

# MPC5777C

## CAN

### 簡易操作手冊

文件標識	MPC5777C_CAN		
當前版本	V1.0	日期	2019-09-17
作者	Michael Gao	聯繫方式	Michael.Gao@wpi-group.com

## 版本歷史

版本	日期	描述	作者
V1.0	2019.09.17	建立文檔	Michael.Gao

# 目錄

## 目錄

1. 概述 .....	6
1.1 功能描述 .....	6
1.2 硬體開發工具 .....	6
1.3 軟體開發工具 .....	6
1.4 參考相關程式 .....	6
2. SDK 的說明與使用 .....	7
2.1 概述 .....	7
2.1.1 SDK 版本說明 .....	7
2.1.2 S32 Design Studio IDE 的使用方式 .....	7
2.1.3 SDK 的取得與安裝方式 .....	7
2.1.4 安裝後 SDK 相關檔案 .....	8
2.2 啟用 Processor Expert 介面檢視 .....	8
2.3 Component Library 頁面 .....	9
2.4 Components 頁面 .....	9
2.5 Component Inspector 頁面 .....	11
3. 原理 .....	13
2.1 Can-bus 車載網路系統架構 .....	13
2.2 Can bus 匯流排設計(抗干擾) .....	14
2.3 Can bus 資料通訊格式 .....	15
SOF 欄位 .....	15
ID 欄位 .....	15
RTR 欄位 .....	15
IDE 欄位 .....	15

R0 欄位 .....	15
DLC 欄位 .....	16
Data 欄位 .....	16
CRC 欄位 .....	16
ACK 欄位: .....	16
EOF 欄位: .....	16
Can bus 填塞規則 .....	17
Can bus 錯誤處理 .....	17
2.4 ISO 實體層介紹 .....	19
高速 (High-Speed) CAN ( 高速匯流排 ) .....	19
低速/容錯 (Low-Speed/Fault-Tolerant) CAN ( 中速匯流排 ) .....	20
單線式 (Single-Wire) CAN ( 低速匯流排 ) .....	21
2.5 CAN-FD 與 CAN 功能比較 .....	21
CAN-FD 優點 .....	21
CAN_FD 資料通訊格式 .....	21
4. 程式碼功能描述 .....	22
3.1 功能概述 .....	22
3.2 程式流程圖 .....	22
5. 硬體接線圖 .....	23
5.1 MPC5777C EVB 發展板內部接線圖 .....	23
5.2 外部接線圖 .....	25
6. 相關 function 介紹 .....	26
6.1 FLEXCAN_DRV_Init .....	26
6.2 FLEXCAN_DRV_ConfigRxMb .....	27
6.3 FLEXCAN_DRV_ConfigTxMb .....	27
6.4 FLEXCAN_DRV_Send .....	27
6.5 FLEXCAN_DRV_Receive .....	27

6.6 FLEXCAN_DRV_GetTransferStatus .....	27
6.7 其他 function .....	27
CAN 實際操作 ( 基於 MPC5777C ).....	27
CAN-Bus 初始化.....	錯誤! 尚未定義書籤。
CAN-Bus clock 來源 .....	錯誤! 尚未定義書籤。
CAN-Bus bit/rate 、Resync jump width 與 Sampling point 設置 .....	錯誤! 尚未定義書籤。
CAN-Bus MBs、RxFIFO 與 filter 相關設置.....	錯誤! 尚未定義書籤。
Can-bus mailbox、ID、data 與資料格式設定 .....	錯誤! 尚未定義書籤。
Can-bus send( TX )資料格式設定 .....	錯誤! 尚未定義書籤。
Can-bus receive( RX )資料格式設定.....	錯誤! 尚未定義書籤。
Pin 腳設定.....	錯誤! 尚未定義書籤。
接收端 ( CAN converter).....	錯誤! 尚未定義書籤。
執行結果 .....	錯誤! 尚未定義書籤。

# 1. 概述

## 1.1 功能描述

在 MPC5777C EVB 板子上面實現 CAN 通訊的發送與接收功能，通過 CAN 的 CONVERTER 連接電腦端與 MPC5777C 板子通訊。

## 1.2 硬體開發工具

MPC5777C EVB 板

CAN to USB converter ( i-7565-H2 )

PEmicro 燒錄器

## 1.3 軟體開發工具

S32 Design Studio IDE for Power Architecture , 版本 2017.R1 , Build id: 171018

S32 SDK PA , BETA 2.9.0

I-7565-H1H2\_Utility\_v1.17

## 1.4 參考相關程式

Example codes / MPC5777C\_CAN

欲了解更多資料請聯絡世平集團 Kevin ( [Kevin.Hong@wpi-group.com](mailto:Kevin.Hong@wpi-group.com) )

Michael ( [Michael.Gao@wpi-group.com](mailto:Michael.Gao@wpi-group.com) )

## 2. SDK 的說明與使用

### 2.1 概述

S32 SDK PA 可以在 S32 Design Studio IDE 開發環境中使用 Processor Expert 工具配置使用者需要的設置，並自動生成設置檔案，提供快速便捷的 Device 與 IP 設置方式。

#### 2.1.1 SDK 版本說明

SDK 的版本分為以下三種：

EAR : Early Assess Release，早期評估版，是 SDK 最先發布的版本，提供基本的 SDK 功能，可以做為早期軟體評估使用。

BETA : Beta 版是修復 EAR 版若干 Bug 後發布的 SDK 版本，相較於 EAR 版本，功能更加完善。

RTM : Ready To Manufacture，最終量產版本，提供最完善的 API 接口及外設 IP 配置選項，修復

#### 2.1.2 S32 Design Studio IDE 的使用方式

關於 S32 Design Studio 的使用方式，請參考文件“S32 Design Studio 使用教學”。

#### 2.1.3 SDK 的取得與安裝方式

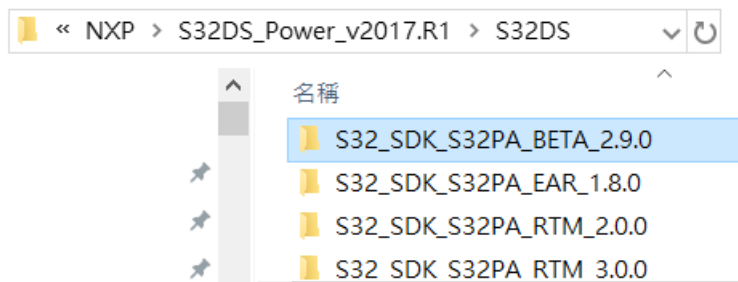
關於 SDK 的取得與安裝方式，請參考文件“MPC57xxx IDE & Tools 下載與安裝教

學”。

## 2.1.4 安裝後 SDK 相關檔案

S32 SDK 預設安裝路徑在 S32 Design Studio IDE 資料夾下，可以同時存在不同的版本。安裝後路徑如下：

C:\NXP\S32DS\_Power\_v2017.R1\S32DS



選取本範例使用的 BETA 2.9.0 版本進入後，可以找到 SDK 的使用文件

Release Note :

C:\NXP\S32DS\_Power\_v2017.R1\S32\_SDK\_S32PA\_BETA\_2.9.0\  
S32\_SDK\_for\_Power\_Architecture\_BETA\_2\_9\_0\_Release\_Notes

網頁版的 User Manual :

..\S32\_SDK\_S32PA\_BETA\_2.9.0\doc\Start\_here.html

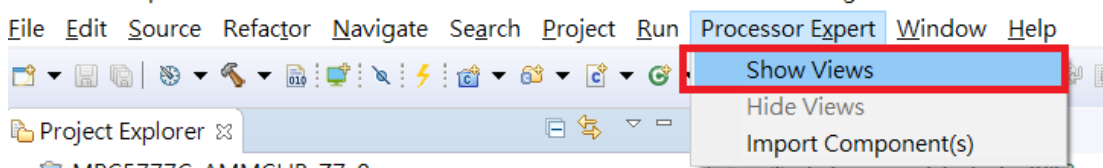
PDF 版的 User Manual :

..\S32\_SDK\_S32PA\_BETA\_2.9.0\doc\S32SDK\_MPC5777C\_UserManual.pdf

## 2.2 啟用 Processor Expert 介面檢視

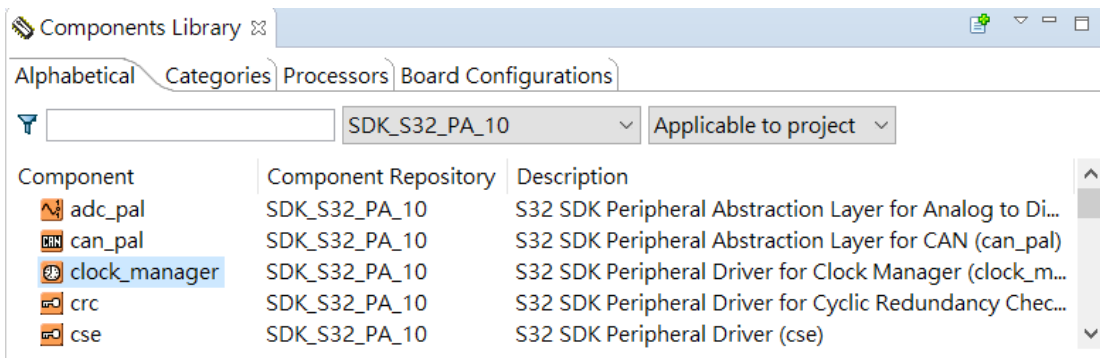
預設的工作站可能沒有開起 Processor Expert 介面。

要開啟介面，請在工作列中選擇 Processor Expert -> Show View





## 2.3 Component Library 頁面



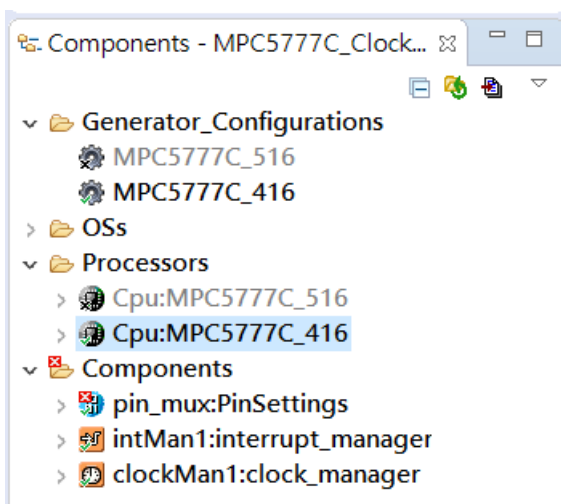
上圖為 Component Library 頁面，內容如下：

Alphabetical：依字母序排列的 SDK 元件列表

Categories：依分類顯示的 SDK 元件列表

Processor：當前專案可使用的處理器配置列表

## 2.4 Components 頁面



上圖為 Components 頁面，包含了目前已加入專案的設置與 SDK 元件。其中：

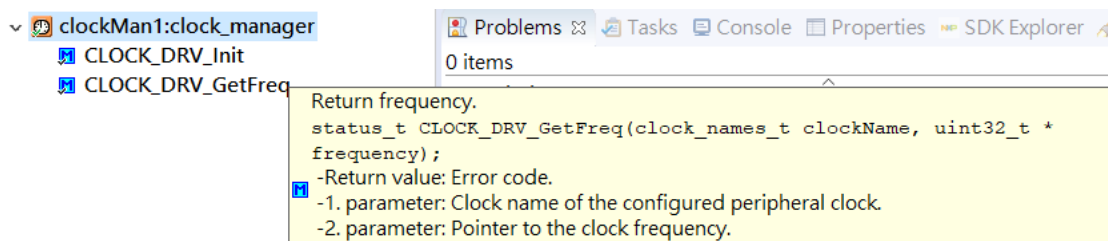
右上角的 Reload Project 按鈕可以將儲存的設置重新載入 Processor Expert

右上角的 Generate Processor Expert Code 按鈕可以將當前的 SDK 設置輸出至

## Project 中

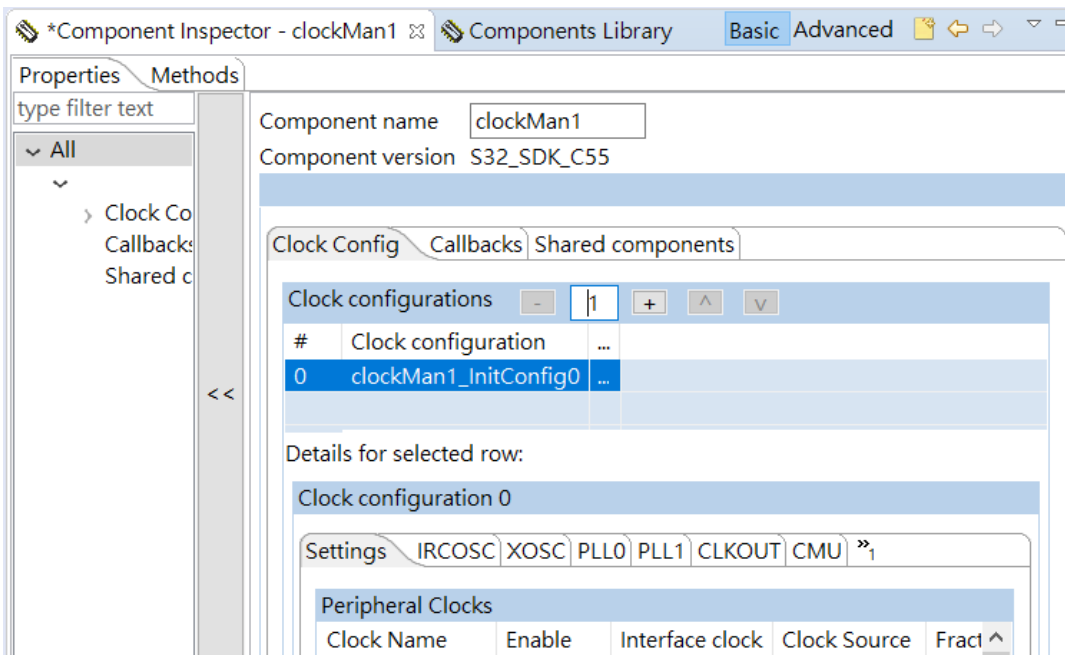
- ❁ 元件或設置圖示，左下角處的綠色“v”代表當前使用中的元件或配置。
- ❁ 元件或設置圖示，左下角處的黑色“x”代表當前該元件或設置不被使用。
- ❁ 元件或設置圖示，左上角處的紅色“x”代表目前的配置內容有錯誤需要修正。
  - › 元件或設置圖是左方的箭頭，單擊可以展開該元件一般使用的 API 選單。

滑鼠停留可以看到 API 所需的輸入參數



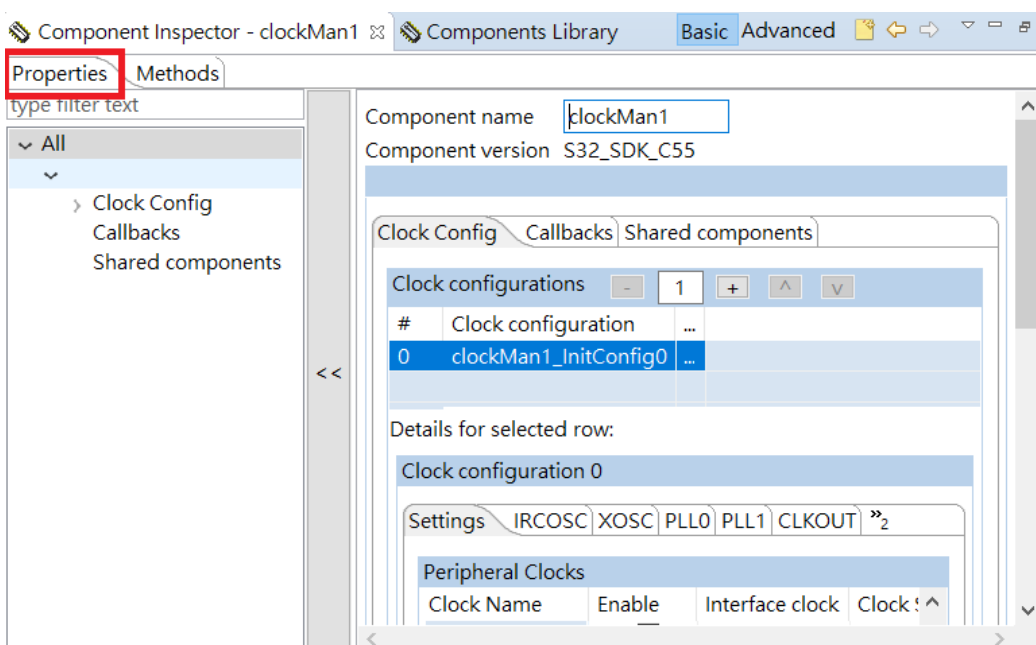
- › 此處顯示的 API 可以直接以滑鼠拖曳到程式當中，不須使用者定義的參數會自動填入，需使用者定義的參數會留白。

## 2.5 Component Inspector 頁面

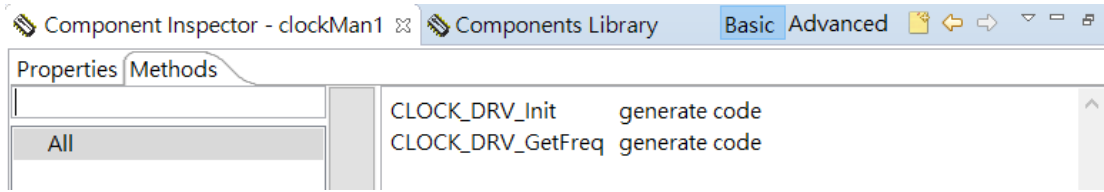


Component Inspector 頁面顯示的是目前在 Components 頁面中選取的項目設置，內容會隨選取的項目不同而切換。上方圖例中，選擇的是命名為 clockMan1 的 Clock\_Manager 元件。

- Properties 分頁顯示的是 SDK Component 的設置介面



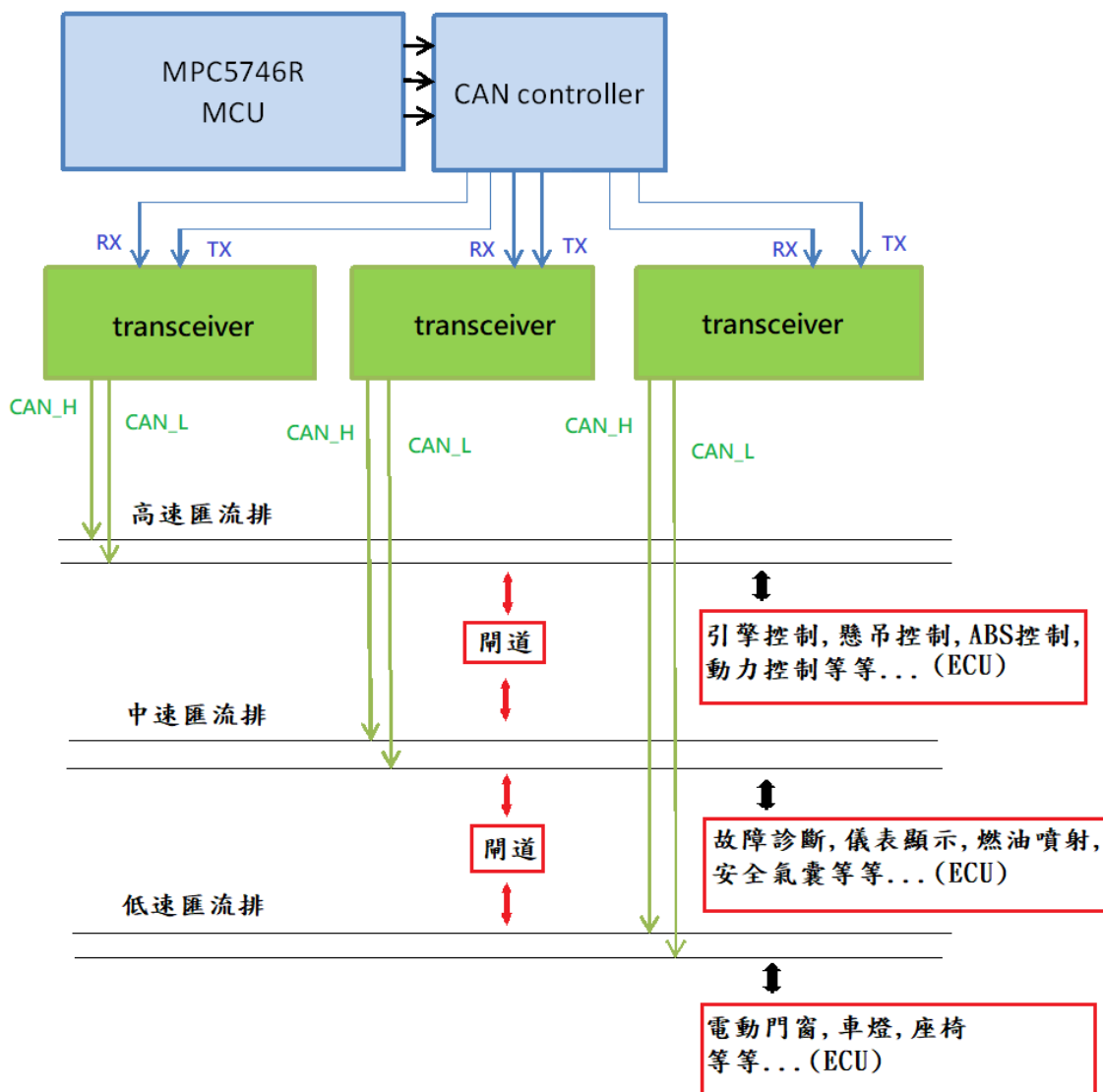
- Methods 頁面顯示的是該元件一般狀況下使用的 API 列表



### 3. 原理

#### 3.1 Can-bus 車載網路系統架構

依功能傳送速率需求來區分為高中低速的匯流排(如圖一)

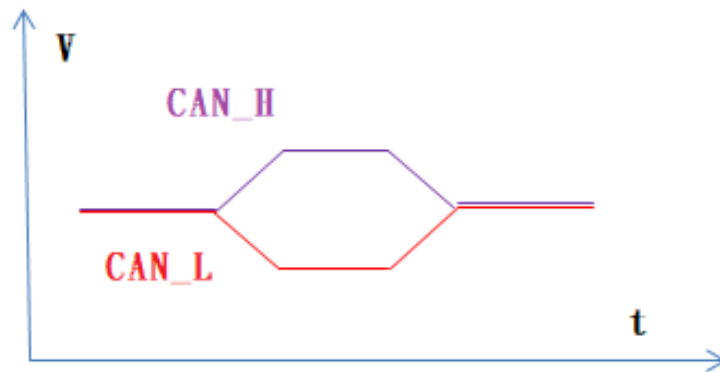


(圖一)

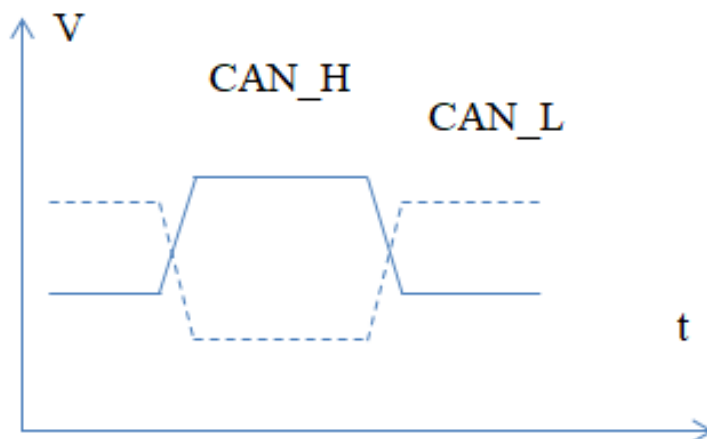
### 3.2 Can bus 匯流排設計(抗干擾)

Can bus 匯流排由 CAN\_L, CAN\_H 所實現, 當兩條訊號電位差小的時候為 1 , 電位差大的時候為 0 , 如圖二所示(高速 Can 訊號)、圖三所示(低速 Can 訊號)

匯流排兩條線路受到干擾相同, 利用電位差輸入可以大幅減少雜訊干擾影響



(圖二)



(圖三)

### 3.3 Can bus 資料通訊格式

Can bus 資料通訊格式如下列所示

SOF	ID	RTR	IDE	Ro/FDF	DLC	Data	CRC	ACK	EOF
1bit	11bit	1bit	1bit	1bit	4bit	0~64bit	16bit	1bit	7bit

#### SOF 欄位

SOF 輸出 1bit dominant(0)，用來同步，代表 CAN frame 開始

#### ID 欄位

表示每個 ECU (節點)優先權，ID 值越小代表優先權越高(11bit)

#### RTR 欄位

RTR 欄位資料通訊框架時為 (0);遠端資料框架時為 (1)(1bit)

#### IDE 欄位

IDE 欄位標準格式為 (0);延伸格式為(1)(1bit)

#### R0 欄位

r0 欄位為預留位通常為 (0)，但是(1)同樣可以接受(1bit)

## DLC 欄位

設定資料長度的程式碼欄位(4bit)(註：資料長度 range 為 0~8byte)

## Data 欄位

傳輸資料欄位(0~64bit)，長度資料由 DLC 指定(0~8byte)

## CRC 欄位

Cyclic redundancy check 用來檢測或校驗資料傳輸可能出現的錯誤

原始資料 = Q 資料 \* K 資料 - R 資料

所以如果原始資料 + R 資料能被 K 資料整除代表資料正確。

最後輸入 CRC Delimiter (1)

## ACK 欄位:

ACK 欄位分為 SKOT 與 Delimiter

ACK SLOT 欄位：發送器傳送端送出 1bit 的(1)訊號，接收器送出 1bit 的(0)訊號告知傳送端

ACK Delimiter：欄位為 1bit 的(1)

## EOF 欄位:

結束位輸出 7bit 的(1)，代表 CAN 資料傳輸結束



## Can bus 填塞規則

傳輸器會在連續 5 個相同的 bit 資料時會填塞一個相反的 bit，接收時接收器會去除填塞的資料(例如：想要傳輸 11111 資料時，傳輸器會傳輸 1111101)，反之如果接收到連續 6 個以上相同訊號代表資料傳輸錯誤。

## Can bus 錯誤處理

### 錯誤狀態

#### 1. 主動錯誤狀態

節點處於正常狀態時，發生 bus error，產生 6 bits 的 error flag (0)，其餘節點也會產生 error flag 延遲 0~6 bits，再產生 8 bits error Delimiter 與 3 bits interframe，資料重傳。

#### 2. 被動錯誤狀態

可接收資料但接收時不能傳送錯誤通知，發生 bus error，產生 6 bits 的 error flag (0)，其餘節點也會產生 error flag 延遲 0~6 bits，再產生 8 bits error Delimiter 與 3 bits interframe，資料重傳。

#### 3. 匯流排關閉狀態

當發送錯誤計數器超過 255 的上限時，匯流排狀態會關閉 (1)，並向 CAN 控制器請求 reset 產生錯誤中斷。

## 錯誤種類

### 1. 位元填塞錯誤(bit-stuffing error)

如果在 SOF 與 CRC 之間有連續 6 個相同訊號(0)或(1)，產生填塞錯誤，在下一個位元，送出 error frame。

### 2. 位元錯誤(bit error)

匯流排傳送(0)但偵測到(1)；或傳送(1)但偵測到(0)，在下一個位元，送出 error frame。

### 3. CRC 錯誤(CRC error)

當節點接收到 CRC 的資料後，計算是否資料錯誤，如果偵測到 CRC 錯誤，在 ACK 之後，送出 error frame。

### 4. 確認 ACK 錯誤(ACK error)

傳送端至少偵測到一個節點在 ACK Slot 送出 1bit 的(0)，代表正確接收，否則在 ACK delimiter 之後送出 error frame。

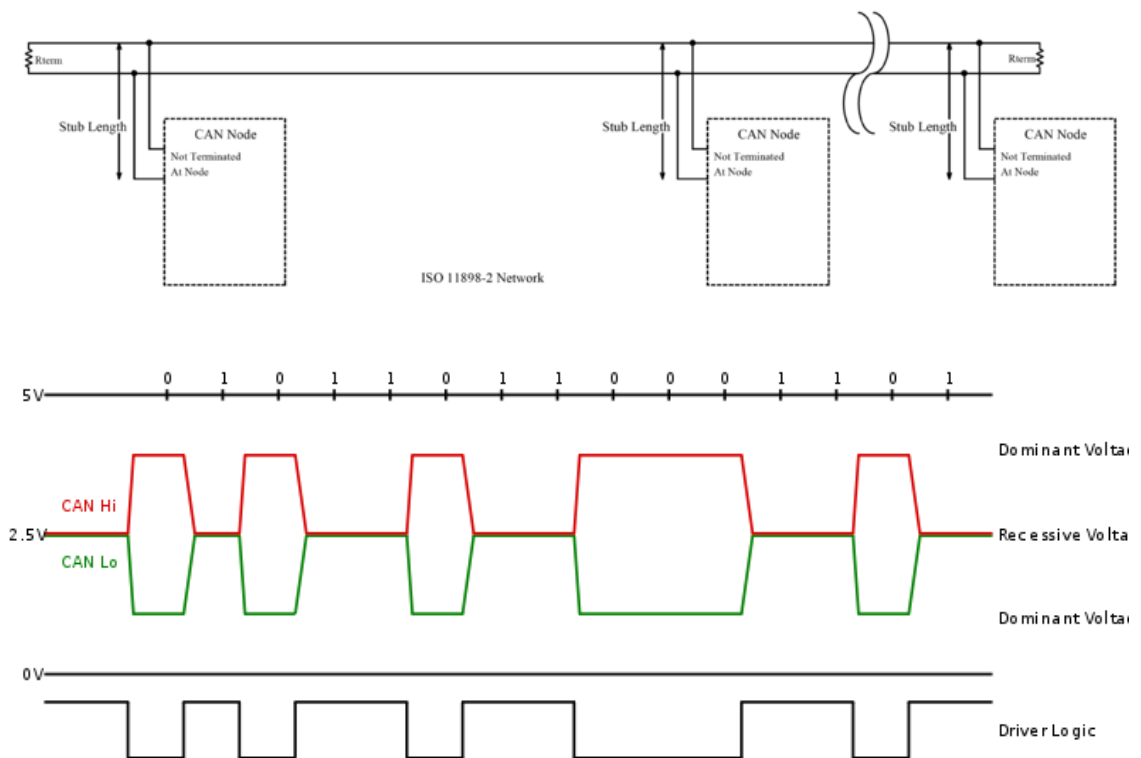
### 5. 格式錯誤(form error)

CRC delimiter(1 bit)、ACK Delimiter(1 bit)、EOF(7 bits)應該都是(1)，如果不是，在下一個位元，送出 error frame。

### 3.4 ISO 實體層介紹

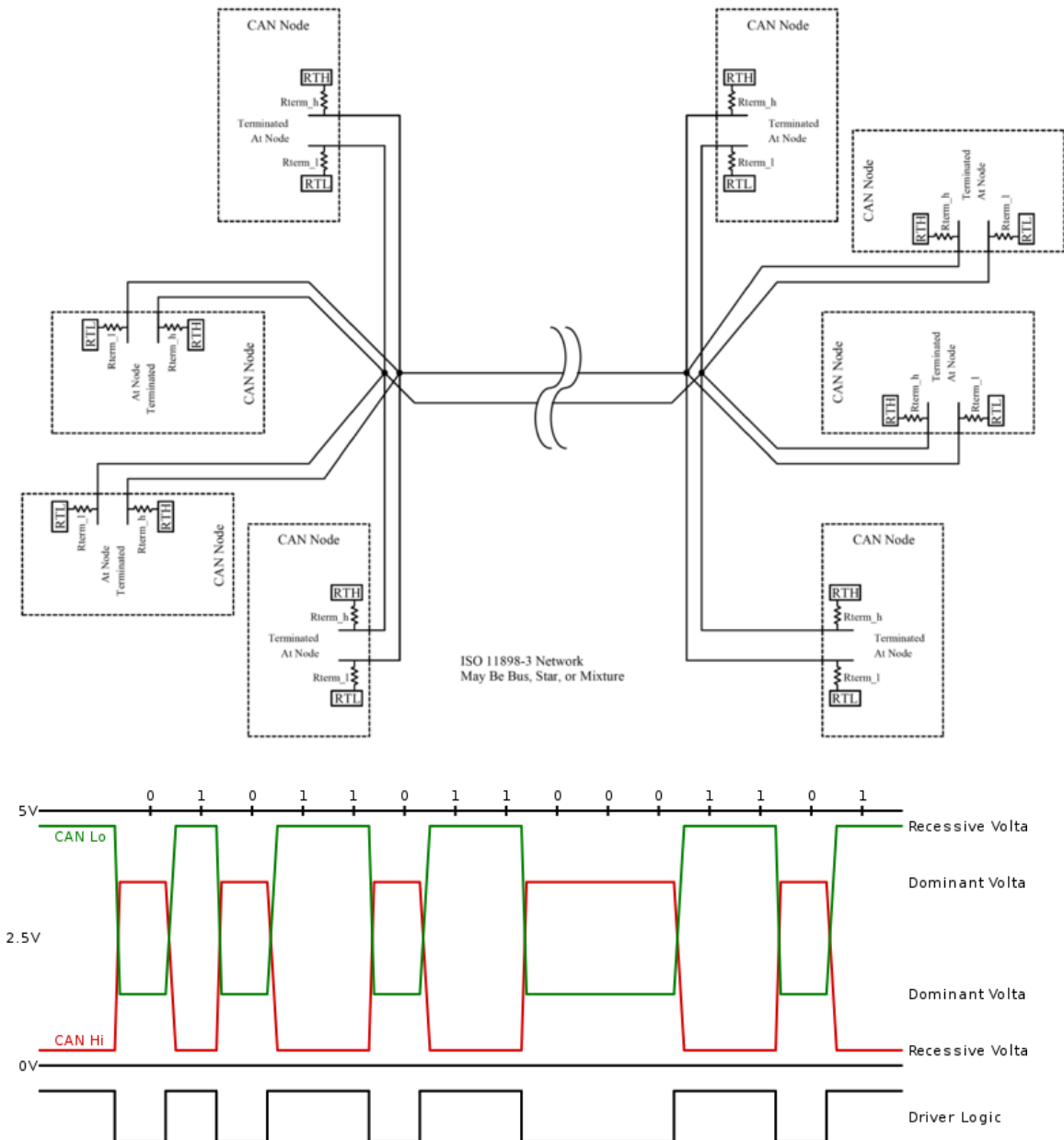
#### 高速 (High-Speed) CAN ( 高速匯流排 )

高速 CAN 可稱為 CAN C 或 ISO 11899-2，匯流排 2 端皆接有 120Ω 電阻，最高可進行 1M bit/s 的通訊傳輸率。



## 低速/容錯 (Low-Speed/Fault-Tolerant) CAN (中速匯流排)

低速/容錯 CAN 可稱為 CAN B 或 ISO 11898-3，最高可高達 125 K bit/s 的通訊傳輸率。



## 單線式 (Single-Wire) CAN (低速匯流排)

單線式可稱為 CAN A 或 SAE-J2411，通訊速率最高可達 33.3 K bit/s

### 3.5 CAN-FD 與 CAN 功能比較

#### CAN-FD 優點

1. 擁有較高的位速率，延遲時間較短
2. 可傳輸更高的數據量
3. 使用更高性能的 CRC 算法，降低未檢測到錯誤的機率

#### CAN\_FD 資料通訊格式

SOF	ID	RRS	IDE	FDF	res	BRS	ESI	DLC	Data	CRC	ACK	EOF
1 bit	11 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	4 bit	0~64 byte	21 bit	1 bit	7 bit

下圖為 CAN-BUS 資料通訊格式

SOF	ID	RTR	IDE	Ro/FDF	DLC	Data	CRC	ACK	EOF
1bit	11bit	1bit	1bit	1bit	4bit	0~64bit	16bit	1bit	7bit

其中 RRS 為遠程請求替換位、CAN-Bus 中 Ro (保留位) 更改為 FDF (靈活數據格式位) 和 res (保留位)、BRS (位速率轉換) 可開啟較高位速率，直到 CRC 結束、ESI (錯誤狀態指示器) 當存在重要通訊問題，訊號為 ( 0 ) 根據汽車製造商需求制訂。

## 4. 程式碼功能描述

### 4.1 功能概述

1. 發送程序：使用 CAN\_3 通道實現發送 CAN 資料，ID 為 0x02，發送 data(0x52)  
使用 I-7565-H1H2\_Utility\_v1.17 介面發送 CAN 資料，ID 為 0x05，發送 data(0x52)
2. 接收程序：使用 CAN\_3 通道實現接收 CAN 資料，ID 設定為 0x02。  
使用 I-7565-H1H2\_Utility\_v1.17 介面接收 CAN 資料

### 4.2 程式流程圖

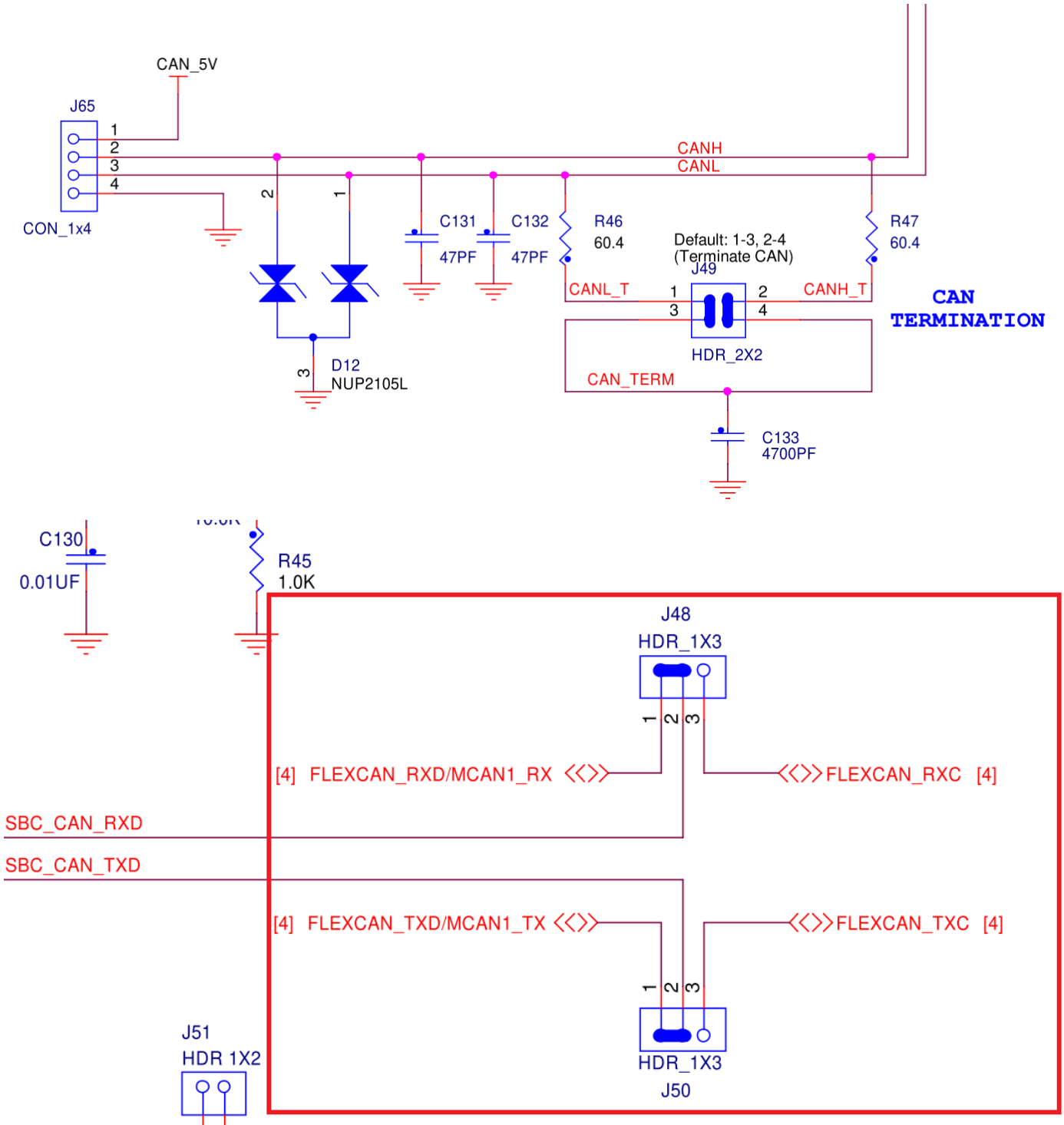
欲了解更多資料請聯絡世平集團 Kevin ( [Kevin.Hong@wpi-group.com](mailto:Kevin.Hong@wpi-group.com) )  
Michael ( [Michael.Gao@wpi-group.com](mailto:Michael.Gao@wpi-group.com) )



此為上圖部分放大圖，CAN 輸出與輸入

MPC5777C 有 2 顆 CAN ic 分別為 TJA1145T 與 MC33FS6520LAE，範例使用

MC33FS6520LAE





## 5.2 外部接線圖

Can-converter 與 MPC5777C\_PVB 外部接線圖

欲了解更多資料請聯絡世平集團 Kevin ( [Kevin.Hong@wpi-group.com](mailto:Kevin.Hong@wpi-group.com) )  
Michael ( [Michael.Gao@wpi-group.com](mailto:Michael.Gao@wpi-group.com) )

## 6. 相關 function 介紹

### 6.1 FLEXCAN\_DRV\_Init

#### 6.1.1 功能描述

選擇 CAN 時鐘來源與頻率

Bit/rate 設置

Rx\_fifo 濾波器設置

Sampling point 設置

#### 6.1.2 函式原型

```
FLEXCAN_DRV_Init(uint8_t instance, flexcan_state_t *state, const flexcan_user_config_t *data);
```

#### 6.1.3 參數

參數名稱	類型	描述
instance	uint8_t	指定 CAN 函式模塊
state	flexcan_state_t	CAN 轉換狀態 (interrupt or DMA)
data	const flexcan_user_config_t *	CAN 使用者參數設定

#### 6.1.4 return 值

無

#### 6.1.5 範例

```
FLEXCAN_DRV_Init(INST_CANCOM1, &canCom1_State, &canCom1_InitConfig0);
```

## 6.2 FLEXCAN\_DRV\_ConfigRxMb

## 6.3 FLEXCAN\_DRV\_ConfigTxMb

## 6.4 FLEXCAN\_DRV\_Send

## 6.5 FLEXCAN\_DRV\_Receive

## 6.6 FLEXCAN\_DRV\_GetTransferStatus

## 6.7 其他 function

## 7. CAN 實際操作 ( 基於 MPC5777C )

欲了解更多資料請聯絡世平集團 Kevin ( [Kevin.Hong@wpi-group.com](mailto:Kevin.Hong@wpi-group.com) )  
Michael ( [Michael.Gao@wpi-group.com](mailto:Michael.Gao@wpi-group.com) )