

ECM Starter Kit

ECM-SK

快速入門

版本 : V.1.7.3

日期 : 2020.12

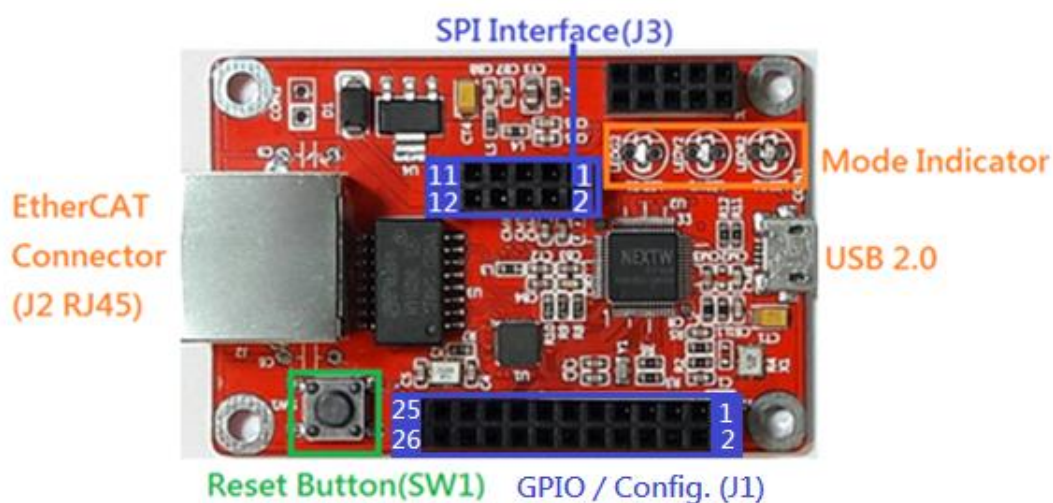
目錄

第 1 章 硬體腳位及使用說明	4
1.1 ECM-SK V1.0 腳位說明	4
1.2 ECM-SK V1.1 腳位說明	5
1.3 LED 狀態顯示及 Config.設定說明.....	6
1.4 系統連接圖	8
1.5 ECM-SK SPI 與控制器間的接線.....	8
1.6 SPI 規格與傳輸模式	9
1.7 SPI 傳輸時序說明	10
1.8 EtherCAT 狀態流程.....	10
1.9 尺寸及固定孔位置	14
第 2 章 命令與回應.....	15
2.1 命令與回應資料結構.....	15
2.1.1 GET_STATUS Command.....	16
2.1.2 SET_STATE Command.....	17
2.1.3 SET_AXIS Command	18
2.1.4 SET_DC Command	20
2.1.5 SET_EX Command.....	21
2.1.6 SET_FIFO Command	22
2.1.7 DRIVE_MODE Command	23
2.1.8 SDO_RD Command.....	24
2.1.9 SDO_WR Command.....	25
2.1.10 ALM_CLR Command	26
2.1.11 SV_ON Command	27
2.1.12 SV_OFF Command	28

2.1.13 IO_RD Command	29
2.1.14 IO_WR Command	29
2.1.15 CSP Command	30
2.1.16 CSV Command	32
2.1.17 CST Command	34
2.1.18 GO_HOME Command	35
2.1.19 ABORT_HOME Command	36
2.1.20 LIO_RD Command	37
2.1.21 LIO_WR Command	38
2.1.22 SW_RESET Command	39
2.2 回應資料	40
第 3 章 流程範例	42
第 4 章 動態函式庫	43
4.1 動態函式庫簡介	43
4.2 NEXTWUSBLib 函式庫	43
4.2.1 OpenECMUSB 函式說明	43
4.2.2 CloseECMUSB 函式說明	44
4.2.3 ECMUSBWrite 函式說明	44
4.2.4 ECMUSBRead 函式說明	44
4.3 NEXTWUSB_dotNET 函式庫	45
4.3.1 OpenECMUSB 函式說明	47
4.3.2 CloseECMUSB 函式說明	48
4.3.3 ECMUSBWrite 函式說明	48
4.3.4 ECMUSBRead 函式說明	48
4.3.5 ClearCmdData 函式說明	49
4.4 Visual Studio 專案環境設定	50

第 1 章 硬體腳位及使用說明

1.1 ECM-SK V1.0 腳位說明



1.1.1 GPIO / CONFIG (J1) 腳位說明

Pin 11	Pin 9	Pin 7	Pin 5	Pin 3	Pin 1
Input 5	Output 0	Input 3	Reserved	Reserved	3.3V Out
Pin 12	Pin 10	Pin 8	Pin 6	Pin 4	Pin 2
CONFIG0	Output 1	Input 1	Input 2	Reserved	Reserved

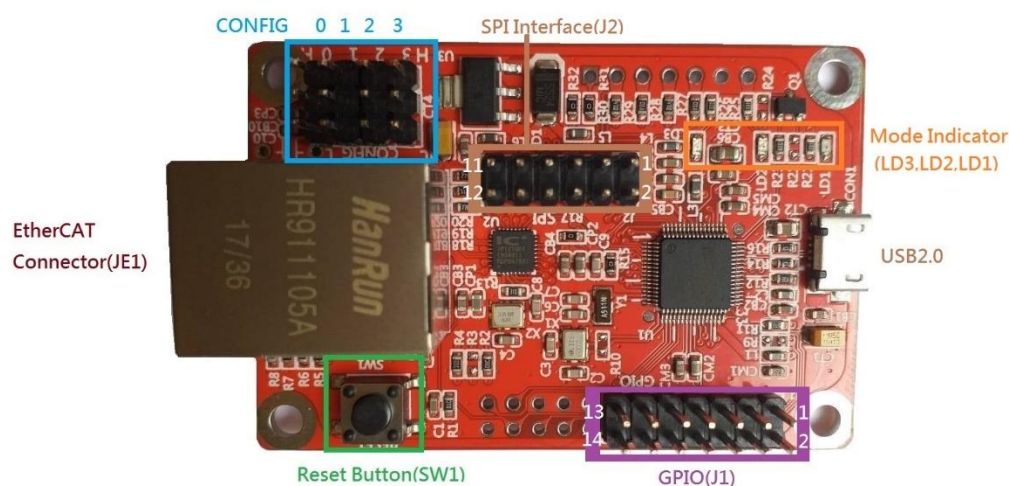
Pin 25	Pin 23	Pin 21	Pin 19	Pin 17	Pin 15	Pin 13
Reserved	Output 5	Output 4	CONFIG3	CONFIG2	Output 2	CONFIG1
Pin 26	Pin 24	Pin 22	Pin 20	Pin 18	Pin 16	Pin 14
GND	Reserved	Input 4	Input 0	Reserved	Busy (Output)	Output 3

* 注意：J1 Input / Output 腳位直接由 ECM IC 提供，請設計適當之隔離電路，以免 IC 損毀，3.3V 為 High，0V 為 Low。請參考 EC01M data sheet。

1.1.2 SPI(J3) 腳位說明

Pin 11	Pin 9	Pin 7	Pin 5	Pin 3	Pin 1
/RESET	SPI_MISO	SPI_/SS	Busy (Output)	Power In (3.3V)	Reserved
Pin 12	Pin 10	Pin 8	Pin 6	Pin 4	Pin 2
Reserved	SPI_MOSI	SPI_CLK	Reserved	Reserved	GND

1.2 ECM-SK V1.1 腳位說明



1.2.1 GPIO (J1) 腳位說明

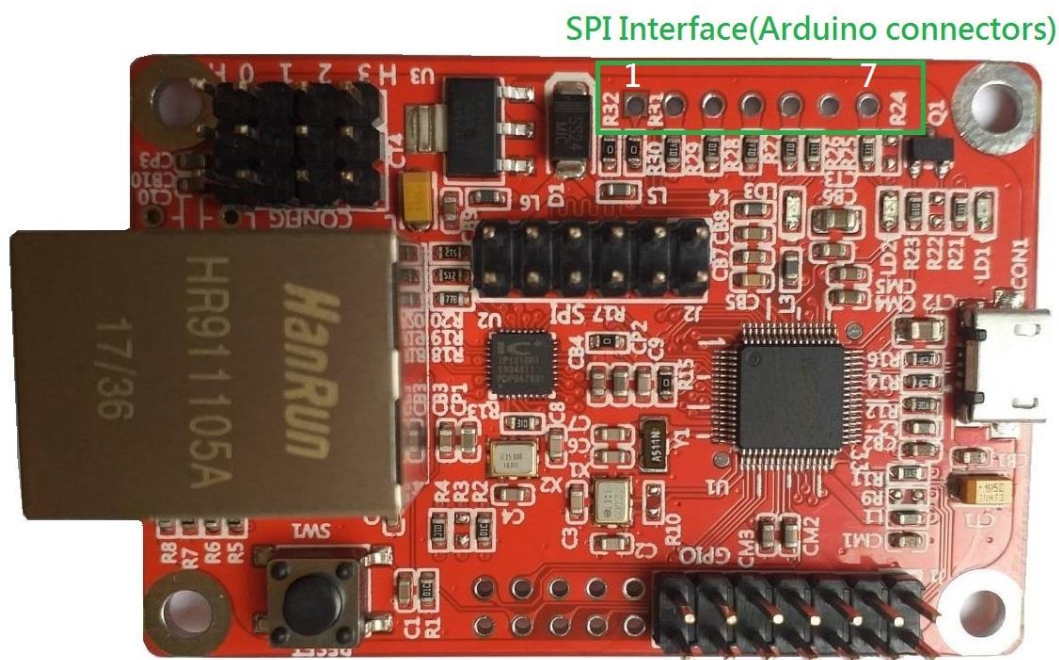
Pin 13	Pin 11	Pin 9	Pin 7	Pin 5	Pin 3	Pin 1
Input 5	Input 4	Input 3	Input 2	Input 1	Input 0	3.3V Out
Pin 14	Pin 12	Pin 10	Pin 8	Pin 6	Pin 4	Pin 2
Output 5	Output 4	Output 3	Output 2	Output 1	Output 0	GND

* 注意：J1 Input / Output 腳位直接由 ECM IC 提供，請設計適當之隔離電路，以免 IC 損毀，3.3V 為 High，0V 為 Low。請參考 EC01M data sheet。

1.2.2 SPI(J2) 腳位說明

Pin 11	Pin 9	Pin 7	Pin 5	Pin 3	Pin 1
/RESET	SPI_MISO	SPI_/SS	Busy (Output)	Power In (3.3V)	Reserved
Pin 12	Pin 10	Pin 8	Pin 6	Pin 4	Pin 2
Reserved	SPI_MOSI	SPI_CLK	Reserved	Reserved	GND

1.2.3 SPI(Arduino Connectors) 腳位說明



Pin 1	Pin 2	Pin 3	Pin 4	Pin 5	Pin 6	Pin 7
Busy (Output)	SPI_/SS	SPI_MOSI	SPI_MISO	SPI_CLK	GND	Power In (3.3V)

* 注意：SPI(Arduino Connectors)與 SPI(J2)作用相同，僅能擇一操作

* Pin7 Power In: 建議提供 3.3V 200mA 以上的電流

1.3 LED 狀態顯示及 Config.設定說明

1.3.1 LED 指示燈號說明

V1.0 組件編號	V1.1 組件編號	顏色	意義
LEDR1	LD1	紅色	電源指示燈
LEDY1	LD2	黃色	狀態指示燈 Y
LEDG1	LD3	綠色	狀態指示燈 G

1.3.2 LED 狀態說明

狀態指示燈 Y (黃色)	狀態指示燈 G (綠色)	狀態
亮 ON	亮 ON	未偵測到 EtherCAT 子站 或 子站未達預期階段
暗 OFF	暗 OFF	Init State 或 Pre-Operational State
亮 ON	暗 OFF	Safe-Operational State
暗 OFF	亮 ON	Operational State

1.3.3 網路口燈號說明

網路口燈號 Y (黃色): 恒亮代表網路速度 100M Hz，網路口正常工作。

網路口燈號 G (綠色): 閃爍代表資料傳輸。

1.3.4 CONFIG Pin 說明

CONFIG	狀態	說明
CONFIG0	L	Host interface is set as USB
	X	Host interface is set as SPI (Default)
	H	
CONFIG1	X	Normal mode (Default)
	L	
	H	Test mode
CONFIG2	X	Data size of each slave is 12 Bytes (Default)
	L	
	H	Data size of each slave is 16 Bytes
CONFIG3	L	FIFO abandon disable
	X	FIFO abandon enable (Default)
	H	

X: Floating, L: Low (0V), H: High (3.3V)

CONFIG0 與上位控制器連結的介面選擇

CONFIG1 模式選擇 Normal Mode – 依據使用者的命令進行控制

Test Mode – 上電後自動依預設值(Slave Type: IO, Cycle Time: 1ms)進入 OP，並週期性傳送不同的 output 命令給所有從站

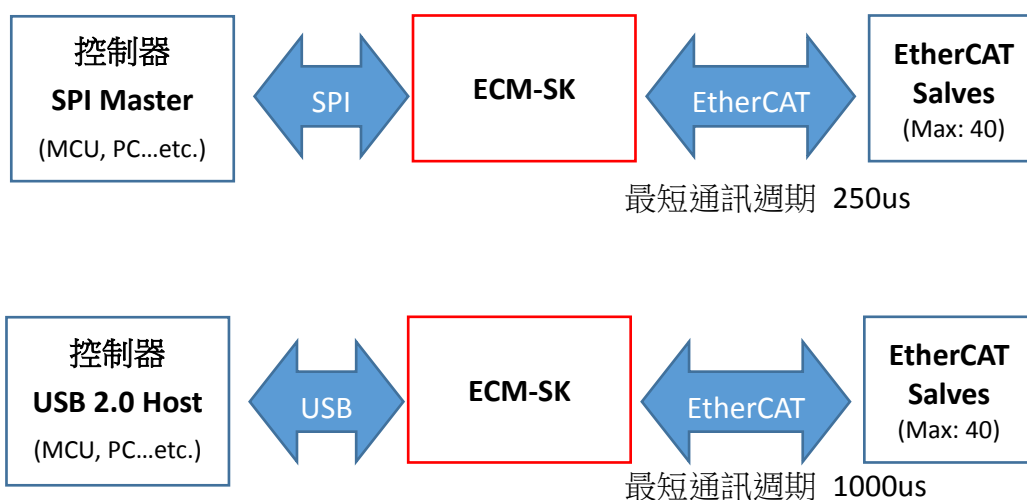
CONFIG2 單一站的資料數選擇

CONFIG3 單筆命令最長傳輸時間選擇

L: disable 單筆命令傳輸時間無限制

H: enable 單筆命令傳輸時間最大為 100 個週期時間，超過則放棄整筆命令

1.4 系統連接圖



* 由 J1 Pin12 CONFIG0 決定 SPI 模式 或 USB 模式

* USB 通訊速度取決於 USB 2.0 Host 效能

1.5 ECM-SK SPI 與控制器間的接線

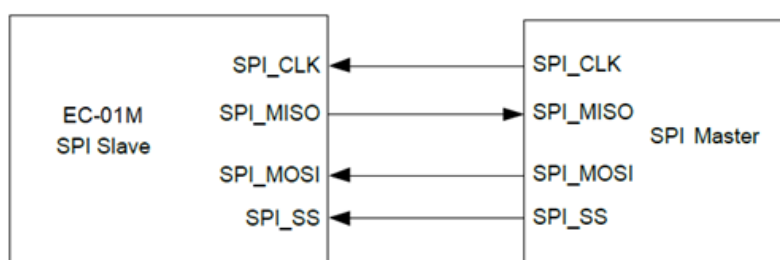


表 1.1 SPI 接腳名稱及意義

名稱	SPI 腳位	意義	說明
Busy	Pin5	ECM 忙碌	ECM 正處於忙碌狀態，不接受新的 SPI 傳輸

SCLK	Pin8	頻率訊號	由 SPI Master 產生並控制，依控制 EtherCAT 通訊週期有最低頻率要求
MOSI	Pin10	主出從入	SPI Master 資料輸出，SPI Slave 資料輸入
MISO	Pin9	主入從出	SPI Master 資料輸入，SPI Slave 資料輸出
/SS	Pin7	晶片致能	選擇信號，由 Master 控制，Slave 只有在 /SS 信號為低電位時，才會對 Master 的操作指令有反應

1.6 SPI 規格與傳輸模式

ECM 與上位控制器間採用 SPI 通訊格式，ECM 與上位控制器間的接線在低頻率時(10MHz 以下)可用杜邦線直接連接，在高頻率時建議直接以銀線焊接方式連接，以免雜訊干擾通訊品質，SCLK 由上位控制器提供，依不同通訊週期有最低頻率要求，SCLK 最高支援至 24MHz，支持 40 子站(每個子站含 12-Byte 資料)，或支援 30 個子站(每個子站含 16-Byte 資料)，通訊週期 SCLK 需求，如表 1.2 所示。

表 1.2 SPI SCLK 最低要求

通訊週期(Cycle Time)	SCLK 最低要求	接線方式
250us	24 MHz	銀線焊接或 PCB (不可用杜邦線)
1ms	6 MHz	可用杜邦線

SPI 模式傳輸說明

ECM 的 SPI 為 Slave 模式，空閒時為低電位，在下降緣發送，並於上升緣接收，高位資料先傳 (MSB)，請參考下圖說明。

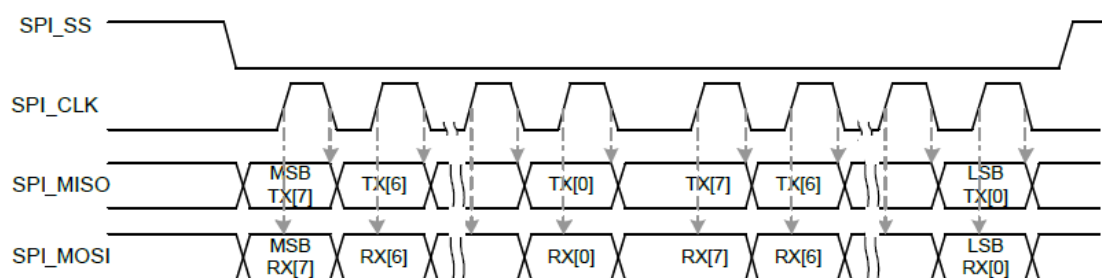


圖 1.1 SPI 時序圖

上位控制器端的 SPI 為 Master 模式，需產生 CLK 並提供給 SPI Slave，並在上升緣發送，並於下降緣接收。SPI 傳輸以 Byte 為單位，每次將由低地址開始傳，依序傳至最高位址，換言之 SPI 傳輸將從 Byte0 開始，再依序傳 Byte1、Byte2...直至最後一個 Byte 為止。而 SPI 傳輸單一 Byte 時，採取 MSB 模式，亦即高位先傳輸。

1.7 SPI 傳輸時序說明

上位控制器可透過 SPI 與 ECM 溝通，上位控制器偵測到 SPI BUSY(J1 Pin16 或 J3 Pin5)為低電位時，可開始進行 SPI 通訊，當進入 SPI 通訊後，SPI BUSY 訊號立刻升為高電位，直至通訊結束後且 ECM 處理完命令封包後 SPI Busy 訊號才會恢復為低電位，之後控制器即可透過 SPI 取得資料或傳送命令。詳細時序圖如下所示：

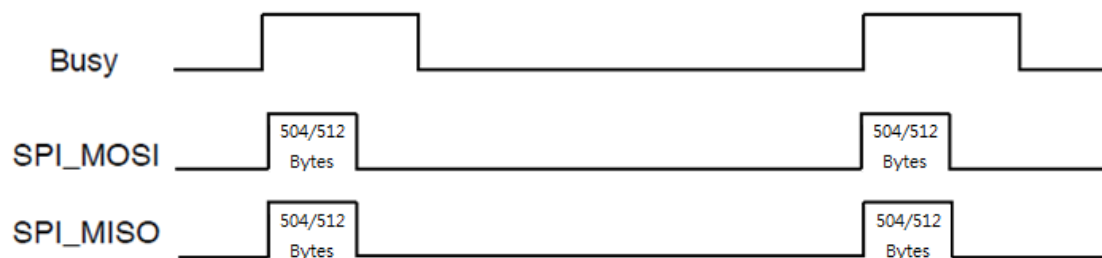


圖 1.2 信號時序示意圖

若欲實現最短週期 250us 的通訊要求，SPI 傳輸頻率必須為 24MHz，且上位控制端偵測到 Busy Pin 為低電位時，必須立即進行下一筆資料的交換。若通訊週期較慢的應用（如慢於 1ms），則上位控制端可於要進行資料傳輸時，再判斷 BUSY 的訊號，若 Busy Pin 為低電位時即可進行 SPI 資料交換。

1.8 EtherCAT 狀態流程

EtherCAT 可分為 4 個執行狀態，Init、PreOP、SafeOP、OP，各狀態能執行的指令不同，在 PreOP 狀態可以進行子站類型及驅動器類型的設置；SafeOP 狀態可進行非週期性資料交換，如讀寫暫存器資料(SDO)；OP 狀態則可進行週期性資料交換，即在固定週期內交換固定資料，OP 狀態亦可進行非週期性資料交換。使用者只需在 PreOP 狀態及 OP 狀態即可完成所有指令操作。

EtherCAT 狀態間的切換時間會因 EtherCAT 子站而有不同（與子站廠牌、性能、數量有關），使用者可從 GET_STATUS 命令的回傳值中 Current State 確認目前狀態，詳細流程與可執行的指令如下圖所示：

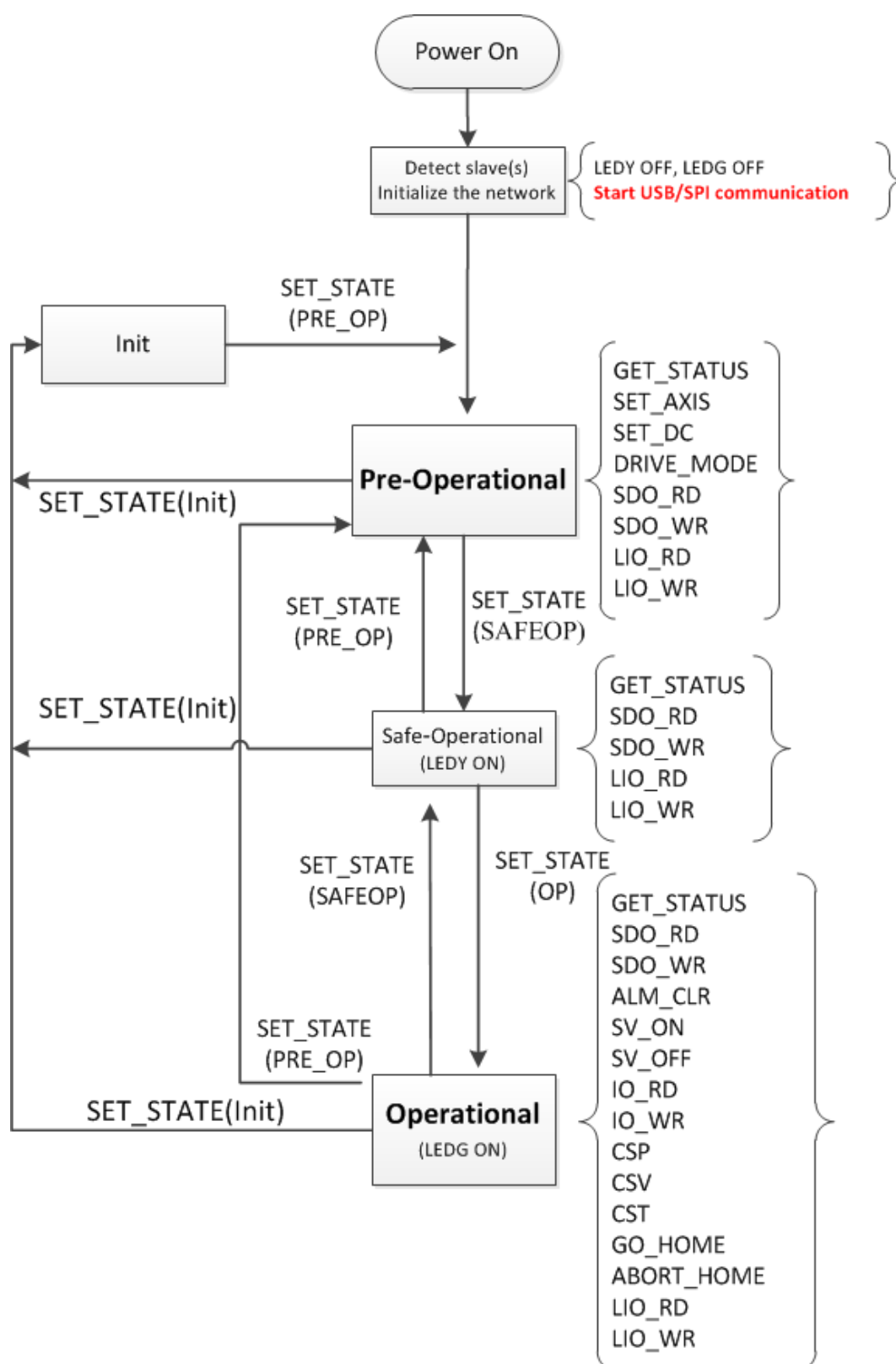


圖 1.3 EtherCAT 狀態流程

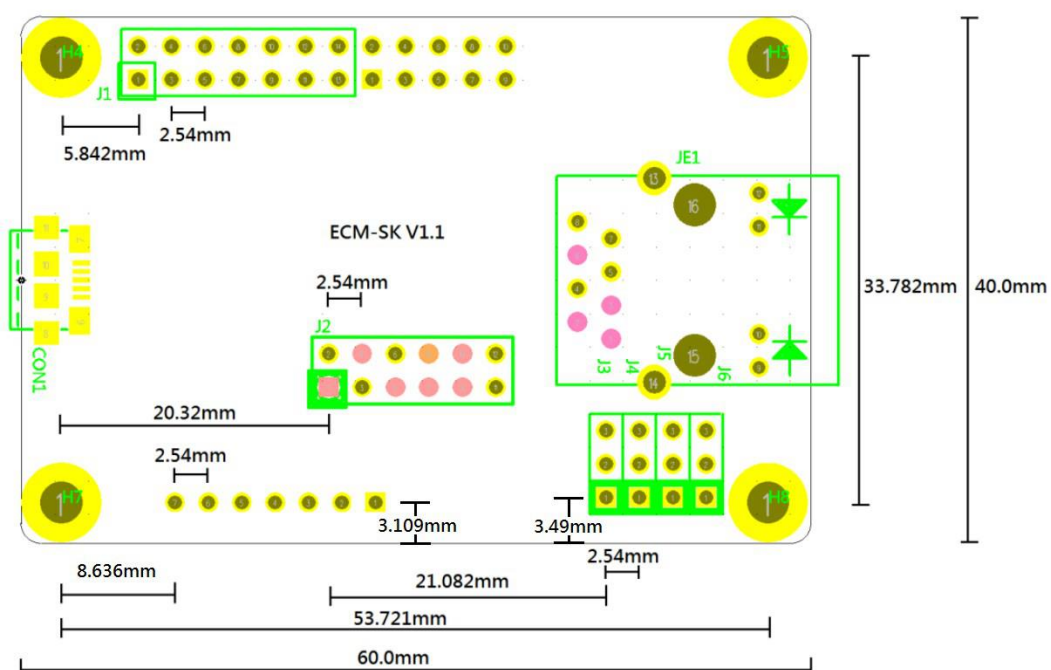
表 1.3 指令清單

Cmd. ID	Command	Operation	ECM	Drive	IO	Init	PreOP	SafeOP	OP	Ref.
0x00	GET_STATUS	Get Status	V	V	V	V	V	V	V	2.1.1
0x01	SET_STATE	Set EtherCAT state	V			V	V	V	V	2.1.2
0x02	SET_AXIS	Set type of slave	V			V	V			2.1.3
0x03	SET_DC	Set DC Mode	V			V	V			2.1.4
0x04	SET_EX	Set Extensions (CRC)	V			V	V	V		2.1.5
0x05	SET_FIFO	Set FIFO	V						V	2.1.6
0x06	DRIVE_MODE	Set mode of servo drive		V		V	V			2.1.7
0x07	SDO_RD	Service data object read		V	V		V	V	V	2.1.8
0x08	SDO_WR	Service data object write		V	V		V	V	V	2.1.9
0x10	ALM_CLR	Alarm clear		V					V	2.1.10
0x11	SV_ON	Servo ON		V					V	2.1.11
0x12	SV_OFF	Servo OFF		V					V	2.1.12
0x13	IO_RD	Digital Input			V				V	2.1.13
0x14	IO_WR	Digital Output			V				V	2.1.14
0x15	CSP	Position control		V					V	2.1.15
0x16	CSV	Velocity control		V					V	2.1.16
0x17	CST	Torque control		V					V	2.1.17
0x18	GO_HOME	Start Homing Procedure		V					V	2.1.18
0x19	ABORT_HOME	Abort Homing Procedure		V					V	2.1.19
0x21	LIO_RD	Read ECM IC input	V			V	V	V	V	2.1.20

NEXTW

ECM-SK 使用説明

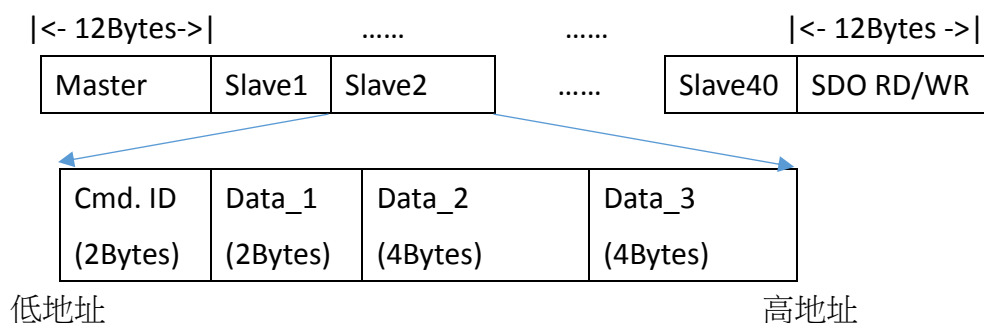
0x22	LIO_WR	Write ECM IC output	V			V	V	V	V	2.1.21
0xBB	SW_RESET	Software Reset	V			V	V	V	V	2.1.22



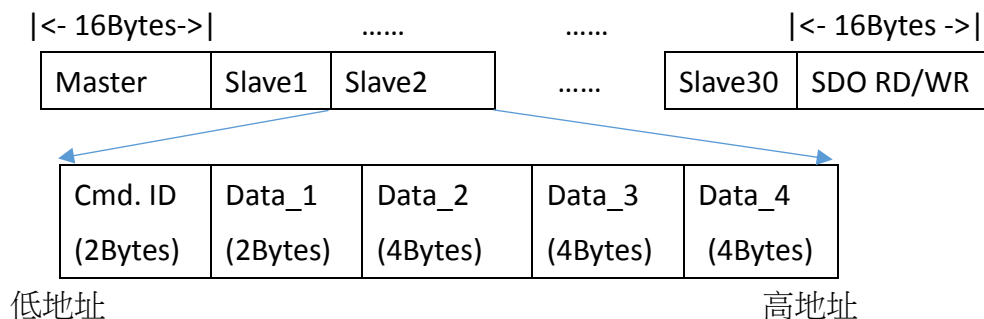
第 2 章 命令與回應

2.1 命令與回應資料結構

上位控制器可在不同的通信階段對各子站下達命令來控制或取得各子站的狀態，子站收到並處理後會響應，每一個命令依 CONFIG 2(J1 Pin17)之設定，為 12Bytes 或 16Bytes 的形式，第一筆命令對應 ECM，第二筆命令對應第一個子站，第三筆命令對應第二個子站，以此類推。資料長度若為 12Bytes 可控制 40 個子站，若為 16Bytes 則可控制 30 個子站，最後一筆命令則是專門對應單一子站進行非週期性的參數讀寫(SDO Read/Write)。在 CONFIG 2 為 Low 時主站最多可連接 40 個子站，因此每次下達命令均會傳遞 $(40+2) * 12 = 504$ Bytes 的資料，亦會得到 $(40+2) * 12 = 504$ Bytes 回應。



若 CONFIG 2 為 High 時主站最多可連接 30 個子站，因此每次下達命令均會傳遞 $(30+2) * 16 = 512$ Bytes 的資料，亦會得到 $(30+2) * 16 = 512$ Bytes 回應。



若資料存放方式為 Little Endian，低位元組放在低位址，例如 Cmd. ID 為 2Byte 資料，假設值為 0x0001 時，低位元組 0x01 會放在 Byte0，高位元組 0x00 會放在

Byte1，又如 Data_2 為 4Byte 資料，假設值為 0x87 65 43 21，最低位元組值 0x21 會放在最低位址 Byte0，之後依序為 0x43、0x65，最高位元組 0x87 則放在最高位址 Byte3。

SPI 傳輸以 Byte 為單位，每次將由低地址開始傳，依序傳至最高位址，換言之 SPI 傳輸將從 Byte0 開始，再依序傳 Byte1、Byte2...直至最後一個 Byte 為止。而 SPI 傳輸單一 Byte 時，採取 MSB 模式，亦即高位先傳輸。

此次命令的結果，將于下次命令時傳回，但部分指令執行可能超過一個運算速度(如 SET_STATE、SDO_RD、SDO_WR、GO_HOME 等)，完成後的結果可在數個週期後利用 GET_STATUS 命令取回。

2.1.1 GET_STATUS Command

- 指令說明：Get Status，不做任何命令，僅取回狀態資料
- Command ID : 0x0000
- 適用階段：所有階段
- 適用類別：所有類別

表 2.1 GET_STATUS Command

BYTE	欄位定義	Command	Response
0	Command ID	0x0000	0x00 或 0xBF
1			
2	Reserved	--	Pram
3	CRC8 Value**	CRC8 Value	
4	Reserved	--	Data1
5			
6			
7			
8	Reserved	--	Data2
9			
10			
11			
12	Reserved*	--	Data3
13			
14			
15			

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

** CRC8 Value 演算法請參考 [2.2 回應資料](#) 關於 CRC 的部分，是否開啟 CRC 檢查可透過 [2.1.5 SET_EX](#) 來設定。

提示：Response 的 Byte 0 於 EtherCAT OP 狀態下回應為 0xBF，其餘狀態回應為 0x00，Byte1~Byte11 各欄位將保持上次命令結果(State 非為 OP)或更新資料(State 為 OP)，回應資料可參考 [2.2 回應資料](#)。

2.1.2 SET_STATE Command

- 指令說明：設定 EtherCAT 狀態
- Command ID : 0x0001
- 適用階段：所有階段
- 適用類別：ECM(第一筆命令)

表 2.2 SET_STATE Command

BYTE	欄位定義	Command	Response
0	Command ID	0x0001	0x01
1			Error Code
2	Reserved	--	Current State
3	CRC8 Value**	CRC8 Value	CRC8 Value
4	EtherCAT State -INIT: 0x01 -PRE_OP: 0x02 -SAFE_OP: 0x04 -OP: 0x08	Requested STATE	--
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

** CRC8 Value 演算法請參考 [2.2 回應資料](#) 關於 CRC 的部分，是否開啟 CRC 檢查可透過 [2.1.5 SET_EX](#) 來設定。

Error Code CRC 錯誤次數，因 CRC 錯誤而被拒絕的命令數

Current State

-INIT: 0x01
-PRE_OP: 0x02
-SAFE_OP: 0x04
-OP: 0x08

提示：變更 EtherCAT State 需耗時數個運算週期，且與總子站數量有關，可對 ECM 下達 GET_STATUS 指令，並取得 Current State 資訊。

2.1.3 SET_AXIS Command

- 指令說明：Set type of slave，設定 ECM 連接之子站型態(IO、Drive、HSP 或 STEP)，最多 40 軸
- Command ID : 0x0002
- 適用階段：Pre-Operational State
- 適用類別：ECM(第一筆命令)

表 2.3 SET_AXIS Command

BYTE	欄位定義	Command	Response
0	Command ID	0x0002	0x02
1			Error Code
2	Group(0~4)	Group (Default: 0)	Current State
3	CRC8 Value**	CRC8 Value	CRC8 Value
4	Topology -Drive : 0x0 -IO : 0x1 -HSP : 0x2 -STEP : 0x3	Topology (Default:0x1)	SlaveCount -偵測到的子站數
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			

14			
15			

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

** CRC8 Value 演算法請參考 [2.2 回應資料](#) 關於 CRC 的部分，是否開啟 CRC 檢查可透過 [2.1.5 SET_EX](#) 來設定。

Group

一顆 ECM 最多可控制 40 個子站(Drive、IO、NEXTW HSP 或 STEP，不支援其他子站種類)，一次可設定 8 子站型態，Group 0 代表設定子站 1~8，Group 1 代表設定子站 9~16，依此類推，Group 的值可為 0~4。

SlaveCount

偵測到的子站數，若於 Init State 一律回傳 0，於 Pre-Operational State 會回傳實際偵測到的子站數量。

Topology

	Bit28	Bit24	Bit20	Bit16	Bit12	Bit8	Bit4	Bit0
Slave	Slave	Slave	Slave	Slave	Slave	Slave	Slave	
n+8	n+7	n+6	n+5	n+4	n+3	n+2	n+1	

$n = \text{Group} * 8$

0: Drive

1: IO

2: NEXTW HSP (High Speed Pulse)

3: Step

Error Code CRC 錯誤次數，因 CRC 錯誤而被拒絕的命令數

Current State

-INIT: 0x01

-PRE_OP: 0x02

-SAFE_OP: 0x04

-OP: 0x08

2.1.4 SET_DC Command

- 指令說明：設定 ECM 週期時間(Cycle Time)
- Command ID : 0x0003
- 適用階段：Pre-Operational State
- 適用類別：ECM(第一筆命令)

表 2.4 SET_DC Command

BYTE	欄位定義	Command	Response
0	Command ID	0x0003	0x03
1			Error Code
2	Reserved	--	Current State
3	CRC8 Value**	CRC8 Value	CRC8 Value
4	Cycle Time (us) 須為 250×2^n (Minimum : 250)	Cycle Time (Default:1000)	--
5			
6			
7			
8	FIX Value	0xFFFF (Default:0xFFFF)	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

** CRC8 Value 演算法請參考 [2.2 回應資料](#) 關於 CRC 的部分，是否開啟 CRC 檢查可透過 [2.1.5 SET_EX](#) 來設定。

FIX Value

固定值，一律為 0x0000FFFF

Error Code CRC 錯誤次數，因 CRC 錯誤而被拒絕的命令數

Current State

-INIT: 0x01

-PRE_OP: 0x02

-SAFE_OP: 0x04

-OP: 0x08

2.1.5 SET_EX Command

- 指令說明：設定 CRC 功能 (設定完成後於下次開機後生效)
- Command ID : 0x0004
- 適用階段：Init State、Pre-Operational State、Safe-Operational State
- 適用類別：ECM(第一筆命令)

表 2.5 SET_EX Command

BYTE	欄位定義	Command	Response
0	Command ID	0x0004	0x04
1			Error Code
2	Reserved	--	Current State
3			CRC8 Value
4	Extension Setting 1~2	Extension Setting Value	--
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

* 此指令將於下次開機後生效

Extension Setting

Bit28	Bit24	Bit20	Bit16	Bit12	Bit8	Bit4	Bit0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Setting2	Setting1

Setting 1: Command CRC Verification

0: Disable Command CRC Verification

1: Enable Command CRC Verification

Setting 2: Response CRC Verification

0: Disable Response CRC Verification

1: Enable Response CRC Verification

2.1.6 SET_FIFO Command

- 指令說明：設定 FIFO
- Command ID : 0x0005
- 適用階段：Operational State
- 適用類別：ECM(第一筆命令)

表 2.6 SET_FIFO Command

BYTE	欄位定義	Command	Response
0	Command ID	0x0005	0x05
1			Error Code
2	Action	Action(1~2)	Current State
3	CRC8 Value	CRC8 Value	CRC8 Value
4	Value	Value (>=0)	--
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

Action

0x0001 : Clear FIFO content and FIFO minimum setting.

清除 FIFO 內所有資料，並清除 FIFO threshold 設定

0x0002 : Set FIFO threshold to **Value**.

設定 FIFO 閾值為 Value。OP 狀態時，當 FIFO 剩餘空間小於 Value 後，才開始在每週期取一筆資料執行，滿足條件後自動將最小數值重設為 160

2.1.7 DRIVE_MODE Command

- 指令說明：Set mode of servo drive，設定 ECM 連接之伺服驅動器命令模式及同步模式
- Command ID : 0x0006
- 適用階段：Pre-Operational State
- 適用類別：Drive 子站、NEXTW HSP 子站、STEP 子站

表 2.7 DRIVE_MODE Command

BYTE	欄位定義	Command	Response
0	Command ID	0x06	0x06
1		Command Index	Command Index
2	Reserved	--	--
3			
4	OP_Mode -CSP: 0x08 -CSV: 0x09 -CST: 0x0A	OP_Mode (Default:0x08)	--
5			
6			
7			
8	DRIVE_TYPE - FREERUN: 0x00 - DCSYNC: 0x01	DRIVE_TYPE (Default:0x01)	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

- Command Index：使用者自訂之數字，命令與回應的 Command Index 會相同。
- OP_MODE：
CSP(Cyclic Synchronous Position Mode)：

ECM在CSP模式下定期發送PDO，在傳送每一筆PDO時，均會將目標絕對位置命令同時傳送至子站。

CSV(Cycle Synchronized Velocity Mode)：

ECM在CSV模式下定期發送PDO，在傳送每一筆PDO時，會將目標速度命令同時傳送至子站。

CST (Cyclic Synchronous Torque Mode)

ECM在CSV模式下定期發送PDO，在傳送每一筆PDO時，會將目標扭力命令同時傳送至子站。

➤ DRIVE_TYPE:

FREE RUN

各子站間非同步，各子站間根據自己的內部時間處理EtherCAT資料，與主站的週期、其他子站的週期及EtherCAT資料到達時間均無關。

DCSYNC

DCSYNC要求主站有很強的即時性能，是高精度的時間同步模式（所有子站與第一個有DC的子站同步），以第一個有DC的子站時間為基準時間，再用此基準時間作為所有子站的參考時間，加上傳輸延時、抖動等時間誤差產生同步信號。

2.1.8 SDO_RD Command

- 指令說明：SDO (Service Data Objects) Read，設定 ECM 對指定的子站下達欲讀取的 Object Index 及 Object Sub Index。
- Command ID : 0x0007
- 適用階段：Pre-Operational State、Safe-Operational State、Operational State
- 適用類別：SDO RD/WR (最後一筆命令)

表 2.8 SDO_RD Command

BYTE	欄位定義	Command	Response
0	Command ID	0x0007	0x0007
1			
2	Slave Index	Slave Index(1~40)	Slave Index
3			
4	Object Index	Object Sub Index	Object Sub Index
5			

6		Object Index	Object Index
7			
8	Value	--	Value
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

Index

Bit16	Bit0
Object Index	Object Sub Index

提示：SDO_RD 需耗時，可在數個週期後對 SDO 下達 GET_STATUS 指令，取回結果。

2.1.9 SDO_WR Command

- 指令說明：SDO (Service Data Objects) Write，設定 ECM 對指定的子站 Object Index 及 Object Sub Index 寫入指定數值。
- Command ID : 0x0008
- 適用階段：Pre-Operational State、Safe-Operational State、Operational State
- 適用類別：SDO RD/WR (最後一筆命令)

表 2.9 SDO_WR Command

BYTE	欄位定義	Command	Response
0	Command ID	0x0008	0x0008
1			
2	Slave Index	Slave Index (1~40)	Slave Index
3	Size	Object Size	
4	Object Index	Object Sub Index	Object Sub Index
5			
6		Object Index	Object Index
7			

8	Value	Object Value	Object Value
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

Index

Bit16		Bit0	
Object Index		Object Sub Index	

提示：SDO_WR 需耗時，可在數個週期後對 SDO 下達 GET_STATUS 指令取回狀態，以確認寫入。

2.1.10 ALM_CLR Command

- 指令說明：Clear alarm of slave，清除子站上的警告訊息
- Command ID : 0x10
- 適用階段：Operational State
- 適用類別：Drive 子站、NEXTW HSP 子站、STEP 子站

表 2.10 ALM_CLR Command

BYTE	欄位定義	Command	Response
0	Command ID	0x10	0x10
1		Command Index	Command Index
2	Reserved	--	Statusword
3			
4	Reserved	--	--
5			
6			
7			
8	Reserved	--	--
9			

10			
11			
12	Reserved*	--	--
13			
14			
15			

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

Command Index 使用者自訂的 Index 值，可藉此值判斷該次回應是針對哪個命令

提示：並非所有的 Alarm 均可以被清除，可以清除的警告請參閱子站說明手冊。

2.1.11 SV_ON Command

- 指令說明：Set Servo ON，設定驅動器子站 Servo ON
- Command ID : 0x11
- 適用階段：Operational State
- 適用類別：Drive 子站、NEXTW HSP 子站、STEP 子站

表 2.11 SV_ON Command

BYTE	欄位定義	Command	Response
0	Command ID	0x11	0x11
1		Command Index	Command Index
2	Reserved	--	Statusword
3			
4	Reserved	--	--
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

Command Index 使用者自訂的 Index 值，可藉此值判斷該次回應是針對哪個命令

提示：SV_ON 指令會在一個通訊週期送達子站，但每個子站進行 Servo on 的程式及所需時間可能不同，使用者可透 GET_STATUS 指定取得最新的 Statusword，Statusword 可判斷子站 Servo on/off 狀態。

2.1.12 SV_OFF Command

- 指令說明：Set Servo OFF，設定驅動器子站 Servo OFF
- Command ID : 0x12
- 適用階段：Operational State
- 適用類別：Drive 子站、NEXTW HSP 子站、STEP 子站

表 2.12 SV_OFF Command

BYTE	欄位定義	Command	Response
0	Command ID	0x12	0x12
1		Command Index	Command Index
2	Reserved	--	Statusword
3			
4	Reserved	--	--
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

Command Index 使用者自訂的 Index 值，可藉此值判斷該次回應是針對哪個命令

提示：SV_OFF 指令會在一個通訊週期送達子站，但每個子站進行 Servo off 所需時間可能不同，使用者可透 GET_STATUS 指定取得最新的 Statusword，Statusword 可判斷子站 Servo on/off 狀態。

2.1.13 IO_RD Command

- 指令說明：Read digital input value，讀取子站數位輸入狀態數值
- Command ID : 0x13
- 適用階段：Operational State
- 適用類別：IO 子站

表 2.13 IO_RD Command

BYTE	欄位定義	Command	Response
0	Command ID	0x0013	0x13
1			--
2	Reserved	--	--
3			
4	Input - 子站上的輸入 狀態值	--	Input
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

2.1.14 IO_WR Command

- 指令說明：Write digital output value，設定子站輸出狀態
- Command ID : 0x14
- 適用階段：Operational State
- 適用類別：IO 子站

表 2.14 IO_WR Command

BYTE	欄位定義	Command	Response
------	------	---------	----------

0	Command ID	0x0014	0x14
1		Command Index	Command Index
2	Reserved	--	--
3			
4	Output - 子站上的輸出 狀態值	Output	--
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

Command Index 使用者自訂的 Index 值，可藉此值判斷該次回應是針對哪個命令

2.1.15 CSP Command

- 指令說明：Cyclic Synchronous Position Command，設定子站之週期性同步位置命令
- Command ID : 0x15
- 適用階段：Operational State
- 適用類別：OP_Mode 為 CSP 的 Drive 或 HSP 或 STEP 子站

表 2.15 CSP Command for Drive and 1 channel HSP

BYTE	欄位定義	Command	Response	Master Cyclic Response
0	Command ID	0x15	0x15	0xBF
1		Cmd. Index	Cmd. Index	Error Code
2	Statusword – Drive 子站狀態	--	Statusword	Current State
3				CRC8 Value
4	Target Position – 目標位置	Target Position (絕對位置)	Current Position	--
5				

6	Current Position			
7	– 目前位置			
8	ALM Code -	Output Value	Current Torque	FIFO Remaining
9	Alarm code 警告		**	FIFO 尚餘空間
10	代碼		ALM Code**	FIFO Full Count
11	Current Torque –			FIFO 空間滿而被拒絕的命令數
	目前扭力值			
12	Reserved*	--	--	--
13				
14				
15				

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

** HSP(High Speed Pulse)子站回傳 Input 狀態，且無 ALM Code，Drive 子站回傳 Current Torque 及 ALM Code。

Command Index 使用者自訂的 Index 值，可藉此值判斷該次回應是針對哪個命令

Error Code CRC 錯誤次數，因 CRC 錯誤而被拒絕的命令數

Current State 應為 OP: 0x08

Output Value 設定 Drive 或 HSP 上的 Output (部分 Drive 可能不支援)

提示：CSP 指令為設定子站目標絕對位置，建議使用者自行規劃加減速，並且至少在一個通訊週期內送出一個 CSP 命令。使用者可以在一個通訊週期內送出兩個或以上的 CSP 命令，尚未送至子站的命令會暫存於 FIFO 中，FIFO 的空間有限，請注意，若 FIFO 已無空間存放則會直接忽略該命令，使用者應檢查 Master 回應中的 FIFO Remaining，若所剩餘空間過小，建議暫停數個通訊週期後，再恢復命令的傳送。

表 2.16 CSP Command for 2-channel HSP

BYTE	欄位定義	Command	Response	Master Cyclic Response
0	Command ID	0x15	0x15	0xBF
1		Cmd. Index	Cmd. Index	Error Code
2	Statusword –	--	Statusword	Current State
3	Drive 子站狀態			CRC8 Value
4	Target Position -	Target Position 1 (絕對位置)	Current Position 1	--
5	Pulse 輸出的目			
6	標絕對位置			

7	Current Position - Encoder 回傳 的目前位置			
8		Output Value 1	Input Status 1	FIFO Remaining FIFO 尚餘空間
9		Target Position 2(絕對位置)	Current Position 2	FIFO Full Count FIFO 空間滿而被拒 絕的命令數
10				
11	Output Value – 輸出值			
12				
13	Input Status – 輸入狀態			
14		Output Value 2	Input Status 2	
15				

提示：若要輸出兩組 Pulse，CONFIG 2 必須為 High。

2.1.16 CSV Command

- 指令說明：Cyclic Synchronous Velocity Command，設定子站之週期性同步速度命令
- Command ID : 0x16
- 適用階段：Operational State
- 適用類別：OP_Mode 為 CSV 的 Drive 或 HSP 或 STEP 子站

表 2.17 CSV Command for Drive and 1 channel HSP

BYTE	欄位定義	Command	Response	Master Cyclic Response
0	Command ID	0x16	0x16	0xBF
1		Cmd. Index	Cmd. Index	Error Code
2	Statusword – Drive 子站狀態	--	Statusword	Current State
3				CRC8 Value
4	Velocity–目標速度 Current Position – 目前位置	Velocity	Current Position	--
5				
6				
7				
8	ALM Code - Alarm code 警告 代碼 Current Torque – 目前扭力值	--	Current Torque **	FIFO Remaining FIFO 尚餘空間
9			ALM Code**	FIFO Full Count FIFO 空間滿而被 拒絕的命令數
10				
11				

12	Reserved*	--	--	--
13				
14				
15				

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

** HSP(High Speed Pulse)子站回傳 Input 狀態，且無 ALM Code，Drive 子站回傳 Current Torque 及 ALM Code。

Command Index 使用者自訂的 Index 值，可藉此值判斷該次回應是針對哪個命令

Error Code CRC 錯誤次數，因 CRC 錯誤而被拒絕的命令數

Current State 應為 OP: 0x08

提示：CSV 指令為設定子站目標轉速，建議使用者自行規劃加減速度，並且至少在一個通訊週期內送出一個 CSV 命令。使用者可以在一個通訊週期內送出兩個或以上的 CSV 命令，尚未送至子站的命令會暫存於 FIFO 中，FIFO 的空間有限，請注意，若 FIFO 已無空間存放則會直接忽略該命令，使用者應檢查 Master 回應中的 FIFO Remaining，若所剩餘空間過小，建議暫停數個通訊週期後，再恢復命令的傳送。

表 2.18 CSV Command for 2 channel HSP

BYTE	欄位定義	Command	Response	Master Cyclic Response
0	Command ID	0x16	0x16	0xBF
1		Cmd. Index	Cmd. Index	Error Code
2	Statusword –	--	Statusword	Current State
3	Drive 子站狀態			CRC8 Value
4	Target Velocity -Pulse 輸出的目標速度	Target Velocity 1	Current Position 1	--
5				
6				
7				
8	Current Position - Encoder 回傳的目前位置	Output Value 1	Input Status 1	FIFO Remaining FIFO 尚餘空間
9				
10	Output Value – 輸出值	Target Velocity 2	Current Position 2	FIFO Full Count FIFO 空間滿而被拒絕的命令數
11				
12				
13				
14	Input Status –	Output Value	Input Status 2	

15	輸入狀態	2		
----	------	---	--	--

提示：若要輸出兩組 Pulse，CONFIG 2 必須為 High。

2.1.17 CST Command

- 指令說明：Cyclic Synchronous Torque Command，設定子站之週期性同步扭力命令
- Command ID : 0x17
- 適用階段：Operational State
- 適用類別：OP_Mode 為 CST 的 Drive 子站

表 2.19 CST Command

BYTE	欄位定義	Command	Response	Master Cyclic Response
0	Command ID	0x17	0x17	0xBF
1		Cmd. Index	Cmd. Index	Error Code
2	Statusword – Drive 子站狀態	--	Statusword	Current State
3				CRC8 Value
4	Torque – 目標 扭力 Current Position – 目前位置	Torque	Current Position	
5				
6				
7				
8	ALM Code - Alarm code 警告代碼 Current Torque – 目前扭力值	--	Current Torque	FIFO Remaining FIFO 尚餘空間
9			ALM Code	FIFO Full Count FIFO 空間滿而被拒絕的命令數
10				
11				
12	Reserved*	--	--	--
13				
14				
15				

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

Command Index 使用者自訂的 Index 值，可藉此值判斷該次回應是針對哪個命令

Error Code CRC 錯誤次數，因 CRC 錯誤而被拒絕的命令數

Current State 應為 OP: 0x08

提示：CST 指令為設定子站的目標扭力，建議使用者自行規劃連續的扭力輸出，並且至少在一個通訊週期內送出一個 CST 命令。使用者可以在一個通訊週期內送出兩個或以上的 CST 命令，尚未送至子站的命令會暫存於 FIFO 中，FIFO 的空間有限，請注意，若 FIFO 已無空間存放則會直接忽略該命令，使用者應檢查 Master 回應中的 FIFO Remaining，若所剩餘空間過小，建議暫停數個通訊週期後，再恢復命令的傳送。

2.1.18 GO_HOME Command

- 指令說明：設定子站開始回 Home 程式
- Command ID : 0x18
- 適用階段：Operational State
- 適用類別：Drive 子站、NEXTW HSP 子站、STEP 子站

表 2.20 GO_HOME Command

BYTE	欄位定義	Command	Response
0	Command ID	0x18	0x18
1		Command Index	Command Index
2	Statusword –	--	Statusword
3	Drive 子站狀態		
4	Current Position –	--	Current Position
5	目前位置		
6			
7			
8	ALM Code - Alarm	--	Current Torque
9	code 警告代碼		
10	Current Torque –		ALM Code
11	目前扭力值		
12	Reserved*	--	--
13			
14			
15			

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

Command Index 使用者自訂的 Index 值，可藉此值判斷該次回應是針對哪個命令

提示 1：執行 GO_HOME 前須透過 SDO_WR 來指定操作模式，請參考子站使用說

明，並確認 Homing Method 種類

Index	SubIndex	Name	Size	Value	Description
6060h	0	Modes of Operation	1 Byte	6	Homing Mode
6098h	0	Homing Method	1 Byte	0~35	Set Homing Method

另亦可透過 SDO_WR 來設定 Homing 相關參數，包括但不限於：

Index	SubIndex	Name	Size
607Ch	0	Home Offset	4 Bytes
6099h	1	Speed during search for switch	4 Bytes
6099h	2	Speed during search for zero	4 Bytes
609Ah	0	Homing Acceleration	4 Bytes

提示 2：GO_HOME 完成後會自動將操作模式切回原設定模式(CSP / CSV / CST)

提示 3：HSP 第二軸相關設定位置為第一軸 Index + 0x800(詳細內容請參考 NEXTW HSP 使用手冊)

2.1.19 ABORT_HOME Command

- 指令說明：設定子站強制終止 Home 程式
- Command ID : 0x19
- 適用階段：Operational State
- 適用類別：Drive 子站、NEXTW HSP 子站、STEP 子站

表 2.21 ABORT_HOME Command

BYTE	欄位定義	Command	Response
0	Command ID	0x19	0x19
1		Command Index	Command Index
2	Statusword –	--	Statusword
3	Drive 子站狀態		
4	Current Position – 目前位置	--	Current Position
5			
6			
7			
8	ALM Code - Alarm	--	Current Torque
9	code 警告代碼		
10	Current Torque –		ALM Code
11	目前扭力值		

12	Reserved*	--	--
13			
14			
15			

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

Command Index 使用者自訂的 Index 值，可藉此值判斷該次回應是針對哪個命令

提示：子站回傳的 Statusword bit10 及 bit12 可判斷回 Home 的狀態，當 bit10 及 bit12 均為 1 時，代表回 Home 程式已經完成。透過 ABORT_HOME 指令可強制所有子站終止 Home 程式。

2.1.20 LIO_RD Command

- 指令說明：讀取 ECM IC 上的 Input 狀態
- Command ID : 0x0021
- 適用階段：所有階段
- 適用類別：ECM(第一筆命令)

表 2.22 LIO_RD Command

BYTE	欄位定義	Command	Response
0	Command ID	0x0021	0x21
1			Error Code
2	Reserved	--	Current State
3	CRC8 Value**	CRC8 Value	CRC8 Value
4	Input Status – Input 狀態	--	Input Status
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

注意：LIO 腳位直接由 ECM IC 提供，請設計適當之隔離電路，以免 IC 損毀，3.3V 為 High，0V 為 Low。

** CRC8 Value 演算法請參考 [2.2 回應資料](#) 關於 CRC 的部分，是否開啟 CRC 檢查可透過 [2.1.5 SET EX](#) 來設定。

2.1.21 LIO_WR Command

- 指令說明：設定 ECM IC 上的 Output 狀態
- Command ID : 0x0022
- 適用階段：所有階段
- 適用類別：ECM(第一筆命令)

表 2.23 LIO_WR Command

BYTE	欄位定義	Command	Response
0	Command ID	0x0022	0x22
1			Error Code
2	Reserved	--	Current State
3	CRC8 Value**	CRC8 Value	CRC8 Value
4	Output Value-- Output 值	Output Value	--
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* Byte12-15 僅在 CONFIG 2 為 High 時才會生效。

注意：LIO 腳位直接由 ECM IC 提供，請設計適當之隔離電路，以免 IC 損毀，3.3V 為 High，0V 為 Low。

** CRC8 Value 演算法請參考 [2.2 回應資料](#) 關於 CRC 的部分，是否開啟 CRC 檢查可透過 [2.1.5 SET EX](#) 來設定。

2.1.22 SW_RESET Command

- 指令說明：Reset，回到初始狀態
- Command ID：0x00BB
- 適用階段：所有階段
- 適用類別：ECM(第一筆命令)

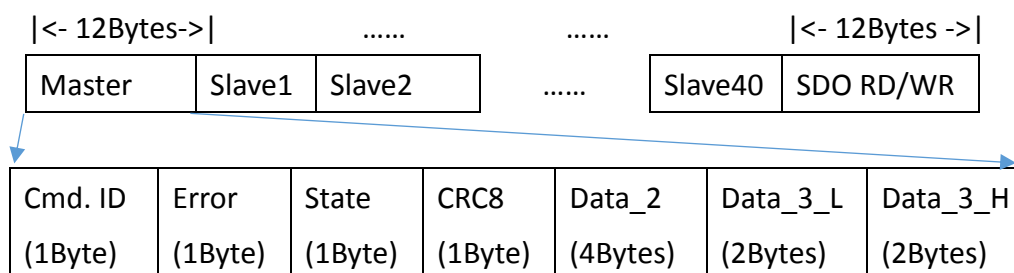
表 2.24 SW_RESET Command

BYTE	欄位定義	Command	Response
0	Command ID	0x00BB	--
1			
2	Reserved	--	--
3			
4	Reserved	--	--
5			
6			
7			
8	Reserved	--	--
9			
10			
11			
12	Reserved*	--	--
13			
14			
15			

* Reset 後將重置所有的變數，已經開啟的 USB 連線會中斷並關閉。要繼續傳輸必須再重新開啟 USB 連線。

2.2 回應資料

每次下達命令後均會得到回應資料，由於資料傳遞週期的關係，當次收到回應資料為上次命令後的執行結果回應，部分命令(如 SET_STATE、SDO_RD、SDO_WR 等)會經過數個運算速度後回應，回應資料的前 12 Bytes 為主站資訊，主站資訊的前 4 Bytes 固定為 Cmd. ID、Error Code、Current State 及 CRC8 Value，當 CONFIG 2 為 Low 時，每子站回應資料為 12Byte，其代表意義分述如下：



Cmd. ID

若對 Master 下達的指令會回傳該指令代碼(如 SET_STATE、SET_AXIS、SET_DC 等)，若 State 為 OP 狀態則會回傳 0xBF

Error CRC 錯誤次數，因 CRC 錯誤而被拒絕的命令數

State 目前主站的執行時間

-INIT: 0x01

-PRE_OP: 0x02

-SAFE_OP: 0x04

-OP: 0x08

CRC8 迴圈冗餘校驗值(Cyclic Redundancy Check)

該次回傳資訊的 CRC8 校驗值，驗算方式如下：

STEP 1：先將 CRC8 值記錄起來，並將該欄位填 0

STEP 2：將回傳值共 504 個 Byte(或 512 個 Byte)依序帶入以下 CRC8 多項式

CRC-8： $X^8 + X^2 + X + 1$ ，初值設為 0x5A

STEP 3：算出 CRC8 值，此值應與 STEP 1 記錄的值相符

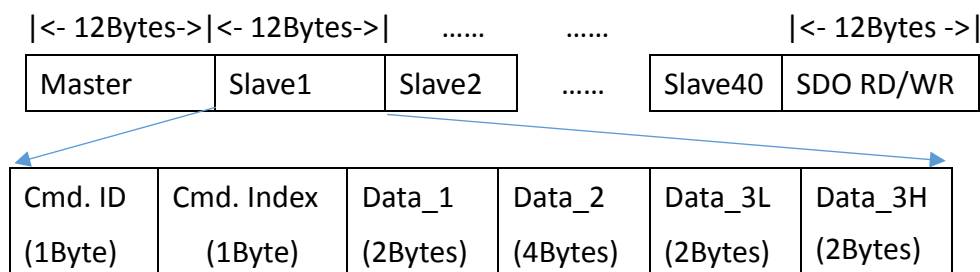
此值是否傳送可透過 [2.1.5 SET_EX](#) 命令來設定

Data_2 顯示回應資料(依命令不同代表不同意義)

Data_3_L 在 OP 狀態下顯示剩餘 FIFO 命令緩衝空間

Data_3_H 在 OP 狀態下顯示因已無命令緩衝空間而被拒絕沒有執行的命令數量

回應資料的第 13 Byte 開始至第 492Byte 為 40 個子站資訊，每個子站資訊為 12 個 Byte，驅動器子站資訊的前 2 Bytes 固定為 Cmd. ID 及 Cmd. Index，其代表意義分述如下：



Cmd. ID 對該子站下達的指令會回傳該指令代碼

Cmd. Index 指此資料為針對哪個 Command Index 命令的回應

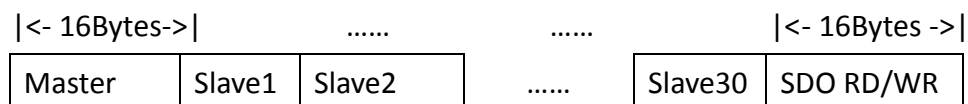
Data_1 在 OP 狀態 下顯示驅動器子站的 Status Word(狀態字)，Status Word 為 EtherCAT CoE Object 0x6041 的標準，相關定義請參考子站使用說明

Data_2 在 OP 狀態 下顯示驅動器子站的 Current Position(目前位置)，此為 EtherCAT CoE Object 0x6064 的標準

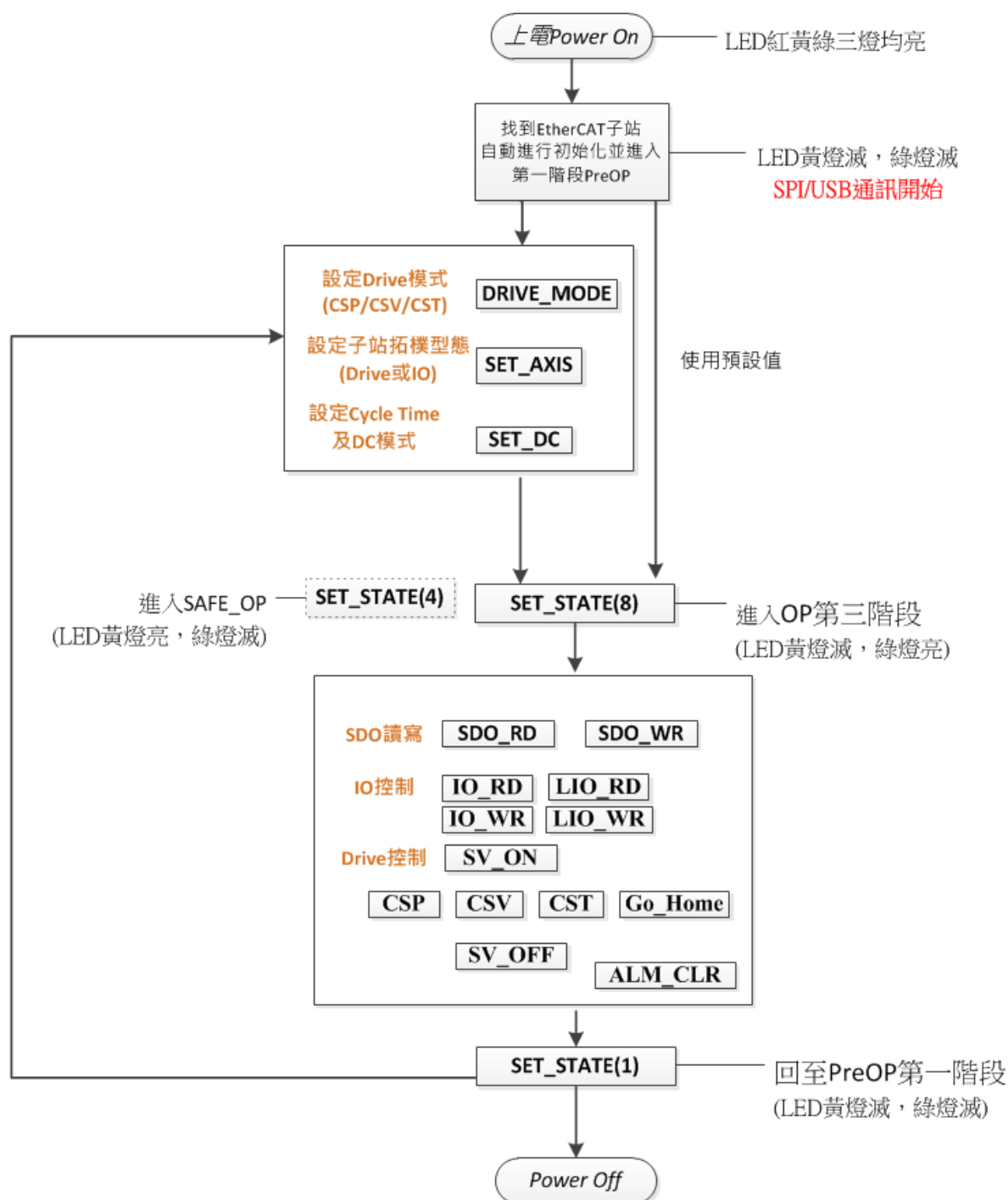
Data_3_L 在 OP 狀態 下顯示驅動器子站的 Current Torque(目前扭力)，此為 EtherCAT CoE Object 0x6077 的標準

Data_3_H 在 OP 狀態 下顯示驅動器子站的 Error Code(錯誤代碼)，此為 EtherCAT CoE Object 0x603F 的標準

當 CONFIG 2 為 High 時，回應資訊均為 16Bytes，前 12Bytes 與上述相同，目前僅 NEXTW HSP 子站在 CSP/CSV 命令時，會使用增加的 4Bytes 空間。



第 3 章 流程範例



第 4 章 動態函式庫

4.1 動態函式庫簡介

「NEXTWUSBLib.dll」為 ECM-SK 使用 USB 做為介面，在 Windows 作業系統中的動態函式庫，支援語言包含 C++、Visual Basic .NET、C# .NET 等，NEXTWUSBLib.dll 須與執行檔.exe 放置相同目錄。「NEXTWUSBLib.lib」為開發 C++ 程式時的參考檔，「NEXTWUSB_dotNET.dll」為開發微軟.NET 程式時的參考函式原型，程式開發時加入專案中的參考。

4.2 NEXTWUSBLib 函式庫

NEXTWUSBLib 函式庫支援 C++、Visual Basic .NET、C# .NET 等.NET 系列程式語言，因在程式執行時間被參考，故須放置於執行檔相同目錄，表 4.1 為「NEXTWUSBLib.dll」動態函式庫主要功能。

表 4.1 「NEXTWUSBLib.dll」主要功能列表

函式名稱	說明	參考
OpenECMUSB()	開啟與 ECM-SK 間的 USB 連線	4.2.1
CloseECMUSB()	關閉與 ECM-SK 間的 USB 連線	4.2.2
ECMUSBWrite()	透過 USB 寫入資料至 ECM-SK	4.2.3
ECMUSBRead()	透過 USB 讀取數據從 ECM-SK	4.2.4

4.2.1 OpenECMUSB 函式說明

函式宣告 **bool** OpenECMUSB()

傳入參數 無

回傳參數 回傳一個 bool 值，指出開啟連線是否成功

說明 使用本函式來開啟與 ECM-SK 間的 USB 連線，若回傳值為 False，有可能有下列原因：

- ECM-SK 並未連接到本機端的任一 USB 埠
- ECM-SK CONFIG 0 未正確設定

- ECM-SK 找不到任何 EtherCAT Slave
- ECM-SK 已經被其他程式開啟
- USB 埠的供電不足，導致 ECM-SK 無法正常運作

4.2.2 CloseECMUSB 函式說明

函式宣告 `void CloseECMUSB()`

傳入參數 無

回傳參數 無

說明 使用本函式來關閉與 ECM-SK 間的 USB 連線，並釋放連線資源

4.2.3 ECMUSBWrite 函式說明

函式宣告 `ECMUSBWrite(unsigned char * data, unsigned long dwlength)`

傳入參數 `data` 指標形態，指向 `unsigned char` 陣列，該陣列中存放欲寫入的資料

`dwlength` `4 bytes unsigned long`，指出欲寫入的資料 `byte` 數，由於每次交換資料為 504 或 512bytes，此值應為 504 或 512

回傳參數 回傳一個 `bool` 值，指出寫入資料是否成功

說明 使用本函式來透過 USB 寫入資料至 ECM-SK，若回傳值為 `False`，有可能有下列原因：

- ECM-SK 未開啟，請先呼叫 `OpenECMUSB` 以開啟連線
- 已開啟的連線失效，可能為 USB 接觸不良或已移除

4.2.4 ECMUSBRead 函式說明

函式宣告 `ECMUSBRead(unsigned char * data, unsigned long dwlength)`

傳入參數 `data` 指標形態，指向 `unsigned char` 陣列，讀取的資料會存放於該陣列中

`dwlength` `4 bytes unsigned long`，指出欲讀出的資料 `byte` 數，由於每次交換資料為 504 或 512bytes，此值應為 504 或 512

回傳參數 回傳一個 `bool` 值，指出讀取資料是否成功

說明 使用本函式來透過 USB 從 ECM-SK 讀取資料，若回傳值為 False，有可能有下列原因：

- ECM-SK 未開啟，請先呼叫 OpenECMUSB 以開啟連線
- 已開啟的連線失效，可能為 USB 接觸不良或已移除

4.3 NEXTWUSB_dotNET 函式庫

NEXTWUSB_dotNET 函式庫支援 Visual Basic .NET、C# .NET 等.NET 系列程式語言，提供.NET 環境下與 ECM-SK 相關的常數定義、結構與函式，提供更為直覺的開發流程，此函式庫在設計階段需加入項目中的參考，執行時間仍會參考

「NEXTWUSBLib.dll」，故「NEXTWUSBLib.dll」須放置於執行檔相同目錄，而

「NEXTWUSB_dotNET_XXB.dll」檔則不須放置於執行檔相同目錄，表 4.2 為

「NEXTWUSB_dotNET_XXB.dll」動態函式庫主要功能，表 4.3 為

「NEXTWUSB_dotNET_12B.dll」動態函式庫命令結構。

表 4.2 「NEXTWUSB_dotNET_XXB.dll」主要常數列表

群組	常數名稱	形態	常數值	說明
最大軸數	DEF_MA_MAX	Int32	42	NEXTWUSB_dotNET_12B.dll 最多 1 ECM、40 Slaves 及 1 組 SDO RD/WR
			32	NEXTWUSB_dotNET_16B.dll 最多 1 ECM、30 Slaves 及 1 組 SDO RD/WR
EtherCAT 狀態	NIC_INIT	Int32	0	初始化網路階段
	STATE_INIT	Int32	1	EtherCAT Init 階段
	STATE_PRE_OP	Int32	2	EtherCAT PreOP 階段
	STATE_SAFE_OP	Int32	4	EtherCAT SafeOP 階段
	STATE_OPERATIONAL	Int32	8	EtherCAT OP 階段
Drive 模式	CSP_MODE	Int32	8	CSP 模式
	CSV_MODE	Int32	9	CSV 模式
	CST_MODE	Int32	10	CST 模式

DC 模式	FREERUN	Int32	0	無 DC 同步
	DCSYNC	Int32	1	DC 同步
子站類別	DRIVE	Int16	0x0	驅動器類型子站
	IO	Int16	0x1	IO 類型子站
	HSP	Int16	0x2	HSP 類型子站
	STEP	Int16	0x3	STEP 類型子站
	None	Int16	0xF	無子站
指令代碼	GET_STATUS	Int16	0x00	取回狀態
	SET_STATE	Int16	0x01	設定 ECM 狀態
	SET_AXIS	Int16	0x02	設定各子站類別
	SET_DC	Int16	0x03	設定週期時間及 DC 模式
	SET_EX	Int16	0x04	設定 CRC 功能
	SET_FIFO	Int16	0x05	設定 FIFO
	DRIVE_MODE	Int16	0x06	設定驅動器模式
	SDO_RD	Int16	0x07	讀取參數
	SDO_WR	Int16	0x08	寫入參數
指令代碼	ALM_CLR	Int16	0x10	清除警告
	SV_ON	Int16	0x11	激磁
	SV_OFF	Int16	0x12	退磁
	IO_RD	Int16	0x13	讀取 IO 子站 Input 狀態
	IO_WR	Int16	0x14	設定 IO 子站 Output 狀態
	CSP	Int16	0x15	CSP 週期指令
	CSV	Int16	0x16	CSV 週期指令
	CST	Int16	0x17	CST 週期指令
	GO_HOME	Int16	0x18	開始回 Home 程式
	ABORT_HOME	Int16	0x19	終止回 Home 程式
指令代碼	LIO_RD	Int16	0x21	讀取 ECM 上的 Input 狀態
	LIO_WR	Int16	0x22	設定 ECM 上的 Output 狀態
指令代碼	SW_RESET	Int16	0xBB	軟體重置

表 4.3 「NEXTWUSB_dotNET_12B.dll」命令結構

結構名稱	成員變數 名稱	形態
transData	CMD	Int16
	Pram	Int16
	Data1	Int32
	Data2	Int32

「NEXTWUSB_dotNET_12B.dll」含有兩個主要的公用變數 cmdData 及 respData，均為 transData 結構陣列，長度均為 42，可存放 1 個 ECM 及 40 個子站的資料，再加上一筆 SDO RD/ WR 專用命令位置。表 4.4 為

「NEXTWUSB_dotNET_16B.dll」動態函式庫命令結構。

表 4.4 「NEXTWUSB_dotNET_16B.dll」命令結構

結構名稱	成員變數 名稱	形態
transData	CMD	Int16
	Pram	Int16
	Data1	Int32
	Data2	Int32
	Data3	Int32

「NEXTWUSB_dotNET_16B.dll」含有兩個主要的公用變數 cmdData 及 respData，均為 transData 結構陣列，長度均為 32，可存放 1 個 ECM 及 30 個子站的資料，再加上一筆 SDO RD/ WR 專用命令位置，表 4.5 為「NEXTWUSB_dotNET_XXB.dll」動態函式庫主要函式。

表 4.5 「NEXTWUSB_dotNET_XXB.dll」動態函式庫主要函式

函式名稱	說明	參考
OpenECMUSB()	開啟與 ECM-SK 間的 USB 連線	4.3.1
CloseECMUSB()	關閉與 ECM-SK 間的 USB 連線	4.3.2
ECMUSBWrite()	透過 USB 寫入資料至 ECM-SK	4.3.3
ECMUSBRead()	透過 USB 讀取數據從 ECM-SK	4.3.4
ClearCmdData()	清除指令	4.3.5

4.3.1 OpenECMUSB 函式說明

函式宣告	bool OpenECMUSB()
傳入參數	無
回傳參數	回傳一個 bool 值，指出開啟連線是否成功
說明	使用本函式來開啟與 ECM-SK 間的 USB 連線，若回傳值為 False，有可能有下列原因： <ul style="list-style-type: none"> ● ECM-SK 未連接到本機端的任一 USB 埠 ● ECM-SK CONFIG 0 未正確設定 ● ECM-SK 找不到任何 EtherCAT Slave ● ECM-SK 已經被其他程式開啟 ● USB 埠的供電不足，導致 ECM-SK 無法正常運作

4.3.2 CloseECMUSB 函式說明

函式宣告	void CloseECMUSB()
傳入參數	無
回傳參數	無
說明	使用本函式來關閉與 ECM-SK 間的 USB 連線，並釋放連線資源

4.3.3 ECMUSBWrite 函式說明

函式宣告	bool ECMUSBWrite()
傳入參數	無
回傳參數	回傳一個 bool 值，指出寫入資料是否成功
說明	使用本函式透過 USB 將 cmdData 的資料寫入至 ECM-SK，若回傳值為 False，有可能有下列原因： <ul style="list-style-type: none"> ● ECM-SK 未開啟，請先呼叫 OpenECMUSB 以開啟連線 ● 已開啟的連線失效，可能為 USB 接觸不良或已移除

4.3.4 ECMUSBRead 函式說明

函式宣告	bool ECMUSBRead()
傳入參數	無
回傳參數	回傳一個 bool 值，指出讀取資料是否成功
說明	使用本函式來透過 USB 從 ECM-SK 讀取數據至 respData，若回傳

值為 False，有可能有下列原因：

- ECM-SK 未開啟，請先呼叫 OpenECMUSB 以開啟連線
- 已開啟的連線失效，可能為 USB 接觸不良或已移除
- 讀回資料的 CRC8 值驗證錯誤

4.3.5 ClearCmdData 函式說明

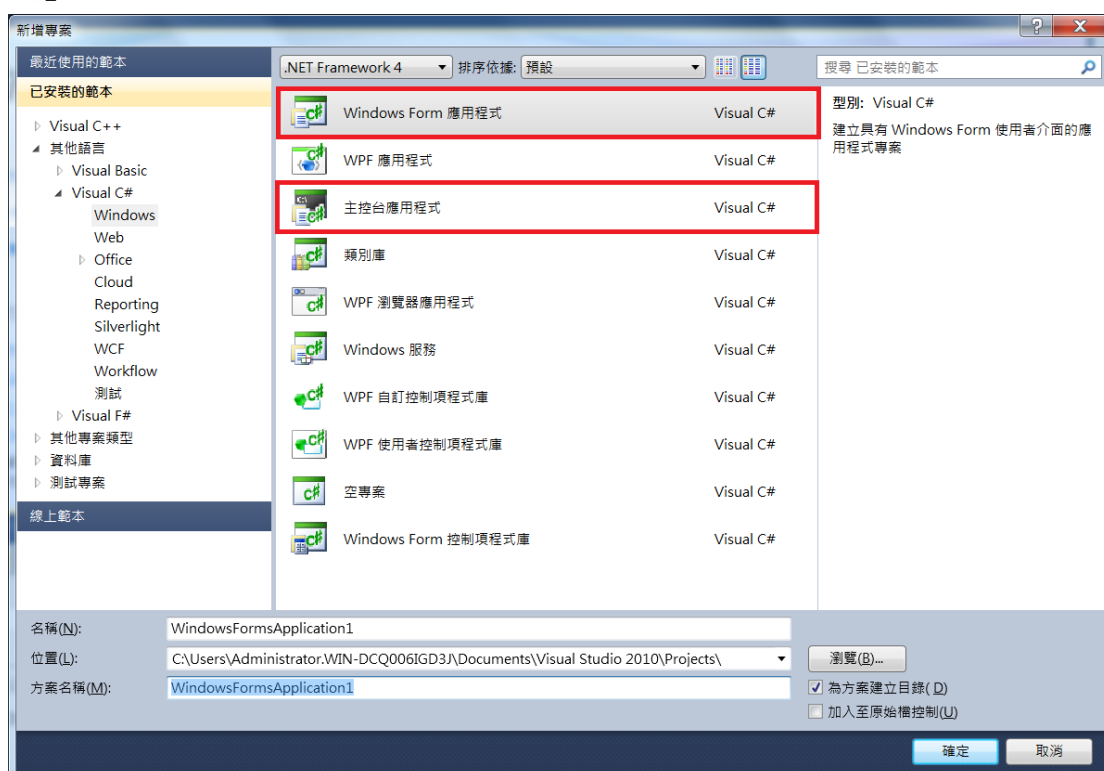
函式宣告	<code>void ClearCmdData()</code>
傳入參數	無
回傳參數	無
說明	使用本函式來清除 cmdData 的內容，使得內容值均為 0

4.4 Visual Studio 專案環境設定

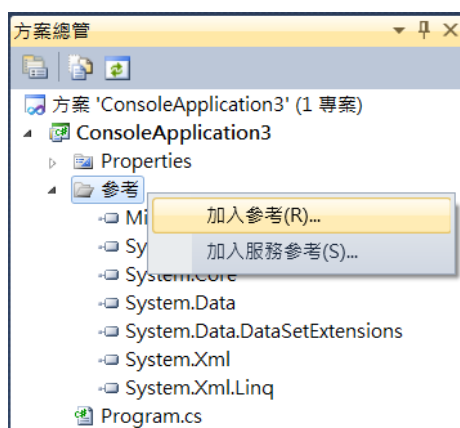
本節以微軟公司的 Visual Studio 2010 的環境為例，逐步說明專案環境設定的步驟。

4.4.1 C# .NET 專案環境設定

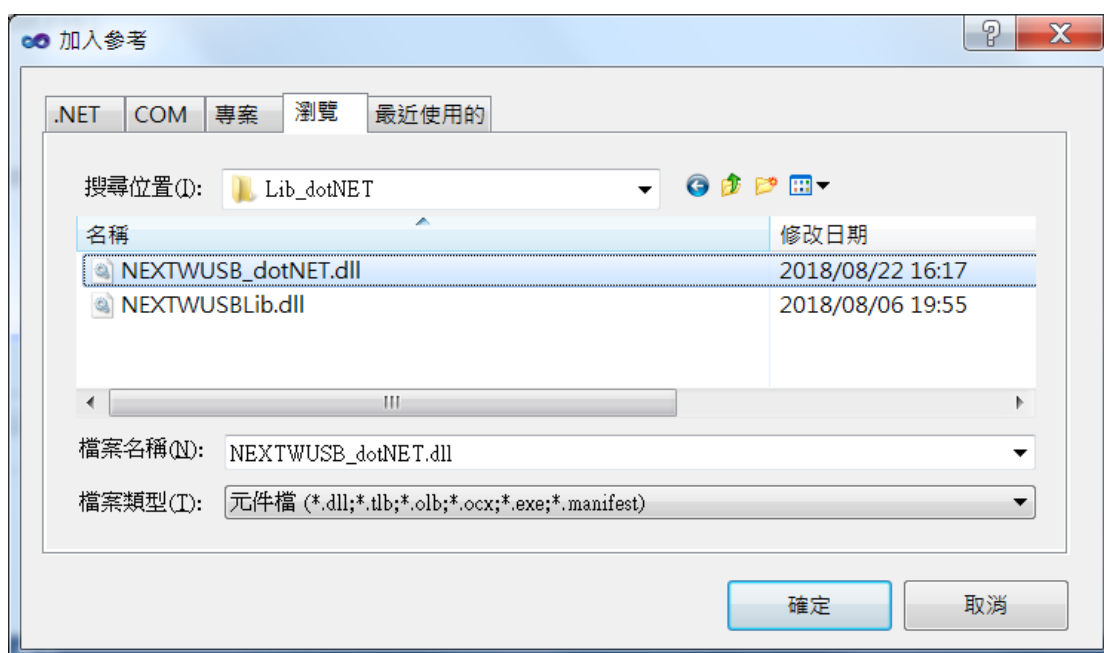
Step 1：新增專案 – 依需求選擇「Windows Form 應用程式」或「控制台應用程式」



Step 2 – 于「方案總管」中的「參考」，按右鍵選擇「加入參考」



Step 3 – 於「流覽」頁籤，選擇「NEXTWUSB_dotNET_XXB.dll」後按確定



Step 4 – 於程式頂端加入命名空間「NEXTWUSB_dotNET_XXB」

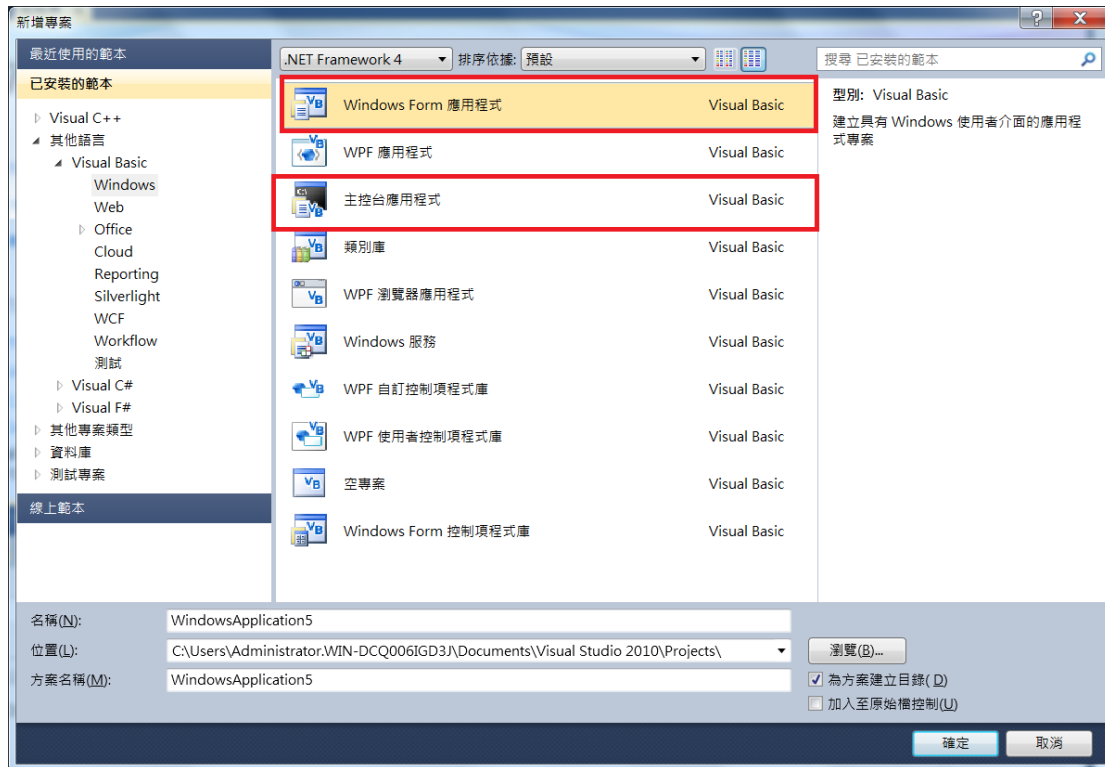
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using NEXTWUSB_dotNET;
```

Step 5 – 即可於主程序使用「NEXTWUSB_dotNET.dll」動態函式庫提供功能

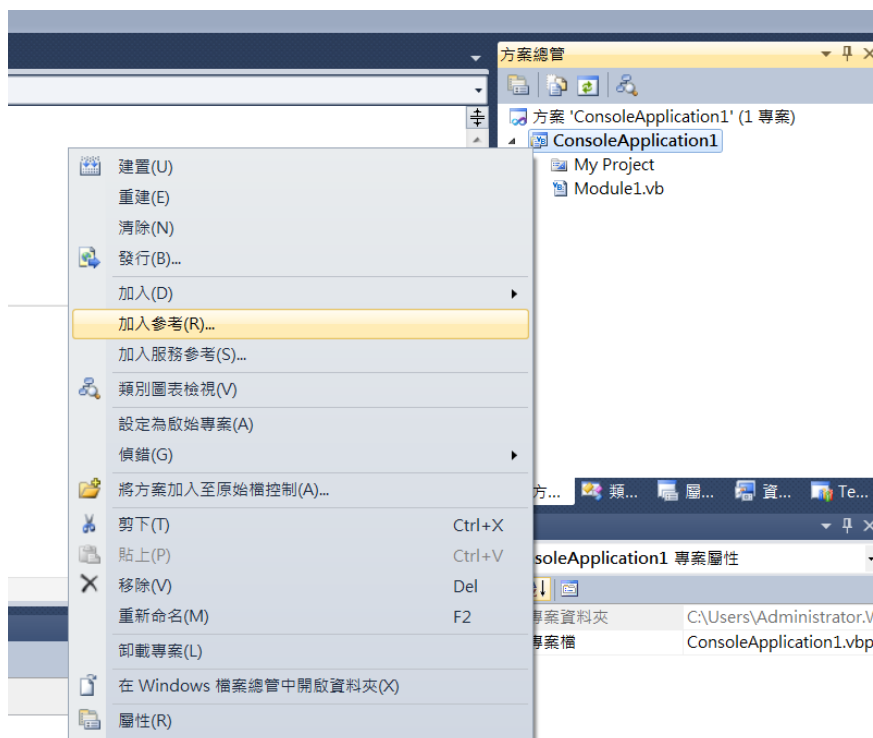
```
class Program
{
    static void Main(string[] args)
    {
        NEXTWUSB.OpenECMUSB( );
        NEXTWUSB.ClearCmdData( );
        NEXTWUSB.ECMUSBWrite( );
        NEXTWUSB.ECMUSBRead( );
        NEXTWUSB.CloseECMUSB( );
    }
}
```

4.4.2 Visual Basic .NET 專案環境設定

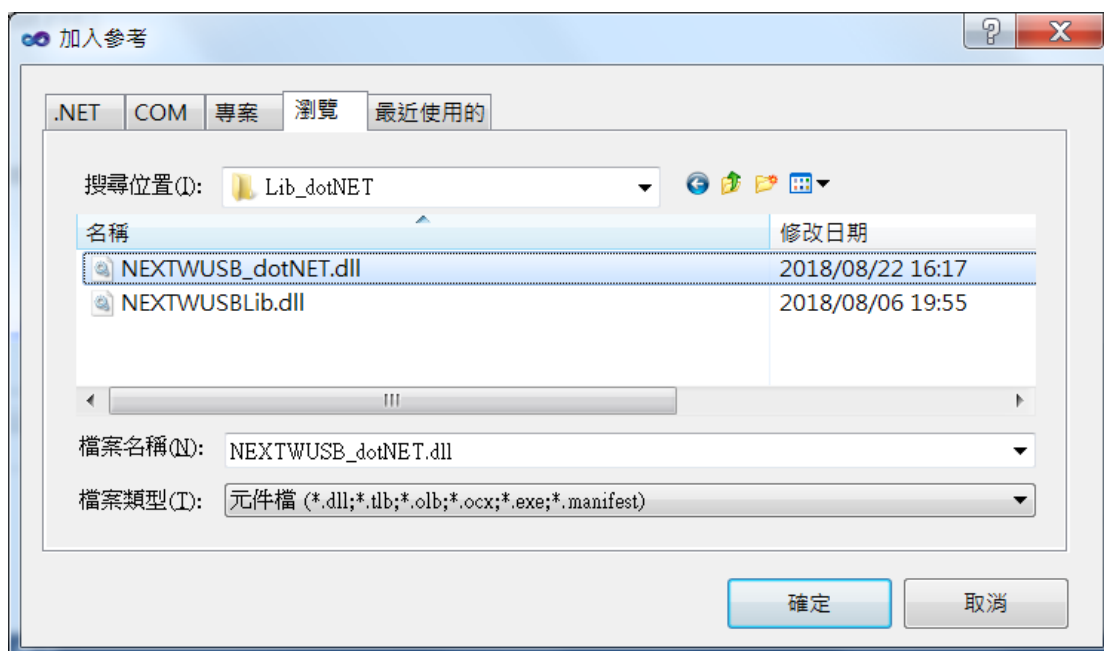
Step 1：新增專案 – 依需求選擇「Windows Form 應用程式」或「控制台應用程式」



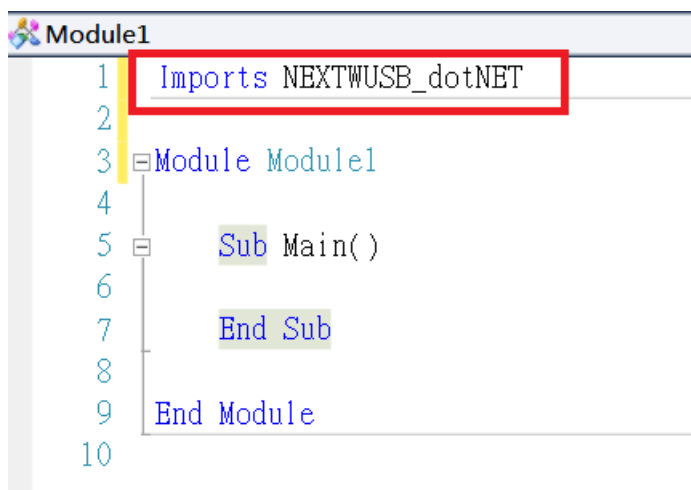
Step 2 – 于「方案總管」中的項目名稱，按右鍵選擇「加入參考」



Step 3 – 於「流覽」頁籤，選擇「NEXTWUSB_dotNET_XXB.dll」後按確定



Step 4 – 於程式頂端加入命名空間「NEXTWUSB_dotNET_XXB」

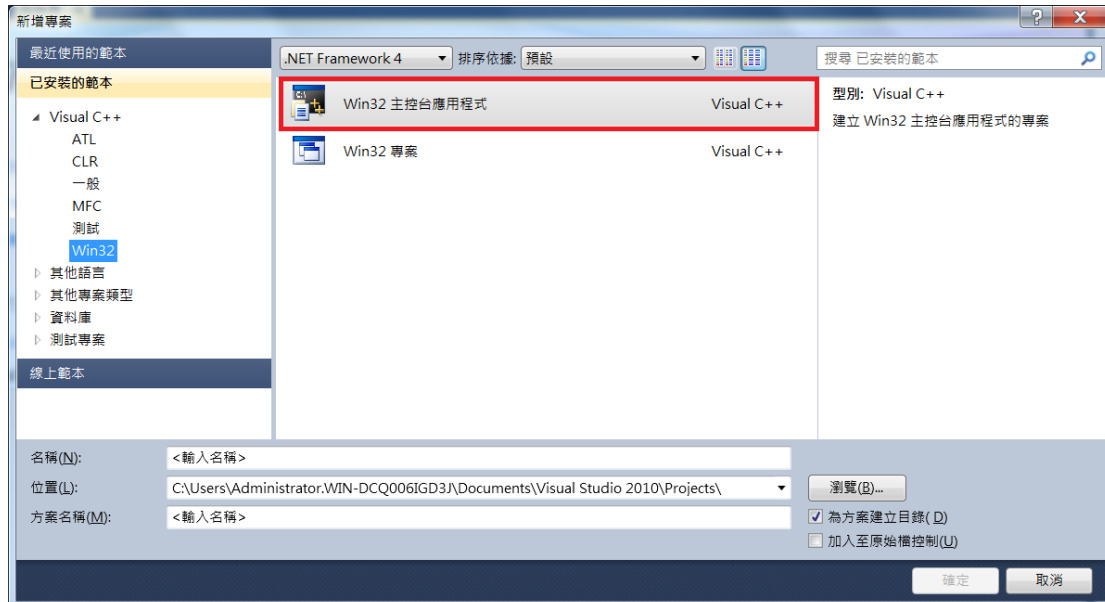


Step 5 – 即可於主程序使用動態函式庫提供功能

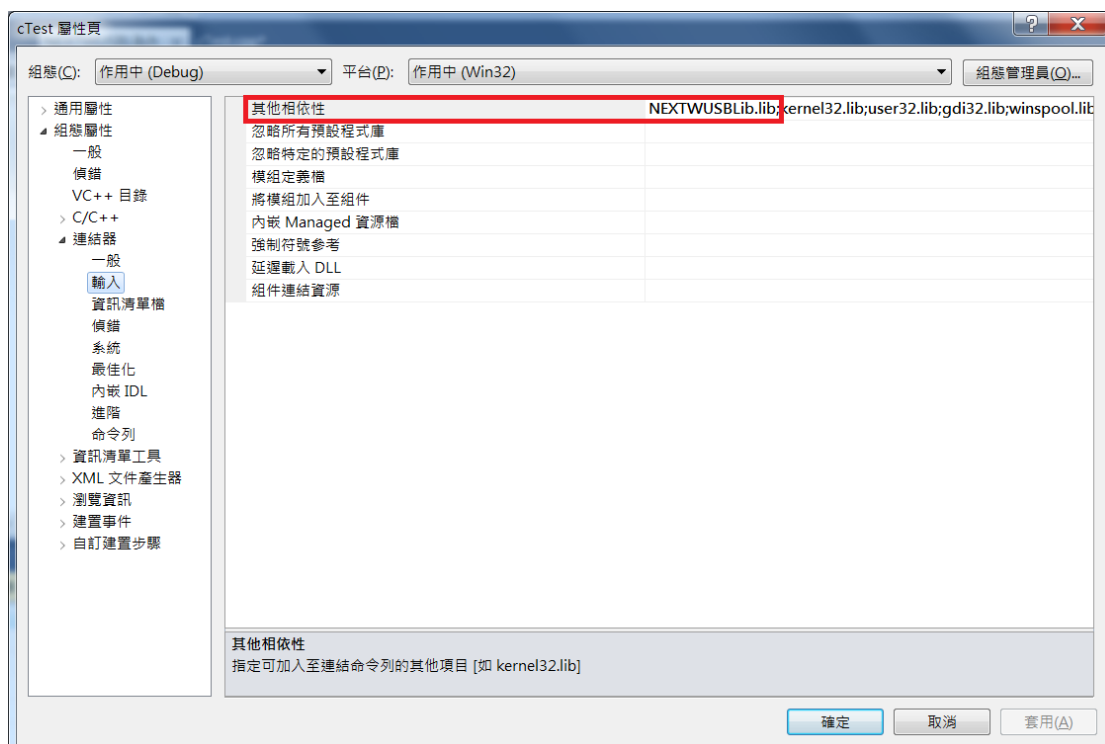
```
Sub Main()  
    NEXTWUSB.OpenECMUSB()  
    NEXTWUSB.ClearCmdData()  
    NEXTWUSB.ECMUSBWrite()  
    NEXTWUSB.ECMUSBRead()  
    NEXTWUSB.CloseECMUSB()  
End Sub
```

4.4.3 C++專案環境設定

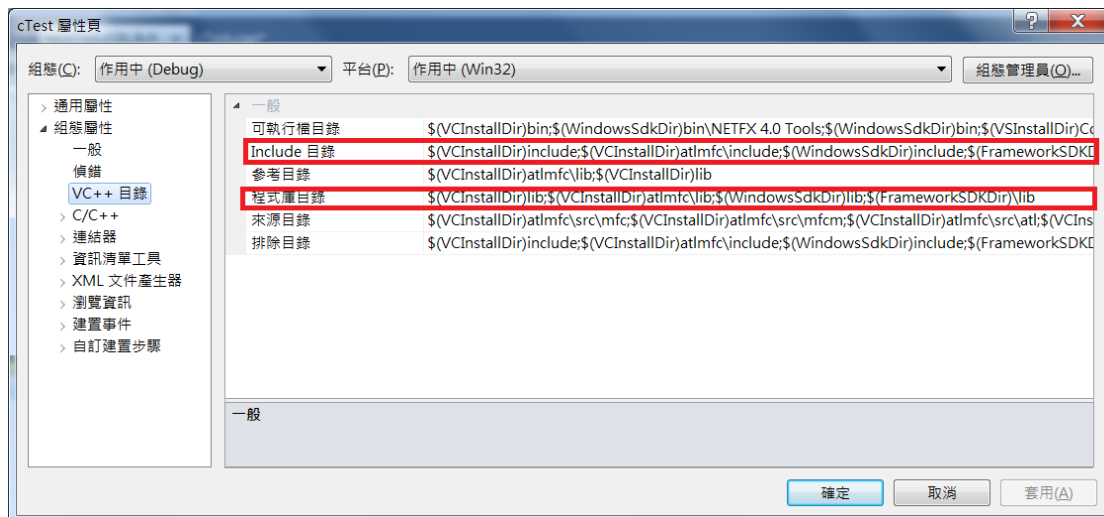
Step 1：新增項目 – 依需求選擇項目類型，此處以「Win32 控制台應用程式」為例



Step 2 – 於專案組態屬性頁中的「連結器」下的「輸入」，右側「其他相依性」欄位，輸入「NEXTWUSBLib.lib」



Step 3 – 於專案組態屬性頁中的「VC++目錄」指定「Include 目錄」及「程式庫目錄」，此處目錄僅供參考，請依實際狀況輸入「NEXTWUSBLib.h」及「NEXTWUSBLib.lib」實際目錄



Step 4 – 於程式頂端加入 Include 定義檔「NEXTWUSBLib.h」

```
#include "stdafx.h"
#include "NEXTWUSBLib.h"
```

Step 5 – 即可於主程序使用「NEXTWUSBLib.dll」動態函式庫提供功能

```
int _tmain(int argc, _TCHAR* argv[])
{
    OpenECMUSB();
    printf("Hello ECM USB");
    CloseECMUSB();
    return 0;
}
```