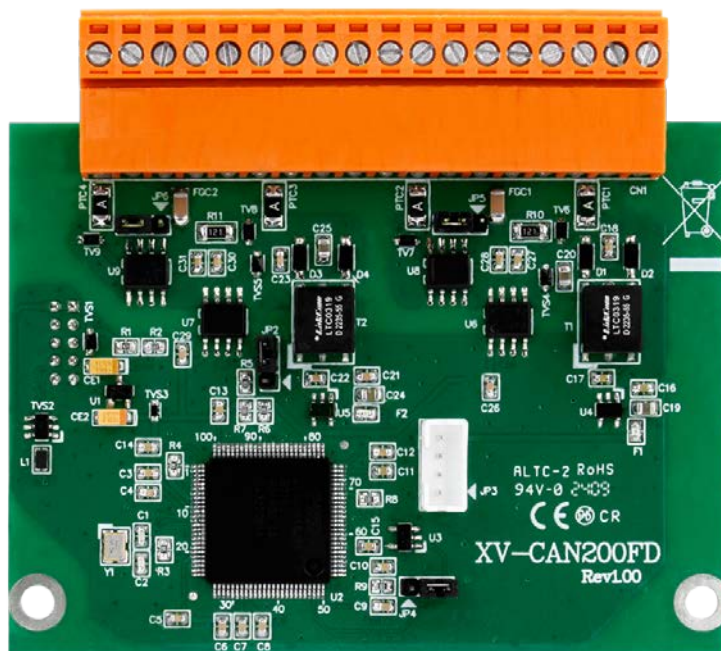


XV-CAN200FD 使用手冊

版本 1.0.1, 2026 年 02 月



產品技術服務與使用資訊

XV-CAN200FD

保固說明

泓格科技股份有限公司（ICP DAS）所生產的產品，均保證原始購買者對於有瑕疵之材料，於交貨日起保有為期一年的保固。

免責聲明

泓格科技股份有限公司對於因為應用本產品所造成的損害並不負任何法律上的責任。本公司保留有任何時間未經通知即可變更與修改本文件內容之權利。本文所含資訊如有變更，恕不予另行通知。本公司盡可能地提供正確與可靠的資訊，但不保證此資訊的使用或其他團體在違反專利或權利下使用。此處包涵的技術或編輯錯誤、遺漏，概不負其法律責任。

版權所有

版權所有 2026 泓格科技股份有限公司保留所有權利。

商標識別

本文件提到的所有公司商標、商標名稱及產品名稱分別屬於該商標或名稱的擁有者所有。

技術服務

如有任何問題，請與本公司客服聯絡，我們將盡速為您服務。

Email 信箱：service@icpdas.com

目錄

1.	簡介	5
1.1.	規格	5
1.2.	特色	6
2.	技術資料	7
2.1.	架構圖	7
2.2.	接腳定義	8
2.3.	終端電阻設定	9
2.4.	接線圖	10
3.	CAN 總線網絡部署	11
3.1.	CAN 總線驅動能力	11
4.	API 函式庫	12
4.1.	API 函式庫概觀	12
4.2.	API 函式庫功能表	14
4.3.	API 函式庫使用流程	17
4.4.	模組初始化功能函式	18
4.4.1.	CANFD_ScanDevice	18
4.4.2.	CANFD_ListDevice	19
4.4.3.	CANFD_OpenDevice	20
4.4.4.	CANFD_CloseDevice	21
4.5.	模組設定功能函式	22
4.5.1.	CANFD_SetCANOPMode	22
4.5.2.	CANFD_GetCANOPMode	24
4.5.3.	CANFD_SetCANADBaudRate	26
4.5.4.	CANFD_GetCANADBaudRate	28
4.5.5.	CANFD_SetCANGlobalFilter	30
4.5.6.	CANFD_GetCANGlobalFilter	31
4.5.7.	CANFD_SetCANSTDIDFilter	33
4.5.8.	CANFD_GetCANSTDIDFilter	35
4.5.9.	CANFD_SetCANEXTIDFilter	36

4.5.10.	CANFD_GetCANEXTIDFilter.....	38
4.5.11.	CANFD_GetCANStatus	39
4.5.12.	CANFD_SetCANReInitialize.....	41
4.6.	模組通訊功能函式	42
4.6.1.	CANFD_SetCANTxMsg	42
4.6.2.	CANFD_GetCANRxMsg.....	44
4.6.3.	CANFD_SetCANHWSendMode	47
4.6.4.	CANFD_GetCANHWSendMode.....	48
4.6.5.	CANFD_SetCANHWSendMsg	49
4.6.6.	CANFD_GetCANRxFramePerSec	52
4.7.	資料緩衝區功能函式	53
4.7.1.	CANFD_GetCANRxMsgCount.....	53
4.7.2.	CANFD_ClearCANRxBuf.....	54
4.7.3.	CANFD_ClearCANTxBuf	55
4.8.	其他功能函式	56
4.8.1.	CANFD_GetDllVersion	56
4.8.2.	CANFD_GetFwVer	57
4.8.3.	CANFD_SetSN	58
4.8.4.	CANFD_ResetModule	59
4.9.	函式回傳碼.....	60
5.	附錄	61
5.1.	文件版本歷史紀錄	61
5.2.	CAN Status 暫存器.....	62
5.3.	CAN Error Counter 暫存器.....	63
5.4.	支援的 CAN FD 資料欄位飽率.....	64

1. 簡介

XV-CAN200FD 為 LP-2241M, LP-2841M 提供兩個隔離的 CAN FD 通訊埠擴充，每個 CAN FD 通訊埠皆有隔離，使得 LP-2241M, LP-2841M 在應用上可以快速及方便的擴充 CAN FD 通訊協定介面。

1.1. 規格

模組	XV-CAN200FD
CAN 介面	
通道數	2
接頭	18-pin terminal-block
通訊速率	CAN bit rates: 10 ~ 1000 kbps, CAN FD bit rates for data field: 100 ~ 10000 kbps
終端電阻	跳線帽開關設定120 Ω 終端電阻
隔離	3000 V DC-DC隔離, 依UL1577規範2500 Vrms持續一分鐘 (光耦合)
規範	ISO-11898-2, 支援 CAN 2.0 A/B 與 CAN FD
電源	
功耗	1.5 W (Max.)
機構	
尺寸(長 x 寬 x 高)	62 mm x 80 mm x 13.5 mm (W x L x H)
環境	
操作溫度	-25 ~ 75 $^{\circ}\text{C}$
儲存溫度	-30 ~ 80 $^{\circ}\text{C}$
濕度	相對溼度 10 ~ 90% RH, 無結露

1.2. 特色

- 相容於ISO 11898-2標準規範
- 支援 CAN2.0A/2.0B及CAN FD協定標準
- CAN FD支援ISO and Non-ISO (Bosch)標準
- CAN FD鮑率範圍(資料欄位)從100kbps至10000kbps
- CAN鮑率範圍從10kbps至1000kbps
- 支援 CAN ID 訊息過濾設定功能
- 接收到之CAN訊息時間戳記，精準度可達±1毫秒。
- CAN總線內建120歐姆終端電阻 (可透過跳線帽開關調整)。

2. 技術資料

2.1. 架構圖

下圖為 XV-CAN200FD 功能架構圖，CAN 埠具備 3000Vrms 隔離電壓保護

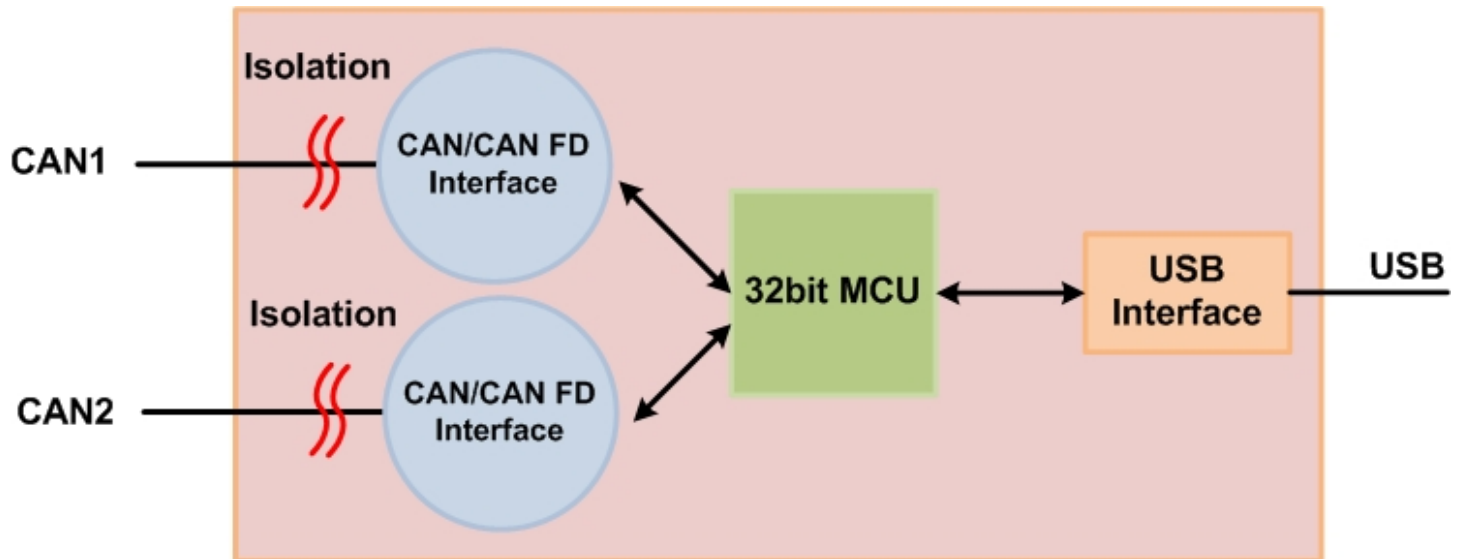
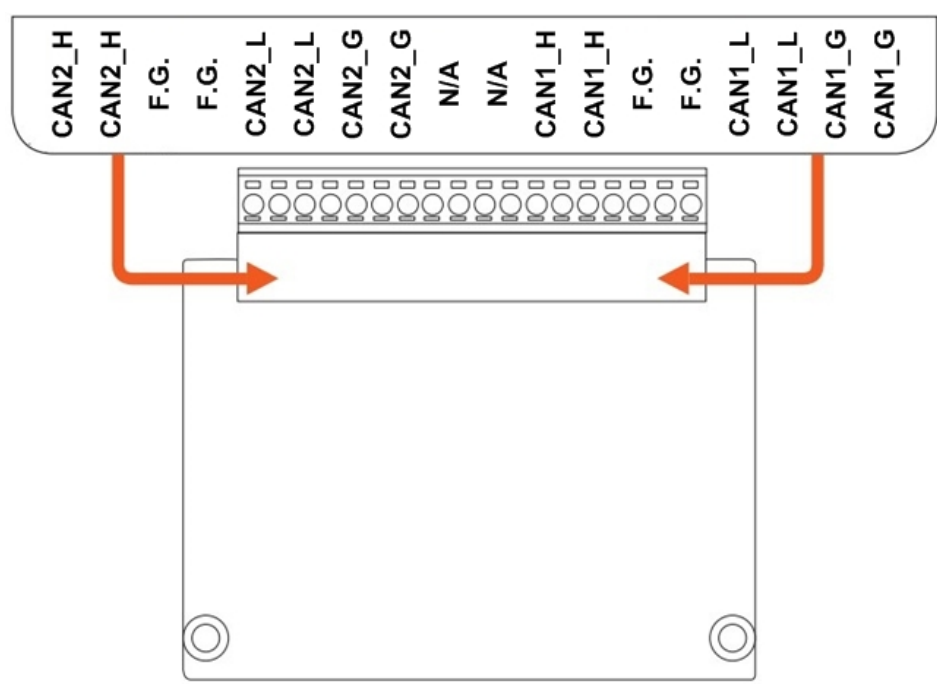


圖 2-1-1 XV-CAN200FD 架構圖

2.2. 接腳定義



XV-CAN200FD 18-pin terminal block 端子接腳定義如下圖所示

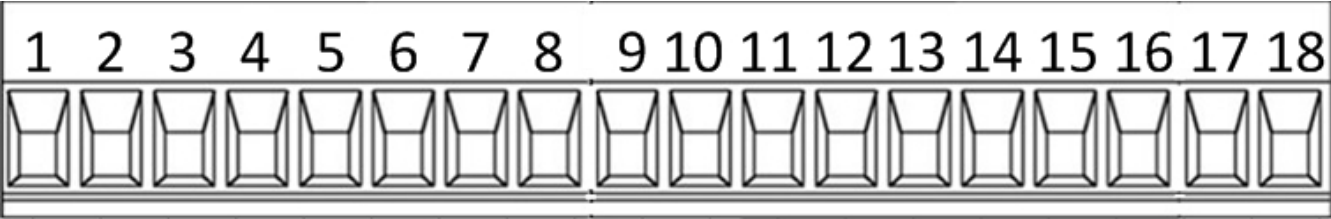


表 2-2-1 接腳定義

腳位	名稱	描述
1, 2	CAN1_G	CAN1 埠的 CAN_Gnd 訊號腳位
3, 4	CAN1_L	CAN1 埠的 CAN_Low 訊號腳位
5, 6	F.G.	大地接地腳位
7, 8	CAN1_H	CAN1 埠的 CAN_High 訊號腳位
9, 10	N/A	N/A
11, 12	CAN2_G	CAN2 埠的 CAN_Gnd 訊號腳位
13, 14	CAN2_L	CAN2 埠的 CAN_Low 訊號腳位
15, 16	F.G.	大地接地腳位
17,18	CAN2_H	CAN2 埠的 CAN_High 訊號腳位

2.3. 終端電阻設定

為了盡量減少 CAN 總線上的反射效果，CAN 總線具有在兩端使用兩個終端電阻將總線終止。根據 ISO11898-2 規範，每一個終端電阻為 120Ω (或 108Ω~132Ω之間)。總線拓撲結構和這些終端電阻的位置，如下圖所示。

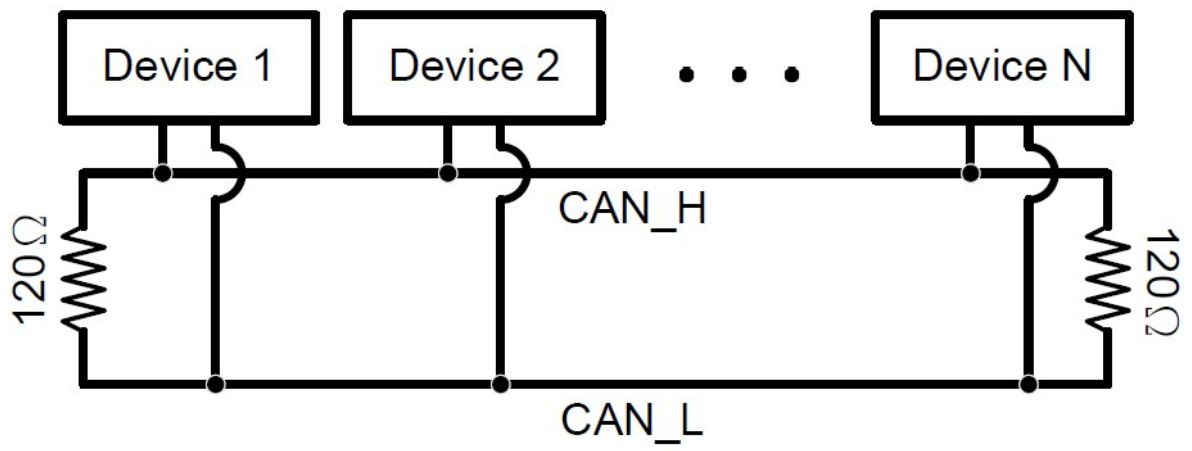


圖 2-3-1 CAN 總線網絡拓撲

XV-CAN200FD 模組每個 CAN 埠均有一個內建的 120Ω終端電阻，使用者可以透過調整跳線帽開關來決定是否啟用。終端電阻的跳線帽開關位於的下圖的 JP5/JP6 位置。

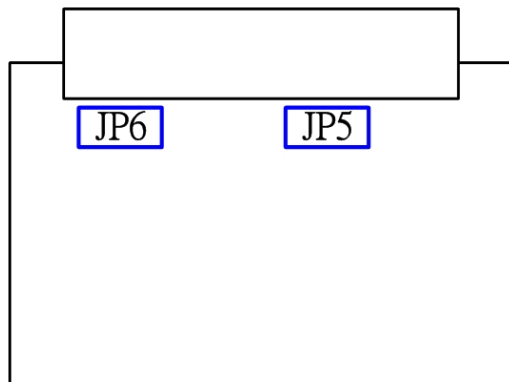
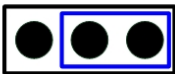


圖 2-3-2 XV-CAN200FD 終端電阻跳線帽開關位址

下圖為終端電阻跳線帽開關啟用及停用的位置圖，預設為啟用狀態。

表 2-3-1 終端電阻對應之指撥開關位置調整說明

JP5/JP6		開關位址	描述
 Enable	 Disable	JP5	Enable: 啟用 CAN1 終端電阻 (預設) Disable: 停用 CAN1 終端電阻
		JP6	Enable: 啟用 CAN2 終端電阻 (預設) Disable: 停用 CAN2 終端電阻

2.4. 接線圖

XV-CAN200FD 模組的 CAN 總線接線圖如下圖所示。

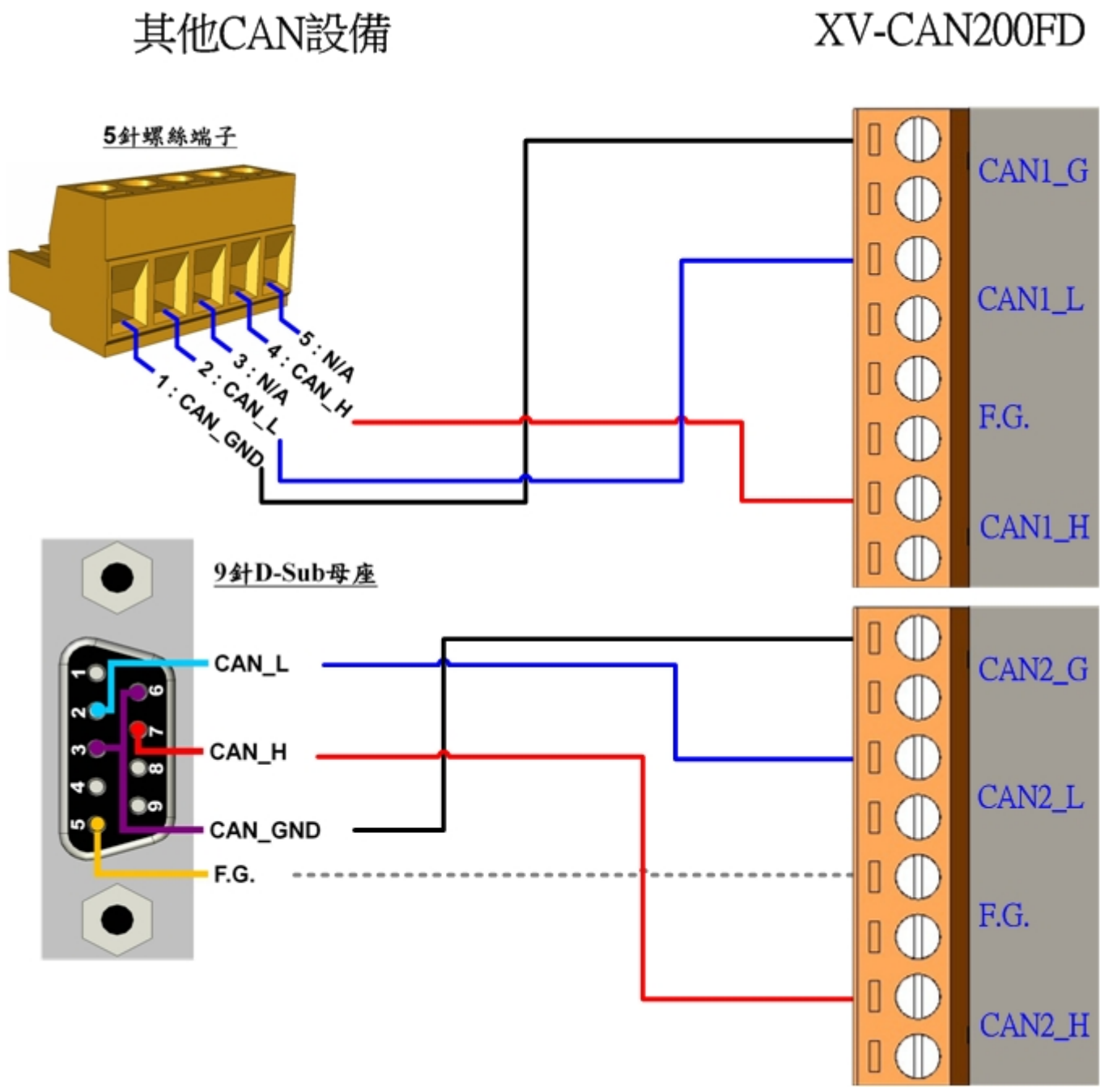


圖 2-4-1 XV-CAN200FD CAN 總線接線圖

3. CAN 總線網絡部署

3.1. CAN 總線驅動能力

在介紹 XV-CAN200FD 模組的 CAN 總線驅動能力之前，必須先假定銅電纜的某些特徵。阻值參數為 120Ω 阻抗和毫秒/線延遲，直流參數如下面的表格所述。

表 3-1-1 CAN 總線導線截面建議的阻值

導線截面[mm ²]	阻值[Ω/km]
~0.25 (AWG23)	< 90
~0.5 (AWG20)	< 50
~0.8 (AWG18)	< 33
~1.3 (AWG16)	< 20

如上所述的條件下，用戶可以參考下表來得知按照 ISO11898-2 規範，每個節段的最大的節點數以及使用不同類型的導線時，最大數據段長度為何。

表 3-1-2 驅動能力

導線截面[mm ²]	在特定節點數量情況下的最大節段長度[m]			
	16 節點	32 節點	64 節點	100 節點
~0.25 (AWG23)	< 220	< 200	< 170	< 150
~0.5 (AWG20)	< 390	< 360	< 310	< 270
~0.8 (AWG18)	< 590	< 550	< 470	< 410
~1.3 (AWG16)	< 980	< 900	< 780	< 670

4. API 函式庫

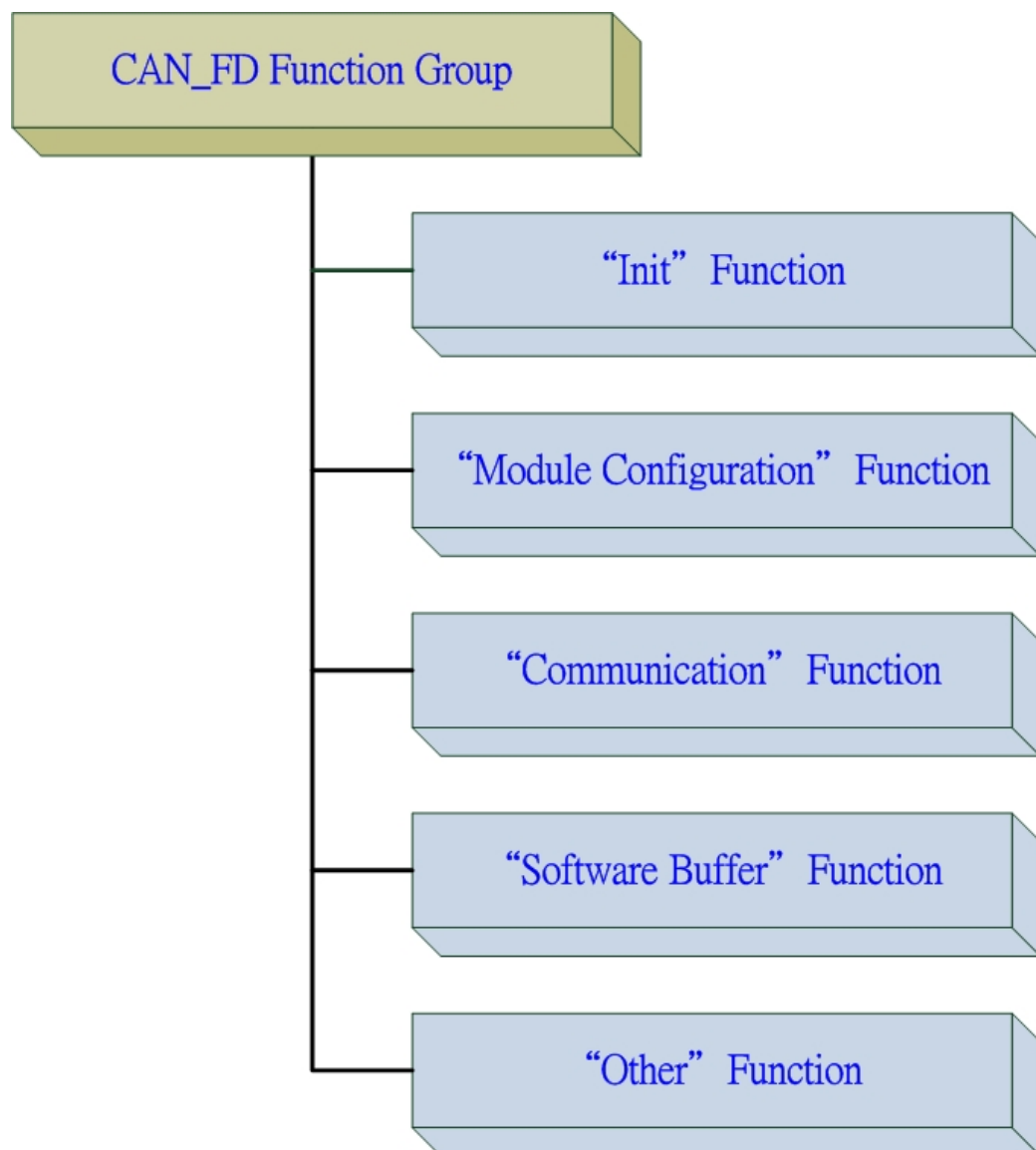
使用者可透過 XV-CAN200FD 應用程式介面函式庫，輕易地自行開發與 CAN Bus 通訊的網路程式。CAN_FD 函式庫與展示範例可從泓格科技網頁上取得：

XV-CAN200FD 使用於 LP-2241M, LP-2841M 的函式庫&範例路徑如下：

<https://www.icpdas.com/tw/download/show.php?num=9581>

4.1. API 函式庫概觀

所有由應用程式介面函式庫提供的函式，大致上可分為 5 個群組：



[Init Function]

該群組函式功能為搜尋及啟用/停用所有可正常使用的 XV-CAN200FD 模組。

[Module Configuration Function]

該群組函式功能為讀取/設定 XV-CAN200FD 模組參數及資訊。

[Communication Function]

該群組函式功能為透過 XV-CAN200FD 模組來傳送及接收 CAN/CAN FD 訊息。

[Software Function]

所有的 CAN 總線傳送/接收的 CAN/CAN FD 訊息均會先存放在 CAN_FD 函式庫的軟體緩衝區上。以便使用者透過此群組功能的函式來取得。

[Other Function]

該群組函式功能為取得 CAN_FD 函式庫資訊或提供使用者撰寫程式上的協助。

4.2. API 函式庫功能表

CAN_FD.dll 提供的函式庫詳列於下表。

“Init” 功能表		
編號	函式名稱	功能描述
1	CANFD_ScanDevice	搜尋電腦上所有支援的設備
2	CANFD_ListDevice	將所有搜尋到的設備 pid/bid 資訊條列至資料表上
3	CANFD_OpenDevice	經由所選擇之設備 pid/bid 資訊來開啟設備
4	CANFD_CloseDevice	停用所選擇之設備

“Module Configuration” 功能表		
編號	函式名稱	功能描述
1	CANFD_SetCANOPMode	在指定的 CAN 埠上，設定 CAN 啟用/監聽及 CAN FD ISO 等操作模式
2	CANFD_GetCANOPMode	在指定的 CAN 埠上，取得模組 CAN 啟用/監聽及 CAN FD ISO 等操作模式
3	CANFD_SetCANADBaudRate	在指定的 CAN 埠上，設定模組實際的 CAN/CAN FD 鮑率及取樣點設定值
4	CANFD_GetCANADBaudRate	在指定的 CAN 埠上，取得模組實際的 CAN/CAN FD 鮑率及取樣點設定值
5	CANFD_SetCANGlobalFilter	在指定的 CAN 埠上，設定模組的遠端 CAN 訊息 ID 過濾器設定參數
6	CANFD_GetCANGlobalFilter	在指定的 CAN 埠上，取得模組的遠端 CAN 訊息 ID 過濾器設定參數及標準/延伸 CAN 訊息過濾器列表的長度
7	CANFD_SetCANSTDIDFilter	在指定的 CAN 埠上，設定模組標準 CAN 訊息過濾 ID 及 ID 列表的長度
8	CANFD_GetCANSTDIDFilter	在指定的 CAN 埠上，取得模組標準 CAN 訊息過濾 ID
9	CANFD_SetCANEXTIDFilter	在指定的 CAN 埠上，設定模組延伸 CAN 訊息

		過濾 ID 及 ID 列表的長度
10	CANFD_GetCANEXTIDFilter	在指定的 CAN 埠上，取得模組延伸 CAN 訊息過濾 ID
11	CANFD_GetCANStatus	在指定的 CAN 埠上，取得 CAN 總線的狀態
12	CANFD_SetCANReInitialize	在指定的 CAN 埠上，將該 CAN 埠重新啟動

“Communication” 功能表		
編號	函式名稱	功能描述
1	CANFD_SetCANTxMsg	傳送一筆 CAN/CAN FD 訊息至所選擇的 CAN 埠的傳送緩衝區中
2	CANFD_GetCANRxMsg	由所選擇的 CAN 埠的接收緩衝區中，取得一筆接收到的 CAN/CAN FD 訊息
3	CANFD_SetCANHWSendMode	啟用/停用模組在指定的 CAN 埠上，利用模組硬體時鐘，周期性的傳送 CAN/CAN FD 訊息
4	CANFD_GetCANHWSendMode	讀取模組在指定的 CAN 埠上，利用模組硬體時鐘，周期性傳送 CAN/CAN FD 訊息的啟用/停用模式
5	CANFD_SetCANHWSendMsg	設定模組在指定的 CAN 埠上，利用模組硬體時鐘，周期性傳送 CAN/CAN FD 訊息的資料內容
6	CANFD_GetCANRxFramePerSec	取得模組在指定的 CAN 埠上，CAN 總線的訊息接收流量值

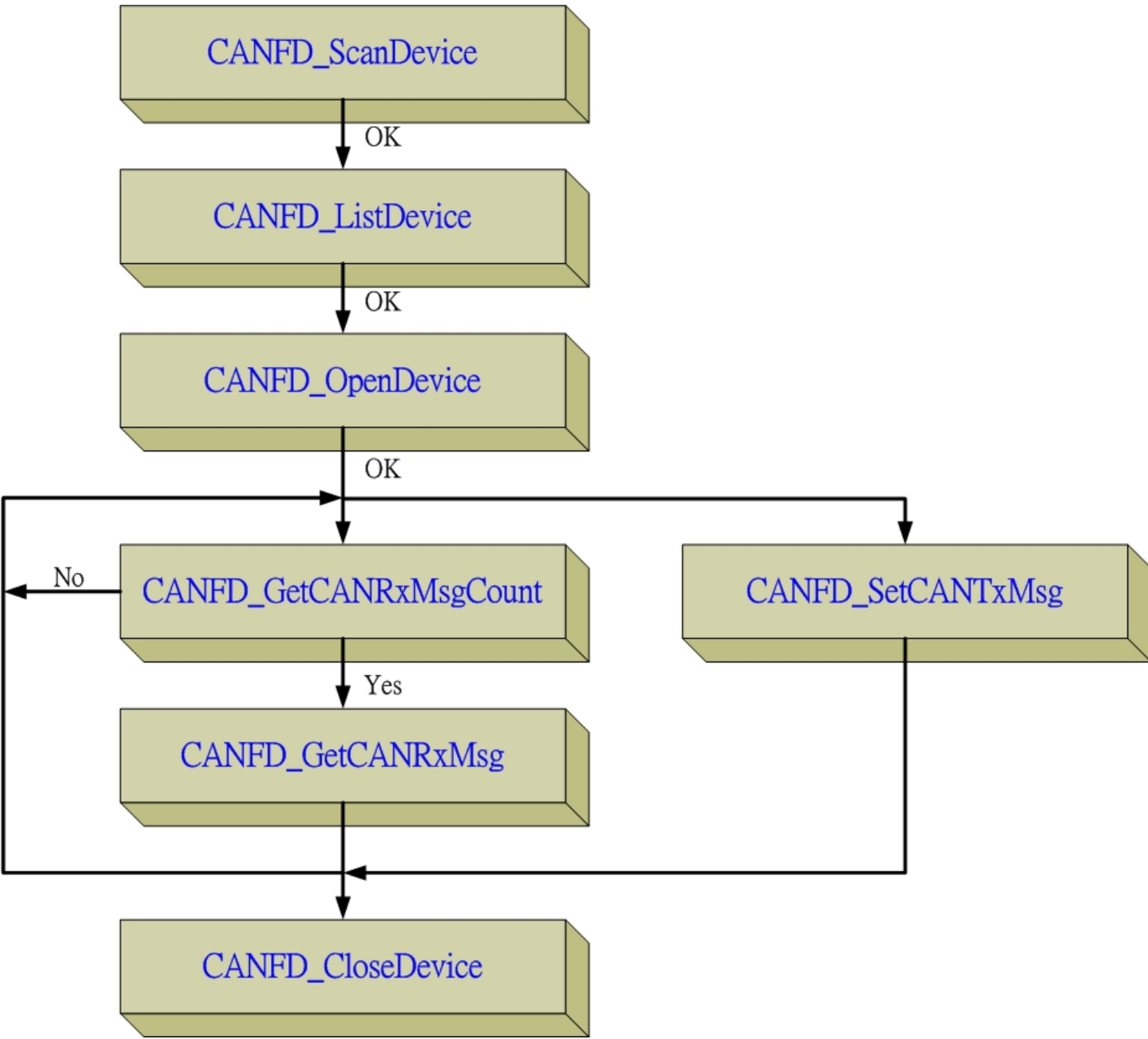
“Software Buffer” 功能表		
編號	函式名稱	功能描述
1	CANFD_GetCANRxMsgCount	取得模組在指定的 CAN 埠上，接收緩衝區內的 CAN/CAN FD 訊息筆數
2	CANFD_ClearCANRxBuf	清除模組在指定的 CAN 埠上，接收緩衝區內

		的所有 CAN/CAN FD 訊息
3	CANFD_ClearCANTxBuf	清除模組在指定的 CAN 埠上，傳送緩衝區內的所有 CAN/CAN FD 訊息

“Other” 功能表		
編號	函式名稱	功能描述
1	CANFD_GetDllVersion	取得函式庫的版本資訊
2	CANFD_GetFwVer	取得模組的韌體版本資訊
3	CANFD_SetSN	設定模組的 bid (board id) 資訊
4	CANFD_ResetModule	重置模組

4.3. API 函式庫使用流程

下圖是使用者使用 CAN_FD 函式庫來開發 CAN 總線程式的基本控制流程圖。



4.4. 模組初始化功能函式

該群組函式功能為搜尋及啟用/停用所有可正常使用的 XV-CAN200FD 模組。

4.4.1. CANFD_ScanDevice

此函式功能為搜尋電腦上所有可正常使用的 XV-CAN200FD 模組。

語法:

C++
int CANFD_ScanDevice(void);
C#
Int32 CANFD_ScanDevice();

參數:

無。

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.4.2. CANFD_ListDevice

此函式功能在於將所有搜尋到的 XV-CAN200FD 設備的 PID (product ID) · BID (board ID)資訊條列至 PID/BID 資料表上。函式庫最多支援 8 個 XV-CAN200FD 設備。

語法:

C++
BYTE CANFD_ListDevice(WORD* o_wPID, DWORD* o_dwBID);
C#
Byte CANFD_ListDevice(UInt16[] o_wPID, UInt32[] o_dwBID);

參數:

***o_wPID**

[out] 此指標指向一個最多可存放 8 筆搜尋到的模組 PID(product ID)資訊的資料陣列。

***o_dwBID**

[out] 此指標指向一個最多可存放 8 筆搜尋到的模組 BID(board ID)資訊的資料陣列。

回傳值:

回傳函式庫所搜尋得的模組個數。

4.4.3. CANFD_OpenDevice

此函式功能在於經由所選擇之設備 pid/bid 資訊來開啟設備。當設備成功開啟時，使用者會得到一組設備編號，使用者可使用此設備編號來使用“Communication”函式庫群組內的 CAN 資料收送函式。

語法:

C++
int CANFD_OpenDevice(WORD *o_wDevice_id, WORD i_wpid, DWORD i_wbid);
C#
Int32 CANFD_OpenDevice(out UInt16 o_wDevice_id, UInt16 i_wpid, UInt32 i_wbid);

參數:

***o_wDevice_id**

[out] 此指標指向一個可存放 經由模組 PID (product ID)/ BID (board ID)所成功開啟的設備編號參數位址。

I_wpid

[in] 欲開啟的 XV-CAN200FD 設備 PID (product ID)。

I_wbid

[in] 欲開啟的 XV-CAN200FD 設備 BID (board ID)。

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.4.4. CANFD_CloseDevice

此函式在停用所選擇之 XV-CAN200FD 設備。在停用後，該設備在函式庫所使用之資源將會被釋放掉。

語法:

C++

```
int CANFD_CloseDevice(WORD i_wDevice_id);
```

C#

```
Int32 CANFD_CloseDevice(UInt16 i_wDevice_id);
```

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.5. 模組設定功能函式

該群組函式功能為讀取/設定 XV-CAN200FD 模組的操作模式、CAN 總線通訊參數、CAN 總線訊息過濾等設定參數及資訊。

4.5.1. CANFD_SetCANOPMode

此功能函式在指定的 CAN 埠上，設定 CAN 埠啟用/監聽及 CAN FD ISO/Non-ISO 等操作模式。

語法:

C++

```
int CANFD_SetCANOPMode(WORD i_wDevice_id, BYTE i_byCANPort, WORD i_wEnable, WORD i_wBusMode, WORD i_wISOMode);
```

C#

```
Int32 CANFD_SetCANOPMode(UInt16 i_wDevice_id, Byte i_byCANPort, UInt16 i_wEnable, UInt16 i_wBusMode, UInt16 i_wISOMode);
```

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

i_byCANPort

[in] 指定的設備 CAN 埠編號。

1: CAN1, 2: CAN2

i_wEnable

[in] 啟用/停用所指定的 CAN 埠功能。

0: 停用, 1: 啟用.

i_wBusMode

[in] 設定所指定的 CAN 埠功能為一般/監聽模式。

0: 一般模式, 可以正常收送 CAN 總線訊息。

1: 監聽模式, 只可以接收 CAN 總線訊息

i_wlISOMode

- [in]** 設定所指定的 CAN 埠之 CAN FD 功能為遵循 Bosch CAN FD Specification V1.0 或 ISO11898-1 之 CAN FD 訊息標準。
- 0: CAN FD 功能為 ISO11898-1 之 CAN FD 訊息標準。
- 1: CAN FD 功能為 Bosch CAN FD Specification V1.0 之 CAN FD 訊息標準。

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.5.2. CANFD_GetCANOPMode

此功能函式在指定的 CAN 埠上，讀取 CAN 埠啟用/監聽及 CAN FD ISO/Non-ISO 等操作模式。

語法:

C++
<pre>int CANFD_GetCANOPMode(WORD i_wDevice_id, BYTE i_byCANPort, WORD *o_wEnable, WORD* o_wBusMode, WORD* o_wISOMode);</pre>
C#
<pre>Int32 CANFD_GetCANOPMode(UInt16 i_wDevice_id, Byte i_byCANPort, out UInt16 o_wEnable, out UInt16 o_wBusMode, out UInt16 o_wISOMode);</pre>

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

I_byCANPort

[in] 指定的設備 CAN 埠編號。

1: CAN1, 2: CAN2

****o_wEnable***

[out] 取得所指定的 CAN 埠上的啟用/停用設定功能。

0: 停用, 1: 啟用.

****o_wBusMode***

[out] 取得所指定的 CAN 埠上的一般/監聽模式設定功能。

0: 一般模式, 可以正常收送 CAN 總線訊息。

1: 監聽模式，只可以接收 CAN 總線訊息

****o_wISOMode***

[out] 取得所指定的 CAN 埠上的 CAN FD 功能為遵循 Bosch CAN FD Specification V1.0 或 ISO11898-1 之 CAN FD 訊息標準設定功能。

0: CAN FD 功能為 ISO11898-1 之 CAN FD 訊息標準。

1: CAN FD 功能為 Bosch CAN FD Specification V1.0 之 CAN FD 訊息標準。

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.5.3. CANFD_SetCANADBaudRate

此函式功能在指定的 CAN 埠上，設定模組欲使用的 CAN/CAN FD 仲裁及資料欄位鮑率及取樣點設定值。

語法:

C++
int CANFD_SetCANADBaudRate(WORD i_wDevice_id, BYTE i_byCANPort, DWORD i_dwArbitrBR, DWORD i_dwDataBR, WORD i_wArbitrBRSP, WORD i_wDataBRSP);
C#
Int32 CANFD_SetCANADBaudRate(UInt16 i_wDevice_id, Byte i_byCANPort, UInt32 i_dwArbitrBR, UInt32 i_dwDataBR, UInt16 i_wArbitrBRSP, UInt16 i_wDataBRSP);

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

I_byCANPort

[in] 指定的設備 CAN 埠編號。

1: CAN1, 2: CAN2

i_dwArbitrBR

[in] 所指定的 CAN 埠 CAN 訊息鮑率設定值及 CAN FD 訊息仲裁欄位的鮑率設定值。單位: bps (bit per second), 有效範圍為 10000 ~ 1000000 (10 kbps ~ 1000 kbps).

I_dwDataBR

[in] 所指定的 CAN 埠 CAN FD 訊息資料欄位的鮑率設定值。

單位: bps (bit per second), 有效範圍為 100000 ~ 10000000 (100 kbps ~ 10000 kbps).

備註:

訊息資料欄位的鮑率設定值(***i_dwDataBR***)需大於或等於仲裁欄位(***i_dwArbitrBR***)的鮑率設定值

i_wArbitrBRSP

[in] 所指定的 CAN 埠 CAN 訊息鮑率取樣點設定值及 CAN FD 訊息仲裁欄位的鮑率取樣點設定值。

單位: 0.01%, 數值 7500 表示 75.00%。建議範圍: 7500 ~ 8750 (75.00% ~ 87.50%).

I_wDataBRSP

[in] 所指定的 CAN 埠 CAN FD 訊息資料欄位的鮑率取樣點設定值。

單位: 0.01%, 數值 7500 表示 75.00%。建議範圍: 7500 ~ 8750 (75.00% ~ 87.50%).

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.5.4. CANFD_GetCANADBaudRate

此函式功能在取得目前所指定的 CAN 埠上的 CAN/CAN FD 仲裁及資料欄位鮑率及取樣點設定值。

語法:

C++
<pre>int CANFD_GetCANADBaudRate(WORD i_wDevice_id, BYTE i_byCANPort, DWORD* o_dwArbitrBR, DWORD* o_dwDataBR, WORD* o_wArbitrBRSP, WORD* o_wDataBRSP);</pre>
C#
<pre>Int32 CANFD_GetCANADBaudRate(UInt16 i_wDevice_id, Byte i_byCANPort, out UInt32 o_dwArbitrBR, out UInt32 o_dwDataBR, out UInt16 o_wArbitrBRSP, out UInt16 o_wDataBRSP);</pre>

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

I_byCANPort

[in] 指定的設備 CAN 埠編號。
1: CAN1, 2: CAN2

****o_dwArbitrBR***

[out] 取得目前所指定的 CAN 埠上，CAN 訊息鮑率設定值及 CAN FD 訊息仲裁欄位鮑率設定值。單位: bps (bit per second), 數值範圍為 10000 ~ 1000000 (10 kbps ~ 1000 kbps).

****o_dwDataBR***

[out] 取得目前所指定的 CAN 埠上，CAN FD 訊息資料欄位鮑率設定值。
單位: bps (bit per second), 數值範圍為 100000 ~ 10000000 (100 kbps ~ 10000 kbps).

***o_wArbitrBRSP**

[out] 取得目前所指定的 CAN 埠上，CAN 訊息鮑率取樣點設定值及 CAN FD 訊息仲裁欄位鮑率取樣點設定值。

單位: 0.01%, 數值 7500 表示 75.00%

***o_wDataBRSP**

[out] 取得目前所指定的 CAN 埠上，CAN FD 訊息資料欄位鮑率取樣點設定值。

單位: 0.01%, 數值 7500 表示 75.00%

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.5.5. CANFD_SetCANGlobalFilter

此函式功能在指定的 CAN 埠上，設定 CAN 訊息過濾功能，包含是否要拒絕或接收標準/延伸(standard/extended)的遠端(remote)CAN 訊息。

語法:

C++
<pre>int CANFD_SetCANGlobalFilter(WORD i_wDevice_id, BYTE i_byCANPort, BYTE i_byRejectRFS, BYTE i_byRejectRFE);</pre>
C#
<pre>Int32 CANFD_SetCANGlobalFilter(UInt16 i_wDevice_id, Byte i_byCANPort, Byte i_byRejectRFS, Byte i_byRejectRFE);</pre>

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

I_byCANPort

[in] 指定的設備 CAN 埠編號。

1: CAN1, 2: CAN2

i_byRejectRFS

[in] 設定是否要拒絕接收標準(standard)的遠端 CAN 訊息。

0: 表示要接收標準的遠端 CAN 訊息。

1: 表示要拒絕接收標準的遠端 CAN 訊息。

I_byRejectRFE

[in] 設定是否要拒絕接收延伸(extended)的遠端 CAN 訊息。

0: 表示要接收延伸的遠端 CAN 訊息。

1: 表示要拒絕接收延伸的遠端 CAN 訊息。

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.5.6. CANFD_GetCANGlobalFilter

此函式功能在指定的 CAN 埠上，取得 CAN 訊息過濾功能設定，包含是否要拒絕或接收標準/延伸(standard/extended)的遠端(remote)CAN 訊息設定參數以及有效的標準/延伸 CAN 訊息 ID 過濾列表資料長度設定參數。

語法:

C++
<pre>int CANFD_GetCANGlobalFilter(WORD i_wDevice_id, BYTE i_byCANPort, BYTE *o_byRejectRFS, BYTE *o_byRejectRFE, WORD* o_wSTDFIDListSize, WORD* o_wEXTFIDListSize);</pre>
C#
<pre>Int32 CANFD_GetCANGlobalFilter(UInt16 i_wDevice_id, Byte i_byCANPort, out Byte o_byRejectRFS, out Byte o_byRejectRFE, out UInt16 o_wSTDFIDListSize, out UInt16 o_wEXTFIDListSize);</pre>

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

I_byCANPort

[in] 指定的設備 CAN 埠編號。

1: CAN1, 2: CAN2

****o_byRejectRFS***

[out] 是否要拒絕接收標準(standard)的遠端 CAN 訊息的設定值。

0: 表示要接收標準的遠端 CAN 訊息。

1: 表示要拒絕接收標準的遠端 CAN 訊息。

****o_byRejectRFE***

[out] 是否要拒絕接收延伸(extended)的遠端 CAN 訊息的設定值。

0: 表示要接收延伸的遠端 CAN 訊息。

1: 表示要拒絕接收延伸的遠端 CAN 訊息。

***o_wSTDFIDListSize**

[out] CAN 訊息 ID 過濾功能中所對應的“標準的 CAN 訊息 ID 列表”中有效的資料個數。此參數將會對應使用在 **CANFD_GetCANSTDIDFilter** 函式中所取得的 CAN 訊息 ID 範圍。

***o_wEXTFIDListSize**

[out] CAN 訊息 ID 過濾功能中所對應的“延伸的 CAN 訊息 ID 列表”中有效的資料個數。此參數將會對應使用在 **CANFD_GetCANEXTIDFilter** 函式中所取得的 CAN 訊息 ID 範圍。

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.5.7. CANFD_SetCANSTDIDFilter

此函式功能在指定的 CAN 埠上，設定 CAN 訊息過濾功能，包含設定要接收的標準(standard)的一般(normal)CAN/CAN FD 訊息 ID 範圍以及 CAN/CAN FD 訊息 ID 對應之範圍個數設定。

語法:

C++
<pre>int CANFD_SetCANSTDIDFilter(WORD i_wDevice_id, BYTE i_byCANPort, WORD i_wSTDFIDListSize, WORD* i_wSTDFID1, WORD* i_wSTDFID2);</pre>
C#
<pre>Int32 CANFD_SetCANSTDIDFilter(UInt16 i_wDevice_id, Byte i_byCANPort, UInt16 i_wSTDFIDListSize, [In, Out] UInt16[] i_wSTDFID1, [In, Out] UInt16[] i_wSTDFID2);</pre>

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

I_byCANPort

[in] 指定的設備 CAN 埠編號。

1: CAN1, 2: CAN2

i_wSTDFIDListSize

[in] 在指定的 CAN 埠上，欲設定至模組的 CAN ID 過濾器中所對應的要接收的標準(standard)的一般(normal)CAN/CAN FD 訊息 ID 範圍的設定個數。模組每個 CAN 埠最多支援 128 組 CAN/CAN FD ID 範圍設定。

**i_wSTDFID1, *i_wSTDFID2*

[in/out] 此指標指向在指定的 CAN 埠上，欲設定至模組的 CAN/CAN FD ID 過濾器中所對應的要接收的標準(standard)的一般(normal)CAN/CAN FD 訊息 ID 範圍設定陣列參數。有效的範圍設定個數取決於 *i_wSTDFIDListSize* 參數數值，最多支援 128 組 CAN/CAN FD ID 範圍設定。範圍設定參數定義如下：

Element0: CAN ID from *i_wSTDFID1*[0] to *i_wSTDFID2*[0]

Element1: CAN ID from *i_wSTDFID1*[1] to *i_wSTDFID2*[1]

...

Element127: CAN ID from i_wSTDFID1[127] to i_wSTDFID2[127]

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.5.8. CANFD_GetCANSTDIDFilter

此函式功能在指定的 CAN 埠上，取得 CAN 訊息過濾功能中要接收的標準 (standard) 的一般(normal)CAN/CAN FD 訊息 ID 範圍設定值。

語法:

C++
<pre>int CANFD_GetCANSTDIDFilter(WORD i_wDevice_id, BYTE i_byCANPort, WORD* o_wSTDFID1, WORD* o_wSTDFID2);</pre>
C#
<pre>Int32 CANFD_GetCANSTDIDFilter(UInt16 i_wDevice_id, Byte i_byCANPort, [In, Out] UInt16[] o_wSTDFID1, [In, Out] UInt16[] o_wSTDFID2);</pre>

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

I_byCANPort

[in] 指定的設備 CAN 埠編號。

1: CAN1, 2: CAN2

****o_wSTDFID1, *o_wSTDFID2***

[in/out] 此指標指向在指定的 CAN 埠上，模組 CAN ID 過濾器中所對應儲存要接收的標準(standard)的一般(normal)CAN/CAN FD 訊息 ID 範圍陣列的初始位址。回傳的有效的 CAN 訊息 ID 範圍陣列個數取決於

CANFD_GetCANGlobalFilter()函式中“o_wSTDFIDListSize”參數的回傳值。

最多支援 128 組 CAN ID 範圍設定。範圍設定參數定義如下:

Element0: CAN ID from o_wSTDFID1[0] to o_wSTDFID2[0]

Element1: CAN ID from o_wSTDFID1[1] to o_wSTDFID2[1]

...

Element127: CAN ID from o_wSTDFID1[127] to o_wSTDFID2[127]

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.5.9. CANFD_SetCANEXTIDFilter

此函式功能在指定的 CAN 埠上，設定 CAN 訊息過濾功能，包含設定要接收的延伸(extended)的一般(normal)CAN/CAN FD 訊息 ID 範圍以及 CAN/CAN FD 訊息 ID 對應之範圍個數設定。

語法:

C++
int CANFD_SetCANEXTIDFilter(WORD i_wDevice_id, BYTE i_byCANPort, WORD i_wEXTFIDListSize, DWORD* i_dwEXTFID1, DWORD* i_dwEXTFID2);
C#
Int32 CANFD_SetCANEXTIDFilter(UInt16 i_wDevice_id, Byte i_byCANPort, UInt16 i_wEXTFIDListSize, [In, Out] UInt16[] i_wEXTFID1, [In, Out] UInt16[] i_wEXTFID2);

參數:

i_wDevice_id
[in] 指定的 XV-CAN200FD 設備編號。

I_byCANPort
[in] 指定的設備 CAN 埠編號。
1: CAN1, 2: CAN2

i_wEXTFIDListSize
[in] 在指定的 CAN 埠上，欲設定至模組的 CAN ID 過濾器中所對應的要接收的延伸(extended)的一般(normal)CAN/CAN FD 訊息 ID 範圍的設定個數。模組每個 CAN 埠最多支援 64 組 CAN ID 範圍設定。

***i_wEXTFID1, *i_wEXTFID2**
[in/out] 此指標指向在指定的 CAN 埠上，欲設定至模組的 CAN ID 過濾器中所對應的要接收的延伸(extended)的一般(normal)CAN/CAN FD 訊息 ID 範圍設定陣列參數。有效的範圍設定個數取決於 **i_wEXTFIDListSize** 參數數值，最多支援 64 組 CAN ID 範圍設定。範圍設定參數定義如下:
Element0: CAN ID from i_wEXTFID1[0] to i_wEXTFID2[0]
Element1: CAN ID from i_wEXTFID1[1] to i_wEXTFID2[1]

...

Element63: CAN ID from i_wEXTFID1[63] to i_wEXTFID2[63]

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.5.10. CANFD_GetCANEXTIDFilter

此函式功能在指定的 CAN 埠上，取得 CAN 訊息過濾功能中要接收的延伸(38extended)的一般(normal)CAN/CAN FD 訊息 ID 範圍設定值。

語法:

C++

```
int CANFD_GetCANEXTIDFilter(WORD i_wDevice_id, BYTE i_byCANPort, WORD* o_wEXTFID1, WORD* o_wEXTFID2);
```

C#

```
Int32 CANFD_GetCANEXTIDFilter(UInt16 i_wDevice_id, Byte i_byCANPort, [In, Out] UInt16[] o_wEXTFID1, [In, Out] UInt16[] o_wEXTFID2);
```

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

I_byCANPort

[in] 指定的設備 CAN 埠編號。

1: CAN1, 2: CAN2

****o_wEXTFID1, *o_wEXTFID2***

[in/out] 此指標指向在指定的 CAN 埠上，模組 CAN ID 過濾器中所對應儲存要接收的延伸(extended)的一般(normal)CAN/CAN FD 訊息 ID 範圍陣列的初始位址。回傳的有效的 CAN 訊息 ID 範圍陣列個數取決於

CANFD_GetCANGlobalFilter()函式中“**o_wEXTFIDListSize**”參數的回傳值。

最多支援 64 組 CAN ID 範圍設定。範圍設定參數定義如下:

Element0: CAN ID from o_wEXTFID1[0] to o_wEXTFID2[0]

Element1: CAN ID from o_wEXTFID1[1] to o_wEXTFID2[1]

...

Element63: CAN ID from o_wEXTFID1[63] to o_wEXTFID2[63]

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.5.11. CANFD_GetCANStatus

此函式功能在於取得模組指定的 CAN 埠上，CAN 總線狀態、CAN 總線傳送/接收的錯誤發生次數、以及傳送/接收資料緩衝區是否溢位等狀態。

語法:

C++

int CANFD_GetCANStatus(WORD i_wDevice_id, BYTE i_byCANPort, DWORD *o_dwCANStatus, DWORD *o_dwErrCnt, DWORD *o_dwBufStatus);

C#

Int32 CANFD_GetCANStatus(UInt16 i_wDevice_id, Byte i_byCANPort, out UInt32 o_dwCANStatus, out UInt32 o_dwErrCnt, out UInt32 o_dwBufStatus);

參數:

i_wDevice_id
[in] 指定的 XV-CAN200FD 設備編號。

i_byCANPort
[in] 指定的設備 CAN 埠編號。
1: CAN1, 2: CAN2

***o_dwCANStatus**
[out] 指定的 CAN 埠的 CAN 總線狀態，詳細內容請參考附錄 5.3 “CAN Status” 定義。

***o_dwErrCnt**
[out] 指定的 CAN 埠的 CAN 總線傳送/接收的錯誤發生次數，詳細內容請參考附錄 5.4 “CAN Error Counter” 定義。

***o_dwBufStatus**
[out] 指定的 CAN 埠的傳送/接收資料緩衝區是否溢位等狀態。

位元	符號	數值	功能描述
0	RX		CAN1/CAN2埠接收緩衝區的狀態
		0	接收緩衝區狀態正常

		1	接收緩衝區資料溢位
1	TX		CAN1/CAN2埠傳送緩衝區的狀態
		0	傳送緩衝區狀態正常
		1	傳送緩衝區資料溢位
3:2	-	-	保留
4	EW		CAN1/CAN2埠的Error Warning狀態
		0	CAN埠Tx/Rx Error Counter小於96
		1	CAN埠Tx/Rx Error Counter大於96
5	EP		CAN1/CAN2埠的Error Passive狀態
		0	CAN埠為Error Active狀態
		1	CAN埠為Error Passive狀態
6	BO		CAN1/CAN2埠的Bus Off狀態
		0	CAN埠為非Bus Off狀態
		1	CAN埠為Bus Off狀態
31:7		-	保留

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.5.12. CANFD_SetCANReInitialize

此函式功能在指定的 CAN 埠上，將該 CAN 埠重新啟動。

語法:

C++

```
int CANFD_SetCANReInitialize(WORD i_wDevice_id, BYTE i_byCANPort);
```

C#

```
Int32 CANFD_SetCANReInitialize(UInt16 i_wDevice_id, Byte i_byCANPort;
```

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

I_byCANPort

[in] 指定的設備 CAN 埠編號。

1: CAN1, 2: CAN2

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.6. 模組通訊功能函式

該群組函式功能為透過 XV-CAN200FD 模組來傳送及接收 CAN 總線上的 CAN/CAN FD 訊息。

4.6.1. CANFD_SetCANTxMsg

此函式功能在於傳送一筆 CAN/CAN FD 訊息至所選擇的 CAN 埠的傳送緩衝區中。

語法:

C++
<pre>int CANFD_SetCANTxMsg(WORD i_wDevice_id, BYTE i_byCANPort, BYTE i_byMode, DWORD i_dwID, BYTE i_byRTR, BYTE i_byFDF, BYTE i_byDlen, BYTE *i_byData);</pre>
C#
<pre>Int32 CANFD_SetCANTxMsg(UInt16 i_wDevice_id, Byte i_byCANPort, Byte i_byMode, UInt32 i_dwID, Byte i_byRTR, Byte i_byFDF, Byte i_byDlen, [In, Out] Byte[] i_byData);</pre>

參數:

i_wDevice_id
[in] 指定的 XV-CAN200FD 設備編號。

I_byCANPort
[in] 指定的設備 CAN 埠編號。
1: CAN1, 2: CAN2

i_byMode
[in] 欲傳送的 CAN 訊息種類。
0: 2.0A, 11-bit CAN ID
1: 2.0B, 29-bit CAN ID

i_dwID
[in] 欲傳送的 CAN 訊息 ID 參數。
有效範圍:
2.0A (11-bit CAN ID) mode ➔ 0x000 ~ 0x7FF

2.0B (29-bit CAN ID) mode → 0x00000000 ~ 0x1FFFFFFF

i_byRTR

[in] 欲傳送的 CAN 訊息 RTR (Remote Transmission Request) 參數。

0: 傳送 CAN 資料訊息

1: 傳送 CAN RTR 訊息。(對於 CAN FD 訊息無效)

i_byFDL

[in] 欲傳送的 CAN/CAN FD 訊息參數。

0: 傳送一般的 CAN 訊息

1: 傳送 CAN FD 訊息

i_byDlen

[in] 欲傳送的 CAN/CAN FD 訊息資料長度參數。

有效範圍:

一般 CAN 訊息: 0x0 ~ 0x8

CAN FD 訊息: 0x0 ~ 0xF

<i>i_byDlen</i> (16 進制)	訊息資料長度 (10 進制)	<i>i_byDlen</i> (16 進制)	訊息資料長度 (10 進制)
0x0	0	0x8	8
0x1	1	0x9	12
0x2	2	0xA	16
0x3	3	0xB	20
0x4	4	0xC	24
0x5	5	0xD	32
0x6	6	0xE	48
0x7	7	0xF	64

****i_byData***

[in/out] 該指標用於取得欲傳送的 8 位元組 CAN 資料參數 或 64 位元組 CAN FD 資料參數陣列的初始位址。

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.6.2. CANFD_GetCANRxMsg

此函式功能在於由所選擇的 CAN 埠的接收緩衝區中，取得一筆接收到的 CAN/CAN FD 訊息。

語法:

C++

```
int CANFD_GetCANRxMsg(WORD i_wDevice_id, BYTE i_byCANPort, BYTE* o_byType, BYTE* o_byMode, DWORD* o_dwID, BYTE* o_byRTR, BYTE* o_byFDF, BYTE* o_byDlen, BYTE *o_byData, DWORD *o_dw_TimeStamp_s, DWORD *o_dw_TimeStamp_us);
```

C#

```
Int32 CANFD_GetCANRxMsg(UInt16 i_wDevice_id, Byte i_byCANPort, out Byte o_byType, out Byte o_byMode, out UInt32 o_dwID, out Byte o_byRTR, out Byte o_byFDF, out Byte o_byDlen, [In, Out] Byte[] o_byData, out UInt32 o_dw_TimeStamp_s, out UInt32 o_dw_TimeStamp_us);
```

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

I_byCANPort

[in] 指定的設備 CAN 埠編號。

1: CAN1, 2: CAN2

**o_byType*

[out] 接收到的訊息格式。

0: 接收到一筆 2.0A/2.0B CAN 訊息。

1: 表示接收到一筆 CAN 錯誤事件訊息。

錯誤事件訊息格式:

Mode: 1 (2.0B, 29-bit CAN ID 訊息格式)

ID: 0xEEEEEEEE

RTR: 0 (No RTR)

Dlen: 0x08

Data: D0~D3 → CAN Bus status in little-endian format

(請參考附錄 7.3 的 “CAN Status” 定義)

D4~D7 → CAN Bus error counter in little-endian format

(請參考附錄 7.4 的 “CAN Error Counter” 定義)

***o_byMode**

[out] CAN 訊息種類。

0: 2.0A, 11-bit CAN ID

1: 2.0B, 29-bit CAN ID

***o_dwID**

[out] 接收到的 CAN 訊息 ID 參數。

2.0A (11-bit CAN ID) CAN ID 訊息 → 0x000 ~ 0x7FF

2.0B (29-bit CAN ID) CAN ID 訊息 → 0x00000000 ~ 0x1FFFFFFF

CAN 錯誤事件訊息 → 0xEEEEEE

***o_byRTR**

[out] 接收到的 CAN 訊息 RTR (Remote Transmission Request) 參數。

0: 此筆訊息為 CAN 資料訊息

1: 此筆訊息為 CAN RTR 訊息。

***o_byFDF**

[out] 接收到的 CAN/CAN FD 訊息參數。

0: 一般的 CAN 訊息

1: CAN FD 訊息

***o_byDlen**

[in] 接收到的 CAN/CAN FD 訊息資料長度參數。

有效範圍:

一般 CAN 訊息: 0x0 ~ 0x8

CAN FD 訊息: 0x0 ~ 0xF

<i>o_byDlen</i> (16 進制)	訊息資料長度 (10 進制)	<i>o_byDlen</i> (16 進制)	訊息資料長度 (10 進制)
0x0	0	0x8	8

0x1	1	0x9	12
0x2	2	0xA	16
0x3	3	0xB	20
0x4	4	0xC	24
0x5	5	0xD	32
0x6	6	0xE	48
0x7	7	0xF	64

***o_byData**

[in/out] 該指標用於取得使用者定義之 64 位元組的陣列空間的初始位址，用來存放接收到的 CAN/CAN FD 資料參數。

***o_dw_TimeStamp_s**

[out] 接收到此訊息時的時間戳記
單位：秒

***o_dw_TimeStamp_us**

[out] 接收到此訊息時的時間戳記
單位：微秒

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.6.3. CANFD_SetCANHWSendMode

此函式用來啟用/停用模組在指定的 CAN 埠上，利用模組硬體時鐘，周期性的傳送 CAN/CAN FD 訊息。使用模組硬體時鐘來傳送 CAN/CAN FD 訊息會比使用電腦端的軟體時鐘還要實時。

語法:

C++
<pre>int CANFD_SetCANHWSendMode(WORD i_wDevice_id, BYTE i_byCANPort, BYTE i_byMode);</pre>
C#
<pre>Int32 CANFD_SetCANHWSendMode(UInt16 i_wDevice_id, Byte i_byCANPort, Byte i_byMode);</pre>

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

i_byCANPort

[in] 指定的設備 CAN 埠編號。

1: CAN1, 2: CAN2

i_byMode

[in] 啟用/停用模組在指定的 CAN 埠上，利用模組硬體時鐘，周期性的傳送 CAN/CAN FD 訊息

0: 停用模組硬體時鐘，周期性的傳送 CAN/CAN FD 訊息

1: 啟用模組硬體時鐘，周期性的傳送 CAN/CAN FD 訊息

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.6.4. CANFD_GetCANHWSendMode

此函式用來取得模組在指定的 CAN 埠上，利用模組硬體時鐘，周期性傳送 CAN/CANFD 訊息的啟用/停用模式。

語法:

C++
<pre>int CANFD_GetCANHWSendMode(WORD i_wDevice_id, BYTE i_byCANPort, BYTE *o_byMode);</pre>
C#
<pre>Int32 CANFD_GetCANHWSendMode(UInt16 i_wDevice_id, Byte i_byCANPort, out Byte o_byMode);</pre>

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

I_byCANPort

[in] 指定的設備 CAN 埠編號。

1: CAN1, 2: CAN2

****o_byMode***

[out] 該指標用於取得目前是停用或啟用模組硬體時鐘傳送 CAN/CAN FD 訊息。

0: 表示目前為停用模組硬體時鐘傳送 CAN/CAN FD 訊息功能。

1: 表示目前為啟用模組硬體時鐘傳送 CAN/CAN FD 訊息功能。

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.6.5. CANFD_SetCANHWSendMsg

此函式用來設定模組在指定的 CAN 埠上，利用模組硬體時鐘，周期性傳送 CAN/CANFD 訊息的資料內容。

語法:

C++
<pre>int CANFD_SetCANHWSendMsg(WORD i_wDevice_id, BYTE i_byCANPort, BYTE i_byMode, DWORD i_dwID, BYTE i_byRTR, i_byFDF, BYTE i_byDlen, BYTE *i_byData, DWORD i_dwTimer, DWORD i_dwCounter);</pre>
C#
<pre>Int32 CANFD_SetCANHWSendMsg(UInt16 i_wDevice_id, Byte i_byCANPort, Byte i_byMode, UInt32 i_dwID, Byte i_byRTR, Byte i_byFDF, Byte i_byDlen, [In, Out] Byte[] i_byData, UInt32 i_dwTimer, UInt32 i_dwCounter);</pre>

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

i_byCANPort

[in] 指定的設備 CAN 埠編號。
1: CAN1, 2: CAN2

i_byMode

[in] CAN 訊息種類。
0: 2.0A, 11-bit CAN ID
1: 2.0B, 29-bit CAN ID

i_dwID

[in] CAN 訊息 ID 參數。
有效範圍:
2.0A (11-bit CAN ID) mode → 0x000 ~ 0x7FF
2.0B (29-bit CAN ID) mode → 0x00000000 ~ 0x1FFFFFFF

i_byRTR

[in] CAN 訊息 RTR (Remote Transmission Request)參數。

- 0: 傳送 CAN 資料訊息
- 1: 傳送 CAN RTR 訊息。(對於 CAN FD 訊息無效)

i_byFDF

[in] 欲傳送的 CAN/CAN FD 訊息參數。

- 0: 傳送一般的 CAN 訊息
- 1: 傳送 CAN FD 訊息

i_byDlen

[in] 欲傳送的 CAN/CAN FD 訊息資料長度參數。

有效範圍:

一般 CAN 訊息: 0x0 ~ 0x8

CAN FD 訊息: 0x0 ~ 0xF

<i>i_byDlen</i> (16 進制)	訊息資料長度 (10 進制)	<i>i_byDlen</i> (16 進制)	訊息資料長度 (10 進制)
0x0	0	0x8	8
0x1	1	0x9	12
0x2	2	0xA	16
0x3	3	0xB	20
0x4	4	0xC	24
0x5	5	0xD	32
0x6	6	0xE	48
0x7	7	0xF	64

****i_byData***

[in/out] 該指標用於取得欲傳送的 8 位元組 CAN 資料參數 或 64 位元組 CAN FD 資料參數陣列的初始位址。

I_dwTimer

[in] 使用模組硬體時鐘傳送該筆 CAN/CAN FD 訊息的時間周期。

單位: 100 微秒

i_dwCounter

[in] 使用模組硬體時鐘傳送該筆 CAN/CAN FD 訊息的次數。

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.6.6. CANFD_GetCANRxFramePerSec

此函式功能在於取得模組在指定的 CAN 埠上，CAN 總線的訊息接收流量值。

語法:

C++
<pre>int CANFD_GetCANRxFramePerSec(WORD i_wDevice_id, BYTE i_byCANPort, WORD *o_wRxFPS);</pre>
C#
<pre>Int32 CANFD_GetCANRxFramePerSec(UInt16 i_wDevice_id, Byte i_byCANPort, out UInt16 o_wRxFPS);</pre>

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

I_byCANPort

[in] 指定的設備 CAN 埠編號。

1: CAN1, 2: CAN2

****o_wRxFPS***

[out] 取得指定的 CAN 埠每秒鐘所接收到的 CAN 總線訊息量，單位為 FPS。

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.7. 資料緩衝區功能函式

所有的 CAN 總線傳送/接收的訊息均會先存放在 CAN_FD 函式庫的軟體緩衝區上。以便使用者透過此群組功能的函式來取得。

4.7.1. CANFD_GetCANRxMsgCount

此函式功能在於取得所指定的 XV-CAN200FD 模組 CAN 埠接收緩衝區內的 CAN 訊息數量。

語法:

C++

```
int CANFD_GetCANRxMsgCount(WORD i_wDevice_id, BYTE i_byCANPort,
DWORD *o_dwCount);
```

C#

```
Int32 CANFD_GetCANRxMsgCount(UInt16 i_wDevice_id, Byte i_byCANPort, out
UInt32 o_dwCount);
```

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

I_byCANPort

[in] 指定的設備 CAN 埠編號。

1: CAN1, 2: CAN2

****o_dwCount***

[out] 取得指定的 CAN 埠接收緩衝區內的 CAN/CAN FD 訊息數量。

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.7.2. CANFD_ClearCANRxBuf

此函式功能在於清除模組在指定的 CAN 埠上，接收緩衝區內的所有 CAN/CAN FD 訊息。

語法:

C++

```
int CANFD_ClearCANRxBuf(WORD i_wDevice_id, BYTE i_byCANPort);
```

C#

```
Int32 CANFD_ClearCANRxBuf(UInt16 i_wDevice_id, Byte i_byCANPort);
```

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

I_byCANPort

[in] 指定的設備 CAN 埠編號。

1: CAN1, 2: CAN2

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.7.3. CANFD_ClearCANTxBuf

此函式功能在於清除模組在指定的 CAN 埠上，傳送緩衝區內的所有 CAN/CAN FD 訊息。

語法:

C++

```
int CANFD_ClearCANTxBuf(WORD i_wDevice_id, BYTE i_byCANPort);
```

C#

```
Int32 CANFD_ClearCANTxBuf(UInt16 i_wDevice_id, Byte i_byCANPort);
```

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

i_byCANPort

[in] 指定的設備 CAN 埠編號。

1: CAN1, 2: CAN2

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.8. 其他功能函式

該群組函式功能為取得 CAN_FD 函式庫資訊或提供使用者撰寫程式上的協助。

4.8.1. CANFD_GetDllVersion

此函式用來取得 CAN_FD 函式庫的版本。

語法:

C++

DWORD CANFD_GetDllVersion(void);

C#

UInt32 CANFD_GetDllVersion();

參數:

None.

回傳值:

回傳 CAN_FD 函式庫的版本。

數值 1000000 (十進制) ➔ 表示 CAN_FD 函式庫的版本為 v1.0.0.0

數值 1000113 (十進制) ➔ 表示 CAN_FD 函式庫的版本為 v1.0.1.13

4.8.2. CANFD_GetFwVer

此函式用來取得 XV-CAN200FD 模組的韌體版本。

語法:

C++
Int CANFD_GetFwVer(WORD i_wDevice_id, WORD* o_wFwVer);
C#
Int32 CANFD_GetFwVer(UInt16 i_wDevice_id, out UInt16 o_wFwVer);

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

****o_wFwVer***

[out] 模組韌體版本。

數值 100 (十進制) ➔ 韌體版本: v1.00

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.8.3. CANFD_SetSN

此函式用來取得 XV-CAN200FD 模組的 BID (board ID)。

語法:

C++
Int CANFD_SetSN(WORD i_wDevice_id, DWORD i_dwSN)
C#
Int32 CANFD_SetSN (UInt16 i_wDevice_id, UInt32 i_dwSN);

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

I_dwSN

[in] XV-CAN200FD 設備的 BID (board ID)。

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.8.4. CANFD_ResetModule

此函式功能在於重置 XV-CAN200FD 模組。

語法:

C++
Int CANFD_ResetModule(WORD i_wDevice_id);
C#
Int32 CANFD_ResetModule(UInt16 i_wDevice_id);

參數:

i_wDevice_id

[in] 指定的 XV-CAN200FD 設備編號。

回傳值:

回傳值為 0 時，代表成功；為其它值時，代表失敗。

4.9. 函式回傳碼

顯示在執行 CAN_FD 函式庫各函式的回傳代碼，可依下表得知各錯誤代碼意義。

回傳碼 (16 進制)	描述
0x0	No error
0x1	OP field of the configuration command error
0x2	FC field of the configuration command error
0x3	DL field of the configuration command error
0x4	Fail to write data into device
0x10001	Invalid device
0x10002	Device already in used
0x10003	Device not exist
0x10004	Get device information error
0x10005	Invalid USB package size
0x10006	Write file fail
0x10007	USB Tx buffer overflow
0x1000A	Exceed maximum supported USB device
0x1000B	USB device not open
0x10100	Communication timeout
0x10101	Invalid CAN port number
0x10102	No data in CAN received buffer
0x10103	CAN transmitted buffer overflow
0x10104	CAN bit rate parameters not support
0x10105	CAN Filter ID list size parameter not support

5. 附錄

5.1. 文件版本歷史紀錄

本章節描述此文件的版本修改紀錄。下表為版本歷史紀錄。

版本	日期	描述
1.0.0	2025/05	初版
1.0.1	2026/02	支援 LP-2841M

5.2. CAN Status 暫存器

位元	符號	數值	描述
2:0	LEC		Last error code These bits indicate the type of the last error to occur on the CAN bus. This bit field will be cleared when a message has been transferred without error. The bits in this bit field will be set upon a read access.
		0x0	No error.
		0x1	Stuff error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed.
		0x2	Form error: A fixed format part of a received frame has the wrong format.
		0x3	Ack Error: The message transmitted by the M_CAN was not acknowledged by another node.
		0x4	Bit1Error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value 1), but the monitored bus value was dominant.
		0x5	Bit0Error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (data or identifier bit logical value 0), but the monitored bus value was recessive. During Bus Off recovery this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the Bus Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed).
		0x6	CRC Error: The CRC check sum of a received message was incorrect. The CRC of an incoming message does not match with the CRC calculated from the received data.
		0x7	Unused: No CAN Bus event was detected
4:3	ACT		Activity. This register monitors the CAN communication state.
		0x0	Synchronizing – node is synchronizing on CAN communication.
		0x1	Idle – node is neither receiver nor transmitter.
		0x2	Receiver – node is operating as receiver
		0x3	Transmitter – node is operating as transmitter.
5	EP		Error passive
		0	The CAN controller is in the error active state.
		1	The CAN controller is in the error passive state as defined in the CAN 2.0 specification.
6	EW		Warning status
		0	Both error counters are below the Error Warning limit of 96
		1	At least one of error counter has reached the Error Warning limit of 96
7	BOFF		Bus off status
		0	The CAN module is not in bus off state.
		1	The CAN controller is in bus off state.
31:8	-	-	Reserved

5.3. CAN Error Counter 暫存器

位元	符號	數值	描述
7:0	TEC		Transmit error counter Current value of the transmit error counter (maximum value 255)
14:8	REC		Receive error counter Current value of the receive error counter (maximum value 127).
15	RP		Receive error passive
		0	Below error level. The receive counter is below the error passive level of 128
		1	At error level. The receive counter has reached the error passive level of 128
31:16	-	-	Reserved

5.4. 支援的 CAN FD 資料欄位鮑率

	支援的資料欄位鮑率 (kbps)				
項次	0	1	2	3	4
0	10000.000	8571.429	7500.000	6666.667	6000.000
5	5454.545	5000.000	4615.385	4285.714	4000.000
10	3750.000	3529.412	3333.333	3157.895	3000.000
15	2857.143	2727.273	2608.696	2500.000	2400.000
20	2307.692	2222.222	2142.857	2068.966	2000.000
25	1935.484	1875.000	1818.182	1764.706	1714.286
30	1666.667	1621.622	1578.947	1538.462	1500.000
35	1463.415	1428.571	1395.349	1363.636	1333.333
40	1304.348	1276.596	1250.000	1224.49	1200.000
45	1176.471	1153.846	1132.075	1111.111	1090.909
50	1071.429	1052.632	1034.483	1016.949	1000.000
55	983.6066	967.7419	952.381	937.500	923.0769
60	909.0909	895.5224	882.3529	869.5652	857.1429
65	845.0704	833.3333	821.9178	810.8108	800.000
70	789.4737	779.2208	769.2308	759.4937	750.000
75	740.7407	731.7073	722.8916	714.2857	705.8824
80	697.6744	689.6552	681.8182	674.1573	666.6667
85	659.3407	652.1739	645.1613	638.2979	631.5789
90	625.000	618.5567	612.2449	606.0606	600.000
95	594.0594	588.2353	582.5243	576.9231	571.4286
100	566.0377	560.7477	555.5556	550.4587	545.4545
105	540.5405	535.7143	530.9735	526.3158	521.7391
110	517.2414	512.8205	508.4746	504.2017	500.000
115	495.8678	491.8033	487.8049	483.871	480.000
120	476.1905	472.4409	468.750	465.1163	461.5385
125	458.0153	454.5455	451.1278	447.7612	444.4444
130	441.1765	437.9562	434.7826	431.6547	428.5714
135	425.5319	422.5352	419.5804	416.6667	413.7931

140	410.9589	408.1633	405.4054	402.6846	400.000
145	397.351	394.7368	392.1569	389.6104	387.0968
150	384.6154	382.1656	379.7468	377.3585	375.000
155	372.6708	370.3704	368.0982	365.8537	363.6364
160	361.4458	359.2814	357.1429	355.0296	352.9412
165	350.8772	348.8372	346.8208	344.8276	342.8571
170	340.9091	338.9831	337.0787	335.1955	333.3333
175	331.4917	329.6703	327.8689	326.087	324.3243
180	322.5806	320.8556	319.1489	317.4603	315.7895
185	314.1361	312.500	310.8808	309.2784	307.6923
190	306.1224	304.5685	303.0303	301.5075	300.000
195	298.5075	297.0297	295.5665	294.1176	292.6829
200	291.2621	289.8551	288.4615	287.0813	285.7143
205	284.3602	283.0189	281.6901	280.3738	279.0698
210	277.7778	276.4977	275.2294	273.9726	272.7273
215	271.4932	270.2703	269.0583	267.8571	266.6667
220	265.4867	264.3172	263.1579	262.0087	260.8696
225	259.7403	258.6207	257.5107	256.4103	255.3191
230	254.2373	253.1646	252.1008	251.046	250.000
235	248.9627	247.9339	246.9136	245.9016	244.898
240	243.9024	242.915	241.9355	240.9639	240.000
245 ~ 290	...				
290	202.7027	202.0202	201.3423	200.6689	200.000
295 ~ 365	...				
365	161.7251	161.2903	160.8579	160.4278	160.000
370~390	...				
390	151.5152	151.1335	150.7538	150.3759	150.000
395~470	...				
470	126.0504	125.7862	125.523	125.261	125.000
475 ~ 490	...				
490	120.9677	120.7243	120.4819	120.2405	120.000
495 ~ 590	...				
590	100.6711	100.5025	100.3344	100.1669	100.000