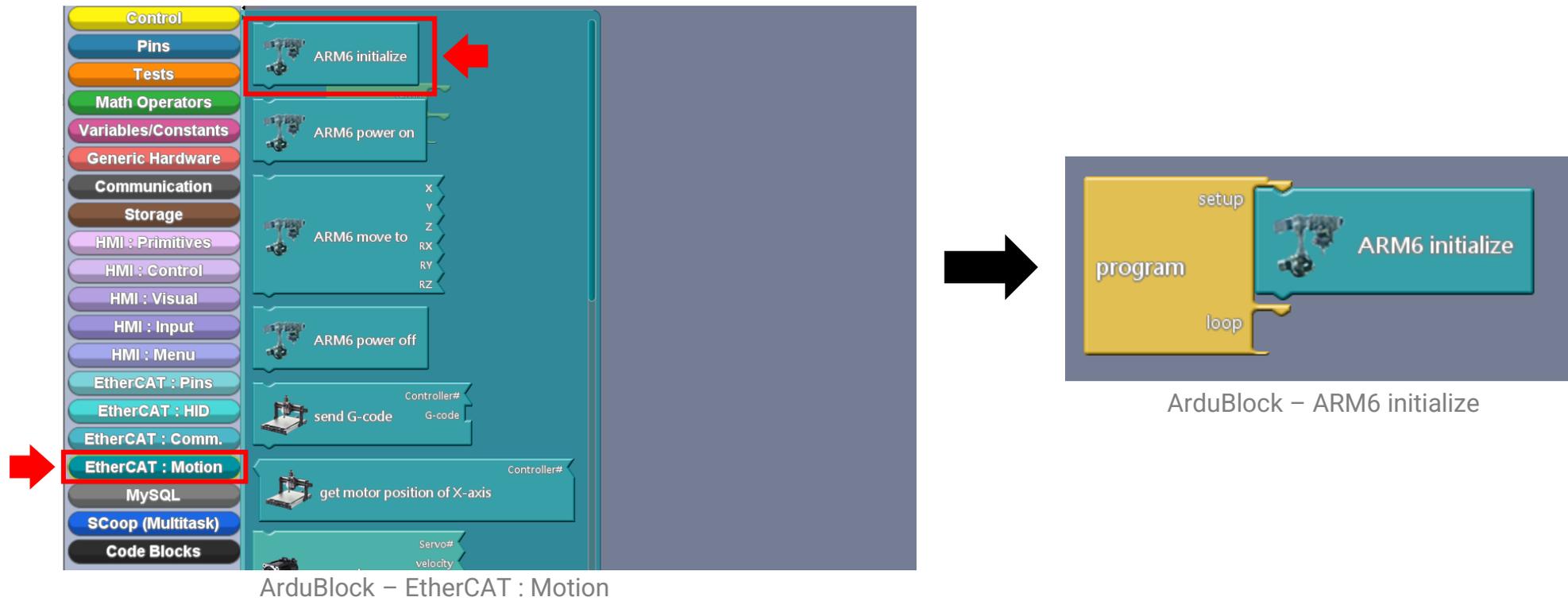
範例：Low code Kinematics - 5

步驟 5：在 **setup** 中放入 **ARM6 Initialize** 積木

先初始化 **Oriental Motor** 驅動器參數。

這些已封裝成一個名為「**ARM6 initialize**」的積木，位於「**EtherCAT: Motion**」類別中。

請將它放到程式的 **setup** 區塊內。

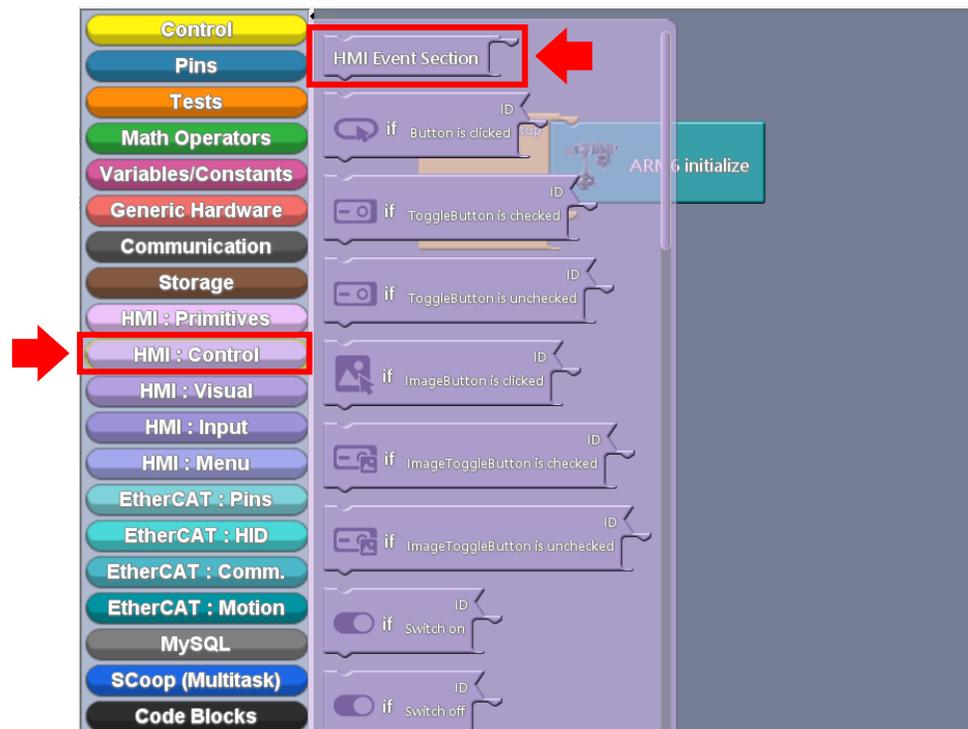


範例：Low code Kinematics - 6

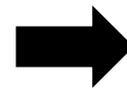
步驟 6：在 loop 中放入 HMI event 積木

因為要與機械手臂和 HMI 互動，必須在 loop 中放入 HMI event。

按鈕事件（如 clicked、pressing）需關聯到「HMI Event Section」，位於「HMI : Control」類別中。



ArduBlock – HMI : Control



ArduBlock – HMI Event section

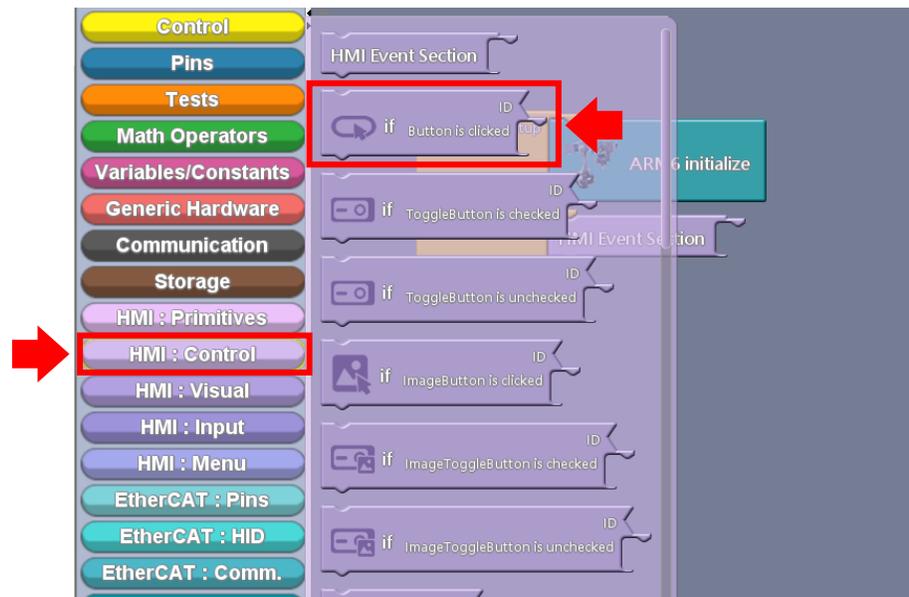
範例：Low code Kinematics - 7

步驟 7：在 HMI event 中放入 Button event

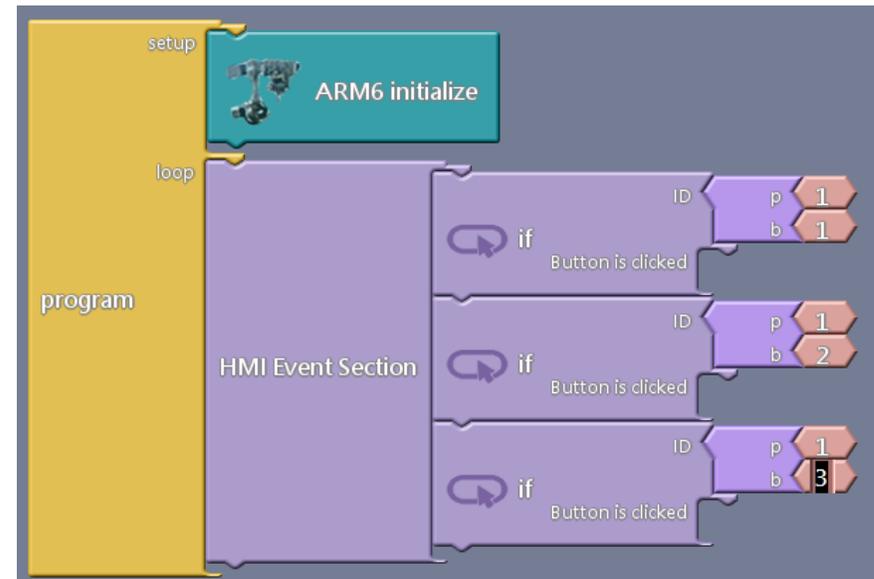
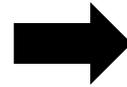
將「if Button is clicked」積木放入「HMI Event Section」中。

當指定按鈕被按下時，HMI event 會回傳 true。

因需要 3 個按鈕，請複製 3 份，並將 ID 分別改成 p1b1、p1b2、p1b3。



ArduBlock – HMI : Control



ArduBlock – if Button is clicked

範例：Low code Kinematics - 8

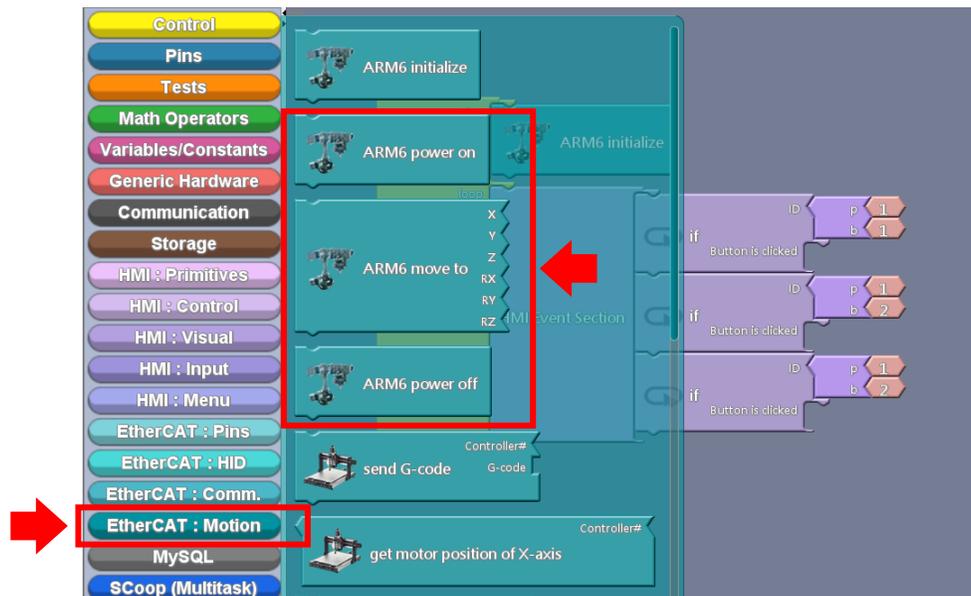
步驟 8：在 Button event 中放入 ARM6 power-on / move-to / power-off

OVR6048K1 的功能已封裝為不同積木，包括：ARM6 power on、ARM6 move to、ARM6 power off。這些都位於「EtherCAT: Motion」類別中。

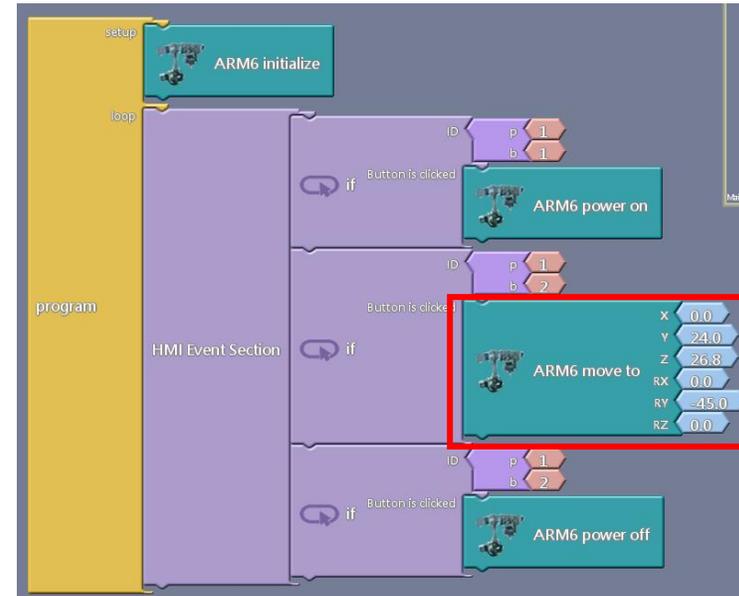
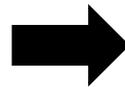
對應方式如下：

- p1b1 → ARM6 power on、p1b2 → ARM6 move to、p1b3 → ARM6 power off

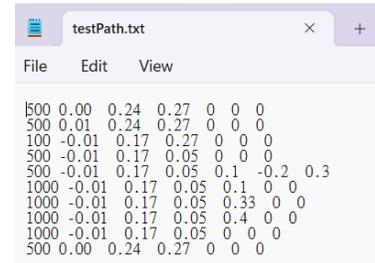
* 註：ARM6 move to 的參數請使用 Kinematics Viewer 的位置值。



ArduBlock – EtherCAT: Motion



ArduBlock – ARM6 move to



範例：Low code Kinematics - 9

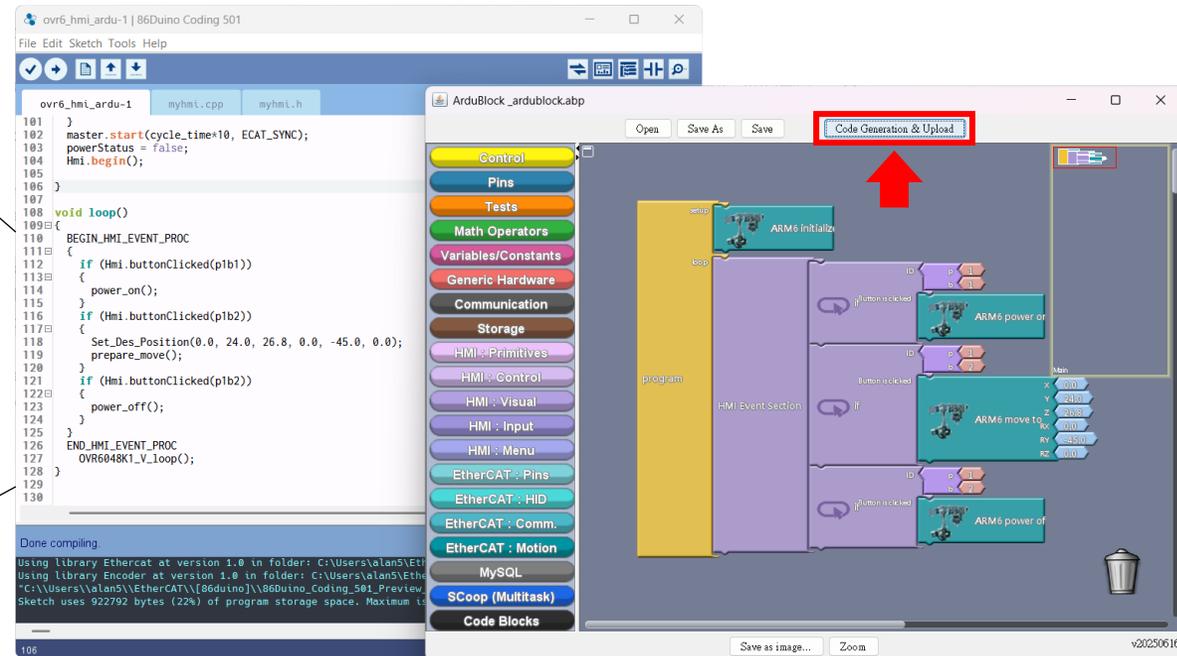
步驟 9：產生程式碼並上傳

完成後可點擊「Code Generate & Upload」按鈕，產生並上傳程式到 QEC-M-090T。

這套積木程式會建立 EtherCAT 通訊、設定 AZ-mini 驅動器，並透過 HMI 按鈕控制 OVR6048K1 機械手臂。

```
107
108 void loop()
109 {
110   BEGIN_HMI_EVENT_PROC
111   {
112     if (Hmi.buttonClicked(p1b1))
113     {
114       power_on();
115     }
116     if (Hmi.buttonClicked(p1b2))
117     {
118       Set_Des_Position(0.0, 24.0, 26.8, 0.0, -45.0, 0.0);
119       prepare_move();
120     }
121     if (Hmi.buttonClicked(p1b2))
122     {
123       power_off();
124     }
125   }
126   END_HMI_EVENT_PROC
127   OVR6048K1_V_loop();
128 }
```

ArduBlock - 86Duino - HMI code

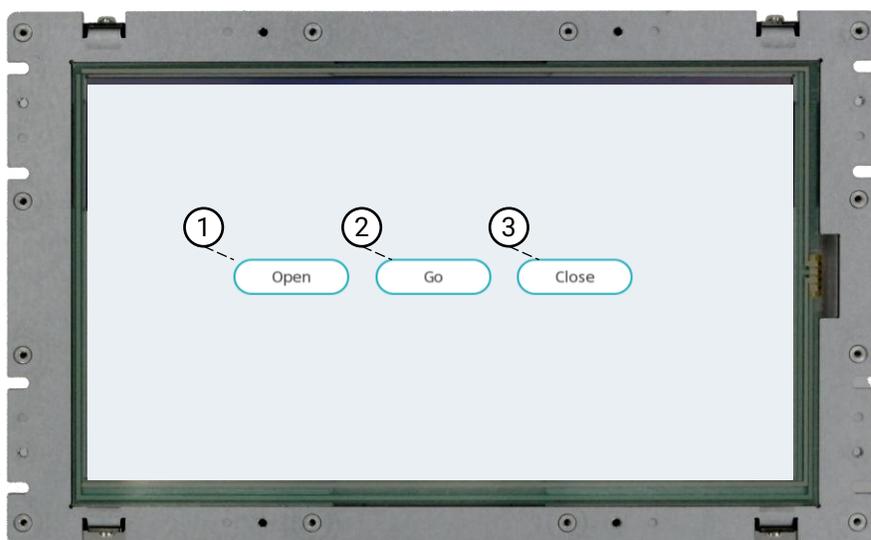


ArduBlock - 86Duino

範例：Low code Kinematics - 10

結論

當成功將程式上傳至 QEC-M-090T 後，即可在裝置上看到對應的低程式碼 HMI 畫面。



QEC-M-090T with low-code example

各按鈕功能

① Open (p1b1)



② Go (p1b2)



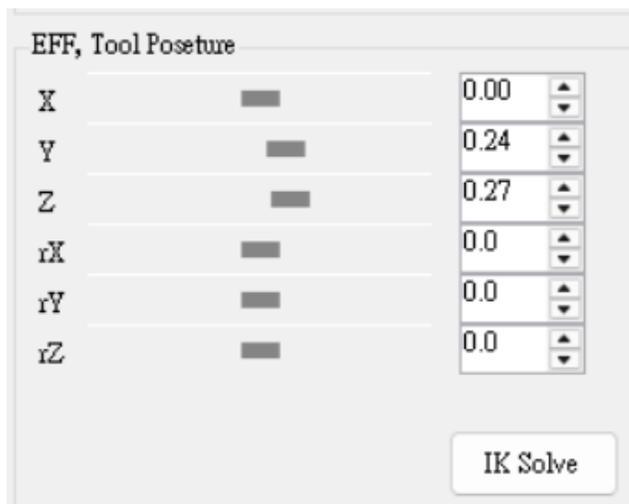
③ Close (p1b3)



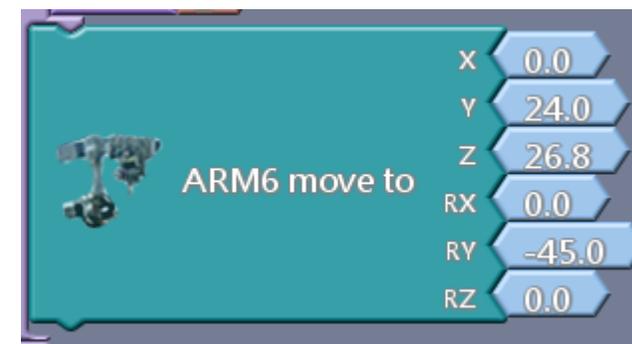
直接將 Viewer 數值用於 ArduBlock

與之前相同，使用者可以從運動學檢視器中取得相同的姿態值，並將其應用到 **API**（或 **ArduBlock**）中來驅動機械手臂。在檢視器中，拖曳滑桿或輸入數值來模擬姿態；然後將這些值複製到 API 中。

「**EFF, Tool Posture**」中的欄位（X, Y, Z, rX, rY, rZ）可 1:1 對應到：



Kinematics viewer : EFF, Tool Posture



ArduBlock Moving the Robotics Arm Block

步驟：

1. 在 EFF 工具姿態中，設定/確認 X、Y、Z、rX、rY、rZ 座標。
2. 將它們 1:1 複製到 **ARM6 move to(X, Y, Z, RX, RY, RZ)** 積木中。

然後，您可以控制機械手臂移動到與運動學檢視器中相同的位置。

5. 入門指南範例 - 2

Robot Arm 6R 範例

The diagram shows a 6-axis articulated robot arm with various components labeled. On the left, a 'Mini Driver' box contains three 'Orientalmotor AZD-KREN' modules. On the right, another 'Mini Driver' box contains three 'Orientalmotor AZD-KREN' modules, and an 'MDevice' box contains one 'iCO' module. The robot arm is shown in a photograph at the bottom, with a large 'Orientalmotor' motor at the base. The background is a light blue gradient.

Mini Driver
Orientalmotor
AZD-KREN

Mini Driver
Orientalmotor
AZD-KREN

Mini Driver
Orientalmotor
AZD-KREN

MDevice
iCO

Orientalmotor

控制架構:

- EtherCAT 控制器: QEC-M-090T
- 伺服馬達: 東方馬達 AZ Mini x6

主要優勢:

- 高速 EtherCAT 控制, 模組化擴展
- 運動控制與運動學 (3D 模擬器)

Control Architecture:

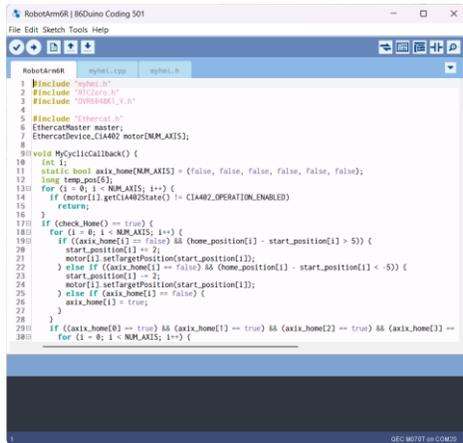
- EtherCAT MDevice: QEC-M-090T(9" L)
- Servo Motor: Oriental Motor AZ Mini

Key Benefits:

- High-speed EtherCAT control with modular expansion
- Motion Control and Kinematics (3D S

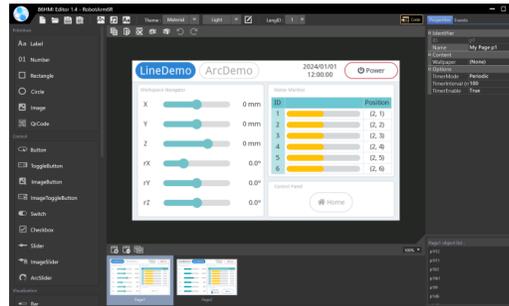
範例：RobotArm6R - 介紹

使用 QEC-M-090T + AZD-KRED × OVR6048K1-V，在 ECAT_SYNC + DC 模式下，我們可執行平滑的末端執行器球面軌跡運動，從規劃、模擬到一鍵 HMI 執行，形成完整流程。

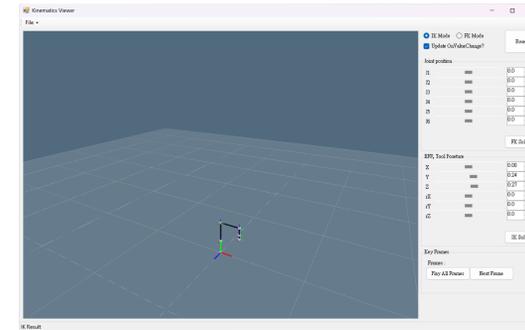


```
1 #include "mymat.h"
2 #include "VICZero.h"
3 #include "OVR6048K1_V.h"
4
5 #include "EtherCAT.h"
6 EtherCATMaster master;
7 EtherCATDevice_I14402 motor(MM_AXIS);
8
9 void MyCyclicCallback() {
10   int i;
11   static bool axis_home[MM_AXIS] = {false, false, false, false, false, false};
12   long temp_pos[3];
13   for (i = 0; i < MM_AXIS; i++) {
14     if (motor[i].getI14402State() != CI14402_OPERATION_ENABLED)
15       return;
16   }
17   if (check_home() == true) {
18     for (i = 0; i < MM_AXIS; i++) {
19       if ((axis_home[i] == false) && (home_position[i] - start_position[i] > 5)) {
20         start_position[i] = 2;
21         motor[i].setTargetPosition(start_position[i]);
22       } else if ((axis_home[i] == false) && (home_position[i] - start_position[i] < -5)) {
23         start_position[i] = -2;
24         motor[i].setTargetPosition(start_position[i]);
25       } else if (axis_home[i] == false) {
26         axis_home[i] = true;
27       }
28     }
29   }
30   if ((axis_home[0] == true) && (axis_home[1] == true) && (axis_home[2] == true) && (axis_home[3] == true) && (axis_home[4] == true) && (axis_home[5] == true)) {
31     for (i = 0; i < MM_AXIS; i++) {
```

RobotArm6R 86Duino example



RobotArm6R HMI example



RobotArm6R Kinematics Viewer



RobotArm6R Zero Post

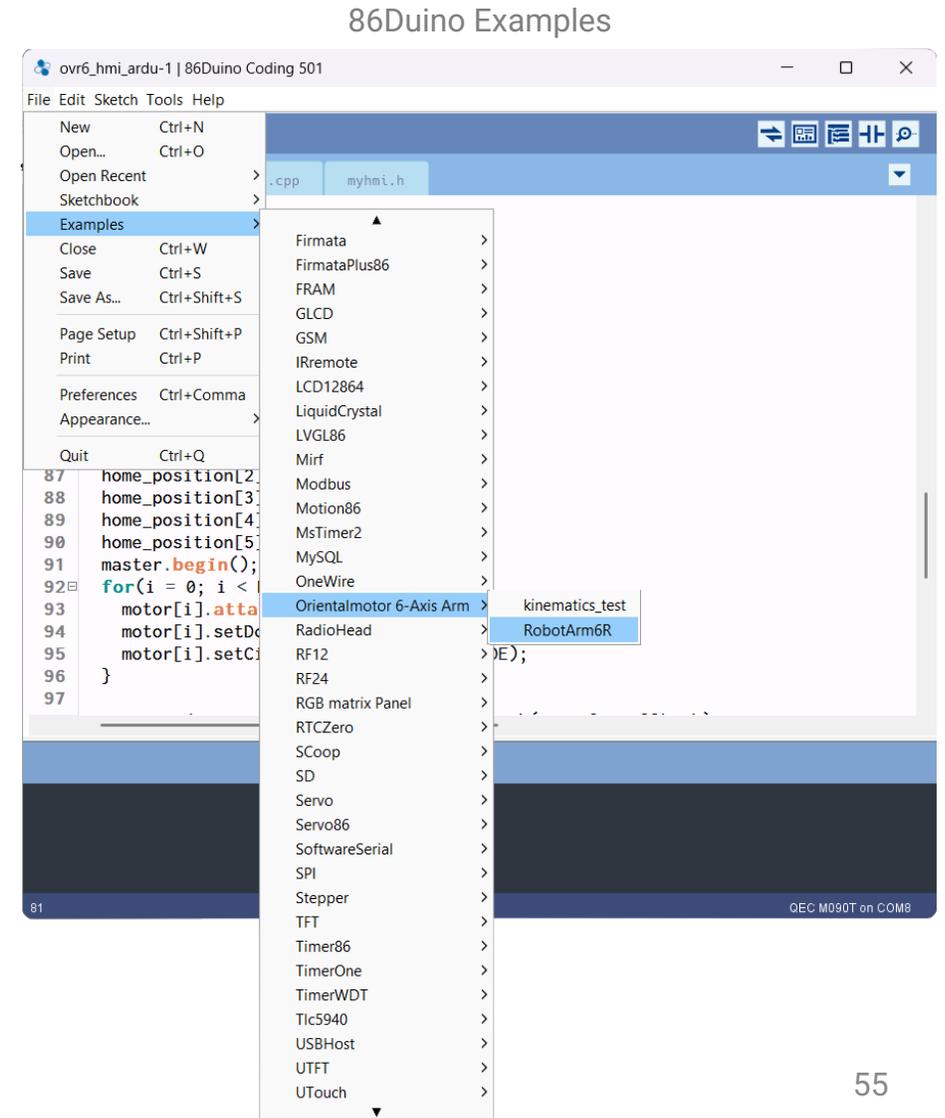


RobotArm6R example

範例：RobotArm6R - 1

步驟 1：開啟範例

- 在 **Tools** → **Board** 選擇 **QEC-M090T**，並設定正確 Port 後即可找到範例集。
- 開啟：**File** → **Examples** → **Orientalmotor 6-Axis Arm** → **RobotArm6R**。



範例：RobotArm6R - 2

步驟 2：查看程式並開啟 HMI

- 開啟 RobotArm6R 後會載入對應 sketch。
使用的函式庫：

- myhmi.h** – HMI helpers
- OVR6048K1_V.h** – robot motion helpers
- Ethercat.h** – EtherCAT protocol
- RTCZero.h** – RTC timer library

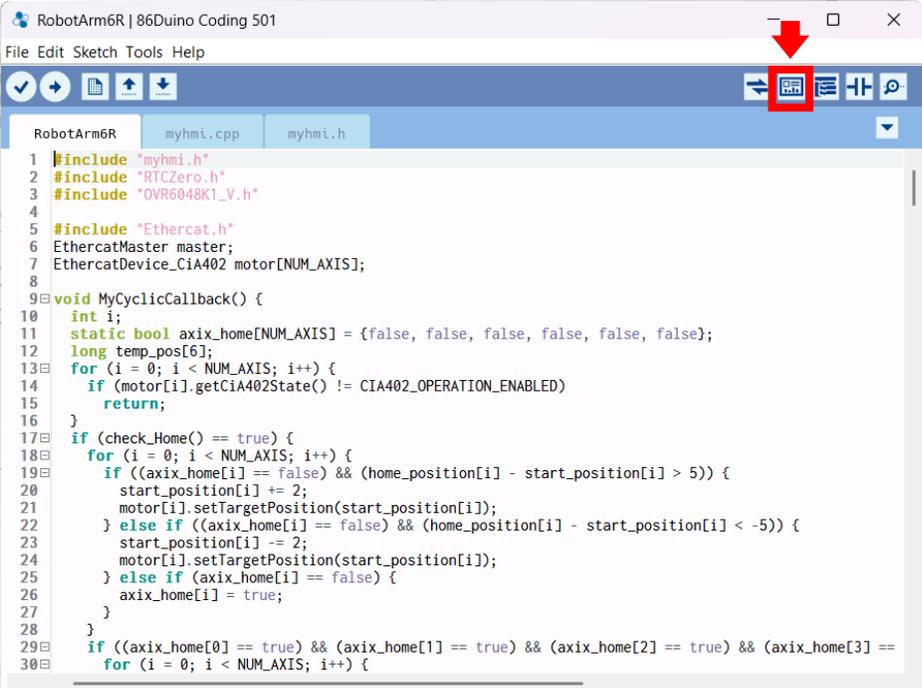
- 本範例設定：

- EtherCAT 週期：**250 microseconds (µs)**.
- EtherCAT 模式：**ECAT_SYNC**.
- Distributed Clock (DC): **Enable (250 µs)**.
- CiA-402 模式：**CSP (Cyclic Synchronous Position)**.

```
91 master.begin();
92 for(i = 0; i < NUM_AXIS; i++) {
93     motor[i].attach(i, master);
94     motor[i].setDc(cycle_time*10);
95     motor[i].setCiA402Mode(CIA402_CSP_MODE);
96 }
97
98 error_code = master.attachRTcyclicCallback(MyCyclicCallback);
99 if (error_code != 0) {
100     printf("error : %d\n", error_code);
101 }
102 master.start(cycle_time*10, ECAT_SYNC); // cycle_time = 25000; // nanosecond
```

RobotArm6R.ino Code about EtherCAT

- 點擊右上角 86HMI 圖示以啟動 HMI Editor。



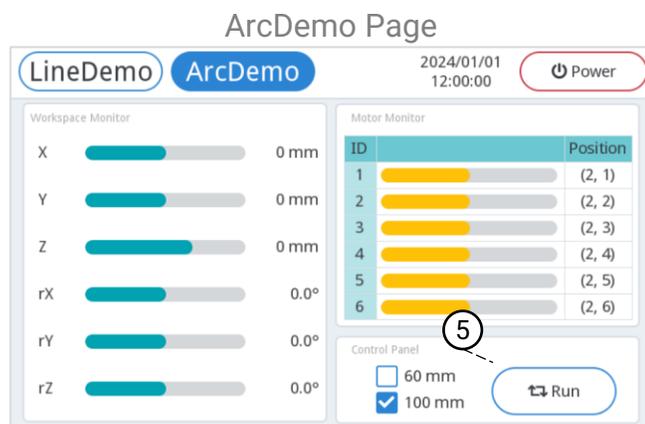
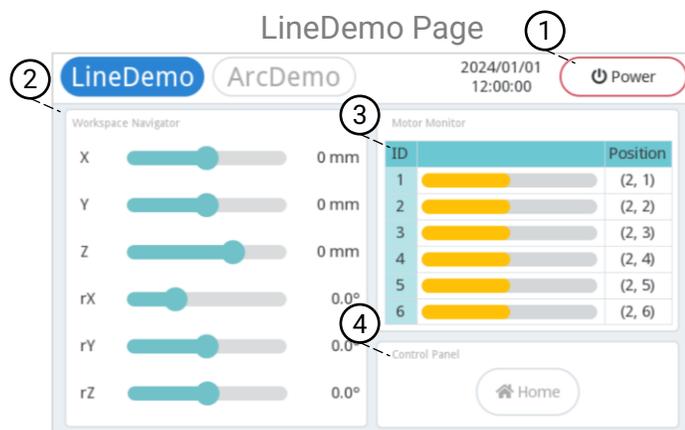
```
RobotArm6R | 86Duino Coding 501
File Edit Sketch Tools Help
[Icons]
RobotArm6R myhmi.cpp myhmi.h
1 #include "myhmi.h"
2 #include "RTCZero.h"
3 #include "OVR6048K1_V.h"
4
5 #include "Ethercat.h"
6 EthercatMaster master;
7 EthercatDevice_CiA402 motor[NUM_AXIS];
8
9 void MyCyclicCallback() {
10     int i;
11     static bool axix_home[NUM_AXIS] = {false, false, false, false, false, false};
12     long temp_pos[6];
13     for (i = 0; i < NUM_AXIS; i++) {
14         if (motor[i].getCiA402State() != CIA402_OPERATION_ENABLED)
15             return;
16     }
17     if (check_Home() == true) {
18         for (i = 0; i < NUM_AXIS; i++) {
19             if ((axix_home[i] == false) && (home_position[i] - start_position[i] > 5)) {
20                 start_position[i] += 2;
21                 motor[i].setTargetPosition(start_position[i]);
22             } else if ((axix_home[i] == false) && (home_position[i] - start_position[i] < -5)) {
23                 start_position[i] -= 2;
24                 motor[i].setTargetPosition(start_position[i]);
25             } else if (axix_home[i] == false) {
26                 axix_home[i] = true;
27             }
28         }
29     }
30     if ((axix_home[0] == true) && (axix_home[1] == true) && (axix_home[2] == true) && (axix_home[3] == true) && (axix_home[4] == true) && (axix_home[5] == true)) {
31         for (i = 0; i < NUM_AXIS; i++) {
```

RobotArm6R.ino

範例：RobotArm6R - 3

步驟 3：86HMI 版面

- HMI 包含 2 頁：LineDemo 和 ArcDemo。



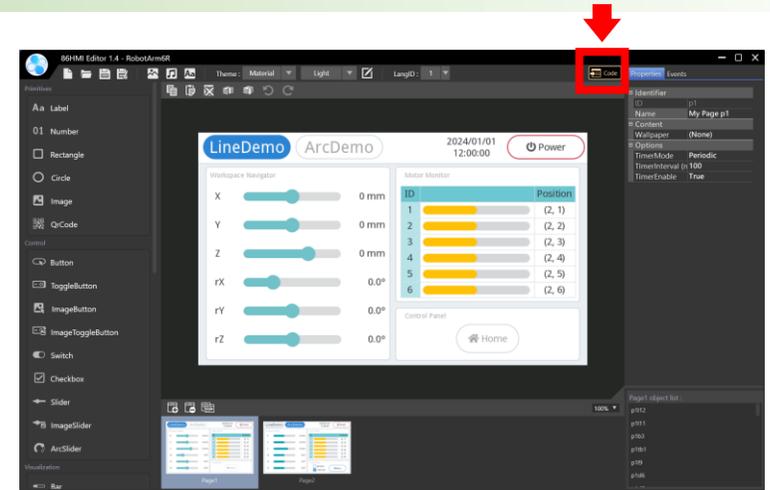
各 HMI 物件功能

- Power**
啟用或停用所有軸驅動器（切換至 CiA402 Enable/Disable）
- Workspace Navigator**
顯示手臂工作空間數值：x、y、z（mm）與 rx、ry、rz（°）。
在 LineDemo 頁中可操作，使用者可直接在工作空間中移動手臂；需先 Homing，且不能與執行中的 demo 同時使用。
- Motor Monitor**
顯示從驅動器讀取的機械手臂位置，並以長條顯示目前位置。
- Control Panel (Home)**
在 LineDemo 中，此區有 Home 按鈕，可讓所有軸回到 home 位置。
- Control Panel (Run)**
在 ArcDemo 中，此區有 Run 按鈕與兩個半徑選項（60mm、10mm）
選擇後按 Run，手臂會沿你選定大小的球面移動；若不再次按 Run，會持續動作。
注意：需先選正確球體尺寸，且必須在 Homing 完成後操作。

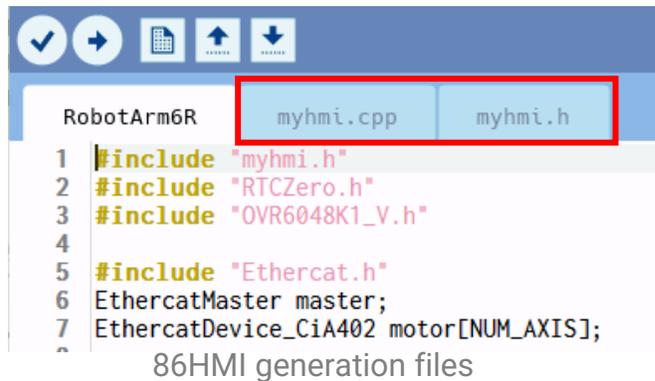
範例：RobotArm6R - 4

步驟 4：86HMI 程式碼產生

- 點擊右上角「Code」按鈕產生事件骨架與 include 敘述，若系統提示請貼入主程式。
- 產生後會看到 **myhmi.h** 與 **myhmi.cpp**，代表 LVGL86 的 helper 與 executor。

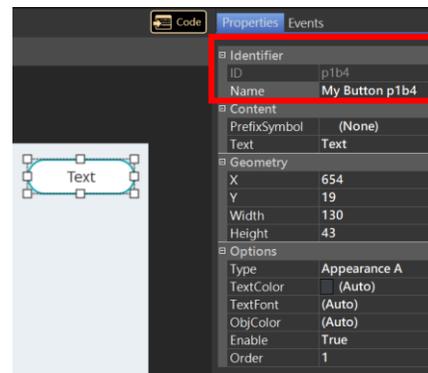


RobotArm6R HMI example

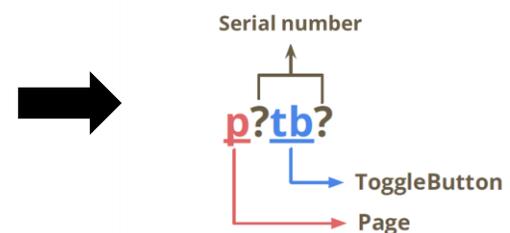


86HMI generation files

86HMI ID Naming rule



Naming rules:



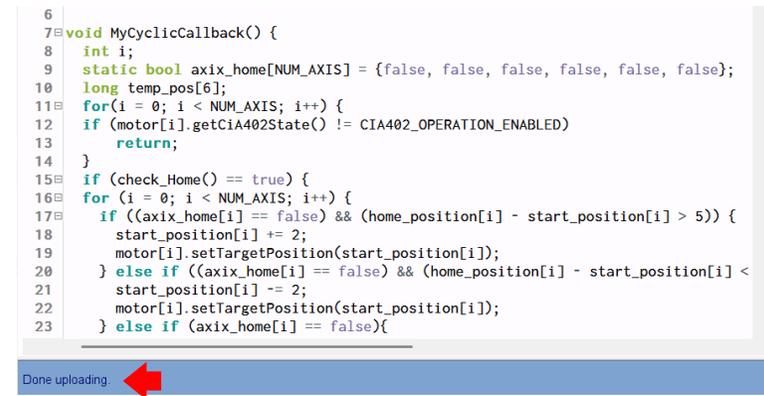
* 補充：在 86HMI 中，每個物件都有唯一 ID。
命名規則為頁碼 + 類別序號。

範例：RobotArm6R - 5

步驟 5：編譯與上傳

程式完成後，點選工具列上的編譯  按鈕進行編譯。
確認沒有錯誤後，再按上傳  按鈕。

上傳完成後，下方視窗會顯示「Done uploading」。



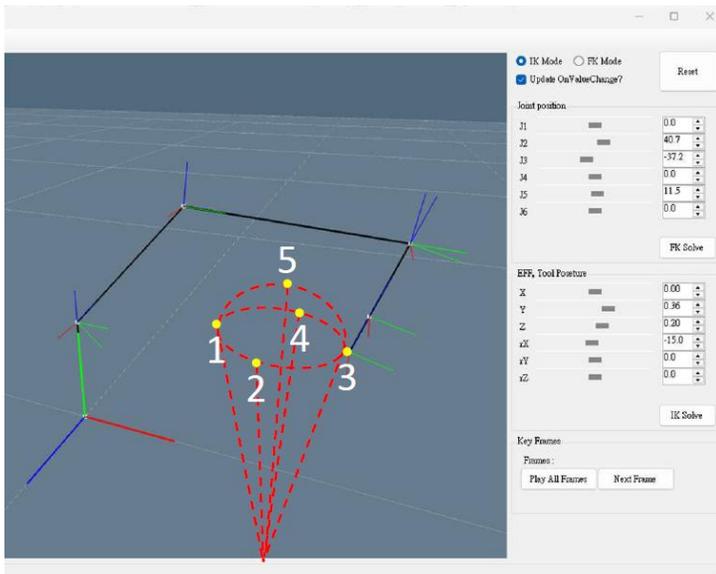
86Duino Done uploading

範例：RobotArm6R 說明

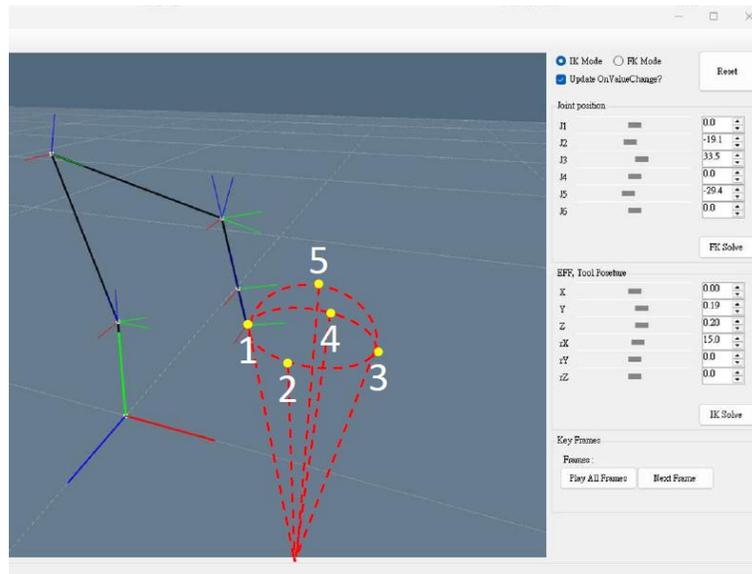
在 RobotArm6R 範例中，手臂會沿著球面半徑移動。共有兩個 demo：

- 60mm 球面
- 100mm 球面

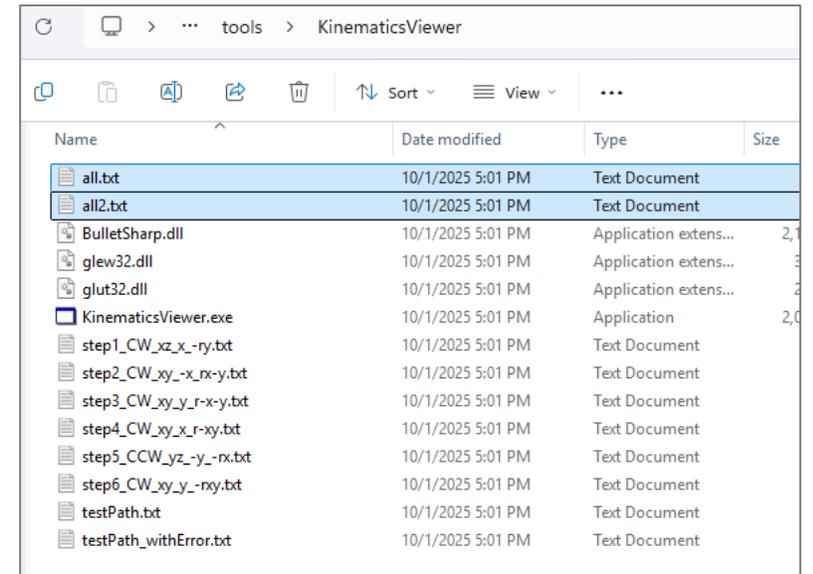
使用者可在 **Kinematics Viewer** 中查看路徑值；對應範例檔為 **all.txt** 與 **all2.txt**。



robotArm6R Moving in Kinematics viewer -1



robotArm6R Moving in Kinematics viewer -2



robotArm6R Sphere demo path

範例：RobotArm6R - 影片

結論



(Video: <https://youtu.be/9-X3WKAfU1o?si=HUMSyUU-MeYdyDMh>)

結束

謝謝！

若需更詳細資訊，請與我們聯繫。

QEC 網站



ICOP 網站



Orientalmotor 網站

