



HY11P 系列

C 函式庫用戶手冊

Table of Contents

1. 導讀	6
1.1. C 函式庫簡介	6
1.2. 相關文檔	6
2. MCU系統控制	7
2.1. 函數簡介	7
2.2. 函數說明	7
3. 晶片時鐘源CLOCK	11
3.1. 函數簡介	11
3.2. CLOCK模組方框圖	12
3.3. 函數說明	13
4. 定時計數器TIMER/WDT	23
4.1. 函數簡介	23
4.2. 功能方框圖	24
4.3. 函數說明	26
5. 晶片IO GPIO	52
5.1. 函數簡介	52
5.2. GPIO模組方框圖	54
5.3. 函數說明	55
6. 模數轉換器ADC	101
6.1. 函數簡介	101
6.2. ADC模組方框圖	101
6.3. 函數說明	102

7. SPI 串列通訊	116
7.1. 函數簡介	116
7.2. SPI 模組方框圖	116
7.3. 函數說明	118
8. 非同步串列通訊 UART	125
8.1. 函數簡介	125
8.2. UART 模組方框圖	126
8.3. 函數說明	127
9. 增強型多功能比較器 ECPA	142
9.1. 函數簡介	142
9.2. ECPA 模組方框圖	143
9.3. 函數說明	144
10. 低雜訊放大器 LNOP1/LNOP2	154
10.1. 功能簡介	154
10.2. LNOP1/LNOP2 模組方框圖	154
10.3. 函數說明	156
11. 電源管理 PMU	164
11.1. 函數簡介	164
11.2. PowerManage & LVD 模組方框圖	164
11.3. 函數說明	165
12. 捕捉/比較器 (CCP)	171
12.1. 函數功能簡介	171
12.2. 捕捉/比較器模組方框圖	172

12.3. 函數說明	173
13. LCD顯示驅動器	180
13.1. 函數簡介	180
13.2. LCD驅動器功能方框圖	180
13.3. 函數說明	181
14. LIBRARY	188
14.1. Library File	188
15. REVISION HISTORY	189
16. C LIBRARY CHANGE LIST	190

注意：

- 1、本說明書中的內容，隨著產品的改進，有可能不經過預告而更改。請客戶及時到本公司網站下載更新 <http://www.hycontek.com>。
- 2、本規格書中的圖形、應用電路等，因第三方工業所有權引發的問題，本公司不承擔其責任。
- 3、本產品在單獨應用的情況下，本公司保證它的性能、典型應用和功能符合說明書中的條件。當使用在客戶的產品或設備中，以上條件我們不作保證，建議客戶做充分的評估和測試。
- 4、請注意輸入電壓、輸出電壓、負載電流的使用條件，使 IC 內的功耗不超過封裝的容許功耗。對於客戶在超出說明書中規定額定值使用產品，即使是瞬間的使用，由此所造成的損失，本公司不承擔任何責任。
- 5、本產品雖內置防靜電保護電路，但請不要施加超過保護電路性能的過大靜電。
- 6、本規格書中的產品，未經書面許可，不可使用在要求高可靠性的電路中。例如健康醫療器械、防災器械、車輛器械、車載器械及航空器械等對人體產生影響的器械或裝置，不得作為其部件使用。
- 7、本公司一直致力於提高產品的品質和可靠度，但所有的半導體產品都有一定的失效概率，這些失效概率可能會導致一些人身事故、火災事故等。當設計產品時，請充分留意冗餘設計並採用安全指標，這樣可以避免事故的發生。
- 8、本規格書中內容，未經本公司許可，嚴禁用於其他目的之轉載或複製。

1. 導讀

1.1. C 函式庫簡介

本檔用於描述 HYCON HY11P 系列 C 函式庫使用的參考手冊，系統端軟體發展人員可以通過使用 C 函式庫直接調用開發替換暫存器操作開發來有效的提高整個產品的開發效率。

檔中 C 函式庫的每一個函數都帶有說明、用法及使用常式，所有的函數都存在我們 HYCON 提供的 C IDE 安裝目錄下的 Driver 資料夾裡。

1.2. 相關文檔

用戶可以在我們公司網站上下載以下所有文檔，獲取其他相關的資料。

下載文檔的網址：

<http://www.hycontek.com/cn/products/720>

- (1)HYCON HY11P Series Data Sheet
- (2)HYCON HY11P Series User's Guide
- (3)HYCON HY11P Series Hardware TOOL User Manual
- (4)HYCON HY11P Series Software TOOL User Manual

2. MCU 系統控制

2.1. 函數簡介

該部分函數描述晶片中斷系統控制及 MCU 狀態讀取，包含：

- 工作模式（休眠模式（sleep）、待機模式（Idle））的控制
- 全域中斷的控制
- MCU 程式狀態讀取
- 包含 RST.h/INT.h 標頭檔

序號	函數名稱	功能描述
01	Sleep	啓動低功耗睡眠模式；
02	Idle	啓動低功耗待機模式；
03	INT_GIE_Enable	使能全域中斷；
04	INT_GIE_Disable	關閉全域中斷；
05	SYS_ReadSKERR	讀取堆疊錯誤旗標(SKERR)
06	SYS_ReadBOR	讀取電源干擾旗標(BOR)
07	SYS_ReadTO	讀取看門狗計數旗標(TO)

2.2. 函數說明

2.2.1. Sleep

- 函數

Sleep();

- 函數功能

啓動低功耗睡眠模式。

- 輸入參數

無

- 包含標頭檔

Driver/RST.h

- 函數返回值

無

- 函數用法

/* 休眠之前需要將所有不用功能關閉，然後調用睡眠函數，使IC進入睡眠模式 */

Sleep();

2.2.2. Idle

- 函數

Idle();

- 函數功能

啓動低功耗待機模式。

- 輸入參數

無

- 包含標頭檔

Driver/RST.h

- 函數返回值

無

- 函數用法

/* 休眠之前需要將所有不用功能關閉，然後調用待機函數，使IC進入待機模式 */

Idle();

2.2.3. INT_GIE_Enable

- 函數

INT_GIE_Enable();

- 函數功能

使能全域中斷，設置暫存器INTE1[7]的值。

- 輸入參數

無

- 包含標頭檔

Driver/INT.h

- 函數返回值

無

- 函數用法

/* 使能全域中斷 */

INT_GIE_Enable();

2.2.4. INT_GIE_Disable

- 函數

INT_GIE_Disable();

- 函數功能

關閉全域中斷，清零暫存器INTE1[7]的值。

- 輸入參數

無

- 包含標頭檔

Driver/INT.h

- 函數返回值

無

- 函數用法

/* 關閉全域中斷 */

INT_GIE_Disable();

2.2.5. SYS_ReadSKERR

- 函數

SYS_ReadSKERR();

- 函數功能

讀取堆疊錯誤旗標 (SKERR)，讀取暫存器PSTATUS[2]的值。

- 輸入參數

無

- 包含標頭檔

Driver/RST.h

- 函數返回值

0x00：清除需透過BOR、RESET或指令

0x04：堆疊錯誤

- 函數用法

/* 讀取堆疊錯誤旗標 */

unsigned char flag;

flag = SYS_ReadSKERR();

2.2.6. SYS_ReadBOR

- 函數

SYS_ReadBOR();

- 函數功能

讀取電源干擾旗標 (BOR)，清零暫存器PSTATUS[4]的值。

- 輸入參數

無

- 包含標頭檔

Driver/RST.h

- 函數返回值

0x00：未發生重置

0x10：發生電源干擾重置

- 函數用法

/* 讀取電源干擾旗標(BOR) */

unsigned char flag;

flag = SYS_ReadBOR();

2.2.7. SYS_ReadTO

- 函數

SYS_ReadTO();

- 函數功能

讀取看門狗計數旗標 (TO)，讀取暫存器PSTATUS[6]的值。

● 輸入參數

無

● 包含標頭檔

Driver/RST.h

● 函數返回值

0x00: 正常，清除需透過BOR、RST或指令

0x40: 看門狗計數溢出重置

● 函數用法

/* 讀取看門狗計數旗標(TO) */

unsigned char flag;

flag = SYS_ReadTO();

3. 晶片時鐘源 CLOCK

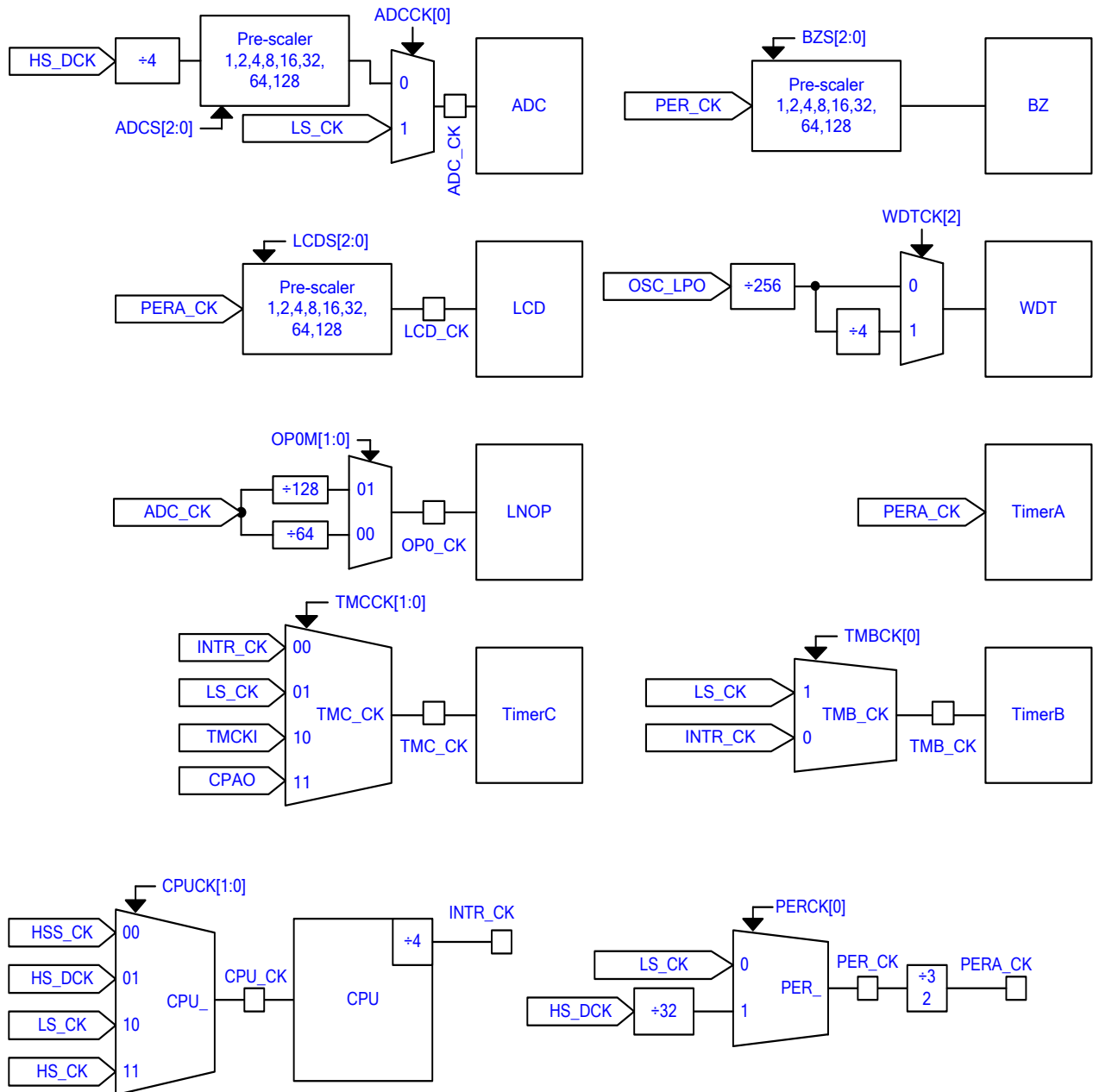
3.1. 函數簡介

函數描述 MCU 及其他功能模組的時鐘源操作，包含：

- 內部高速及低速頻率的控制
- 外部高速及低速晶振的控制
- MCU 周邊功能模組時鐘源控制

序號	函數名稱	功能描述
01	CLK_GPIO_OpenXTInput	設置PT2.0/PT2.1作為外部晶振輸入引腳
02	CLK_CPUCKOpen	設置晶片工作頻率及外部晶振
03	CLK_ExtHSCKEnable	開啓外部高速晶振
04	CLK_ExtLSCKEnable	開啓外部低速晶振
05	CLK_ExtCKDisable	關閉外部晶振
06	CLK_ExtHSCKSelect	選擇外部高速晶振作為晶片高速頻率來源
07	CLK_ExtLSCKSelect	選擇外部低速晶振作為晶片低速頻率來源
08	CLK_HAOEnable	開啓內部高速頻率HAO
09	CLK_HAODisable	關閉內部高速HAO
10	CLK_HAOHSCKSel	選擇內部HAO作為晶片高速頻率來源
11	CLK_LPOHSCKSel	選擇內部LPO作為晶片高速頻率來源
12	CLK_CPUCKSel	選擇晶片CPU頻率來源，HSS_CK頻率分頻
13	CLK_PERCKHSDCKSel	選擇HSD_CK作為周邊頻率PERCK的頻率來源
14	CLK_PERCKLSCKSel	選擇LS_CK作為周邊頻率PERCK的頻率來源
15	CLK_LPOHSCKSel	選擇內部LPO作為晶片高速頻率來源
16	CLK_LCDCKSel	設置LCD工作頻率預分頻
17	CLK_BZCKSel	設置蜂鳴器驅動頻率預分頻
18	CLK_ADCKSel	設置ADC工作頻率來源
19	CLK_ADCKDivSel	設置ADC工作頻率預分頻

3.2. CLOCK 模組方框圖



3.3. 函數說明

3.3.1. CLK_GPIO_OpenXTInput

- 函數

```
void CLK_GPIO_OpenXTInput(void);
```

- 函數功能

設置PT2.0/PT2.1作為外部晶振輸入引腳。

- 輸入參數

無

- 包含標頭檔

Driver/CLK.h

- 函數返回值

無

- 函數用法

```
/* 設置PT2.0/PT2.1作為外部晶振輸入引腳 */  
CLK_GPIO_OpenXTInput();
```

3.3.2. CLK_CPUCKOpen

- 函數

```
void CLK_CPUCKOpen(  
    unsigned char CPUCKSel,  
    unsigned char ExtCKEn,  
    unsigned char HSSCKDiv,  
    unsigned char VXTSP,  
    unsigned char oscSOURCE) ;
```

- 函數功能

設置晶片的CPU工作頻率來源及高速頻率HSS的分頻設置，設置晶片外部晶振開關機高低頻率選擇，設置暫存器MCKCN1/MCKCN2。

- 函數輸入參數

CPUCKSel [in] : 選擇CPU_CK來源

MCKCN2_CPUCK_HSCK : CPU_CK頻率來源於HS_CK

MCKCN2_CPUCK_LSCK : CPU_CK頻率來源於LS_CK

MCKCN2_CPUCK_HSDCK : CPU_CK頻率來源於HSD_CK

MCKCN2_CPUCK_HSSCK : CPU_CK頻率來源於HSS_CK

ExtCKEn [in] : 開關外部晶振

MCKCN1_ENXT_ENABLE : 開啓外部晶振

MCKCN1_ENXT_DISABLE : 關閉外部晶振

HSSCKDiv [in] : 設置高速頻率HSS分頻

MCKCN2_HSS_HSCKDIV8 : 高速頻率HSS進行8分頻

MCKCN2_HSS_HSCKDIV4：高速頻率HSS進行4分頻

MCKCN2_HSS_HSCKDIV2：高速頻率HSS進行2分頻

MCKCN2_HSS_HSCKDIV1：高速頻率HSS進行1分頻

VXTSP [in]：選擇外部晶振高頻或低頻

MCKCN1_XTSP_XTS：外部高速頻率2~8MHZ

MCKCN1_XTSP_XTL：外部低速頻率32768HZ

OscSOURCE [in]：選擇晶片工作頻率來源

當外部晶振開啓時：

MCKCN2_LSCK_CY：晶片低速頻率來源為外部晶振

MCKCN2_LSCK_LPO：晶片低速頻率拉遠為內部LPO

當外部晶振關閉時：

MCKCN2_LSCK_CY：不能設置

MCKCN2_LSCK_LPO：晶片低速頻率拉遠為內部LPO

當外部晶振開啓時：

MCKCN2_HSCK_CY：晶片高速頻率來源為外部晶振

MCKCN2_HSCK_HAO：晶片高速頻率來源為內部HAO

當外部晶振關閉時：

MCKCN2_HSCK_CY：不能設置

MCKCN2_HSCK_HAO：晶片高速頻率來源為內部HAO

- 包含標頭檔

Driver/ CLK.h

- 函數返回值

無

- 函數用法

/* 關閉外部高速晶震，選擇HAO作為MCU頻率源，HSS進行1分頻，且高速頻率HSD_CK作為CPU時鐘源 */

```
CLK_CPUCKOpen(
```

```
MCKCN2_CPUCK_HSDCK,
```

```
MCKCN1_ENXT_DISABLE,
```

```
MCKCN2_HSS_HSCKDIV1,
```

```
MCKCN1_XTSP_XTL,
```

```
MCKCN2_HSCK_HAO);
```

3.3.3. CLK_ExtHSCKEnable

- 函數

```
CLK_ExtHSCKEnable();
```

- 函數功能

開啓外部高速晶振，操作暫存器MCKCN1[2:1]=11B。

- 函數輸入參數

無

- 包含標頭檔

Driver/CKL.h

- 函數返回值

無

- 函數用法

/* 開啓外部晶振 */

CLK_GPIO_OpenXTInput(); //開啓外部高速晶振引腳

CLK_ExtHSCKEnable(); //開啓外部高速晶振

3.3.4. CLK_ExtLSCKEnable

- 函數

CLK_ExtLSCKEnable();

- 函數功能

開啓外部低速晶振，操作暫存器MCKCN1[2:1]=10B。

- 輸入參數

無

- 包含標頭檔

Driver/CLK.h

- 函數返回值

無

- 函數用法

/* 開啓外部低速晶振 */

CLK_GPIO_OpenXTInput(); //開啓外部高速晶振引腳

CLK_ExtLSCKEnable(); //開啓外部低速晶振

3.3.5. CLK_ExtCKDisable

- 函數

CLK_ExtCKDisable();

- 函數功能

關閉外部晶振，操作暫存器MCKCN1[1]=0B。

- 輸入參數

無

- 包含標頭檔

Driver/CLK.h

- 函數返回值

無

- 函數用法

/* 關閉外部晶振 */

CLK_ExtCKDisable(); //關閉外部晶振

3.3.6. CLK_ExtHSCKSelect

- 函數

CLK_ExtHSCKSelect();

- 函數功能

選擇外部高速晶振作為晶片高速頻率來源，在設置前需要先開啓外部高速晶振並等待穩定，暫存器MCKCN2[4]=1。

- 輸入參數

無

- 包含標頭檔

Driver/CLK.h

- 函數返回值

無

- 函數用法

```
/* 選擇外部高速晶震作為晶片高速頻率來源 */  
CLK_GPIO_OpenXTInput();           //開啓外部高速晶振引腳  
CLK_ExtHSCKEnable();              //開啓外部高速晶振  
CLK_ExtHSCKSelect();              //選擇晶片高速頻率源為外部晶振
```

3.3.7. CLK_ExtLSCKSelect

- 函數

CLK_ExtLSCKSelect();

- 函數功能

選擇外部低速晶振作為晶片低速頻率來源，在設置前需要先開啓外部低速晶振並等待穩定，暫存器MCKCN2[5]=1。

- 輸入參數

無

- 包含標頭檔

Driver/CLK.h

- 函數返回值

無

- 函數用法

```
/* 設置外部低速晶振作為晶片低速頻率來源 */  
CLK_GPIO_OpenXTInput();           //開啓外部高速晶振引腳  
CLK_ExtLSCKEnable();              //開啓外部低速晶振；  
CLK_ExtLSCKSelect();              //設置外部低速晶振作為晶片低速頻率來源
```

3.3.8. CLK_HAOEnable

- 函數

CLK_HAOEnable();

● 函數功能

開啓內部高速頻率HAO，設置MCKCN1[0]=1。

● 輸入參數

無

● 包含標頭檔

Driver/CLK.h

● 函數返回值

無

● 函數用法

/* 使能內部HAO */

CLK_HAOEnable(); //使能內部高速 HAO

3.3.9. CLK_HAODisable

● 函數

CLK_HAODisable();

● 函數功能

關閉內部高速HAO，設置MCKCN1[0]=0。

● 輸入參數

無

● 包含標頭檔

Driver/CLK.h

● 函數返回值

無

● 函數用法

/* 關閉內部高速晶振HAO */

CLK_HAODisable(); //關閉內部高速頻率 HAO

3.3.10. CLK_HAOHSCkSel

● 函數

CLK_HAOHSCkSel();

● 函數功能

選擇內部HAO作為晶片高速頻率來源，在設置前需要先開啓HAO並等待穩定，暫存器MCKCN2[4]=0。

● 輸入參數

無

● 包含標頭檔

Driver/CLK.h

● 函數返回值

無

● 函數用法

```
/* 開啟內部高速晶振HAO並設置為晶片高速頻率來源 */  
CLK_HAOEnable();           //開啟內部高速頻率 HAO  
CLK_HAOHSCkSel();         //設置 HAO 作為晶片高速頻率來源
```

3.3.11. CLK_LPOHSCkSel

● 函數

```
CLK_LPOHSCkSel();
```

● 函數功能

選擇內部LPO作為晶片高速頻率來源，暫存器MCKCN2[5]=0。

● 輸入參數

無

● 包含標頭檔

Driver/CLK.h

● 函數返回值

無

● 函數用法

```
/* 設置LPO作為晶片低速頻率來源 */  
CLK_LPOHSCkSel();           //設置 LPO 作為晶片低速頻率來源
```

3.3.12. CLK_CPUCKSel

● 函數

```
CLK_CPUCKSel(hssdiv, cpucksel);
```

● 函數功能

選擇晶片CPU頻率來源，並設置高速頻率HSS_CK頻率分頻，暫存器MCKCN2[1:0]及MCKCN2[3:2]。

● 輸入參數

hssdiv [in] : 設置高速頻率HSS_CK頻率分頻

MCKCN2_HSS_HSCKDIV8 : 高速頻率HSS進行8分頻

MCKCN2_HSS_HSCKDIV4 : 高速頻率HSS進行4分頻

MCKCN2_HSS_HSCKDIV2 : 高速頻率HSS進行2分頻

MCKCN2_HSS_HSCKDIV1 : 高速頻率HSS進行1分頻

cpucksel [in] : 選擇晶片CPU頻率來源

MCKCN2_CPUCK_HSCK : CPU_CK頻率來源於HS_CK

MCKCN2_CPUCK_LSCK : CPU_CK頻率來源於LS_CK

MCKCN2_CPUCK_HSDCK : CPU_CK頻率來源於HSD_CK

MCKCN2_CPUCK_HSSCK : CPU_CK頻率來源於HSS_CK

● 包含標頭檔

Driver/CLK.h

● 函數返回值

無

- 函數用法

/* 設置CPU_CK來源HSD_CK，並設置HSS_CK為1分頻 */

CLK_CPUCKSel(MCKCN2_HSS_HSCKDIV1, MCKCN2_CPUCK_HSDCK);

3.3.13. CLK_PERCKHSDCKSel

- 函數

CLK_PERCKHSDCKSel();

- 函數功能

選擇HSD_CK作為周邊頻率PERCK的頻率來源，暫存器MCKCN3[3]=1。

- 輸入參數

無

- 包含標頭檔

Driver/CLK.h

- 函數返回值

無

- 函數用法

/* 選擇HSD_CK作為周邊頻率PERCK的頻率來源 */

CLK_PERCKHSDCKSel();

3.3.14. CLK_PERCKLSCKSel

- 函數

CLK_PERCKLSCKSel();

- 函數功能

選擇LS_CK作為周邊頻率PERCK的頻率來源，暫存器MCKCN3[3]=0。

- 輸入參數

無

- 包含標頭檔

Driver/CLK.h

- 函數返回值

無

- 函數用法

/* 選擇LS_CK作為周邊頻率PERCK的頻率來源 */

CLK_PERCKLSCKSel();

3.3.15. CLK_LPOHSCkSel

- 函數

CLK_LPOHSCkSel();

- 函數功能

選擇內部LPO作為晶片高速頻率來源，暫存器MCKCN2[5]=0。

● 輸入參數

無

● 包含標頭檔

Driver/CLK.h

● 函數返回值

無

● 函數用法

/* 設置LPO作為晶片低速頻率來源 */

CLK_LPOHSckSel(); //設置 LPO 作為晶片低速頻率來源

3.3.16. CLK_LCDCKSel

● 函數

CLK_LCDCKSel(LCDCKSel);

● 函數功能

設置LCD工作頻率預分頻，設置暫存器MCKCN3[7:5]。

● 輸入參數

LCDCkSel [in] : LCD工作頻率LCD_CK的預分頻設置

MCKCN3_LCDS_PERACKDIV128 : LCD_CK=Pera_CK/128

MCKCN3_LCDS_PERACKDIV64 : LCD_CK=Pera_CK/64

MCKCN3_LCDS_PERACKDIV32 : LCD_CK=Pera_CK/32

MCKCN3_LCDS_PERACKDIV16 : LCD_CK=Pera_CK/16

MCKCN3_LCDS_PERACKDIV8 : LCD_CK=Pera_CK/8

MCKCN3_LCDS_PERACKDIV4 : LCD_CK=Pera_CK/4

MCKCN3_LCDS_PERACKDIV2 : LCD_CK=Pera_CK/2

MCKCN3_LCDS_PERACKDIV1 : LCD_CK=Pera_CK/1

● 包含標頭檔

Driver/CLK.h

● 函數返回值

無

● 函數用法

/* 設置LCD工作頻率為Pera_CK/16預分頻 */

CLK_LCDCKSel(MCKCN3_LCDS_PERACKDIV16); //設置 LCD_CK= Pera_CK/16

3.3.17. CLK_BZCKSel

● 函數

CLK_BZCKSel(BZCKSel);

● 函數功能

設置蜂鳴器驅動頻率預分頻，設置暫存器MCKCN3[2:0]。

● 輸入參數

BZCKSel [in] : 設置蜂鳴器驅動頻率預分頻

MCKCN3_BZS_PERCKDIV128	: 蜂鳴器驅動頻率為PER_CK/128
MCKCN3_BZS_PERCKDIV64	: 蜂鳴器驅動頻率為PER_CK/64
MCKCN3_BZS_PERCKDIV32	: 蜂鳴器驅動頻率為PER_CK/32
MCKCN3_BZS_PERCKDIV16	: 蜂鳴器驅動頻率為PER_CK/16
MCKCN3_BZS_PERCKDIV8	: 蜂鳴器驅動頻率為PER_CK/8
MCKCN3_BZS_PERCKDIV4	: 蜂鳴器驅動頻率為PER_CK/4
MCKCN3_BZS_PERCKDIV3	: 蜂鳴器驅動頻率為PER_CK/3
MCKCN3_BZS_PERCKDIV1	: 蜂鳴器驅動頻率為PER_CK/1

● 包含標頭檔

Driver/CLK.h

● 函數返回值

無

● 函數用法

/* 設置蜂鳴器頻率為PER_CK/32 */

CLK_BZCKSel(MCKCN3_BZS_PERCKDIV32);

3.3.18. CLK_ADCCkSel

● 函數

CLK_ADCCkSel(ADCCkSel);

● 函數功能

設置ADC工作頻率來源，設置暫存器MCKCN1[4]。

● 輸入參數

ADCCkSel [in] : 設置ADC工作頻率來源

MCKCN1_ADCCk_LSCK	: ADC工作頻率來源於LS_CK
MCKCN1_ADCCk_HSDCK	: ADC工作頻率來源於HSD_CK

● 包含標頭檔

Driver/CLK.h

● 函數返回值

無

● 函數用法

/* 設置ADC工作頻率來源於HSD_CK */

CLK_ADCCkSel(MCKCN1_ADCCk_HSDCK); //設置 ADC 工作頻率來源於 HSD_CK

3.3.19. CLK_ADCCkDivSel

● 函數

CLK_ADCCkDivSel(ADCCkDivSel);

● 函數功能

設置ADC工作頻率預分頻，設置暫存器MCKCN1[7:5]。

● 輸入參數

ADCDivSel [in] : 設置ADC工作頻率預分頻

MCKCN1_ADCS_DIV128	: ADC_CK/128
MCKCN1_ADCS_DIV64	: ADC_CK/64
MCKCN1_ADCS_DIV32	: ADC_CK/32
MCKCN1_ADCS_DIV16	: ADC_CK/16
MCKCN1_ADCS_DIV8	: ADC_CK/8
MCKCN1_ADCS_DIV4	: ADC_CK/4
MCKCN1_ADCS_DIV2	: ADC_CK/2
MCKCN1_ADCS_DIV1	: ADC_CK/1

● 包含標頭檔

Driver/CLK.h

● 函數返回值

無

● 函數用法

/* 設置ADC工作頻率進行32預分頻 */

```
CLK_ADCKDivSel( MCKCN1_ADCS_DIV32 );           //設置 ADC_CK/32
```

4. 定時計數器 TIMER/WDT

4.1. 函數簡介

該部分函數描述看門狗(WDT)/定時計數器 A(Timer A)/ 定時計數器 B(Timer B) /定時計數器 C(Timer C)的功能控制，包含：

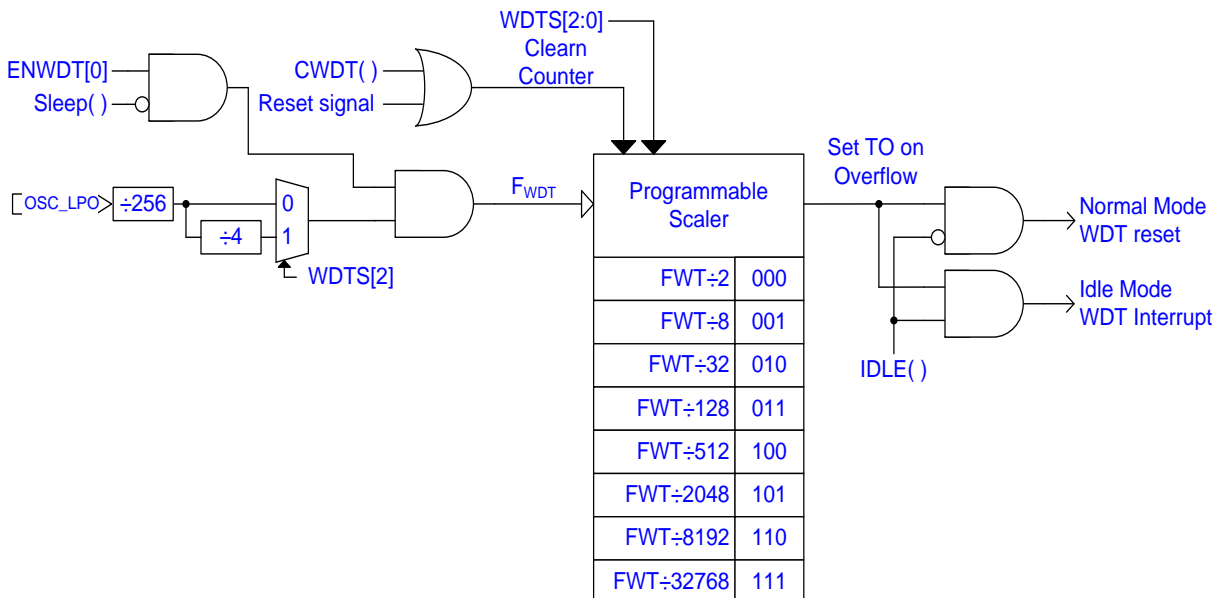
- 看門狗(WDT)的配置控制、啓動控制、中斷控制
- 定時計數器 A(Timer A)的配置控制、啓動控制、定時中斷控制
- 定時計數器 B(Timer B)的配置控制、啓動控制、定時控制
- 定時計數器 C(Timer C)的配置控制及 PWM 模式控制
- 包含 TMA.h/ TMB.h/ TMC.h/ PWM.h

序號	函數名稱	功能描述
01	WDT_Open	使能看門狗(WDT)，設定計數溢出值
02	WDT_INT_Enable	使能看門狗定時計數中斷功能
03	WDT_INT_Disable	關閉看門狗定時計數中斷功能
04	WDT_INT_IsFlag	讀取看門狗中斷請求旗標
05	WDT_INT_ClearFlag	清除看門狗中斷請求旗標
06	WDT_Enable	啓動看門狗功能
07	WDT_OFTimeSel	設置看門狗定時計數溢出時間
08	WDT_Clear	清零看門狗計數值
09	TMA_Open	設置 TMA 頻率源及計數溢出值啓動 TMA
10	TMA_INT_Enable	使能 TMA 定時中斷功能
11	TMA_INT_Disable	關閉 TMA 定時中斷功能
12	TMA_INT_IsFlag	讀取 TMA 定時中斷請求旗標
13	TMA_INT_ClearFlag	清零 TMA 中斷請求旗標
14	TMA_Enable	啓動 TMA 定時計數功能
15	TMA_Disable	關閉 TMA 定時計數功能
16	TMA_CLKSel	設置TMA的時鐘頻率源
17	TMA_OFControlSel	設置 TMA 定時計數溢出值
18	TMB_Open	使能TMB，設置TMB頻率源和頻率分頻器，及TMB計數模式
19	TMB_INT_Enable	使能TMB定時中斷功能
20	TMB_INT_Disable	關閉TMB定時中斷功能
21	TMB_INT_IsFlag	讀取TMB中斷請求旗標
22	TMB_INT_ClearFlag	清零TMB定時中斷旗標
23	TMB_Enable	使能TMB定時計數功能
24	TMB_Disable	關閉TMB定時計數功能
25	TMB_CLKSel	設置 TMB 工作頻率源
26	TMB_CLKPrescalerSel	設置 TMB 工作頻率源分頻器
27	TMB_SynchConfig	TMB工作頻率與CPU同步處理設置
28	TMB_TMBROperateMode	設置TMB計數方式
29	TMC_Open	啓動TMC定時計數功能，設置TMC的時鐘源及時鐘源預分頻值，設置TMC定時計數溢出值
30	TMC_INT_Enable	使能 TMC 定時計數中斷功能
31	TMC_INT_Disable	關閉 TMC 定時計數中斷功能
32	TMC_INT_IsFlag	讀取 TMC 定時計數中斷請求旗標
33	TMC_INT_ClearFlag	清除 TMC 中斷請求旗標
34	TMC_Enable	啓動 TMC 定時計數功能

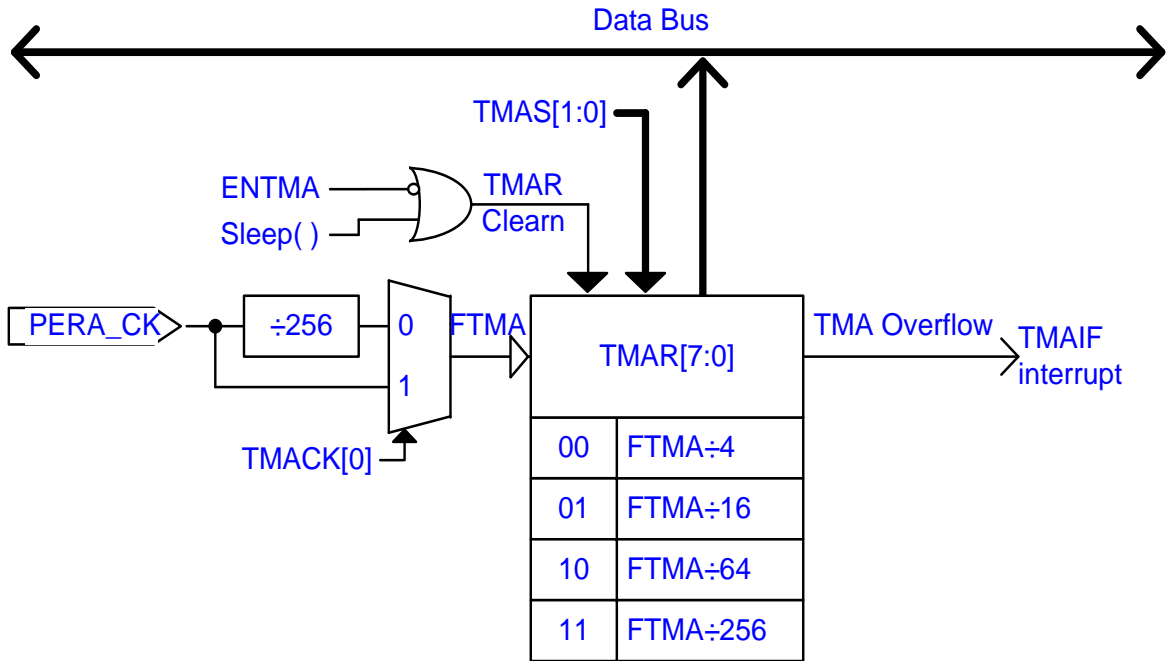
序號	函數名稱	功能描述
35	TMC_Disable	關閉TMC定時計數功能
36	TMC_CLKSel	設置TMC工作頻率源
37	TMC_OFControlSel	設置TMC計數溢出值
38	TMC_CLKPrescalerSel	TMC工作頻率預分頻設置
39	PWM_Open	使能PWM功能;設置PWM輸出模式及對應引腳複用功能,輸出引腳有效准位,PWM週期及占空比,死區延遲時間
40	PWM_Enable	使能PWM功能
41	PWM_Disable	關閉PWM功能
42	PWM_ShutoffEnable	使能PWM自動關閉功能
43	PWM_ShutoffDisable	關閉PWM自動關閉功能
44	PWM_ShutoffConfig	設置PWM自動關閉功能觸發事件
45	PWM_ShutoffDefine0	設置PWM自動關閉時PWM0/PWM2輸出引腳狀態定義
46	PWM_ShutoffDefine1	設置PWM自動關閉時PWM1/PWM3輸出引腳狀態定義
47	PWM_OutMode	設置PWM輸出模式
48	PWM_OutStateSelect	設置PWM輸出引腳有效准位
49	PWM_StartConfig	設置PWM自動開啓控制條件
50	PWM_DBDTime	設置PWM死區延遲時間
51	PWM_PWMRL	寫入PWM週期控制值PWMR[9:0]低 2bit
52	PWM_PWMRH	寫入PWM週期控制值PWMR[9:0]高 8bit
53	PFD_Enable	使能PFD功能
54	PFD_Disable	關閉PFD功能
55	PFD_Open	使能PFD功能並設置PFD輸出波形週期

4.2. 功能方框圖

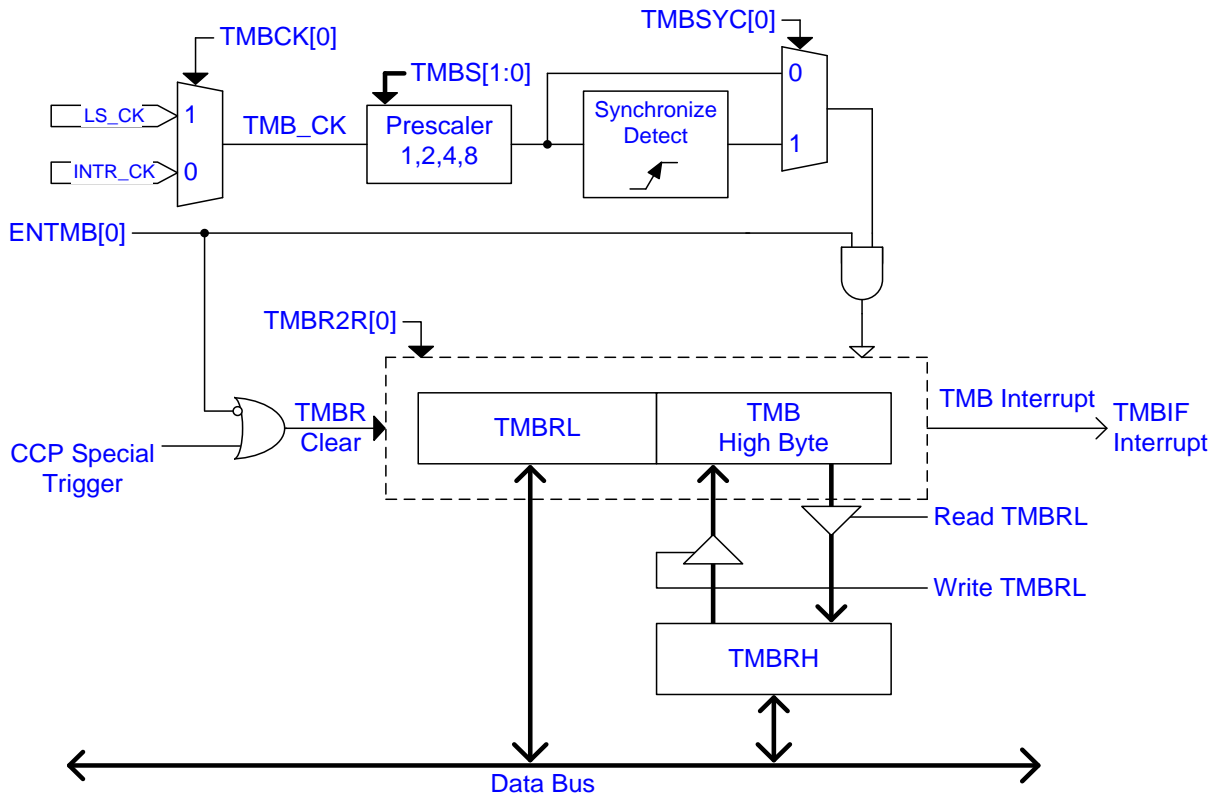
4.2.1. 看門狗(WDT) 模組方框圖



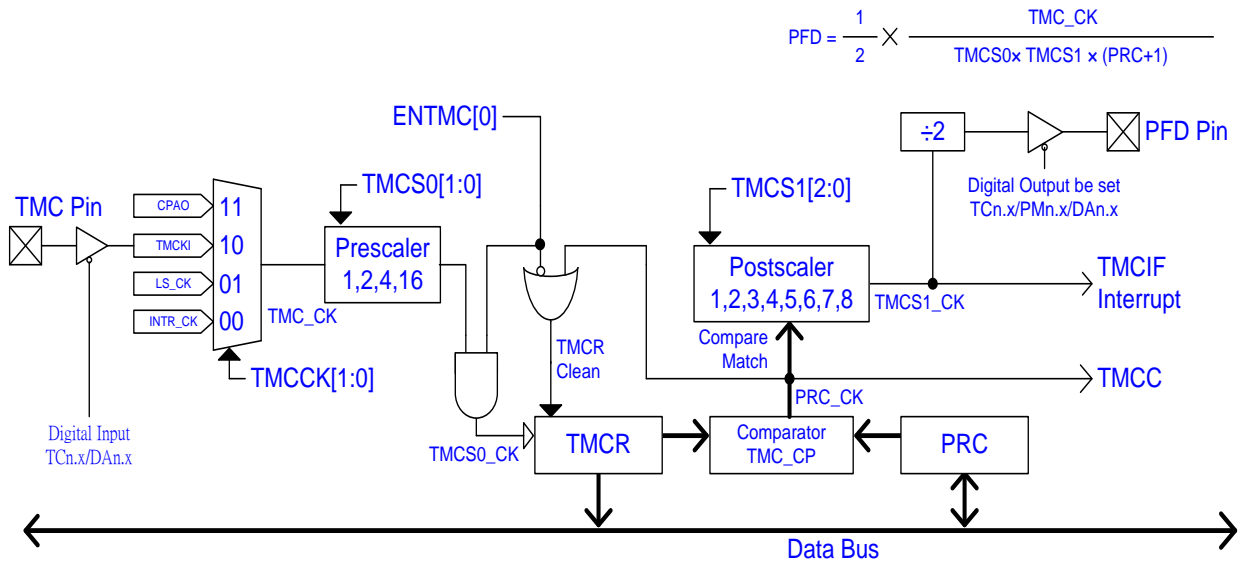
4.2.2. 定時計數器 A(Timer A) 模組方框圖



4.2.3. 定時計數器 B(Timer B) 模組方框圖



4.2.4. 定時計數器 C(Timer C) 模組方框圖



Note: PWM just work at TMCCCK=00 mode (clock source = INTR_CK)

4.3. 函數說明

4.3.1. WDT_Open

- 函數

```
void WDT_Open(unsigned char wdt);
```

- 函數功能

使能看門狗(WDT)，設定計數溢出值，設置暫存器TMACN[3:0]。

- 輸入參數

wdts [in] 看門狗時鐘源分頻設置

TMACN_WDTS_32768 : Fwdt / 32768

TMACN_WDTS_8192 : Fwdt / 8192

TMACN_WDTS_2048 : Fwdt / 2048

TMACN_WDTS_512 : Fwdt / 512

TMACN_WDTS_128 : Fwdt / 128

TMACN_WDTS_32 : Fwdt / 32

TMACN_WDTS_8 : Fwdt / 8

TMACN_WDTS_2 : Fwdt / 2

- 包含標頭檔

Driver/WDT.h

- 函數返回值

無

- 函數用法

```
/* 設置看門狗溢出時間Fwdt/32 */
```

```
WDT_Open ( TMACN_WDTS_32 );
```

4.3.2. WDT_INT_Enable

- 函數

WDT_INT_Enable();

- 函數功能

使能看門狗定時計數中斷功能，設置暫存器INTE1[2]=1。

- 輸入參數

無

- 包含標頭檔

Driver/WDT.h

- 函數返回值

無

- 函數用法

```
/* 使能看門狗定時計數中斷 */  
unsigned int data ;  
data = DrvWDT_CounterRead();
```

4.3.3. WDT_INT_Disable

- 函數

WDT_INT_Disable();

- 函數功能

關閉看門狗定時計數中斷功能，設置暫存器INTE1[2]=0。

- 輸入參數

無

- 包含標頭檔

Driver/WDT.h

- 函數返回值

無

- 函數用法

```
/* 關閉看門狗定時中斷功能 */  
WDT_INT_Disable();
```

4.3.4. WDT_INT_IsFlag

- 函數

WDT_INT_IsFlag();

- 函數功能

讀取看門狗中斷請求旗標，讀取暫存器INTF1[2]。

- 輸入參數

無

- 包含標頭檔

Driver/WDT.h

● 函數返回值

0x00 : 看門狗沒產生中斷請求旗標

0x04 : 看門狗產生中斷請求旗標

● 函數用法

/* 讀取看門中斷請求旗標 */

unsigned char flag;

Flag = WDT_INT_IsFlag();

4.3.5. WDT_INT_ClearFlag

● 函數

WDT_INT_ClearFlag()

● 函數功能

清除看門狗中斷請求旗標，設置暫存器INTF1[2]=0。

● 輸入參數

無

● 包含標頭檔

Driver/WDT.h

● 函數返回值

無

● 函數用法

/* 清除看門狗中斷請求旗標 */

WDT_INT_ClearFlag();

4.3.6. WDT_Enable

● 函數

WDT_Enable();

● 函數功能

啟動看門狗功能，設置暫存器TMACN[3]=1。

● 輸入參數

無

● 包含標頭檔

Driver/WDT.h

● 函數返回值

無

● 函數用法

/* 啟動看門狗定時計數功能 */

WDT_Enable();

4.3.7. WDT_OFTimeSel

- 函數

WDT_OFTimeSel(WDTSel);

- 函數功能

設置看門狗定時計數溢出時間，設置暫存器TMACN[2:0]。

- 輸入參數

WDTSel [in] 看門狗時鐘源分頻設置

TMACN_WDTS_32768 : Fwdt / 32768

TMACN_WDTS_8192 : Fwdt / 8192

TMACN_WDTS_2048 : Fwdt / 2048

TMACN_WDTS_512 : Fwdt / 512

TMACN_WDTS_128 : Fwdt / 128

TMACN_WDTS_32 : Fwdt / 32

TMACN_WDTS_8 : Fwdt / 8

TMACN_WDTS_2 : Fwdt / 2

- 包含標頭檔

Driver/WDT.h

- 函數返回值

無

- 函數用法

/* 設置看門狗定時計數溢出為Fwdt/32 */

WDT_OFTimeSel(TMACN_WDTS_32);

4.3.8. WDT_Clear

- 函數

WDT_Clear();

- 函數功能

清零看門狗計數值。

- 輸入參數

無

- 包含標頭檔

Driver/WDT.h

- 函數返回值

無

- 函數用法

/* 清零看門狗定時計數值 */

WDT_Clear();

4.3.9. TMA_Open

- 函數

void TMA_Open(unsigned char ck, unsigned char cks);

- 函數功能

設置TMA時鐘頻率源及定時計數溢出值，並啟動TMA定時功能，設置暫存器TMACN[7:4]。

- 輸入參數

ck [in] : TMA時鐘頻率源選擇

TMACN_TMACK_DIV1 : $F_{TMA} = PERA_CK$

TMACN_TMACK_DIV256 : $F_{TMA} = PERA_CK/256$

cks [in] : TMA定時計數溢出值設置

TMACN_TMAS_DIV256 : $F_{TMA} / 256$ ，每次溢出發生中斷， $TMAR[7:0]=TMAR[7:0]+256$

TMACN_TMAS_DIV64 : $F_{TMA} / 64$ ，每次溢出發生中斷， $TMAR[7:0]=TMAR[7:0]+64$

TMACN_TMAS_DIV16 : $F_{TMA} / 16$ ，每次溢出發生中斷， $TMAR[7:0]=TMAR[7:0]+16$

TMACN_TMAS_DIV4 : $F_{TMA} / 4$ ，每次溢出發生中斷， $TMAR[7:0]=TMAR[7:0]+4$

- 包含標頭檔

Driver/TMA.h

- 函數返回值

無

- 函數用法

/* 設置TMA時鐘源為PERA_CK，及計數溢出值為 $F_{TMA} / 16$ */

TMA_Open(TMACN_TMACK_DIV1, TMACN_TMAS_DIV16);

4.3.10. TMA_INT_Enable

- 函數

TMA_INT_Enable();

- 函數功能

使能TMA定時中斷功能，設置暫存器INTE1[3]=1。

- 輸入參數

無

- 包含標頭檔

Driver/TMA.h

- 函數返回值

無

- 函數用法

/* 使能TMA定時中斷功能 */

TMA_INT_Enable();

4.3.11. TMA_INT_Disable

- 函數

TMA_INT_Disable();

- 函數功能

關閉TMA定時中斷功能，設置暫存器INTE1[3]=0。

- 輸入參數

無

- 包含標頭檔

Driver/TMA.h

- 函數返回值

無

- 函數用法

```
/* 關閉TMA定時中斷功能 */  
TMA_INT_Disable();
```

4.3.12. TMA_INT_IsFlag

- 函數

TMA_INT_IsFlag();

- 函數功能

讀取TMA定時中斷請求旗標，讀取暫存器INTF1[3]。

- 輸入參數

無

- 包含標頭檔

Driver/TMA.h

- 函數返回值

返回TMA中斷請求旗標：

0x00：TMA沒有中斷請求

0x08：TMA產生中斷請求

- 函數用法

```
/* 讀取TMA定時中斷請求旗標 */  
unsigned char flag;  
flag = TMA_INT_IsFlag();
```

4.3.13. TMA_INT_ClearFlag

- 函數

TMA_INT_ClearFlag();

- 函數功能

清零TMA中斷請求旗標，設置暫存器INTF1[3]=0。

- 輸入參數

無

- 包含標頭檔

Driver/TMA.h

- 函數返回值

無

- 函數用法

/* 清零TMA中斷請求旗標 */

TMA_INT_ClearFlag();

4.3.14. TMA_Enable

- 函數

TMA_Enable()

- 函數功能

啓動TMA定時計數功能，設置暫存器TMACN[7]=1。

- 輸入參數

無

- 包含標頭檔

Driver/TMA.h

- 函數返回值

無

- 函數用法

/* 啓動TMA定時計數功能 */

TMA_Enable();

4.3.15. TMA_Disable

- 函數

TMA_Disable();

- 函數功能

關閉TMA定時計數功能，設置暫存器TMACN[7]=0。

- 輸入參數

無

- 包含標頭檔

Driver/TMA.h

- 函數返回值

無

- 函數用法

/* 關閉TMA定時計數功能 */

TMA_Disable();

4.3.16. TMA_CLKSel

- 函數

TMA_CLKSel(CLKSel);

- 函數功能

設置TMA的時鐘頻率源，設置暫存器TMACN[6]。

- 輸入參數

CLKSel [in]：設置TMA時鐘頻率源

TMACN_TMACK_DIV1 : $F_{TMA} = PERA_CK$

TMACN_TMACK_DIV256 : $F_{TMA} = PERA_CK/256$

- 包含標頭檔

Driver/TMA.h

- 函數返回值

無

- 函數用法

/* 設置TMA時鐘頻率源為PERA_CK */

TMA_CLKSel(TMACN_TMACK_DIV1);

4.3.17. TMA_OFControlSel

- 函數

TMA_OFControlSel(TMASel);

- 函數功能

設置TMA定時計數溢出值，設置暫存器TMACN[5:4]。

- 輸入參數

TMASel [in]：設置TMA定時計數溢出值

TMACN_TMAS_DIV256 : $F_{TMA} / 256$ ，每次溢出發生中斷， $TMAR[7:0]=TMAR[7:0]+256$

TMACN_TMAS_DIV64 : $F_{TMA} / 64$ ，每次溢出發生中斷， $TMAR[7:0]=TMAR[7:0]+64$

TMACN_TMAS_DIV16 : $F_{TMA} / 16$ ，每次溢出發生中斷， $TMAR[7:0]=TMAR[7:0]+16$

TMACN_TMAS_DIV4 : $F_{TMA} / 4$ ，每次溢出發生中斷， $TMAR[7:0]=TMAR[7:0]+4$

- 包含標頭檔

Driver/TMA.h

- 函數返回值

無

- 函數用法

/* 設定TMA定時計數溢出值為 $F_{TMA} / 16$ */

TMA_OFControlSel(TMACN_TMAS_DIV16);

4.3.18. TMB_Open

- 函數

void TMB_Open(unsigned char ck, unsigned char cks, unsigned char mode);

- 函數功能

使能TMB，設置TMB時鐘頻率源和時鐘頻率分頻器，及TMB計數模式，設置暫存器TMBCN[7:4]/ TMBCN[2]。

- 輸入參數

ck [in] : 設置TMB工作頻率源

TMBCN_TMBCK_LSCK : TMB工作頻率為LS_CK

TMBCN_TMBCK_INTRCK : TMB工作頻率為INTR_CK

cks [in] : 設置TMB工作頻率分頻器

TMBCN_TMBS_DIV8 : TMB_CK /8

TMBCN_TMBS_DIV4 : TMB_CK /4

TMBCN_TMBS_DIV2 : TMB_CK /2

TMBCN_TMBS_DIV1 : TMB_CK /1

mode [in] : 設置TMB計數模式

TMBCN_TMBR2R_16BIT : TMB作為16bit定時計數器使用

TMBCN_TMBR2R_8BIT : TMB作為8bit定時計數器使用

● 包含標頭檔

Driver/TMB.h

● 函數返回值

無

● 函數用法

/* 使能TMB，設置TMB時鐘源為INTR_CK，時鐘源進行2分頻，作為16bit計數器 */

TMB_Open(TMBCN_TMBCK_INTRCK, TMBCN_TMBS_DIV2, TMBCN_TMBR2R_16BIT);

4.3.19. TMB_INT_Enable

● 函數

TMB_INT_Enable();

● 函數功能

使能TMB定時中斷功能，設置暫存器INTE1[4]=1。

● 輸入參數

無

● 包含標頭檔

Driver/TMB.h

● 函數返回值

無

● 函數用法

/* 使能TMB定時中斷功能 */

TMB_INT_Enable();

4.3.20. TMB_INT_Disable

● 函數

TMB_INT_Disable();

● 函數功能

關閉TMB定時中斷功能，設置暫存器INTE1[4]=0。

- 輸入參數

無

- 包含標頭檔

Driver/TMB.h

- 函數返回值

無

- 函數用法

```
/* 關閉TMB定時中斷功能 */
```

```
TMB_INT_Disable();
```

4.3.21. TMB_INT_IsFlag

- 函數

```
TMB_INT_IsFlag();
```

- 函數功能

讀取TMB中斷請求旗標，讀取暫存器INTF1[4]。

- 輸入參數

無

- 包含標頭檔

Driver/TMB.h

- 函數返回值

返回TMB定時中斷請求旗標值

0x00：TMB沒有產生中斷請求

0x10：TMB產生中斷請求

- 函數用法

```
/* 讀取TMB中斷請求旗標 */
```

```
unsigned char flag;
```

```
flag = TMB_INT_IsFlag();
```

4.3.22. TMB_INT_ClearFlag

- 函數

```
TMB_INT_ClearFlag();
```

- 函數功能

清零TMB定時中斷旗標，設置暫存器INTF1[4]=0。

- 輸入參數

無

- 包含標頭檔

Driver/TMB.h

- 函數返回值

無

- 函數用法

/*清零TMB定時中斷旗標*/

TMB_INT_ClearFlag();

4.3.23. TMB_Enable

- 函數

TMB_Enable();

- 函數功能

使能TMB定時計數功能，設置暫存器TMBCN[7]=1。

- 輸入參數

無

- 包含標頭檔

Driver/TMB.h

- 函數返回值

無

- 函數用法

/* 使能TMB定時計數功能 */

TMB_Enable();

4.3.24. TMB_Disable

- 函數

TMB_Disable();

- 函數功能

關閉TMB定時計數功能，設置暫存器TMBCN[7]=0。

- 輸入參數

無

- 包含標頭檔

Driver/TMB.h

- 函數返回值

無

- 函數用法

/* 關閉TMB定時計數功能 */

TMB_Disable();

4.3.25. TMB_CLKSel

- 函數

TMB_CLKSel(CLKSel);

- 函數功能

設置TMB工作頻率源，設置暫存器TMBCN[6]。

- 輸入參數

CLKSel [in] : 設置TMB工作頻率源

TMBCN_TMBCK_LSCK : TMB工作頻率為LS_CK

TMBCN_TMBCK_INTRCK : TMB工作頻率為INTR_CK

- 包含標頭檔

Driver/TMB.h

- 函數返回值

無

- 函數用法

/* 設置TMB工作頻率源為INTR_CK */

TMB_CLKSel(TMBCN_TMBCK_INTRCK);

4.3.26. TMB_CLKPrescalerSel

- 函數

TMB_CLKPrescalerSel(CLKPreSel);

- 函數功能

設置TMB工作頻率源分頻器，設置暫存器TMBCN[5:4]。

- 輸入參數

CLKPreSel [in] : 設置TMB工作頻率分頻器

TMBCN_TMBS_DIV8 : TMB_CK /8

TMBCN_TMBS_DIV4 : TMB_CK /4

TMBCN_TMBS_DIV2 : TMB_CK /2

TMBCN_TMBS_DIV1 : TMB_CK /1

- 包含標頭檔

Driver/TMB.h

- 函數返回值

無

- 函數用法

/* 設置TMB工作頻率源為INTR_CK，並進行2分頻 */

TMB_CLKSel(TMBCN_TMBCK_INTRCK);

TMB_CLKPrescalerSel(TMBCN_TMBS_DIV2);

4.3.27. TMB_SynchConfig

- 函數

TMB_SynchConfig(SynchCon);

- 函數功能

TMB工作頻率與CPU同步處理設置，若TMB的工作頻率與CPU工作頻率不一致時，需要進行同步處理，設置暫存器TMBCN[3]。

- 輸入參數

TMBCN_TMBSYC_SYNCH : 與CPU同步處理
TMBCN_TMBSYC_ASYNCH : 與CPU不同步處理

● 包含標頭檔

Driver/TMB.h

● 函數返回值

無

● 函數用法

/* 設置TMB工作頻率為LS_CK，同時設置同步處理 */

TMB_CLKSel(TMBCN_TMCK_LCK);

TMB_SynchConfig(TMBCN_TMBSYC_SYNCH);

4.3.28. TMB_TMBROperateMode

● 函數

TMB_TMBROperateMode(TMBRSel);

● 函數功能

設置TMB計數方式，設置暫存器TMBCN[2]。

● 輸入參數

TMBRSel [in] : 設置TMB計數模式

TMBCN_TMBR2R_16BIT : TMB作為16bit定時計數器使用

TMBCN_TMBR2R_8BIT : TMB作為8bit定時計數器使用

● 包含標頭檔

Driver/TMB.h

● 函數返回值

無

● 函數用法

/* 設置TMB計數器為16bit */

TMB_TMBROperateMode(TMBCN_TMBR2R_16BIT);

4.3.29. TMC_Open

● 函數

void TMC_Open(unsigned char tmck, unsigned char tmcs0, unsigned char tmcs1);

● 函數功能

啓動TMC定時計數功能，設置TMC的時鐘源及時鐘源預分頻值，設置TMC定時計數溢出值，設置暫存器TMCCN[7:0]。

● 輸入參數

tmck [in] : TMC工作頻率選擇

TMCCN_TMCK_CPAO : TMC 工作頻率為 CPAO

TMCCN_TMCK_TMCKI : TMC 工作頻率為 TMCKI

TMCCN_TMCK_LSCK : TMC 工作頻率為 LS_CK

TMCCN_TMCK_INTRCK : TMC 工作頻率為 INTR_CK

tmcs0 [in] : TMC 工作頻率預分頻設置

TMCCN_TMCS0_DIV16 : TMC_CK/16

TMCCN_TMCS0_DIV4 : TMC_CK/4

TMCCN_TMCS0_DIV2 : TMC_CK/2

TMCCN_TMCS0_DIV1 : TMC_CK/1

tmcs1 [in] : TMC 定時計數溢出值設置

TMCCN_TMCS1_DIV8 : PRC_CK/8

TMCCN_TMCS1_DIV7 : PRC_CK/7

TMCCN_TMCS1_DIV6 : PRC_CK/6

TMCCN_TMCS1_DIV5 : PRC_CK/5

TMCCN_TMCS1_DIV4 : PRC_CK/4

TMCCN_TMCS1_DIV3 : PRC_CK/3

TMCCN_TMCS1_DIV2 : PRC_CK/2

TMCCN_TMCS1_DIV1 : PRC_CK/1

● 包含標頭檔

Driver/TMC.h

● 函數返回值

無

● 函數用法

/* 設置TMC工作頻率為LS_CK/2,溢出控制值為PRC_CK/4 */

TMCCN_Open(TMCCN_TMCK_LSCK, TMCCN_TMCS0_DIV2, TMCCN_TMCS1_DIV4);

4.3.30. TMC_INT_Enable

● 函數

TMC_INT_Enable();

● 函數功能

使能TMC定時計數中斷功能，設置暫存器INTE1[5]=1。

● 輸入參數

無

● 包含標頭檔

Driver/TMC.h

● 函數返回值

無

● 函數用法

/* 使能TMC定時計數中斷功能 */

TMC_INT_Enable();

4.3.31. TMC_INT_Disable

- 函數

TMC_INT_Disable();

- 函數功能

關閉TMC定時計數中斷功能，設置暫存器INTE1[5]=0。

- 輸入參數

無

- 包含標頭檔

Driver/TMC.h

- 函數返回值

無

- 函數用法

```
/* 關閉TMC定時中斷功能 */
```

```
TMC_INT_Disable();
```

4.3.32. TMC_INT_IsFlag

- 函數

TMC_INT_IsFlag();

- 函數功能

讀取TMC定時計數中斷請求旗標，讀取暫存器INTF1[5]。

- 輸入參數

無

- 包含標頭檔

Driver/TMC.h

- 函數返回值

0x00 : TMC沒中斷請求

0x20 : TMC產生中斷請求

- 函數用法

```
/* 讀取TMC中斷請求旗標 */
```

```
unsigned char flag ;
```

```
flag = TMC_INT_IsFlag();
```

4.3.33. TMC_INT_ClearFlag

- 函數

TMC_INT_ClearFlag();

- 函數功能

清除TMC中斷請求旗標，設置暫存器INTF1[5]=0。

- 輸入參數

無

- 包含標頭檔

Driver/TMC.h

- 函數返回值

無

- 函數用法

/* 清除TMC中斷請求旗標 */

TMC_INT_ClearFlag();

4.3.34. TMC_Enable

- 函數

TMC_Enable();

- 函數功能

啓動TMC定時計數功能，設置暫存器TMCCN[7]=1。

- 輸入參數

無

- 包含標頭檔

Driver/TMC.h

- 函數返回值

無

- 函數用法

/* 啓動TMC定時計數功能 */

TMC_Enable();

4.3.35. TMC_Disable

- 函數

TMC_Disable();

- 函數功能

關閉TMC定時計數功能，設置TMCCN[7]=0。

- 輸入參數

無

- 包含標頭檔

Driver/TMC.h

- 函數返回值

無

- 函數用法

/* 關閉TMC定時計數功能 */

TMC_Disable();

4.3.36. TMC_CLKSel

- 函數

TMC_CLKSel(CLKSel);

● 函數功能

設置TMC工作頻率源，設置暫存器TMCCN[6:5]。

● 輸入參數

CLKSel [in] :

TMCCN_TMCK_CPAO : TMC 工作頻率為 CPAO
TMCCN_TMCK_TMCKI : TMC 工作頻率為 TMCKI
TMCCN_TMCK_LSCK : TMC 工作頻率為 LS_CK
TMCCN_TMCK_INTRCK : TMC 工作頻率為 INTR_CK

● 包含標頭檔

Driver/TMC.h

● 函數返回值

無

● 函數用法

```
/* 設置TMC工作頻率源為INTR_CK */  
TMC_CLKSel( TMCCN_TMCK_INTRCK );
```

4.3.37. TMC_OFControlSel

● 函數

TMC_OFControlSel(TMCSel1);

● 函數功能

設置TMC計數溢出值，設置暫存器TMCCN[4:2]。

● 輸入參數

TMCSel1 [in] : TMC計數溢出值設置

TMCCN_TMCS1_DIV8 : PRC_CK/8
TMCCN_TMCS1_DIV7 : PRC_CK/7
TMCCN_TMCS1_DIV6 : PRC_CK/6
TMCCN_TMCS1_DIV5 : PRC_CK/5
TMCCN_TMCS1_DIV4 : PRC_CK/4
TMCCN_TMCS1_DIV3 : PRC_CK/3
TMCCN_TMCS1_DIV2 : PRC_CK/2
TMCCN_TMCS1_DIV1 : PRC_CK/1

● 包含標頭檔

Driver/TMC.h

● 函數返回值

無

● 函數用法

```
/* 設置TMC計數溢出為PRC_CK/5 */  
TMC_OFControlSel( TMCCN_TMCS1_DIV5 );
```

4.3.38. TMC_CLKPrescalerSel

- 函數

TMC_CLKPrescalerSel(TMCSel0);

- 函數功能

TMC工作頻率預分頻設置，設置暫存器TMCCN[1 :0]。

- 輸入參數

TMCSel0 [in] : TMC工作頻率預分頻設置

TMCCN_TMCS0_DIV16 : TMC_CK/16

TMCCN_TMCS0_DIV4 : TMC_CK/4

TMCCN_TMCS0_DIV2 : TMC_CK/2

TMCCN_TMCS0_DIV1 : TMC_CK/1

- 包含標頭檔

Driver/TMC.h

- 函數返回值

無

- 函數用法

/* 設置TMC工作預分頻為TMC_CK/4 */

TMC_CLKPrescalerSel(TMCCN_TMCS0_DIV4);

4.3.39. PWM_Open

- 函數

```
void PWM_Open(unsigned char mode,  
              unsigned char state,  
              unsigned char prc,  
              unsigned int pwmr,  
              unsigned char dbd)
```

- 函數功能

使能PWM功能，設置PWM輸出模式及對應引腳複用功能，設置PWM輸出引腳有效准位，設置PWM週期及占空比，設置PWM死區延遲時間，設置暫存器TMCCN/PRC/PWMR/PWMCN/PDBD/PT2M1/PT2M2/TRISC2

- 輸入參數

mode [in] : 設置PWM輸出模式

PWMCN_PWMCG_REVERSE : 全橋反向輸出，PWM1調變輸出，PWM2有效准位輸出

PWMCN_PWMCG_HALF : 半橋輸出，PWM0/PWM1死區延遲調變輸出，PWM2/PWM3作為IO引腳

PWMCN_PWMCG_FULL : 全橋正向輸出，PWM3調變輸出，PWM0有效准位輸出

PWMCN_PWMCG_SINGLE : 單輸出，PWM0調變輸出，PWM1/PWM2/PWM3作為IO引腳

state [in] : 設置PWM輸出引腳有效准位

PWMCN_PWMM_M3 : PWM0/PWM2有效准位為Low，PWM1/PWM3有效准位為Low

PWMCN_PWMM_M2 : PWM0/PWM2有效准位為Low，PWM1/PWM3有效准位為High

PWMCN_PWMM_M1 : PWM0/PWM2有效准位為High，PWM1/PWM3有效准位為Low

PWMCN_PWMM_M0 : PWM0/PWM2有效准位為High，PWM1/PWM3有效准位為High

prc [in] : 設置PWM週期，8bit的設置值，設置範圍是0x0~0xFF

pwmr [in] : 設置PWM占空比，10bit的設置值，設置範圍是0x0~0x3FF

dbd [in] : 設置PWM死區延遲時間，7bit的設置值，設置範圍是0x0~0x7F

- 包含標頭檔

Driver/PWM.h

- 函數返回值

無

- 函數用法

```
/* 設置PWM為單端輸出模式，PWM0/PWM1/PWM2/PWM3有效准位Low，周圍為0xFF，占空比為0x7F，  
死區延時為0x0A */
```

```
void PWM_Open( PWMCN_PWMCG_SINGLE, PWMCN_PWMM_M3, 0xFF, 0x7F, 0x0A );
```

4.3.40. PWM_Enable

- 函數

```
PWM_Enable();
```

- 函數功能

使能PWM功能，設置暫存器PWMCN[7]=1。

- 輸入參數

無

- 包含標頭檔

Driver/PWM.h

- 函數返回值

無

- 函數用法

```
/* 使能PWM功能 */
```

```
PWM_Enable();
```

4.3.41. PWM_Disable

- 函數

```
PWM_Disable();
```

- 函數功能

關閉PWM功能，設置暫存器PWMCN[7]=0。

- 輸入參數

無

- 包含標頭檔

Driver/PWM.h

- 函數返回值

無

- 函數用法

/* 關閉PWM功能 */

PWM_Disable();

4.3.42. PWM_ShutoffEnable

- 函數

PWM_ShutoffEnable();

- 函數功能

使能PWM自動關閉功能，設置暫存器PASC[7]=1。

- 輸入參數

無

- 包含標頭檔

Driver/PWM.h

- 函數返回值

無

- 函數用法

/* 使能PWM自動關閉功能 */

PWM_ShutoffEnable();

4.3.43. PWM_ShutoffDisable

- 函數

PWM_ShutoffDisable();

- 函數功能

關閉PWM自動關閉功能，設置暫存器PASC[7]=0。

- 輸入參數

無

- 包含標頭檔

Driver/PWM.h

- 函數返回值

無

- 函數用法

/* 關閉PWM自動關閉功能 */

PWM_ShutoffDisable();

4.3.44. PWM_ShutoffConfig

- 函數

PWM_ShutoffConfig(SCFCon);

- 函數功能

設置PWM自動關閉功能觸發事件，設置暫存器PASC[5:4]。

● 輸入參數

SCFCon [in]：設置PWM自動關閉功能觸發事件

PASC_PASCF_CPAO：觸發信號為CPAO

PASC_PASCF_MIX：觸發信號為CPAO或FIL0

PASC_PASCF_FIL0：觸發信號為FIL0

PASC_PASCF_CLOSE：關閉自動關閉事件

● 包含標頭檔

Driver/PWM.h

● 函數返回值

無

● 函數用法

/* 設置PWM自動關閉功能觸發事件為CPAO */

PWM_ShutoffConfig(PASC_PASCF_CPAO);

4.3.45. PWM_ShutoffDefine0

● 函數

PWM_ShutoffDefine0(SCN0Sel);

● 函數功能

設置PWM自動關閉時PWM0/PWM2輸出引腳狀態定義，設置暫存器PASC[3:2]。

● 輸入參數

SCN0Sel [in]：設置PWM自動關閉時PWM0/PWM2輸出引腳狀態定義

PASC_PSSCN0_OTHER：自動關閉時，PWM0 & PWM2輸出引腳的為三態狀態

PASC_PSSCN0_HIGH：自動關閉時，PWM0 & PWM2輸出引腳的為高電位

PASC_PSSCN0_LOW：自動關閉時，PWM0 & PWM2輸出引腳的為低電位

● 包含標頭檔

Driver/PWM.h

● 函數返回值

無

● 函數用法

/* 設置自動關閉時，PWM0 & PWM2輸出引腳的為高電位 */

PWM_ShutoffDefine0(PASC_PSSCN0_HIGH);

4.3.46. PWM_ShutoffDefine1

● 函數

PWM_ShutoffDefine1(SCN1Sel);

● 函數功能

設置PWM自動關閉時PWM1/PWM3輸出引腳狀態定義，設置暫存器PASC[1 :0]。

● 輸入參數

SCN1Sel [in] : 設置PWM自動關閉時PWM1/PWM3輸出引腳狀態定義

PASC_PSSCN1_OTHER : 自動關閉時，PWM1 & PWM3輸出引腳的為三態狀態

PASC_PSSCN1_HIGH : 自動關閉時，PWM1 & PWM3輸出引腳的為高電位

PASC_PSSCN1_LOW : 自動關閉時，PWM1 & PWM3輸出引腳的為低電位

● 包含標頭檔

Driver/PWM.h

● 函數返回值

無

● 函數用法

/* 設置自動關閉時，PWM1 & PWM3輸出引腳的為高電位 */

PWM_ShutoffDefine1(PASC_PSSCN1_HIGH);

4.3.47. PWM_OutMode

● 函數

PWM_OutMode(CGSel);

● 函數功能

設置PWM輸出模式，設置暫存器PWMCN[3:2]。

● 輸入參數

CGSel [in] : 設置PWM輸出模式

PWMCN_PWMCG_REVERSE : 全橋反向輸出，PWM1調變輸出，PWM2有效准位輸出

PWMCN_PWMCG_HALF : 半橋輸出，PWM0/PWM1死區延遲調變輸出，PWM2/PWM3作為IO引腳

PWMCN_PWMCG_FULL : 全橋正向輸出，PWM3調變輸出，PWM0有效准位輸出

PWMCN_PWMCG_SINGLE : 單輸出，PWM0調變輸出，PWM1/PWM2/PWM3作為IO引腳

● 包含標頭檔

Driver/PWM.h

● 函數返回值

無

● 函數用法

/* 設置PWM為單端輸出 */

PWM_OutMode(PWMCN_PWMCG_SINGLE);

4.3.48. PWM_OutStateSelect

● 函數

PWM_OutStateSelect(MSel);

● 函數功能

設置PWM輸出引腳有效准位，設置暫存器PWMCN[1 :0]。

● 輸入參數

MSel [in] : 設置PWM輸出引腳有效准位

PWMCN_PWMM_M3 : PWM0/PWM2有效准位為Low，PWM1/PWM3有效准位為Low

PWMCN_PWMM_M2 : PWM0/PWM2有效准位為Low , PWM1/PWM3有效准位為High

PWMCN_PWMM_M1 : PWM0/PWM2有效准位為High , PWM1/PWM3有效准位為Low

PWMCN_PWMM_M0 : PWM0/PWM2有效准位為High , PWM1/PWM3有效准位為High

- 包含標頭檔

Driver/PWM.h

- 函數返回值

無

- 函數用法

/* 設置PWM0/PWM2有效准位為High , PWM1/PWM3有效准位為Low */

PWM_OutStateSelect(PWMCN_PWMM_M1);

4.3.49. PWM_OutStateSelect

- 函數

PWM_StartConfig(PRSCon);

- 函數功能

設置PWM自動開啓控制條件，設置暫存器PDBD[7]。

- 輸入參數

PRSCon [in] : 設置PWM自動開啓控制條件

PDBD_ENPRS_AUTO : 硬體自動將PASF置0，PWMx在下一個週期自動開啓

PDBD_ENPRS_USER : 使用者程式上將PASF置0，PWMx在下一個週期自動開啓

- 包含標頭檔

Driver/PWM.h

- 函數返回值

無

- 函數用法

/* 設置PWM由硬體控制自動開啓 */

PWM_StartConfig(PDBD_ENPRS_AUTO);

4.3.50. PWM_OutStateSelect

- 函數

PWM_DBDCSel(DBDCSel);

- 函數功能

設置PWM死區延遲時間，設置暫存器PDBD[6:0]。

- 輸入參數

DBDCSel [in] : 設置PWM死區延遲時間，7bit的設置值，設置範圍是0x0~0xF

- 包含標頭檔

Driver/PWM.h

- 函數返回值

無

- 函數用法

/* 設置PWM死區延遲時間為0xA */

PWM_DBTime(0x0A);

4.3.51. PWM_PWMRL

- 函數

PWM_PWMRL(RLSel);

- 函數功能

寫入PWM週期控制值PWMR[9:0]的低2位PWMRL[1:0]， $PWMR = PWMRH[9:2] + PWMRL[1:0]$ ，設置暫存器PWMCN[5:4]。

- 輸入參數

RLSel [in]：設置PWM週期控制值PWMR[9:0]的低2位PWMRL[1:0]，設置範圍是0x0~0x03

- 包含標頭檔

Driver/PWM.h

- 函數返回值

無

- 函數用法

/* 設置PWMR[9:0]為0xAA，則低2位PWMRL[1:0]=0x02 */

PWM_PWMRL(0x02);

4.3.52. PWM_PWMRH

- 函數

PWM_PWMRH(RHSel);

- 函數功能

寫入PWM週期控制值PWMR[9:0]的高8位PWMRH[9:2]， $PWMR = PWMRH[9:2] + PWMRL[1:0]$ ，設置暫存器PWMR[7:0]。

- 輸入參數

RHSel [in]：設置PWM週期控制值PWMR[9:0]的高8位PWMRH[9:2]，設置範圍是0x0~0xFF

- 包含標頭檔

Driver/PWM.h

- 函數返回值

無

- 函數用法

/* 設置PWMR[9:0]為0xAA，則PWMRH[9:2]=0x2A */

PWM_PWMRL(0x2A);

4.3.53. PFD_Enable

- 函數

PFD_Enable();

- 函數功能

使能PFD功能，設置暫存器PWMCN[6]=1。

- 輸入參數

無

- 包含標頭檔

Driver/PWM.h

- 函數返回值

無

- 函數用法

```
/* 使能PFD功能 */  
PFD_Enable();
```

4.3.54. PFD_Disable

- 函數

```
PFD_Disable();
```

- 函數功能

關閉PFD功能，設置暫存器PWMCN[6]=0。

- 輸入參數

無

- 包含標頭檔

Driver/PWM.h

- 函數返回值

無

- 函數用法

```
/* 關閉PFD功能 */  
PFD_Disable();
```

4.3.55. PFD_Open

- 函數

```
void PFD_Open(unsigned char prc);
```

- 函數功能

使能PFD功能並設置PFD輸出波形週期，設置暫存器PWMCN[6]=1，PRC[7:0]，PT2M1[4]。

- 輸入參數

prc [in]：設置PFD輸出波形週期控制值，輸入範圍0x0~0xFF

- 包含標頭檔

Driver/PWM.h

- 函數返回值

無

- 函數用法

/* 使能PFD功能，設置PFD頻率控制值為0x2F */

PFD_Open(0x2F);

5. 晶片 IO GPIO

5.1. 函數簡介

該部分函數描述 GPIO 的工作模式控制，包含：

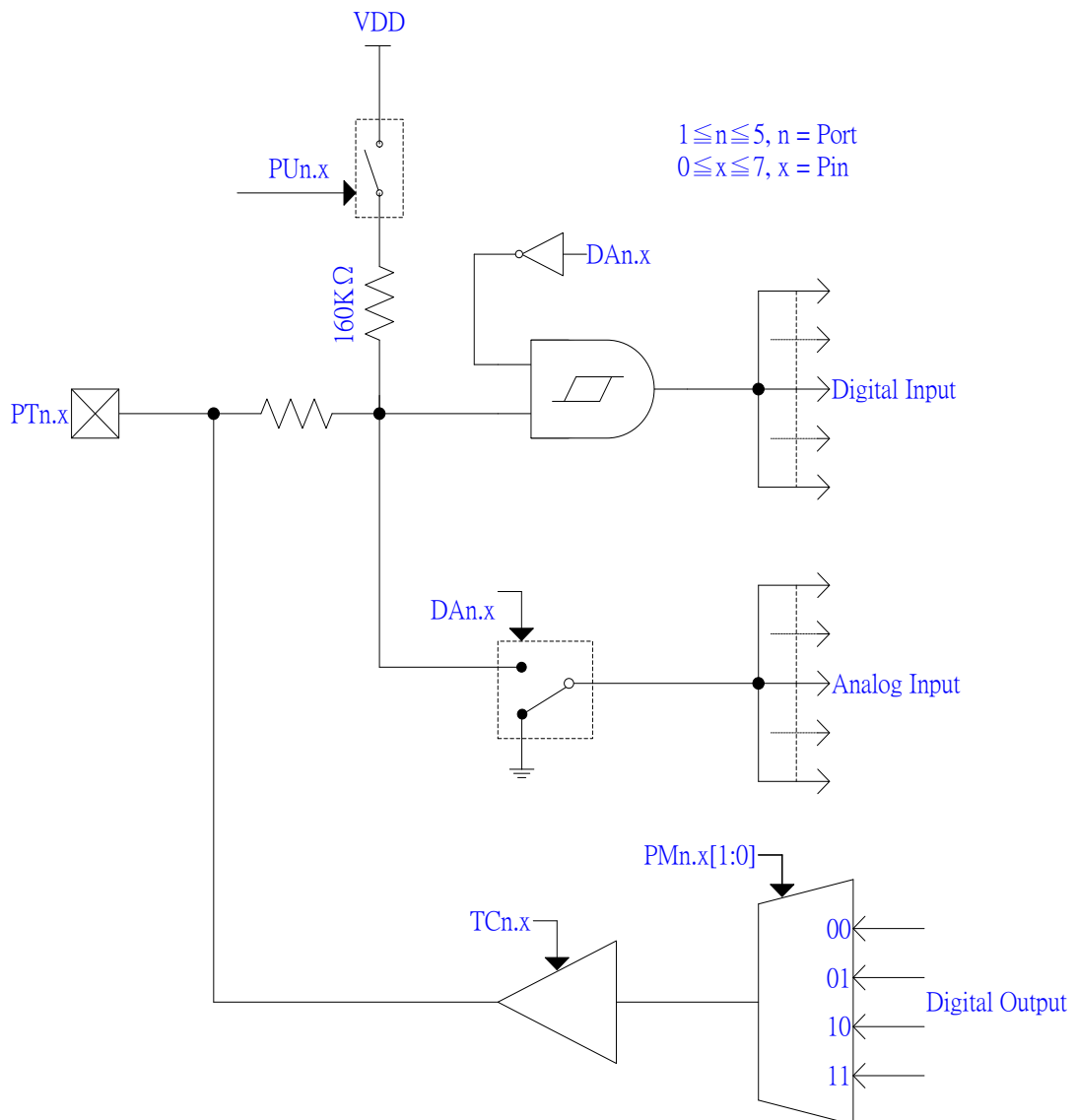
- GPIO 的工作模式控制
- GPIO 的上拉控制
- GPIO 的外部中斷功能控制
- GPIO 複用功能控制

序號	函數名稱	功能描述
01	GPIO_OpenPT1Input	設置 PT1 對應引腳輸入模式及使能上拉電阻
02	GPIO_OpenPT2Input	設置PT2對應引腳輸入模式及使能上拉電阻
03	GPIO_OpenPT3Input	設置PT3對應引腳輸入模式及使能上拉電阻
04	GPIO_OpenPT4Input	設置PT4對應引腳輸入模式及使能上拉電阻
05	GPIO_OpenPT5Input	設置PT5對應引腳輸入模式及使能上拉電阻
06	GPIO_PT1OutputMode	設置PT1作為輸出模式
07	GPIO_PT1OutputHigh	設置PT1對應IO輸出高電位
08	GPIO_PT1OutputLow	設置PT1對應IO輸出低電位
09	GPIO_PT1InputMode	設置PT1對應IO作為輸入模式
10	GPIO_PT1InputPullHight	使能PT1對應IO的輸入上拉電阻
11	GPIO_PT1InputPullHightClear	關閉PT1對應IO pin的輸入上拉電阻
12	GPIO_GetPT1Data	讀取PT1輸入狀態
13	GPIO_PT1AnalogMode	使能PT1.0/PT1.1/PT1.2類比的功能
14	GPIO_PT1DigitalMode	使能PT1.0/PT1.1/PT1.2數位的功能
15	BZ_Open	設置蜂鳴器驅動頻率預分頻，及PT1.7作為輸出
16	BZ_OutEnable	使能BZ蜂鳴器功能
17	BZ_OutDisable	關閉BZ蜂鳴器輸出功能
18	INT0_Enable	使能PT1.0外部中斷功能
19	INT0_Disable	關閉PT1.0外部中斷功能
20	INT0_IsFlag	讀取PT1.0外部中斷旗標
21	INT0_ClearFlag	清除PT1.0外部中斷請求旗標
22	GPIO_INTEG0Sel	設置PT1.0外部中斷觸發沿
23	INT1_Enable	使能PT1.1外部中斷功能
24	INT1_Disable	關閉PT1.1外部中斷功能
25	INT1_IsFlag	讀取PT1.1外部中斷旗標
26	INT1_ClearFlag	清除PT1.1外部中斷請求旗標
27	GPIO_INTEG1Sel	設置PT1.1外部中斷觸發沿
28	GPIO_PM14Sel	設置PT1.4數位的複用功能
29	GPIO_PM15Sel	設置PT1.5數位的複用功能
30	GPIO_PM16Sel	設置PT1.6數位的複用功能
31	GPIO_PM17Sel	設置PT1.7數位的複用功能
32	GPIO_PT2OutputMode	設置PT2作為輸出模式
33	GPIO_PT2OutputHigh	設置PT2對應IO輸出高電位
34	GPIO_PT2OutputLow	設置PT2對應IO輸出低電位
35	GPIO_PT2InputMode	設置PT2對應IO作為輸入模式.
36	GPIO_PT2InputPullHight	使能PT2 對應IO的輸入上拉電阻
37	GPIO_PT2InputPullHightClear	關閉PT2對應IO pin的輸入上拉電阻
38	GPIO_GetPT2Data	讀取PT2輸入狀態

序號	函數名稱	功能描述
39	GPIO_PM22Sel	設置PT2.2數位的複用功能
40	GPIO_PM23Sel	設置PT2.3數位的複用功能
41	GPIO_PM24Sel	設置PT2.4數位的複用功能
42	GPIO_PM25Sel	設置PT2.5數位的複用功能
43	GPIO_PM26Sel	設置PT2.6數位的複用功能
44	GPIO_PM27Sel	設置PT2.7數位的複用功能
45	GPIO_PT3OutputMode	設置PT3作為輸出模式
46	GPIO_PT3OutputHigh	設置PT3對應IO輸出高電位
47	GPIO_PT3OutputLow	設置PT3對應IO輸出低電位
48	GPIO_PT3InputMode	設置PT3對應IO作為輸入模式
49	GPIO_PT3InputPullHight	使能PT3 對應IO的輸入上拉電阻
50	GPIO_PT3InputPullHightClear	關閉PT3對應IO pin的輸入上拉電阻
51	GPIO_GetPT3Data	讀取PT3輸入狀態
52	GPIO_PT4InputPullHight	使能PT4 對應IO的輸入上拉電阻
53	GPIO_PT4InputPullHightClear	關閉PT4對應IO pin的輸入上拉電阻
54	GPIO_GetPT4Data	讀取PT4輸入狀態
55	GPIO_PT4AnalogMode	使能PT4類比的功能
56	GPIO_PT4DigitalMode	使能PT4數位的功能
57	GPIO_PT5InputPullHight	使能PT5 對應IO的輸入上拉電阻
58	GPIO_PT5InputPullHightClear	關閉PT5對應IO pin的輸入上拉電阻
59	GPIO_GetPT5Data	讀取PT5輸入狀態
60	GPIO_PT5AnalogMode	使能PT5類比的功能
61	GPIO_PT5DigitalMode	使能PT5數位的功能
62	GPIO_INT0_Enable	使能PT1.0外部中斷功能
63	GPIO_INT0_Disable	關閉PT1.0外部中斷功能
64	GPIO_INT1_Enable	使能PT1.1外部中斷功能
65	GPIO_INT1_Disable	關閉PT1.1外部中斷功能
66	GPIO_INT_TYPE_PT10	設置PT1.0外部中斷觸發沿
67	GPIO_INT_TYPE_PT11	設置PT1.1外部中斷觸發沿
68	GPIO_INT_Low2BitEnable	使能PT1.1/PT1.0外部中斷功能
69	GPIO_AI0Enable	使能PT4.0 (AI0) 類比的功能
70	GPIO_AI1Enable	使能PT4.1 (AI1) 類比的功能
71	GPIO_AI2Enable	使能PT4.2 (AI2) 類比的功能
72	GPIO_AI3Enable	使能PT4.3 (AI3) 類比的功能
73	GPIO_AI4Enable	使能PT4.4 (AI4) 類比的功能
74	GPIO_AI5Enable	使能PT4.5 (AI5) 類比的功能
75	GPIO_AI6Enable	使能PT4.6 (AI6) 類比的功能
76	GPIO_AI7Enable	使能PT4.7 (AI7) 類比的功能
77	GPIO_AI8Enable	使能PT5.0 (AI8) 類比的功能
78	GPIO_AI9Enable	使能PT5.1 (AI9) 類比的功能
79	GPIO_AI10Enable	使能PT5.2 (AI10) 類比的功能
80	GPIO_AI11Enable	使能PT5.3 (AI11) 類比的功能
81	GPIO_CPAI0Enable	使能PT2.2複用為比較器信號輸入端CPAI0
82	GPIO_CPAI0Disable	關閉PT2.2複用為比較器信號輸入端CPAI0
83	GPIO_CPAI1Enable	使能PT2.3複用為比較器信號輸入端CPAI1
84	GPIO_CPAI1Disable	關閉PT2.3複用為比較器信號輸入端CPAI1
85	GPIO_CPAI2Enable	使能PT2.4複用為比較器信號輸入端CPAI2
86	GPIO_CPAI2Disable	關閉PT2.4複用為比較器信號輸入端CPAI2
87	GPIO_CPAI3Enable	使能PT2.5複用為比較器信號輸入端CPAI3
88	GPIO_CPAI3Disable	關閉PT2.5複用為比較器信號輸入端CPAI3
89	GPIO_CPAI4Enable	使能PT2.6複用為比較器信號輸入端CPAI4

序號	函數名稱	功能描述
90	GPIO_CPAI4Disable	關閉PT2.6複用為比較器信號輸入端CPAI4
91	GPIO_CPAI5Enable	使能PT2.7複用為比較器信號輸入端CPAI5
92	GPIO_CPAI5Disable	關閉PT2.7複用為比較器信號輸入端CPAI5
93	GPIO_CPAI6Enable	使能PT1.0複用為比較器信號輸入端CPAI6
94	GPIO_CPAI6Disable	關閉PT1.0複用為比較器信號輸入端CPAI6
95	GPIO_CPAI7Enable	使能PT1.1複用為比較器信號輸入端CPAI7
96	GPIO_CPAI7Disable	關閉PT1.1複用為比較器信號輸入端CPAI7

5.2. GPIO 模組方框圖



5.3. 函數說明

5.3.1. GPIO_OpenPT1Input

- 函數

```
void GPIO_OpenPT1Input(unsigned char InputBits, unsigned char PullHighBits);
```

- 函數功能

設置PT1對應引腳作為GPIO的輸入模式及使能上拉電阻，設置PT1暫存器TRISC1/PT1PU/PT1DA。

- 輸入參數

InputBitst [in]：代表GPIO port，PT1.0~PT1.3預設為輸入模式，輸入參數如下

TRISC1_TC14_INPUT：PT1.4作為輸入模式

TRISC1_TC15_INPUT：PT1.5作為輸入模式

TRISC1_TC16_INPUT：PT1.6作為輸入模式

TRISC1_TC17_INPUT：PT1.7作為輸入模式

注意：PT1.0~PT1.3預設為輸入模式，不用設置

PullHighBits [in]：代表 GPIO port，設置輸入上拉電阻，輸入參數如下

PT1PU_PU10_ENABLE：使能PT1.0上拉電阻； PT1PU_PU10_DISABLE：關閉PT1.0上拉電阻

PT1PU_PU11_ENABLE：使能PT1.1上拉電阻； PT1PU_PU11_DISABLE：關閉PT1.1上拉電阻

PT1PU_PU12_ENABLE：使能PT1.2上拉電阻； PT1PU_PU12_DISABLE：關閉PT1.2上拉電阻

PT1PU_PU13_ENABLE：使能PT1.3上拉電阻； PT1PU_PU13_DISABLE：關閉PT1.3上拉電阻

PT1PU_PU14_ENABLE：使能PT1.4上拉電阻； PT1PU_PU14_DISABLE：關閉PT1.4上拉電阻

PT1PU_PU15_ENABLE：使能PT1.5上拉電阻； PT1PU_PU15_DISABLE：關閉PT1.5上拉電阻

PT1PU_PU16_ENABLE：使能PT1.6上拉電阻； PT1PU_PU16_DISABLE：關閉PT1.6上拉電阻

PT1PU_PU17_ENABLE：使能PT1.7上拉電阻； PT1PU_PU17_DISABLE：關閉PT1.7上拉電阻

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 設置PT1.4/PT1.5作為輸入模式及使能輸入上拉電阻 */
```

```
GPIO_OpenPT1Input(TRISC1_TC14_INPUT, PT1PU_PU14_ENABLE); //PT1.4打開輸入模式及使能上拉
```

```
GPIO_OpenPT1Input(TRISC1_TC15_INPUT, PT1PU_PU15_ENABLE); //PT1.5打開輸入模式及使能上拉
```

5.3.2. GPIO_OpenPT2Input

- 函數

```
void GPIO_OpenPT2Input(unsigned char InputBits, unsigned char PullHighBits);
```

- 函數功能

設置PT2對應引腳作為GPIO的輸入模式及使能上拉電阻，設置PT2暫存器TRISC2/PT2PU。

- 輸入參數

InputBitst [in]：代表GPIO port PT2，輸入參數如下

TRISC2_TC20_INPUT : PT2.0作為輸入模式
TRISC2_TC21_INPUT : PT2.1作為輸入模式
TRISC2_TC22_INPUT : PT2.2作為輸入模式
TRISC2_TC23_INPUT : PT2.3作為輸入模式
TRISC2_TC24_INPUT : PT2.4作為輸入模式
TRISC2_TC25_INPUT : PT2.5作為輸入模式
TRISC2_TC26_INPUT : PT2.6作為輸入模式
TRISC2_TC27_INPUT : PT2.7作為輸入模式

PullHighBits [in] : 代表 GPIO port PT2，設置輸入上拉電阻，輸入參數如下

PT2PU_PU20_ENABLE : 使能PT2.0上拉電阻 ; PT2PU_PU20_DISABLE : 關閉PT2.0上拉電阻
PT2PU_PU21_ENABLE : 使能PT2.1上拉電阻 ; PT2PU_PU21_DISABLE : 關閉PT2.1上拉電阻
PT2PU_PU22_ENABLE : 使能PT2.2上拉電阻 ; PT2PU_PU22_DISABLE : 關閉PT2.2上拉電阻
PT2PU_PU23_ENABLE : 使能PT2.3上拉電阻 ; PT2PU_PU23_DISABLE : 關閉PT2.3上拉電阻
PT2PU_PU24_ENABLE : 使能PT2.4上拉電阻 ; PT2PU_PU24_DISABLE : 關閉PT2.4上拉電阻
PT2PU_PU25_ENABLE : 使能PT2.5上拉電阻 ; PT2PU_PU25_DISABLE : 關閉PT2.5上拉電阻
PT2PU_PU26_ENABLE : 使能PT2.6上拉電阻 ; PT2PU_PU26_DISABLE : 關閉PT2.6上拉電阻
PT2PU_PU27_ENABLE : 使能PT2.7上拉電阻 ; PT2PU_PU27_DISABLE : 關閉PT2.7上拉電阻

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 設置PT2.4/PT2.5作為輸入模式及使能輸入上拉電阻 */
```

```
GPIO_OpenPT2Input(TRISC2_TC24_INPUT, PT2PU_PU24_ENABLE); //PT2.4打開輸入模式及使能上拉
```

```
GPIO_OpenPT2Input(TRISC2_TC25_INPUT, PT2PU_PU25_ENABLE); //PT2.5打開輸入模式及使能上拉
```

5.3.3. GPIO_OpenPT3Input

- 函數

```
void GPIO_OpenPT3Input(unsigned char InputBits, unsigned char PullHighBits);
```

- 函數功能

設置PT3對應引腳作為GPIO的輸入模式及使能上拉電阻，設置PT3暫存器TRISC3/PT3PU。

- 輸入參數

InputBits [in] : 代表GPIO port PT3，輸入參數如下

TRISC3_TC30_INPUT : PT3.0作為輸入模式
TRISC3_TC31_INPUT : PT3.1作為輸入模式
TRISC3_TC32_INPUT : PT3.2作為輸入模式
TRISC3_TC33_INPUT : PT3.3作為輸入模式
TRISC3_TC34_INPUT : PT3.4作為輸入模式
TRISC3_TC35_INPUT : PT3.5作為輸入模式

TRISC3_TC36_INPUT : PT3.6作為輸入模式

TRISC3_TC37_INPUT : PT3.7作為輸入模式

PullHighBits [in] : 代表 GPIO port PT3，設置輸入上拉電阻，輸入參數如下

PT3PU_PU30_ENABLE : 使能PT3.0上拉電阻 ; PT3PU_PU30_DISABLE : 關閉PT3.0上拉電阻

PT3PU_PU31_ENABLE : 使能PT3.1上拉電阻 ; PT3PU_PU31_DISABLE : 關閉PT3.1上拉電阻

PT3PU_PU32_ENABLE : 使能PT3.2上拉電阻 ; PT3PU_PU32_DISABLE : 關閉PT3.2上拉電阻

PT3PU_PU33_ENABLE : 使能PT3.3上拉電阻 ; PT3PU_PU33_DISABLE : 關閉PT3.3上拉電阻

PT3PU_PU34_ENABLE : 使能PT3.4上拉電阻 ; PT3PU_PU34_DISABLE : 關閉PT3.4上拉電阻

PT3PU_PU35_ENABLE : 使能PT3.5上拉電阻 ; PT3PU_PU35_DISABLE : 關閉PT3.5上拉電阻

PT3PU_PU36_ENABLE : 使能PT3.6上拉電阻 ; PT3PU_PU36_DISABLE : 關閉PT3.6上拉電阻

PT3PU_PU37_ENABLE : 使能PT3.7上拉電阻 ; PT3PU_PU37_DISABLE : 關閉PT3.7上拉電阻

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 設置PT3.4/PT3.5作為輸入模式及使能輸入上拉電阻 */
```

```
GPIO_OpenPT3Input(TRISC3_TC34_INPUT, PT3PU_PU34_ENABLE); //PT3.4打開輸入模式及使能上拉
```

```
GPIO_OpenPT3Input(TRISC3_TC35_INPUT, PT3PU_PU35_ENABLE); //PT3.5打開輸入模式及使能上拉
```

5.3.4. GPIO_OpenPT4Input

- 函數

```
void GPIO_OpenPT4Input(unsigned char InputBits, unsigned char PullHighBits);
```

- 函數功能

設置PT4對應引腳作為GPIO的輸入模式及使能上拉電阻，設置PT4暫存器PT4DA/PT4PU。

- 輸入參數

InputBitst [in] : 代表GPIO port PT4，PT4預設作為輸入，需要關閉類比類功能，輸入參數如下

PT4DA_DA40_DIG : 設置PT4.0作為數位口

PT4DA_DA41_DIG : 設置PT4.1作為數位口

PT4DA_DA42_DIG : 設置PT4.2作為數位口

PT4DA_DA43_DIG : 設置PT4.3作為數位口

PT4DA_DA44_DIG : 設置PT4.4作為數位口

PT4DA_DA45_DIG : 設置PT4.5作為數位口

PT4DA_DA46_DIG : 設置PT4.6作為數位口

PT4DA_DA47_DIG : 設置PT4.7作為數位口

PullHighBits [in] : 代表 GPIO port PT4，設置輸入上拉電阻，輸入參數如下

PT4PU_PU40_ENABLE : 使能PT4.0上拉電阻 ; PT4PU_PU40_DISABLE : 關閉PT4.0上拉電阻

PT4PU_PU41_ENABLE : 使能PT4.1上拉電阻 ; PT4PU_PU41_DISABLE : 關閉PT4.1上拉電阻

PT4PU_PU42_ENABLE : 使能PT4.2上拉電阻 ; PT4PU_PU42_DISABLE : 關閉PT4.2上拉電阻

PT4PU_PU43_ENABLE : 使能PT4.3上拉電阻 ; PT4PU_PU43_DISABLE : 關閉PT4.3上拉電阻
PT4PU_PU44_ENABLE : 使能PT4.4上拉電阻 ; PT4PU_PU44_DISABLE : 關閉PT4.4上拉電阻
PT4PU_PU45_ENABLE : 使能PT4.5上拉電阻 ; PT4PU_PU45_DISABLE : 關閉PT4.5上拉電阻
PT4PU_PU46_ENABLE : 使能PT4.6上拉電阻 ; PT4PU_PU46_DISABLE : 關閉PT4.6上拉電阻
PT4PU_PU47_ENABLE : 使能PT4.7上拉電阻 ; PT4PU_PU47_DISABLE : 關閉PT4.7上拉電阻

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 設置PT4.4/PT4.5作為輸入模式及使能輸入上拉電阻 */

GPIO_OpenPT4Input(PT4DA_DA44_DIG, PT4PU_PU44_ENABLE); //PT4.4打開輸入模式及使能上拉

GPIO_OpenPT4Input(PT4DA_DA45_DIG, PT4PU_PU45_ENABLE); //PT4.5打開輸入模式及使能上拉

5.3.5. GPIO_OpenPT5Input

- 函數

void GPIO_OpenPT5Input(unsigned char InputBits, unsigned char PullHighBits);

- 函數功能

設置PT5對應引腳作為GPIO的輸入模式及使能上拉電阻，設置PT5暫存器PT5DA/PT5PU。

- 輸入參數

InputBits [in] : 代表GPIO port PT5，PT4預設作為輸入口，需要關閉類比類功能，輸入參數如下

PT5DA_DA50_DIG : 設置PT5.0作為數位口

PT5DA_DA51_DIG : 設置PT5.1作為數位口

PT5DA_DA52_DIG : 設置PT5.2作為數位口

PT5DA_DA53_DIG : 設置PT5.3作為數位口

PullHighBits [in] : 代表 GPIO port PT5，設置輸入上拉電阻，輸入參數如下

PT5PU_PU50_ENABLE : 使能PT5.0上拉電阻 ; PT5PU_PU50_DISABLE : 關閉PT5.0上拉電阻

PT5PU_PU51_ENABLE : 使能PT5.1上拉電阻 ; PT5PU_PU51_DISABLE : 關閉PT5.1上拉電阻

PT5PU_PU52_ENABLE : 使能PT5.2上拉電阻 ; PT5PU_PU52_DISABLE : 關閉PT5.2上拉電阻

PT5PU_PU53_ENABLE : 使能PT5.3上拉電阻 ; PT5PU_PU53_DISABLE : 關閉PT5.3上拉電阻

PT5PU_PU54_ENABLE : 使能PT5.4上拉電阻 ; PT5PU_PU54_DISABLE : 關閉PT5.4上拉電阻

PT5PU_PU55_ENABLE : 使能PT5.5上拉電阻 ; PT5PU_PU55_DISABLE : 關閉PT5.5上拉電阻

PT5PU_PU56_ENABLE : 使能PT5.6上拉電阻 ; PT5PU_PU56_DISABLE : 關閉PT5.6上拉電阻

PT5PU_PU57_ENABLE : 使能PT5.7上拉電阻 ; PT5PU_PU57_DISABLE : 關閉PT5.7上拉電阻

- 包含標頭檔

Driver/GPIO.c

- 函數返回值

無

- 函數用法

```
/* 設置PT5.4/PT5.5作為輸入模式及使能輸入上拉電阻 */
```

```
GPIO_OpenPT5Input(PT4DA_DA54_DIG, PT5PU_PU54_ENABLE); //PT5.4打開輸入模式及使能上拉
```

```
GPIO_OpenPT5Input(PT4DA_DA55_DIG, PT5PU_PU55_ENABLE); //PT5.5打開輸入模式及使能上拉
```

5.3.6. GPIO_PT1OutputMode

- 函數

```
GPIO_PT1OutputMode(BitSet);
```

- 函數功能

設置PT1作為輸出模式，操作暫存器TRISC1。

- 輸入參數

BitSet [in]：代表GPIO port PT1，PT1.0~PT1.3默認為輸入，設置參數如下

TRISC1_TC14_OUTPUT：PT1.4作為輸出模式

TRISC1_TC15_OUTPUT：PT1.5作為輸出模式

TRISC1_TC16_OUTPUT：PT1.6作為輸出模式

TRISC1_TC17_OUTPUT：PT1.7作為輸出模式

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 設置PT1.5/PT1.4的輸出模式 */
```

```
GPIO_PT1OutputMode( TRISC1_TC14_OUTPUT );
```

```
GPIO_PT1OutputMode( TRISC1_TC15_OUTPUT );
```

```
//或是如下用法
```

```
GPIO_PT1OutputMode( TRISC1_TC15_OUTPUT | TRISC1_TC14_OUTPUT );
```

5.3.7. GPIO_PT1OutputHigh

- 函數

```
GPIO_PT1OutputHigh(BitSet);
```

- 函數功能

設置PT1對應IO輸出高電位，操作暫存器PT1[7 :4]。

- 輸入參數

BitSet [in]：代表GPIO port PT1，8bit數據bit0~bit7分別對應PT1.0~PT1.7，輸入參數如下

PT1_PT14_H：PT1.4輸出高電位

PT1_PT15_H：PT1.5輸出高電位

PT1_PT16_H：PT1.6輸出高電位

PT1_PT17_H：PT1.7輸出高電位

注意：PT1.0~PT1.3預設是輸出模式，所以不能設置輸出及輸出高電位

- 包含標頭檔

Driver/GPIO.h

● 函數返回值

無

● 函數用法

/* PT1.4輸出高電位 */

GPIO_PT1OutputMode(TRISC1_TC14_OUTPUT); //設置PT1.4作為輸出模式

GPIO_PT1OutputHigh(PT1_PT14_H); //設置PT1.4輸出高電位

5.3.8. GPIO_PT1OutputLow

● 函數

GPIO_PT1OutputLow(BitSet);

● 函數功能

設置PT1對應IO口輸出低電位，操作暫存器PT1[7:4]。

● 輸入參數

BitSet [in]：代表GPIO port PT1，8bit數據bit0~bit7分別對應PT1.0~PT1.7，輸入參數如下

PT1_PT14_L：PT1.4輸出低電位

PT1_PT15_L：PT1.5輸出低電位

PT1_PT16_L：PT1.6輸出低電位

PT1_PT17_L：PT1.7輸出低電位

注意：PT1.0~PT1.3預設是輸出模式，所以不能設置輸出及輸出低電位

● 包含標頭檔

Driver/GPIO.h

● 函數返回值

無

● 函數用法

/* PT1.4輸出低電位 */

GPIO_PT1OutputMode(TRISC1_TC14_OUTPUT); //設置PT1.4作為輸出模式

GPIO_PT1OutputHigh(PT1_PT14_L); //設置PT1.4輸出低電位

5.3.9. GPIO_PT1InputMode

● 函數

GPIO_PT1InputMode(BitSet);

● 函數功能

設置PT1對應IO口作為輸入模式，設置暫存器TRISC1。

● 輸入參數

BitSet [in]：代表PT1對應IO pin，PT1.0~PT1.3默認是輸入口，只需針對PT1.4~PT1.7設置，輸入參數如下

TRISC1_TC14_INPUT：PT1.4作為輸入模式

TRISC1_TC15_INPUT：PT1.5作為輸入模式

TRISC1_TC16_INPUT：PT1.6作為輸入模式

TRISC1_TC17_INPUT : PT1.7作為輸入模式

注意：PT1.0~PT1.3預設為輸入模式，不用設置

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 設置PT1.4作為輸入口 */

```
GPIO_PT1InputMode( TRISC1_TC14_INPUT );
```

5.3.10. GPIO_PT1InputPullHight

- 函數

```
GPIO_PT1InputPullHight(BitSet);
```

- 函數功能

使能PT1 對應IO口的輸入上拉電阻，設置暫存器PT1PU。

- 輸入參數

BitSet [in] : 代表GPIO PT1，輸入值如下

PT1PU_PU10_ENABLE : 使能PT1.0上拉電阻

PT1PU_PU11_ENABLE : 使能PT1.1上拉電阻

PT1PU_PU12_ENABLE : 使能PT1.2上拉電阻

PT1PU_PU13_ENABLE : 使能PT1.3上拉電阻

PT1PU_PU14_ENABLE : 使能PT1.4上拉電阻

PT1PU_PU15_ENABLE : 使能PT1.5上拉電阻

PT1PU_PU16_ENABLE : 使能PT1.6上拉電阻

PT1PU_PU17_ENABLE : 使能PT1.7上拉電阻

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 使能 PT1.4 輸入上拉電阻 */

```
GPIO_PT1InputPullHight( PT1PU_PU14_ENABLE );
```

5.3.11. GPIO_PT1InputPullHightClear

- 函數

```
GPIO_PT1InputPullHightClear(BitSet);
```

- 函數功能

關閉PT1對應IO pin的輸入上拉電阻，設置暫存器PT1PU。

- 輸入參數

BitSet [in] : 代表GPIO port PT1輸入值如下

PT1PU_PU10_DISABLE : 關閉PT1.0上拉電阻
PT1PU_PU11_DISABLE : 關閉PT1.1上拉電阻
PT1PU_PU12_DISABLE : 關閉PT1.2上拉電阻
PT1PU_PU13_DISABLE : 關閉PT1.3上拉電阻
PT1PU_PU14_DISABLE : 關閉PT1.4上拉電阻
PT1PU_PU15_DISABLE : 關閉PT1.5上拉電阻
PT1PU_PU16_DISABLE : 關閉PT1.6上拉電阻
PT1PU_PU17_DISABLE : 關閉PT1.7上拉電阻

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉 PT1.4 輸入上拉電阻 */  
GPIO_PT1InputPullHightClear( PT1PU_PU14_DISABLE );
```

5.3.12. GPIO_GetPT1Data

- 函數

GPIO_GetPT1Data(PT1Data);

- 函數功能

讀取PT1輸入狀態，讀取暫存器PT1。

- 輸入參數

PT1Data [in] : 用於存放讀取到的PT1輸入狀態值

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

PT1Data : 返回 PT1 輸入狀態值

- 函數用法

```
/* 讀取 PT1 輸入狀態值 */  
unsigned char PT1_DATA;  
GPIO_GetPT1Data( PT1_DATA );
```

5.3.13. GPIO_PT1AnalogMode

- 函數

GPIO_PT1AnalogMode(BitSet);

- 函數功能

使能PT1.0/PT1.1/PT1.2類比的功能，設置暫存器PT1DA[2:0]。

- 輸入參數

BitSet [in] : 代表GPIO port PT1輸入值如下

PT1DA_DA10_CPAI6 : PT1.0作為CPAI6輸入口

PT1DA_DA11_CPAI7 : PT1.1作為CPAI7輸入口

PT1DA_DA12_LVDIN : PT1.2作為LVD輸入引腳

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 使能PT1.2的類比的功能 */

GPIO_PT1AnalogMode(PT1DA_DA12_LVDIN); //使能PT1.2類比的功能 ;

5.3.14. GPIO_PT1DigitalMode

- 函數

GPIO_PT1DigitalMode(BitSet);

- 函數功能

使能PT1.0/PT1.1/PT1.2數位的功能，設置暫存器PT1DA[2:0]。

- 輸入參數

BitSet [in] : 代表GPIO port PT1，輸入值如下

PT1DA_DA10_DIG : 使能PT1.0數位的功能

PT1DA_DA11_DIG : 使能PT1.1數位的功能

PT1DA_DA12_DIG : 使能PT1.2數位的功能

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 使能PT1.2 數位的功能 */

GPIO_PT1DigitalMode(PT1DA_DA12_DIG); //使能PT1.2數位的功能

5.3.15. BZ_Open

- 函數

void BZ_Open(unsigned char cks);

- 函數功能

設置蜂鳴器驅動頻率預分頻及設置PT1.7作為輸出模式，PT10暫存器0X40890[19:18][3:2]。

- 輸入參數

cks [in] : 設置蜂鳴器驅動頻率預分頻

MCKCN3_BZS_PERCKDIV128 : 蜂鳴器驅動頻率為PER_CK/128

MCKCN3_BZS_PERCKDIV64 : 蜂鳴器驅動頻率為PER_CK/64

MCKCN3_BZS_PERCKDIV32 : 蜂鳴器驅動頻率為PER_CK/32

MCKCN3_BZS_PERCKDIV16 : 蜂鳴器驅動頻率為PER_CK/16

MCKCN3_BZS_PERCKDIV8 : 蜂鳴器驅動頻率為PER_CK/8

MCKCN3_BZS_PERCKDIV4 : 蜂鳴器驅動頻率為PER_CK/4

MCKCN3_BZS_PERCKDIV3 : 蜂鳴器驅動頻率為PER_CK/3
MCKCN3_BZS_PERCKDIV1 : 蜂鳴器驅動頻率為PER_CK/1

● 包含標頭檔

Driver/GPIO.h

● 函數返回值

無

● 函數用法

/* 設置蜂鳴器驅動頻率為PER_CK/16 */

BZ_Open(MCKCN3_BZS_PERCKDIV16); //設置蜂鳴器驅動頻率16分頻

5.3.16. BZ_OutEnable

● 函數

BZ_OutEnable();

● 函數功能

使能BZ蜂鳴器功能，設置暫存器P1M2[6]=1。

● 輸入參數

無

● 包含標頭檔

Driver/GPIO.h

● 函數返回值

無

● 函數用法

/* 使能BZ輸出功能 */

BZ_Open(MCKCN3_BZS_PERCKDIV16); //設置蜂鳴器驅動頻率16分頻
BZ_OutEnable();

5.3.17. BZ_OutDisable

● 函數

BZ_OutDisable();

● 函數功能

關閉BZ蜂鳴器輸出功能，設置暫存器PT1M2[6]=0。

● 輸入參數

無

● 包含標頭檔

Driver/GPIO.h

● 函數返回值

無

● 函數用法

/* 關閉BZ輸出功能 */


```
BZ_OutDisable(); //關閉蜂鳴器輸出功能
GPIO_PT1InputMode( TRISC1_TC17_INPUT ); //設置PT1.7作為輸入
```

5.3.18. INT0_Enable

- 函數

```
INT0_Enable();
```

- 函數功能

使能PT1.0外部中斷功能，設置暫存器INTE1[0]=1。

- 輸入參數

無。

- 包含標頭檔

Driver/INT.h

- 函數返回值

無

- 函數用法

```
/* 使能PT1.0外部中斷功能 */
```

```
INT0_Enable();
```

5.3.19. INT0_Disable

- 函數

```
INT0_Disable();
```

- 函數功能

關閉PT1.0外部中斷功能，設置暫存器INTE1[0]=0。

- 輸入參數

無

- 包含標頭檔

Driver/INT.h

- 函數返回值

無

- 函數用法

```
/* 關閉PT1.0外部中斷功能 */
```

```
INT0_Disable();
```

5.3.20. INT0_IsFlag

- 函數

```
INT0_IsFlag();
```

- 函數功能

讀取PT1.0外部中斷旗標，讀取暫存器INTF0[0]。

- 輸入參數

無

- 包含標頭檔

Driver/INT.h

- 函數返回值

0x00 : PT1.0沒有中斷請求

0x01 : PT1.0產生中斷請求

- 函數用法

```
/* 讀取PT1.0的外部中斷旗標 */
```

```
unsigned char flag;
```

```
flag = INT0_IsFlag();
```

5.3.21. INT0_ClearFlag

- 函數

```
INT0_ClearFlag();
```

- 函數功能

清除PT1.0外部中斷請求旗標，設置暫存器INTF0[0]=0。

- 輸入參數

無

- 包含標頭檔

Driver/INT.h

- 函數返回值

無

- 函數用法

```
/* 清除PT1.0外部中斷請求旗標 */
```

```
INT0_ClearFlag();
```

5.3.22. GPIO_INTEG0Sel

- 函數

```
GPIO_INTEG0Sel(EG0Sel);
```

- 函數功能

設置PT1.0外部中斷觸發沿，設置暫存器PT1M1[1:0]。

- 輸入參數

EG0Sel [in] : PT1.0外部中斷觸發沿選擇

PT1M1_INTEG0_LEV_ : 電位變化觸發

PT1M1_INTEG0_LEV : 電位變化觸發

PT1M1_INTEG0_EDGERISE : 上升沿觸發

PT1M1_INTEG0_EDGEFALL : 下降沿觸發

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 設置PT1.0外部中斷觸發沿為下降沿觸發 */

```
GPIO_INTEG0Sel( PT1M1_INTEG0_EDGEFALL );
```

5.3.23. INT1_Enable

- 函數

```
INT1_Enable();
```

- 函數功能

使能PT1.1外部中斷功能，設置暫存器INTE1[1]=1。

- 輸入參數

無。

- 包含標頭檔

Driver/INT.h

- 函數返回值

無

- 函數用法

/* 使能PT1.1外部中斷功能 */

```
INT1_Enable();
```

5.3.24. INT1_Disable

- 函數

```
INT1_Disable();
```

- 函數功能

關閉PT1.1外部中斷功能，設置暫存器INTE1[1]=0。

- 輸入參數

無

- 包含標頭檔

Driver/INT.h

- 函數返回值

無

- 函數用法

/* 關閉PT1.1外部中斷功能 */

```
INT0_Disable();
```

5.3.25. INT1_IsFlag

- 函數

```
INT1_IsFlag();
```

- 函數功能

讀取PT1.1外部中斷旗標，讀取暫存器INTF0[1]。

- 輸入參數

無

- 包含標頭檔

Driver/INT.h

- 函數返回值

0x00 : PT1.1沒有中斷請求

0x02 : PT1.1產生中斷請求

- 函數用法

```
/* 讀取PT1.1的外部中斷旗標 */
```

```
unsigned char flag;
```

```
flag = INT1_IsFlag();
```

5.3.26. INT1_ClearFlag

- 函數

```
INT1_ClearFlag();
```

- 函數功能

清除PT1.1外部中斷請求旗標，設置暫存器INTF0[1]=0。

- 輸入參數

無

- 包含標頭檔

Driver/INT.h

- 函數返回值

無

- 函數用法

```
/* 清除PT1.1外部中斷請求旗標 */
```

```
INT1_ClearFlag();
```

5.3.27. GPIO_INTEG1Sel

- 函數

```
GPIO_INTEG1Sel(EG1Sel);
```

- 函數功能

設置PT1.1外部中斷觸發沿，設置暫存器PT1M1[3:2]。

- 輸入參數

EG1Sel [in] : PT1.1外部中斷觸發沿選擇

PT1M1_INTEG1_LEV_ : 電位變化觸發

PT1M1_INTEG1_LEV : 電位變化觸發

PT1M1_INTEG1_EDGERISE : 上升沿觸發

PT1M1_INTEG1_EDGEFALL : 下降沿觸發

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 設置PT1.1外部中斷觸發沿為下降沿觸發 */

GPIO_INTEG1Sel(PT1M1_INTEG1_EDGEFALL);

5.3.28. GPIO_PM14Sel

- 函數

GPIO_PM14Sel(PM14Sel);

- 函數功能

設置PT1.4數位的複用功能，設置暫存器PT1M2[0]。

- 輸入參數

PM14Sel [in] : PT1.4複用功能選擇

PT1M2_PM14_TX : PT1.4複用為TX 引腳

PT1M2_PM14_GPIO : PT1.4複用為GPIO

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* PT1.4複用TX引腳 */

GPIO_PM14Sel(PT1M2_PM14_TX);

5.3.29. GPIO_PM15Sel

- 函數

GPIO_PM15Sel(PM15Sel);

- 函數功能

設置PT1.5數位的複用功能，設置暫存器PT1M2[2]。

- 輸入參數

PM15Sel [in] PT1.5數字類複用功能設置

PT1M2_PM15_SDO : PT1.5複用SPI通訊的SDO引腳

PT1M2_PM15_GPIO : PT1.5複用為GPIO

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* PT1.5複用為SPI的SDO引腳 */  
GPIO_PM15Sel( PT1M2_PM15_SDO );
```

5.3.30. GPIO_PM16Sel

- 函數

```
GPIO_PM16Sel(PM16Sel);
```

- 函數功能

設置PT1.6數位的複用功能，設置暫存器PT1M2[4]。

- 輸入參數

```
PM16Sel [in] : PT1.6複用功能設置  
PT1M2_PM16_SCK : PT1.6複用為SPI通訊引腳SCK  
PT1M2_PM16_GPIO : PT1.6複用GPIO
```

- 包含標頭檔

```
Driver/GPIO.h
```

- 函數返回值

無

- 函數用法

```
/* PT1.6複用為SPI的SCK引腳 */  
GPIO_PM16Sel( PT1M2_PM16_SCK );
```

5.3.31. GPIO_PM17Sel

- 函數

```
GPIO_PM17Sel(PM17Sel);
```

- 函數功能

設置PT1.7數位的複用功能，設置暫存器PT1M2[6]。

- 輸入參數

```
PM17Sel [in] : PT1.7複用功能設置  
PT1M2_PM17_BZ : PT1.7複用為蜂鳴器驅動引腳  
PT1M2_PM17_GPIO : PT1.7複用為GPIO
```

- 包含標頭檔

```
Driver/GPIO.h
```

- 函數返回值

無

- 函數用法

```
/* 設置PT1.7作為BZ驅動引腳 */  
GPIO_PM17Sel( PT1M2_PM17_BZ );
```

5.3.32. GPIO_PT2OutputMode

- 函數

GPIO_PT2OutputMode(BitSet);

- 函數功能

設置PT2作為輸出模式，操作暫存器TRISC2。

- 輸入參數

BitSet [in]：代表GPIO port PT2，設置參數如下

TRISC2_TC20_OUTPUT：PT2.0作為輸出模式

TRISC2_TC21_OUTPUT：PT2.1作為輸出模式

TRISC2_TC22_OUTPUT：PT2.2作為輸出模式

TRISC2_TC23_OUTPUT：PT2.3作為輸出模式

TRISC2_TC24_OUTPUT：PT2.4作為輸出模式

TRISC2_TC25_OUTPUT：PT2.5作為輸出模式

TRISC2_TC26_OUTPUT：PT2.6作為輸出模式

TRISC2_TC27_OUTPUT：PT2.7作為輸出模式

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 設置PT2.5/PT2.4的輸出模式 */

```
GPIO_PT2OutputMode( TRISC2_TC24_OUTPUT );
```

```
GPIO_PT2OutputMode( TRISC2_TC25_OUTPUT );
```

//或改成以下用法

```
GPIO_PT2OutputMode( TRISC2_TC25_OUTPUT | TRISC2_TC24_OUTPUT );
```

5.3.33. GPIO_PT2OutputHigh

- 函數

GPIO_PT2OutputHigh(BitSet);

- 函數功能

設置PT2對應IO輸出高電位，操作暫存器PT2[7:0]。

- 輸入參數

BitSet [in]：代表GPIO port PT2，8bit數據bit0~bit7分別對應PT2.0~PT2.7，輸入參數如下

PT2_PT20_H：PT2.0輸出高電位

PT2_PT21_H：PT2.1輸出高電位

PT2_PT22_H：PT2.2輸出高電位

PT2_PT23_H：PT2.3輸出高電位

PT2_PT24_H：PT2.4輸出高電位

PT2_PT25_H：PT2.5輸出高電位

PT2_PT26_H：PT2.6輸出高電位

PT2_PT27_H : PT2.7輸出高電位

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* PT2.4輸出高電位 */

```
GPIO_PT2OutputMode( TRISC2_TC24_OUTPUT );           //設置PT2.4作為輸出模式
```

```
GPIO_PT2OutputHigh( PT2_PT24_H );                   //設置PT2.4輸出高電位
```

5.3.34. GPIO_PT2OutputLow

- 函數

```
GPIO_PT2OutputLow(BitSet);
```

- 函數功能

設置PT2對應IO口輸出低電位，操作暫存器PT2[7:0]。

- 輸入參數

BitSet [in] : 代表GPIO port PT2，8bit數據bit0~bit7分別對應PT2.0~PT2.7，輸入參數如下

PT2_PT20_L : PT2.0輸出低電位

PT2_PT21_L : PT2.1輸出低電位

PT2_PT22_L : PT2.2輸出低電位

PT2_PT23_L : PT2.3輸出低電位

PT2_PT24_L : PT2.4輸出低電位

PT2_PT25_L : PT2.5輸出低電位

PT2_PT26_L : PT2.6輸出低電位

PT2_PT27_L : PT2.7輸出低電位

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* PT2.4輸出低電位 */

```
GPIO_PT2OutputMode( TRISC2_TC24_OUTPUT );           //設置PT2.4作為輸出模式
```

```
GPIO_PT2OutputLow( PT2_PT24_L );                    //設置PT2.4輸出低電位
```

5.3.35. GPIO_PT2InputMode

- 函數

```
GPIO_PT2InputMode(BitSet);
```

- 函數功能

設置PT2對應IO口作為輸入模式，設置暫存器TRISC2。

● 輸入參數

BitSet [in]：代表PT2對應IO pin，針對PT2.0~PT2.7設置，輸入參數如下

TRISC2_TC20_INPUT：PT2.0作為輸入模式

TRISC2_TC21_INPUT：PT2.1作為輸入模式

TRISC2_TC22_INPUT：PT2.2作為輸入模式

TRISC2_TC23_INPUT：PT2.3作為輸入模式

TRISC2_TC24_INPUT：PT2.4作為輸入模式

TRISC2_TC25_INPUT：PT2.5作為輸入模式

TRISC2_TC26_INPUT：PT2.6作為輸入模式

TRISC2_TC27_INPUT：PT2.7作為輸入模式

● 包含標頭檔

Driver/GPIO.h

● 函數返回值

無

● 函數用法

/* 設置PT2.4作為輸入口 */

GPIO_PT2InputMode(TRISC2_TC24_INPUT);

5.3.36. GPIO_PT2InputPullHight

● 函數

GPIO_PT2InputPullHight(BitSet);

● 函數功能

使能PT2 對應IO口的輸入上拉電阻，設置暫存器PT2PU。

● 輸入參數

BitSet [in]：代表GPIO PT2，輸入值如下

PT2PU_PU20_ENABLE：使能PT2.0上拉電阻

PT2PU_PU21_ENABLE：使能PT2.1上拉電阻

PT2PU_PU22_ENABLE：使能PT2.2上拉電阻

PT2PU_PU23_ENABLE：使能PT2.3上拉電阻

PT2PU_PU24_ENABLE：使能PT2.4上拉電阻

PT2PU_PU25_ENABLE：使能PT2.5上拉電阻

PT2PU_PU26_ENABLE：使能PT2.6上拉電阻

PT2PU_PU27_ENABLE：使能PT2.7上拉電阻

● 包含標頭檔

Driver/GPIO.h

● 函數返回值

無

● 函數用法

/* 使能 PT2.4 輸入上拉電阻 */

GPIO_PT2InputPullHight(PT2PU_PU24_ENABLE);

5.3.37. GPIO_PT2InputPullHightClear

- 函數

GPIO_PT2InputPullHightClear(BitSet);

- 函數功能

關閉PT2對應IO pin的輸入上拉電阻，設置暫存器PT2PU。

- 輸入參數

BitSet [in]：代表GPIO port PT2輸入值如下

PT2PU_PU20_DISABLE：關閉PT2.0上拉電阻

PT2PU_PU21_DISABLE：關閉PT2.1上拉電阻

PT2PU_PU22_DISABLE：關閉PT2.2上拉電阻

PT2PU_PU23_DISABLE：關閉PT2.3上拉電阻

PT2PU_PU24_DISABLE：關閉PT2.4上拉電阻

PT2PU_PU25_DISABLE：關閉PT2.5上拉電阻

PT2PU_PU26_DISABLE：關閉PT2.6上拉電阻

PT2PU_PU27_DISABLE：關閉PT2.7上拉電阻

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉 PT2.4 輸入上拉電阻 */  
GPIO_PT2InputPullHightClear( PT2PU_PU24_DISABLE );
```

5.3.38. GPIO_GetPT2Data

- 函數

GPIO_GetPT2Data(PT2Data);

- 函數功能

讀取PT2輸入狀態，讀取暫存器PT2。

- 輸入參數

PT2Data [in]：用於存放讀取到的PT2輸入狀態值

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

PT2Data：返回 PT2 輸入狀態值

- 函數用法

```
/* 讀取 PT2 輸入狀態值 */  
unsigned char PT2_DATA;  
GPIO_GetPT2Data( PT2_DATA );
```

5.3.39. GPIO_PM22Sel

- 函數

GPIO_PM22Sel(PM22Sel);

- 函數功能

設置PT2.2數位的複用功能，設置暫存器PT2M1[5:4]。

- 輸入參數

PM22Sel [in]：設置PT2.2複用功能

PT2M1_PM22_PFD : PT2.2複用為PFD輸出引腳

PT2M1_PM22_PWM0 : PT2.2複用為PWM0輸出引腳

PT2M1_PM22_GPIO : PT2.2複用為GPIO

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 設置PT2.2複用為PWM0輸出引腳 */

GPIO_PM22Sel(PT2M1_PM22_PWM0);

//PT2.2複用為PWM0輸出引腳

5.3.40. GPIO_PM23Sel

- 函數

GPIO_PM23Sel(PM23Sel);

- 函數功能

設置PT2.3數位的複用功能，設置暫存器PT2M1[6]。

- 輸入參數

PM23Sel [in]：設置PT2.3複用功能

PT2M1_PM23_PWM1 : PT2.3複用為PWM1輸出引腳

PT2M1_PM23_GPIO : PT2.3複用為GPIO

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 設置PT2.3複用為PWM1輸出引腳 */

GPIO_PM23Sel(PT2M1_PM23_PWM1);

//PT2.3複用為PWM1輸出引腳

5.3.41. GPIO_PM24Sel

- 函數

GPIO_PM24Sel(PM24Sel);

- 函數功能

設置PT2.4數位的複用功能，設置暫存器PT2M2[1:0]。

- 輸入參數

PM24Sel [in] : 設置PT2.4複用功能

PT2M2_PM24_CCP0 : PT2.4複用為CCP0引腳

PT2M2_PM24_PWM2 : PT2.4複用為PWM2輸出引腳

PT2M2_PM24_GPIO : PT2.4複用為GPIO

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 設置PT2.4複用為PWM2輸出引腳 */

GPIO_PM24Sel(PT2M2_PM24_PWM2); //PT2.4複用為PWM2輸出引腳

5.3.42. GPIO_PM25Sel

- 函數

GPIO_PM25Sel(PM25Sel);

- 函數功能

設置PT2.5數位的複用功能，設置暫存器PT2M2[3:2]。

- 輸入參數

PM25Sel [in] : 設置PT2.5複用功能

PT2M2_PM25_CCP1 : PT2.5複用為CCP1引腳

PT2M2_PM25_PWM3 : PT2.5複用為PWM3輸出引腳

PT2M2_PM25_GPIO : PT2.5複用為GPIO

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 設置PT2.5複用為PWM3輸出引腳 */

GPIO_PM25Sel(PT2M2_PM25_PWM3); //PT2.5複用為PWM3輸出引腳

5.3.43. GPIO_PM26Sel

- 函數

GPIO_PM26Sel(PM26Sel);

- 函數功能

設置PT2.6數位的複用功能，設置暫存器PT2M2[4]。

- 輸入參數

PM26Sel [in] : 設置PT2.6複用功能

PT2M2_PM26_CPAO : PT2.6複用為CPAO引腳

PT2M2_PM26_GPIO : PT2.6複用為GPIO

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 設置PT2.6複用為CPAO輸出引腳 */
```

```
GPIO_PM26Sel( PT2M2_PM26_CPAO );
```

```
//PT2.6複用為CPAO輸出引腳
```

5.3.44. GPIO_PM27Sel

- 函數

```
GPIO_PM27Sel(PM27Sel);
```

- 函數功能

設置PT2.7數位的複用功能，設置暫存器PT2M2[6]。

- 輸入參數

PM27Sel [in] : 設置PT2.7複用功能

PT2M2_PM27_CPAO : PT2.7複用為CPAO引腳

PT2M2_PM27_GPIO : PT2.7複用為GPIO

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 設置PT2.7複用為CPAO輸出引腳 */
```

```
GPIO_PM27Sel( PT2M2_PM27_CPAO );
```

```
//PT2.7複用為CPAO輸出引腳
```

5.3.45. GPIO_PT3OutputMode

- 函數

```
GPIO_PT3OutputMode(BitSet);
```

- 函數功能

設置PT3作為輸出模式，操作暫存器TRISC3。

- 輸入參數

BitSet [in] : 代表GPIO port PT3，設置參數如下

TRISC3_TC30_OUTPUT : PT3.0作為輸出模式

TRISC3_TC31_OUTPUT : PT3.1作為輸出模式

TRISC3_TC32_OUTPUT : PT3.2作為輸出模式

TRISC3_TC33_OUTPUT : PT3.3作為輸出模式

TRISC3_TC34_OUTPUT : PT3.4作為輸出模式

TRISC3_TC35_OUTPUT : PT3.5作為輸出模式

TRISC3_TC36_OUTPUT : PT3.6作為輸出模式

TRISC3_TC37_OUTPUT : PT3.7作為輸出模式

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 設置PT3.5/PT3.4的輸出模式 */

```
GPIO_PT3OutputMode( TRISC3_TC34_OUTPUT );
```

```
GPIO_PT3OutputMode( TRISC3_TC35_OUTPUT );
```

//或改成以下用法

```
GPIO_PT3OutputMode( TRISC3_TC35_OUTPUT | TRISC3_TC34_OUTPUT );
```

5.3.46. GPIO_PT3OutputHigh

- 函數

```
GPIO_PT3OutputHigh(BitSet);
```

- 函數功能

設置PT3對應IO輸出高電位，操作暫存器PT3[7:0]。

- 輸入參數

BitSet [in] : 代表GPIO port PT3，8bit數據bit0~bit7分別對應PT3.0~PT3.7，輸入參數如下

PT3_PT30_H : PT3.0輸出高電位

PT3_PT31_H : PT3.1輸出高電位

PT3_PT32_H : PT3.2輸出高電位

PT3_PT33_H : PT3.3輸出高電位

PT3_PT34_H : PT3.4輸出高電位

PT3_PT35_H : PT3.5輸出高電位

PT3_PT36_H : PT3.6輸出高電位

PT3_PT37_H : PT3.7輸出高電位

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* PT3.4輸出高電位 */

```
GPIO_PT3OutputMode( TRISC3_TC34_OUTPUT );
```

//設置PT3.4作為輸出模式

```
GPIO_PT3OutputHigh( PT3_PT34_H );
```

//設置PT3.4輸出高電位

5.3.47. GPIO_PT3OutputLow

- 函數

GPIO_PT3OutputLow(BitSet);

● 函數功能

設置PT3對應IO口輸出低電位，操作暫存器PT3[7:0]。

● 輸入參數

BitSet [in]：代表GPIO port PT3，8bit數據bit0~bit7分別對應PT3.0~PT3.7，輸入參數如下

PT3_PT30_L：PT3.0輸出低電位
PT3_PT31_L：PT3.1輸出低電位
PT3_PT32_L：PT3.2輸出低電位
PT3_PT33_L：PT3.3輸出低電位
PT3_PT34_L：PT3.4輸出低電位
PT3_PT35_L：PT3.5輸出低電位
PT3_PT36_L：PT3.6輸出低電位
PT3_PT37_L：PT3.7輸出低電位

● 包含標頭檔

Driver/GPIO.h

● 函數返回值

無

● 函數用法

/* PT3.4輸出低電位 */

```
GPIO_PT3OutputMode( TRISC3_TC34_OUTPUT );           //設置PT3.4作為輸出模式  
GPIO_PT3OutputHigh( PT3_PT34_L );                 //設置PT3.4輸出低電位
```

5.3.48. GPIO_PT3InputMode

● 函數

GPIO_PT3InputMode(BitSet);

● 函數功能

設置PT3對應IO口作為輸入模式，設置暫存器TRISC3。

● 輸入參數

BitSet [in]：代表PT3對應IO pin，針對PT3.0~PT3.7設置，輸入參數如下

TRISC3_TC30_INPUT：PT3.0作為輸入模式
TRISC3_TC31_INPUT：PT3.1作為輸入模式
TRISC3_TC32_INPUT：PT3.2作為輸入模式
TRISC3_TC33_INPUT：PT3.3作為輸入模式
TRISC3_TC34_INPUT：PT3.4作為輸入模式
TRISC3_TC35_INPUT：PT3.5作為輸入模式
TRISC3_TC36_INPUT：PT3.6作為輸入模式
TRISC3_TC37_INPUT：PT3.7作為輸入模式

● 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 設置PT3.4作為輸入口 */

```
GPIO_PT3InputMode( TRISC3_TC34_INPUT );
```

5.3.49. GPIO_PT3InputPullHight

- 函數

```
GPIO_PT3InputPullHight(BitSet);
```

- 函數功能

使能PT3 對應IO口的輸入上拉電阻，設置暫存器PT3PU。

- 輸入參數

BitSet [in]：代表GPIO PT3，輸入值如下

PT3PU_PU30_ENABLE：使能PT3.0上拉電阻

PT3PU_PU31_ENABLE：使能PT3.1上拉電阻

PT3PU_PU32_ENABLE：使能PT3.2上拉電阻

PT3PU_PU33_ENABLE：使能PT3.3上拉電阻

PT3PU_PU34_ENABLE：使能PT3.4上拉電阻

PT3PU_PU35_ENABLE：使能PT3.5上拉電阻

PT3PU_PU36_ENABLE：使能PT3.6上拉電阻

PT3PU_PU37_ENABLE：使能PT3.7上拉電阻

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 使能 PT3.4 輸入上拉電阻 */

```
GPIO_PT3InputPullHight( PT3PU_PU34_ENABLE );
```

5.3.50. GPIO_PT3InputPullHightClear

- 函數

```
GPIO_PT3InputPullHightClear(BitSet);
```

- 函數功能

關閉PT3對應IO pin的輸入上拉電阻，設置暫存器PT3PU。

- 輸入參數

BitSet [in]：代表GPIO port PT3輸入值如下

PT3PU_PU30_DISABLE：關閉PT3.0上拉電阻

PT3PU_PU31_DISABLE：關閉PT3.1上拉電阻

PT3PU_PU32_DISABLE：關閉PT3.2上拉電阻

PT3PU_PU33_DISABLE：關閉PT3.3上拉電阻

PT3PU_PU34_DISABLE : 關閉PT3.4上拉電阻
PT3PU_PU35_DISABLE : 關閉PT3.5上拉電阻
PT3PU_PU36_DISABLE : 關閉PT3.6上拉電阻
PT3PU_PU37_DISABLE : 關閉PT3.7上拉電阻

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉 PT3.4 輸入上拉電阻 */  
GPIO_PT3InputPullHightClear( PT3PU_PU34_DISABLE );
```

5.3.51. GPIO_GetPT3Data

- 函數

GPIO_GetPT3Data(PT3Data);

- 函數功能

讀取PT3輸入狀態，讀取暫存器PT3。

- 輸入參數

PT3Data [in] : 用於存放讀取到的PT3輸入狀態值

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

PT3Data : 返回 PT3 輸入狀態值

- 函數用法

```
/* 讀取 PT3 輸入狀態值 */  
unsigned char PT3_DATA;  
GPIO_GetPT3Data( PT3_DATA );
```

5.3.52. GPIO_PT4InputPullHight

- 函數

GPIO_PT4InputPullHight(BitSet);

- 函數功能

使能PT4 對應IO的輸入上拉電阻，設置暫存器PT4PU。

- 輸入參數

BitSet [in] : 代表GPIO PT4，輸入值如下

PT4PU_PU40_ENABLE : 使能PT4.0上拉電阻
PT4PU_PU41_ENABLE : 使能PT4.1上拉電阻
PT4PU_PU42_ENABLE : 使能PT4.2上拉電阻
PT4PU_PU43_ENABLE : 使能PT4.3上拉電阻
PT4PU_PU44_ENABLE : 使能PT4.4上拉電阻
PT4PU_PU45_ENABLE : 使能PT4.5上拉電阻
PT4PU_PU46_ENABLE : 使能PT4.6上拉電阻

PT4PU_PU47_ENABLE : 使能PT4.7上拉電阻

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 使能 PT4.4 輸入上拉電阻 */  
GPIO_PT4InputPullHight( PT4PU_PU44_ENABLE );
```

5.3.53. GPIO_PT4InputPullHightClear

- 函數

GPIO_PT4InputPullHightClear(BitSet) ;

- 函數功能

關閉PT4對應IO pin的輸入上拉電阻，設置暫存器PT4PU。

- 輸入參數

BitSet [in] : 代表GPIO port PT4輸入值如下

PT4PU_PU40_DISABLE : 關閉PT4.0上拉電阻
PT4PU_PU41_DISABLE : 關閉PT4.1上拉電阻
PT4PU_PU42_DISABLE : 關閉PT4.2上拉電阻
PT4PU_PU43_DISABLE : 關閉PT4.3上拉電阻
PT4PU_PU44_DISABLE : 關閉PT4.4上拉電阻;
PT4PU_PU45_DISABLE : 關閉PT4.5上拉電阻
PT4PU_PU46_DISABLE : 關閉PT4.6上拉電阻
PT4PU_PU47_DISABLE : 關閉PT4.7上拉電阻

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉 PT4.4 輸入上拉電阻 */  
GPIO_PT4InputPullHightClear( PT4PU_PU44_DISABLE );
```

5.3.54. GPIO_GetPT4Data

- 函數

GPIO_GetPT4Data(PT4Data);

- 函數功能

讀取PT4輸入狀態，讀取暫存器PT4。

- 輸入參數

PT4Data [in] : 用於存放讀取到的PT4輸入狀態值

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

PT4Data : 返回 PT4 輸入狀態值

- 函數用法

```
/* 讀取 PT4 輸入狀態值 */  
unsigned char PT4_DATA;  
GPIO_GetPT4Data( PT4_DATA );
```

5.3.55. GPIO_PT4AnalogMode

- 函數

GPIO_PT4AnalogMode(BitSet);

- 函數功能

使能PT4類比的功能，設置暫存器PT4DA[7:0]。

- 輸入參數

BitSet [in] : 代表GPIO port PT4輸入值如下

PT4DA_DA40_AI0 : PT4.0複用為ADC輸入口AI0

PT4DA_DA41_AI1 : PT4.1複用為ADC輸入口AI1

PT4DA_DA42_AI2 : PT4.2複用為ADC輸入口AI2

PT4DA_DA43_AI3 : PT4.3複用為ADC輸入口AI3

PT4DA_DA44_AI4 : PT4.4複用為ADC輸入口AI4

PT4DA_DA45_AI5 : PT4.5複用為ADC輸入口AI5

PT4DA_DA46_AI6 : PT4.6複用為ADC輸入口AI6

PT4DA_DA47_AI7 : PT4.7複用為ADC輸入口AI7

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 使能PT4.2類比的功能 */
```

```
GPIO_PT4AnalogMode( PT4DA_DA42_AI2 );
```

```
//使能PT4.2類比的功能
```

5.3.56. GPIO_PT4DigitalMode

- 函數

GPIO_PT4DigitalMode(BitSet);

- 函數功能

使能PT4數位的功能，設置暫存器PT4DA[7:0]。

- 輸入參數

BitSet [in] : 代表GPIO port PT4，輸入值如下

PT4DA_DA40_DIG : 使能PT4.0數位的功能

PT4DA_DA41_DIG : 使能PT4.1數位的功能

PT4DA_DA42_DIG : 使能PT4.2數位的功能

PT4DA_DA43_DIG : 使能PT4.3數位的功能

PT4DA_DA44_DIG : 使能PT4.4數位的功能
PT4DA_DA45_DIG : 使能PT4.5數位的功能
PT4DA_DA46_DIG : 使能PT4.6數位的功能
PT4DA_DA47_DIG : 使能PT4.7數位的功能

● 包含標頭檔

Driver/GPIO.h

● 函數返回值

無

● 函數用法

/* 使能PT4.2數位的功能 */

GPIO_PT4DigitalMode(PT4DA_DA42_DIG); //使能PT4.2數位的功能

5.3.57. GPIO_PT5InputPullHight

● 函數

GPIO_PT5InputPullHight(BitSet);

● 函數功能

使能PT5 對應IO口的輸入上拉電阻，設置暫存器PT5PU。

● 輸入參數

BitSet [in] : 代表GPIO PT5，輸入值如下

PT5PU_PU50_ENABLE : 使能PT5.0上拉電阻
PT5PU_PU51_ENABLE : 使能PT5.1上拉電阻
PT5PU_PU52_ENABLE : 使能PT5.2上拉電阻
PT5PU_PU53_ENABLE : 使能PT5.3上拉電阻
PT5PU_PU54_ENABLE : 使能PT5.4上拉電阻
PT5PU_PU55_ENABLE : 使能PT5.5上拉電阻
PT5PU_PU56_ENABLE : 使能PT5.6上拉電阻
PT5PU_PU57_ENABLE : 使能PT5.7上拉電阻

● 包含標頭檔

Driver/GPIO.h

● 函數返回值

無

● 函數用法

/* 使能 PT5.4 輸入上拉電阻 */

GPIO_PT5InputPullHight(PT5PU _PU54_ENABLE);

5.3.58. GPIO_PT5InputPullHightClear

● 函數

GPIO_PT5InputPullHightClear(BitSet);

● 函數功能

關閉PT5對應IO pin的輸入上拉電阻，設置暫存器PT5PU。

- 輸入參數

BitSet [in] : 代表GPIO port PT5輸入值如下

PT5PU_PU50_DISABLE : 關閉PT5.0上拉電阻

PT5PU_PU51_DISABLE : 關閉PT5.1上拉電阻

PT5PU_PU52_DISABLE : 關閉PT5.2上拉電阻

PT5PU_PU53_DISABLE : 關閉PT5.3上拉電阻

PT5PU_PU54_DISABLE : 關閉PT5.4上拉電阻

PT5PU_PU55_DISABLE : 關閉PT5.5上拉電阻

PT5PU_PU56_DISABLE : 關閉PT5.6上拉電阻

PT5PU_PU57_DISABLE : 關閉PT5.7上拉電阻

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉 PT5.4 輸入上拉電阻 */
```

```
GPIO_PT5InputPullHightClear( PT5PU_PU54_DISABLE );
```

5.3.59. GPIO_GetPT5Data

- 函數

```
GPIO_GetPT5Data(PT5Data);
```

- 函數功能

讀取PT5輸入狀態，讀取暫存器PT5。

- 輸入參數

PT5Data [in] : 用於存放讀取到的PT5輸入狀態值

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

PT5Data : 返回 PT5 輸入狀態值

- 函數用法

```
/* 讀取 PT5 輸入狀態值 */
```

```
unsigned char PT5_DATA;
```

```
GPIO_GetPT5Data( PT5_DATA );
```

5.3.60. GPIO_PT5AnalogMode

- 函數

```
GPIO_PT5AnalogMode(BitSet);
```

- 函數功能

使能PT5類比的功能，設置暫存器PT5DA[3:0]。

- 輸入參數

BitSet [in] : 代表GPIO port PT5.0~PT5.3輸入值如下

PT5DA_DA50_AI8 : PT5.0複用為ADC輸入口AI8

PT5DA_DA51_AI9 : PT5.1複用為ADC輸入口AI9
PT5DA_DA52_AI10 : PT5.2複用為ADC輸入口AI10
PT5DA_DA53_AI11 : PT5.3複用為ADC輸入口AI11

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 使能PT5.2的類比的功能 */

GPIO_PT5AnalogMode(PT5DA_DA52_AI10); //使能PT5.2類比的功能

5.3.61. GPIO_PT5DigitalMode

- 函數

GPIO_PT5DigitalMode(BitSet);

- 函數功能

使能PT5數位的功能，設置暫存器PT5DA[3:0]。

- 輸入參數

BitSet [in] : 代表GPIO port PT5，輸入值如下

PT5DA_DA50_DIG : 使能PT5.0數位的功能

PT5DA_DA51_DIG : 使能PT5.1數位的功能

PT5DA_DA52_DIG : 使能PT5.2數位的功能

PT5DA_DA53_DIG : 使能PT5.3數位的功能

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 使能PT5.2數位的功能 */

GPIO_PT5DigitalMode(PT5DA_DA52_DIG); //使能PT5.2數位的功能

5.3.62. GPIO_INT0_Enable

- 函數

GPIO_INT0_Enable();

- 函數功能

使能PT1.0外部中斷功能，設置暫存器INTE1[0]=1。

- 輸入參數

無。

- 包含標頭檔

Driver/INT.h

- 函數返回值

無

- 函數用法

```
/* 使能PT1.0外部中斷功能 */
```

```
GPIO_INT0_Enable();
```

5.3.63. GPIO_INT0_Disable

- 函數

```
GPIO_INT0_Disable();
```

- 函數功能

關閉PT1.0外部中斷功能，設置暫存器INTE1[0]=0。

- 輸入參數

無

- 包含標頭檔

Driver/INT.h

- 函數返回值

無

- 函數用法

```
/* 關閉PT1.0外部中斷功能 */
```

```
GPIO_INT0_Disable();
```

5.3.64. GPIO_INT1_Enable

- 函數

```
GPIO_INT1_Enable();
```

- 函數功能

使能PT1.1外部中斷功能，設置暫存器INTE1[1]=1。

- 輸入參數

無

- 包含標頭檔

Driver/INT.h

- 函數返回值

無

- 函數用法

```
/* 使能PT1.1外部中斷功能 */
```

```
GPIO_INT1_Enable();
```

5.3.65. GPIO_INT1_Disable

- 函數

```
GPIO_INT1_Disable();
```

- 函數功能

關閉PT1.1外部中斷功能，設置暫存器INTE1[1]=0。

- 輸入參數

無

- 包含標頭檔

Driver/INT.h

- 函數返回值

無

- 函數用法

```
/* 關閉PT1.1外部中斷功能 */  
GPIO_INT1_Disable();
```

5.3.66. GPIO_INT_TYPE_PT10

- 函數

```
GPIO_INT_TYPE_PT10(PT10INTType);
```

- 函數功能

設置PT1.0外部中斷觸發沿，設置暫存器PT1M1[1:0]。

- 輸入參數

PT10INTType [in] : PT1.0外部中斷觸發沿選擇

PT1M1_INTEG0_LEV_ : 電位變化觸發

PT1M1_INTEG0_LEV : 電位變化觸發

PT1M1_INTEG0_EDGERISE : 上升沿觸發

PT1M1_INTEG0_EDGEFALL : 下降沿觸發

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 設置PT1.0外部中斷觸發沿為下降沿觸發 */  
GPIO_INT_TYPE_PT10( PT1M1_INTEG0_EDGEFALL );
```

5.3.67. GPIO_INT_TYPE_PT11

- 函數

```
GPIO_INT_TYPE_PT11(PT11INTType);
```

- 函數功能

設置PT1.1外部中斷觸發沿，設置暫存器PT1M1[3:2]。

- 輸入參數

PT11INTType [in] : PT1.1外部中斷觸發沿選擇

PT1M1_INTEG1_LEV_ : 電位變化觸發

PT1M1_INTEG1_LEV : 電位變化觸發
PT1M1_INTEG1_EDGERISE : 上升沿觸發
PT1M1_INTEG1_EDGEFALL : 下降沿觸發

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 設置PT1.1外部中斷觸發沿為下降沿觸發 */  
GPIO_INT_TYPE_PT11( PT1M1_INTEG1_EDGEFALL );
```

5.3.68. GPIO_INT_Low2BitEnable

- 函數

GPIO_INT_Low2BitEnable(L2BSet);

- 函數功能

使能PT1.1/PT1.0外部中斷功能，設置暫存器INTE1[1:0]。

- 輸入參數

L2Bset [in] : 使能PT1.0/PT1.1外部中斷

INTE1_E0IE_ENABLE : 使能PT1.0外部中斷

INTE1_E1IE_ENABLE : 使能PT1.1外部中斷

- 包含標頭檔

Driver/INT.h

- 函數返回值

無

- 函數用法

```
/* 使能PT1.1/PT1.0外部中斷功能 */  
GPIO_INT_Low2BitEnable( INTE1_E0IE_ENABLE | INTE1_E1IE_ENABLE );
```

5.3.69. GPIO_AI0Enable

- 函數

GPIO_AI0Enable();

- 函數功能

使能PT4.0 (AI0) 類比的功能，設置PT4DA[0]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 使能PT4.0類比的功能 */

GPIO_AI0Enable();

5.3.70. GPIO_AI1Enable

- 函數

GPIO_AI1Enable();

- 函數功能

使能PT4.1 (AI1) 類比的功能，設置PT4DA[1]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 使能PT4.1類比的功能 */

GPIO_AI1Enable();

5.3.71. GPIO_AI2Enable

- 函數

GPIO_AI2Enable();

- 函數功能

使能PT4.2 (AI2) 類比的功能，設置PT4DA[2]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 使能PT4.2類比的功能 */

GPIO_AI2Enable();

5.3.72. GPIO_AI3Enable

- 函數

GPIO_AI3Enable();

- 函數功能

使能PT4.3 (AI3) 類比的功能，設置PT4DA[3]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 使能PT4.3類比的功能 */  
GPIO_AI3Enable();
```

5.3.73. GPIO_AI4Enable

- 函數

GPIO_AI4Enable();

- 函數功能

使能PT4.4 (AI4) 類比的功能，設置PT4DA[4]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 使能PT4.4類比的功能 */  
GPIO_AI4Enable();
```

5.3.74. GPIO_AI5Enable

- 函數

GPIO_AI5Enable();

- 函數功能

使能PT4.5 (AI5) 類比的功能，設置PT4DA[5]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 使能PT4.5類比的功能 */  
GPIO_AI5Enable();
```

5.3.75. GPIO_AI6Enable

- 函數

GPIO_AI6Enable();

- 函數功能

使能PT4.6 (AI6) 類比的功能，設置PT4DA[6]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/ 使能PT4.6類比的功能 */*

GPIO_AI6Enable();

5.3.76. GPIO_AI7Enable

- 函數

GPIO_AI7Enable();

- 函數功能

使能PT4.7 (AI7) 類比的功能，設置PT4DA[7]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/ 使能PT4.7類比的功能 */*

GPIO_AI7Enable();

5.3.77. GPIO_AI8Enable

- 函數

GPIO_AI8Enable();

- 函數功能

使能PT5.0 (AI8) 類比的功能，設置PT5DA[0]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 使能PT5.0類比的功能 */

GPIO_AI8Enable();

5.3.78. GPIO_AI9Enable

- 函數

GPIO_AI9Enable();

- 函數功能

使能PT5.1 (AI9) 類比的功能，設置PT5DA[1]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 使能PT5.1類比的功能 */

GPIO_AI9Enable();

5.3.79. GPIO_AI10Enable

- 函數

GPIO_AI10Enable();

- 函數功能

使能PT5.2 (AI10) 類比的功能，設置PT5DA[2]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 使能PT5.2類比的功能 */

GPIO_AI10Enable();

5.3.80. GPIO_AI8Enable

- 函數

```
GPIO_AI11Enable();
```

- 函數功能

使能PT5.3 (AI11) 類比的功能，設置PT5DA[3]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 使能PT5.3類比的功能 */
```

```
GPIO_AI11Enable();
```

5.3.81. GPIO_CPAI0Enable

- 函數

```
GPIO_CPAI0Enable();
```

- 函數功能

使能PT2.2複用為增強型比較器信號輸入端CPAI0，設置PT2M1[5:4]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 使能PT2.2複用為CPAI0引腳 */
```

```
GPIO_CPAI0Enable();
```

5.3.82. GPIO_CPAI0Disable

- 函數

```
GPIO_CPAI0Disable();
```

- 函數功能

關閉PT2.2複用為增強型比較器信號輸入端CPAI0功能，設置PT2M1[5:4]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉PT2.2複用為CPAI0引腳 */
```

```
GPIO_CPAI0Disable();
```

5.3.83. GPIO_CPAI1Enable

- 函數

```
GPIO_CPAI1Enable();
```

- 函數功能

使能PT2.3複用為增強型比較器信號輸入端CPAI1，設置PT2M1[6]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 使能PT2.3複用為CPAI1引腳 */
```

```
GPIO_CPAI1Enable();
```

5.3.84. GPIO_CPAI1Disable

- 函數

```
GPIO_CPAI1Disable();
```

- 函數功能

關閉PT2.3複用為增強型比較器信號輸入端CPAI1功能，設置PT2M1[6]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉PT2.3複用為CPAI1引腳 */
```

```
GPIO_CPAI1Disable();
```

5.3.85. GPIO_CPAI2Enable

- 函數

```
GPIO_CPAI2Enable();
```

- 函數功能

使能PT2.4複用為增強型比較器信號輸入端CPAI2，設置PT2M2[1:0]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 使能PT2.4複用為CPAI2引腳 */  
GPIO_CPAI2Enable();
```

5.3.86. GPIO_CPAI2Disable

- 函數

GPIO_CPAI2Disable();

- 函數功能

關閉PT2.4複用為增強型比較器信號輸入端CPAI2功能，設置PT2M2[1:0]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉PT2.4複用為CPAI2引腳 */  
GPIO_CPAI2Disable();
```

5.3.87. GPIO_CPAI3Enable

- 函數

GPIO_CPAI3Enable();

- 函數功能

使能PT2.5複用為增強型比較器信號輸入端CPAI3，設置PT2M2[3:2]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 使能PT2.5複用為CPAI3引腳 */  
GPIO_CPAI3Enable();
```


5.3.88. GPIO_CPAI3Disable

- 函數

GPIO_CPAI3Disable();

- 函數功能

關閉PT2.5複用為增強型比較器信號輸入端CPAI3功能，設置PT2M2[3:2]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 關閉PT2.5複用為CPAI3引腳 */
```

```
GPIO_CPAI3Disable();
```

5.3.89. GPIO_CPAI4Enable

- 函數

GPIO_CPAI4Enable();

- 函數功能

使能PT2.6複用為增強型比較器信號輸入端CPAI4，設置PT2M2[4]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

```
/* 使能PT2.6複用為CPAI4引腳 */
```

```
GPIO_CPAI4Enable();
```

5.3.90. GPIO_CPAI4Disable

- 函數

GPIO_CPAI4Disable();

- 函數功能

關閉PT2.6複用為增強型比較器信號輸入端CPAI4功能，設置PT2M2[4]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 關閉PT2.6複用為CPAI4引腳 */

GPIO_CPAI4Disable();

5.3.91. GPIO_CPAI5Enable

- 函數

GPIO_CPAI5Enable();

- 函數功能

使能PT2.7複用為增強型比較器信號輸入端CPAI5，設置PT2M2[6]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 使能PT2.7複用為CPAI5引腳 */

GPIO_CPAI5Enable();

5.3.92. GPIO_CPAI5Disable

- 函數

GPIO_CPAI5Disable();

- 函數功能

關閉PT2.7複用為增強型比較器信號輸入端CPAI5功能，設置PT2M2[6]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 關閉PT2.7複用為CPAI5引腳 */

GPIO_CPAI5Disable();

5.3.93. GPIO_CPAI6Enable

- 函數

GPIO_CPAI6Enable();

● 函數功能

使能PT1.0複用為增強型比較器信號輸入端CPAI6，設置PT1DA[0]。

● 輸入參數

無

● 包含標頭檔

Driver/GPIO.h

● 函數返回值

無

● 函數用法

/* 使能PT1.0複用為CPAI6引腳 */

GPIO_CPAI6Enable();

5.3.94. GPIO_CPAI6Disable

● 函數

GPIO_CPAI6Disable();

● 函數功能

關閉PT1.0複用為增強型比較器信號輸入端CPAI6功能，設置PT1DA[0]。

● 輸入參數

無

● 包含標頭檔

Driver/GPIO.h

● 函數返回值

無

● 函數用法

/* 關閉PT1.0複用為CPAI6引腳 */

GPIO_CPAI6Disable();

5.3.95. GPIO_CPAI7Enable

● 函數

GPIO_CPAI7Enable();

● 函數功能

使能PT1.1複用為增強型比較器信號輸入端CPAI7，設置PT1DA[1]。

● 輸入參數

無

● 包含標頭檔

Driver/GPIO.h

● 函數返回值

無

- 函數用法

/* 使能PT1.1複用為CPAI7引腳 */

```
GPIO_CPAI7Enable();
```

5.3.96. GPIO_CPAI7Disable

- 函數

```
GPIO_CPAI7Disable();
```

- 函數功能

關閉PT1.1複用為增強型比較器信號輸入端CPAI7功能，設置PT1DA[1]。

- 輸入參數

無

- 包含標頭檔

Driver/GPIO.h

- 函數返回值

無

- 函數用法

/* 關閉PT1.1複用為CPAI7引腳 */

```
GPIO_CPAI7Disable();
```

6. 模數轉換器 ADC

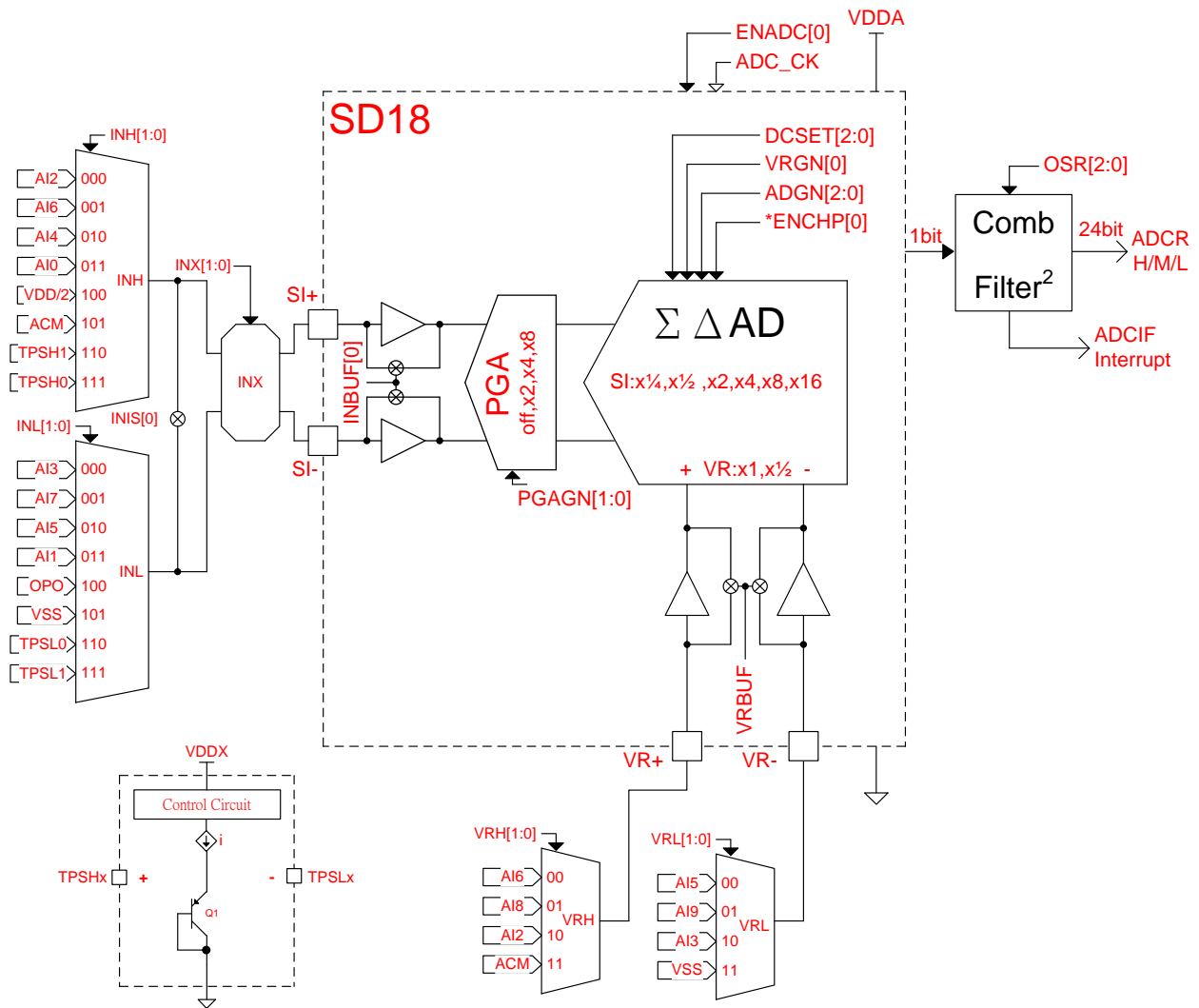
6.1. 函數簡介

該部分函數描述ADC 系統的控制，包含：

- ADC的信號輸入埠與參考輸入埠的配置與切換
- ADC放大倍數的設置
- ADC中斷配置
- ADC轉換值的讀取
- 包含ADC.h

序號	函數名稱	功能描述
01	ADC_Open	啓動ADC，並設置ADC的工作頻率、信號輸入、參考電壓輸入端、放大倍數、資料輸出速率、直流偏置電壓
02	ADC_GetData	讀取ADC轉換值
03	ADC_CLK_Enable	設置ADC工作頻率及預分頻值
04	ADC_CLK_Disable	關閉ADC工作頻率
05	ADC_INT_Enable	使能ADC中斷功能
06	ADC_INT_Disable	關閉ADC中斷功能
07	ADC_INT_IsFlag	讀取ADC中斷請求旗標
08	ADC_INT_ClearFlag	清除ADC中斷請求旗標
09	ADC_INSEnable	使能ADC信號輸入端內部短路功能
10	ADC_INSDisable	關閉ADC信號輸入端內部短路功能
11	ADC_Enable	啓動ADC轉換功能
12	ADC_Disable	關閉ADC轉換功能
13	ADC_VRbufEnable	使能ADC的參考電壓輸入端的輸入緩衝器功能
14	ADC_VRbufDisable	關閉ADC參考電壓輸入端輸入緩衝器功能
15	ADC_INbufEnable	使能ADC信號輸入端的輸入緩衝器功能
16	ADC_INbufDisable	關閉ADC信號輸入端的輸入緩衝器功能
17	ADC_AINConfig	設置ADC信號輸入端的來源
18	ADC_VRINConfig	設置ADC參考電壓輸入端來源
19	ADC_VRGainSelect	設置ADC參考電壓輸入端放大倍數
20	ADC_GainConfig	設置ADC信號輸入端放大倍數ADGN * PGA
21	ADC_OSRConfig	設置ADC轉換輸出速率，輸出速率為ADC_CK /OSR
22	ADC_DCSetConfig	設置ADC信號輸入端的直流偏置電壓
23	ADC_INXConfig	設置ADC信號輸入端的輸入信號轉置器

6.2. ADC 模組方框圖



6.3. 函數說明

6.3.1. ADC_Open

● 函數

```
void ADC_Open(unsigned char ck, unsigned char cks, unsigned char inh,
              unsigned char inl, unsigned char vrh, unsigned char vrl,
              unsigned char adgn, unsigned char pgagn, unsigned char vrgn,
              unsigned char dcset, unsigned char osr);
```

● 函數功能

啓動ADC，並設置ADC的工作頻率、信號輸入、參考電壓輸入端、放大倍數、資料輸出速率、直流偏置電壓等，設置暫存器MCKCN1[7:4]/ADCCN1/ADCCN2/ADCCN3/AINET1/AINET2。

● 輸入參數

ck [in] : 設置ADC工作頻率源

MCKCN1_ADCK_LSCK : ADC工作頻率源為LS_CK

MCKCN1_ADCK_HSDCK : ADC工作頻率源為HSD_CK

cks [in] : 設置ADC工作頻率源預分頻

MCKCN1_ADGS_DIV128 : ADC工作頻率預分頻ADC_CK/128
MCKCN1_ADGS_DIV64 : ADC工作頻率預分頻ADC_CK/64
MCKCN1_ADGS_DIV32 : ADC工作頻率預分頻ADC_CK/32
MCKCN1_ADGS_DIV16 : ADC工作頻率預分頻ADC_CK/16
MCKCN1_ADGS_DIV8 : ADC工作頻率預分頻ADC_CK/8
MCKCN1_ADGS_DIV4 : ADC工作頻率預分頻ADC_CK/4
MCKCN1_ADGS_DIV2 : ADC工作頻率預分頻ADC_CK/2
MCKCN1_ADGS_DIV1 : ADC工作頻率預分頻ADC_CK/1

inh [in] : 設置ADC信號輸入端的正向輸入通道

AINET1_INH_AI2 : AI2
AINET1_INH_AI6 : AI6
AINET1_INH_AI4 : AI4
AINET1_INH_AI0 : AI0
AINET1_INH_HALF_VDDA : VDDA/2
AINET1_INH_ADM : ACM
AINET1_INH_TPSH1 : TPSH1
AINET1_INH_TPSH0 : TPSH0

inl [in] : 設置ADC信號輸入端的負向輸入通道

AINET1_INL_AI3 : AI3
AINET1_INL_AI7 : AI7
AINET1_INL_AI5 : AI5
AINET1_INL_AI1 : AI1
AINET1_INL_OPO : OPO
AINET1_INL_VSS : VSS
AINET1_INL_TPSSL0 : TPSSL0
AINET1_INL_TPSSL1 : TPSSL1

vrh [in] : 設置ADC參考電壓輸入端的正向輸入通道

AINET2_VRH_AI6 : AI6
AINET2_VRH_AI8 : AI8
AINET2_VRH_AI2 : AI2
AINET2_VRH_ACM : ACM

vrl [in] : 設置ADC參考電壓輸入端的負向輸入通道

AINET2_VRL_AI5 : AI5
AINET2_VRL_AI9 : AI9
AINET2_VRL_AI3 : AI3
AINET2_VRL_VSS : VSS

adgn [in] : 設置ADC的ADGN放大倍數

ADCCN1_ADGN_1DIV4 : x 1/4

ADCCN1_ADGN_1DIV2 : x 1/2

ADCCN1_ADGN_1 : x 1

ADCCN1_ADGN_2 : x 2

ADCCN1_ADGN_4 : x 4

ADCCN1_ADGN_8 : x 8

ADCCN1_ADGN_16 : x 16

pgagn [in] : 設置ADC的PGA放大倍數

ADCCN1_PGAGN_1 : x 1

ADCCN1_PGAGN_2 : x 2

ADCCN1_PGAGN_4 : x 4

ADCCN1_PGAGN_8 : x 8

vrgn [in] : 設置參考電壓端放大倍數

ADCCN2_VREGN_DIV2 : VREF * 1/2

ADCCN2_VREGN_DIV1 : VREF* 1

dcset [in] : 設置ADC的信號輸入端直流偏置電壓

ADCCN2_DCSET_0 : 不偏壓

ADCCN2_DCSET_P1DIV4 : + VREF* 1/4

ADCCN2_DCSET_P1DIV2 : + VREF* 1/2

ADCCN2_DCSET_P3DIV4 : + VREF* 3/4

ADCCN2_DCSET_00 : 不偏壓

ADCCN2_DCSET_N1DIV4 : -VREF* 1/4

ADCCN2_DCSET_N1DIV2 : -VREF* 1/2

ADCCN2_DCSET_N3DIV4 : -VREF* 3/4

osr [in] : 設置ADC轉換輸出速率控制值

ADCCN3_OSR_256 : 轉換輸出速率為ADC_CK/256

ADCCN3_OSR_512 : 轉換輸出速率為ADC_CK/512

ADCCN3_OSR_1024 : 轉換輸出速率為ADC_CK/1024

ADCCN3_OSR_2048 : 轉換輸出速率為ADC_CK/2048

ADCCN3_OSR_4096 : 轉換輸出速率為ADC_CK/4096

ADCCN3_OSR_8192 : 轉換輸出速率為ADC_CK/8192

ADCCN3_OSR_16384 : 轉換輸出速率為ADC_CK/16384

ADCCN3_OSR_32768 : 轉換輸出速率為ADC_CK/32768

● 包含標頭檔

Driver/ADC.h

● 函數返回值

無

● 函數用法

/* 設置ADC工作頻率為HSD_CK/4，信號輸入端為AI0---AI1，參考電壓輸入端為AI2—AI3，參考電壓衰減倍數1/2，ADC放大倍數為16*8，偏置電壓為0，OSR為32768 */


```
ADC_Open( MCKCN1_ADCK_HSDCK, MCKCN1_ADCS_DIV4,  
          AINET1_INH_AI0, AINET1_INH_AI1, AINET2_VRH_AI2, AINET2_VRH_AI3,  
          ADCCN1_ADGN_16, ADCCN1_PGAGN_8,  
          ADCCN2_VREGN_DIV2, ADCCN2_DCSET_0,  
          ADCCN3_OSR_32768 );
```

6.3.2. ADC_GetData

- 函數

```
long ADC_GetData(void);
```

- 函數功能

讀取ADC轉換值，讀取暫存器ADCRH:ADCRM:ADCRL

- 輸入參數

無

- 包含標頭檔

Diriver/ADC.h

- 函數返回值

24位的ADC轉換值

- 函數用法

```
/* 讀取ADC轉換值 */
```

```
long ADC_DATA;
```

```
ADC_DATA = ADC_GetData();
```

6.3.3. ADC_CLK_Enable

- 函數

```
ADC_CLK_Enable(adck,adcs);
```

- 函數功能

設置ADC工作頻率及預分頻值，設置暫存器MCKCN1[7:4]。

- 輸入參數

adck [in]：設置ADC工作頻率源

MCKCN1_ADCK_LSCK : ADC工作頻率源為LS_CK

MCKCN1_ADCK_HSDCK : ADC工作頻率源為HSD_CK

adcs [in]：設置ADC工作頻率預分頻值

MCKCN1_ADACS_DIV128 : ADC工作頻率預分頻ADC_CK/128

MCKCN1_ADACS_DIV64 : ADC工作頻率預分頻ADC_CK/64

MCKCN1_ADACS_DIV32 : ADC工作頻率預分頻ADC_CK/32

MCKCN1_ADACS_DIV16 : ADC工作頻率預分頻ADC_CK/16

MCKCN1_ADACS_DIV8 : ADC工作頻率預分頻ADC_CK/8

MCKCN1_ADACS_DIV4 : ADC工作頻率預分頻ADC_CK/4

MCKCN1_ADACS_DIV2 : ADC工作頻率預分頻ADC_CK/2

MCKCN1_ADCS_DIV1 : ADC工作頻率預分頻ADC_CK/1

- 包含標頭檔

Driver/ADC.h

- 函數返回值

無

- 函數用法

```
/* 設定ADC工作頻率為HSD_CK/4 */
```

```
ADC_CLK_Enable( MCKCN1_ADCK_HSDCK, MCKCN1_ADCS_DIV4 );
```

6.3.4. ADC_CLK_Disable

- 函數

```
ADC_CLK_Disable();
```

- 函數功能

關閉ADC工作頻率，設置暫存器MCKCN1[4]。

- 輸入參數

無

- 包含標頭檔

Driver/ADC.h

- 函數返回值

無

- 函數用法

```
/* 關閉ADC 工作頻率 */
```

```
ADC_CLK_Disable();
```

6.3.5. ADC_INT_Enable

- 函數

```
ADC_INT_Enable();
```

- 函數功能

使能ADC中斷功能，設置暫存器INTE1[6]=1。

- 輸入參數

無

- 包含標頭檔

Driver/ADC.h

- 函數返回值

無

- 函數用法

```
/* 使能ADC中斷功能 */
```

```
ADC_INT_Enable();
```

6.3.6. ADC_INT_Disable

- 函數

ADC_INT_Disable();

- 函數功能

關閉ADC中斷功能，設置暫存器INTE1[6]=0。

- 輸入參數

無

- 包含標頭檔

Driver/ADC.h

- 函數返回值

無

- 函數用法

```
/* 關閉ADC中斷功能 */
```

```
ADC_INT_Disable();
```

6.3.7. ADC_INT_IsFlag

- 函數

ADC_INT_IsFlag()

- 函數功能

讀取ADC中斷請求旗標，讀取暫存器INTF1[6]。

- 輸入參數

無

- 包含標頭檔

Driver/ADC.h

- 函數返回值

0x00 : ADC不產生中斷

0x40 : ADC產生中斷請求

- 函數用法

```
/* 讀取ADC中斷請求旗標 */
```

```
unsigned char flag;
```

```
flag = ADC_INT_IsFlag();
```

6.3.8. ADC_INT_ClearFlag

- 函數

ADC_INT_ClearFlag();

- 函數功能

清除ADC中斷請求旗標，設置暫存器INTF1[6]=0。

- 輸入參數

無

- 包含標頭檔

Driver/ADC.h

- 函數返回值

無

- 函數用法

/* 清零ADC中斷請求旗標 */

ADC_INT_ClearFlag();

6.3.9. ADC_INSEnable

- 函數

ADC_INSEnable();

- 函數功能

使能ADC信號輸入端內部短路功能，設置暫存器AINET1[1]=1。

- 輸入參數

無

- 包含標頭檔

Driver/ADC.h

- 函數返回值

無

- 函數用法

/* 設置ADC信號輸入端內部短路 */

ADC_INSEnable();

6.3.10. ADC_INSDisable

- 函數

ADC_INSDisable();

- 函數功能

關閉ADC信號輸入端內部短路功能，設置暫存器AINET1[1]=0。

- 輸入參數

無

- 包含標頭檔

Driver/ADC.h

- 函數返回值

無

- 函數用法

/* 關閉ADC信號輸入端內部短路功能 */

ADC_INSDisable();

6.3.11. ADC_Enable

- 函數

ADC_Enable();

- 函數功能

啓動ADC轉換功能;

設置暫存器ADCCN1[7]=1 。

- 輸入參數

無

- 包含標頭檔

Drirver/ADC.h

- 函數返回值

無

- 函數用法

```
/* 啓動ADC轉換功能 */
```

```
ADC_Enable();
```

6.3.12. ADC_Disable

- 函數

ADC_Disable();

- 函數功能

關閉ADC轉換功能，設置暫存器ADCCN1[7]=0 。

- 輸入參數

無

- 包含標頭檔

Drirver/ADC.h

- 函數返回值

無

- 函數用法

```
/* 關閉ADC轉換功能 */
```

```
ADC_Disable();
```

6.3.13. ADC_VRbufEnable

- 函數

ADC_VRbufEnable();

- 函數功能

使能ADC的參考電壓輸入端的輸入緩衝器功能，設置暫存器ADCCN2[4]=1 。

- 輸入參數

無

- 包含標頭檔

Drirver/ADC.h

- 函數返回值

無

- 函數用法

/* 使能ADC參考電壓輸入端輸入緩衝器功能 */

ADC_VRbufEnable();

6.3.14. ADC_VRbufDisable

- 函數

ADC_VRbufDisable();

- 函數功能

關閉ADC參考電壓輸入端輸入緩衝器功能，設置暫存器ADCCN2[4]=0。

- 輸入參數

無

- 包含標頭檔

Driver/ADC.h

- 函數返回值

無

- 函數用法

/* 關閉ADC參考電壓輸入端輸入緩衝器功能 */

ADC_VRbufDisable();

6.3.15. ADC_INbufEnable

- 函數

ADC_INbufEnable();

- 函數功能

使能ADC信號輸入端的輸入緩衝器功能，設置暫存器ADCCN2[5]=1。

- 輸入參數

無

- 包含標頭檔

Driver/ADC.h

- 函數返回值

無

- 函數用法

/* 使能ADC信號輸入端的輸入緩衝器功能 */

ADC_INbufEnable();

6.3.16. ADC_INbufDisable

- 函數

ADC_INbufDisable();

- 函數功能

關閉ADC信號輸入端的輸入緩衝器功能，設置暫存器ADCCN2[5]=0。

- 輸入參數

無

- 包含標頭檔

Driver/ADC.h

- 函數返回值

無

- 函數用法

```
/* 關閉ADC信號輸入端的輸入緩衝器功能 */  
ADC_INbufDisable();
```

6.3.17. ADC_AINConfig

- 函數

ADC_AINConfig(inh,inl);

- 函數功能

設置ADC信號輸入端的來源，設置暫存器AINET1[7:5]。

- 輸入參數

inh[in]：設置ADC信號輸入端的正向輸入通道

AINET1_INH_AI2	: AI2
AINET1_INH_AI6	: AI6
AINET1_INH_AI4	: AI4
AINET1_INH_AI0	: AI0
AINET1_INH_HALF_VDDA	: VDDA/2
AINET1_INH_ADM	: ACM
AINET1_INH_TPSH1	: TPSH1
AINET1_INH_TPSH0	: TPSH0

inl[in]：設置ADC信號輸入端的正向輸入通道

AINET1_INL_AI3	: AI3
AINET1_INL_AI7	: AI7
AINET1_INL_AI5	: AI5
AINET1_INL_AI1	: AI1
AINET1_INL_OPO	: OPO
AINET1_INL_VSS	: VSS
AINET1_INL_TPSL0	: TPSL0
AINET1_INL_TPSL1	: TPSL1

- 包含標頭檔

Driver/ADC.h

- 函數返回值

無

- 函數用法

/* 設置ADC信號輸入端為AI0---AI1 */

ADC_AINConfig(AINET1_INH_AI0, AINET1_INL_AI1);

6.3.18. ADC_VRINConfig

- 函數

ADC_VRINConfig(vrh,vrl);

- 函數功能

設置ADC參考電壓輸入端來源，設置暫存器AINET2[6:5]/AINET2[2:1]。

- 輸入參數

vrh [in] : 設置ADC參考電壓輸入端的正向輸入通道

AINET2_VRH_AI6 : AI6

AINET2_VRH_AI8 : AI8

AINET2_VRH_AI2 : AI2

AINET2_VRH_ACM : ACM

vrl [in] : 設置ADC參考電壓輸入端的負向輸入通道

AINET2_VRL_AI5 : AI5

AINET2_VRL_AI9 : AI9

AINET2_VRL_AI3 : AI3

AINET2_VRL_VSS : VSS

- 包含標頭檔

Driver/ADC.h

- 函數返回值

無

- 函數用法

/* 設置ADC參考電壓輸入端為AI2—AI3 */

ADC_VRINConfig(AINET2_VRH_AI2, AINET2_VRL_AI3);

6.3.19. ADC_VRGainSelect

- 函數

ADC_VRGainSelect(VRGSel);

- 函數功能

設置ADC參考電壓輸入端放大倍數，設置暫存器ADCCN2[3]。

- 輸入參數

VRGSel [in] : 設置ADC參考電壓輸入端放大倍數

ADCCN2_VREGN_DIV2 : VREF * 1/2

ADCCN2_VREGN_DIV1 : VREF* 1

- 包含標頭檔

Driver/ADC.h

- 函數返回值

無

- 函數用法

/* 設置ADC參考電壓輸入端放大倍數為1/2 */

```
ADC_VRGainSelect( ADCCN2_VREGN_DIV2 );
```

6.3.20. ADC_GainConfig

- 函數

```
ADC_GainConfig(adgn,pgagn);
```

- 函數功能

設置ADC信號輸入端放大倍數ADGN * PGA，設置暫存器ADCCN1[4 :0]。

- 輸入參數

adgn [in] : 設置ADC的ADGN放大倍數

ADCCN1_ADGN_1DIV4 : x 1/4

ADCCN1_ADGN_1DIV2 : x 1/2

ADCCN1_ADGN_1 : x 1

ADCCN1_ADGN_2 : x 2

ADCCN1_ADGN_4 : x 4

ADCCN1_ADGN_8 : x 8

ADCCN1_ADGN_16 : x 16

pgagn [in] : 設置ADC的PGA放大倍數

ADCCN1_PGAGN_1 : x 1

ADCCN1_PGAGN_2 : x 2

ADCCN1_PGAGN_4 : x 4

ADCCN1_PGAGN_8 : x 8

- 包含標頭檔

Driver/ADC.h

- 函數返回值

無

- 函數用法

/* 設置ADC放大倍數為16*8 */

```
ADC_GainConfig( ADCCN1_ADGN_16, ADCCN1_PGAGN_8 );
```

6.3.21. ADC_OSRConfig

- 函數

```
ADC_OSRConfig(osr);
```

- 函數功能

設置ADC轉換輸出速率，輸出速率為ADC_CK/ OSR，設置暫存器ADCCN3[7 :5]。

● 輸入參數

osr [in] : 設置ADC轉換輸出速率控制值

ADCCN3_OSR_256 : 轉換輸出速率為ADC_CK/256
ADCCN3_OSR_512 : 轉換輸出速率為ADC_CK/512
ADCCN3_OSR_1024 : 轉換輸出速率為ADC_CK/1024
ADCCN3_OSR_2048 : 轉換輸出速率為ADC_CK/2048
ADCCN3_OSR_4096 : 轉換輸出速率為ADC_CK/4096
ADCCN3_OSR_8192 : 轉換輸出速率為ADC_CK/8192
ADCCN3_OSR_16384 : 轉換輸出速率為ADC_CK/16384
ADCCN3_OSR_32768 : 轉換輸出速率為ADC_CK/32768

● 包含標頭檔

Driver/ADC.h

● 函數返回值

無

● 函數用法

```
/* 設置ADC轉換輸出速率為ADC_CK / 32768 */  
ADC_OSRConfig( ADCCN3_OSR_32768 );
```

6.3.22. ADC_DCSetConfig

● 函數

ADC_DCSetConfig(dcset) ;

● 函數功能

設置ADC信號輸入端的直流偏置電壓，設置暫存器ADCCN2[2 :0]。

● 輸入參數

dcset [in] : 設置ADC的信號輸入端直流偏置電壓

ADCCN2_DCSET_0 : 不偏壓
ADCCN2_DCSET_P1DIV4 : + VREF* 1/4
ADCCN2_DCSET_P1DIV2 : + VREF* 1/2
ADCCN2_DCSET_P3DIV4 : + VREF* 3/4
ADCCN2_DCSET_00 : 不偏壓
ADCCN2_DCSET_N1DIV4 : -VREF* 1/4
ADCCN2_DCSET_N1DIV2 : -VREF* 1/2
ADCCN2_DCSET_N3DIV4 : -VREF* 3/4

● 包含標頭檔

Driver/ADC.h

● 函數返回值

無

● 函數用法

```
/* 設置ADC的直流偏置電壓為+ VREF* 1/4 */
```

```
ADC_DCSetConfig( ADCCN2_DCSET_P1DIV4 );
```

6.3.23. ADC_INXConfig

- 函數

```
ADC_INXConfig(inxmode);
```

- 函數功能

設置ADC信號輸入端的輸入信號轉置器，設置暫存器AINET2[4:3]。

- 輸入參數

inxmode [in]：輸入信號轉置器工作模式選擇

AINET2_INX_MODE0：INH→ADH，INL→ADL

AINET2_INX_MODE1：INL→ADH & ADL，INH浮接

AINET2_INX_MODE2：INH→ADH & ADL，INL浮接

AINET2_INX_MODE3：INH→ADL，INL→ADH

- 包含標頭檔

Driver/ADC.h

- 函數返回值

無。

- 函數用法

```
/* 設置ADC輸入信號轉置器模式為INH→ADH，INL→ADL */
```

```
ADC_INXConfig( AINET2_INX_MODE0 );
```

7. SPI 串列通訊

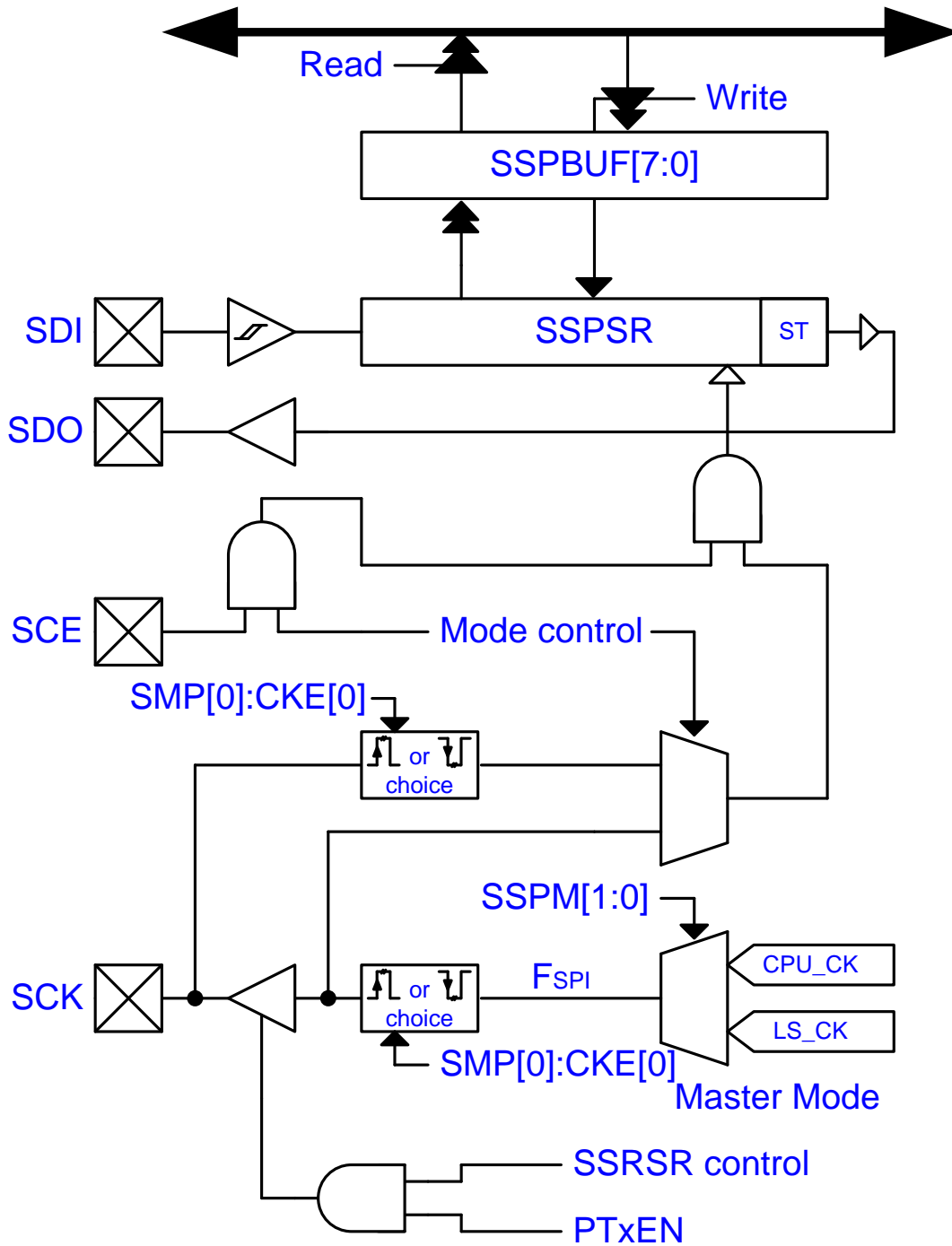
7.1. 函數簡介

該部分函數描述 SPI 功能的控制，包括：

- SPI 功能的開啓控制及中斷向量的控制
- SPI 的工作模式配置及工作狀態控制
- SPI 的資料的收發
- 包含 SPI.h

序號	函數名稱	功能描述
01	SPI_Open	開啓SPI功能，設置SPI工作模式，設置SPI通訊IO，設置SPI在收發資料完成後SCK電位狀態及SPI收發資料時SCK電位狀態
02	SPI_INT_Enable	使能 SPI 中斷功能
03	SPI_INT_Disable	關閉 SPI 中斷功能
04	SPI_INT_IsFlag	讀取SPI 中斷請求旗標
05	SPI_INT_ClearFlag	清除 SPI 中斷請求旗標
06	SPI_Enable	開啓SPI 功能
07	SPI_Disable	關閉 SPI 功能
08	SPI_CLKIdleConfig	設置資料收發完整後SCK電位狀態
09	SPI_CLKConfig	設置資料收發時SCK電位狀態
10	SPI_MasterMode	設置SPI 主動模式下輸入資料採樣時間點
11	SPI_SlaveMode	設置SPI 被動模式下輸入資料採樣時間點
12	SPI_Mode	設置SPI 工作模式
13	SPI_BUYCheck	讀取SPI寫入衝突狀態旗標
14	SPI_BFCheck	接收緩衝器滿旗標
15	SPI_ClearPOV	讀取SPI 接收溢出旗標

7.2. SPI 模組方框圖



7.3. 函數說明

7.3.1. SPI_Open

● 函數

```
void SPI_Open(unsigned char sspm, unsigned char ckp, unsigned char cke, unsigned char smp);
```

● 函數功能

開啓SPI功能，設置SPI工作模式，設置SPI通訊IO，設置SPI在收發資料完成後SCK電位狀態及SPI收發資料時SCK電位狀態，設置暫存器SSPCON1/PT1DA/PT1M2/TRISC1。

● 輸入參數

sspm [in] : SPI工作模式設置

SSPCON1_SSPM_LSCK : SPI主動模式，時鐘源為LS_CK

SSPCON1_SSPM_CPUCK : SPI主動模式，時鐘源為CPU_CK

SSPCON1_SSPM_SCK : SPI 3-線被動模式，時鐘源由SCK引腳輸入，SCE作為IO引腳使用

SSPCON1_SSPM_4WIRE : SPI 4-線被動模式，時鐘源由SCK引腳輸入，SCE作為使能控制引腳

ckp [in] : 設置資料收發完整後SCK電位狀態

SSPCON1_CKP_HI : 時鐘源為高電位時為空間狀態

SSPCON1_CKP_LOW : 時鐘源為低電位時為空間狀態

cke [in] : 設置資料收發時SCK電位狀態

SSPCON1_CKE_IDLE : 當時鐘源從有效狀態變為空間狀態時發送

SSPCON1_CKE_EFFECTIVE : 當時鐘源從空間狀態變為有效狀態時發送

smp Div [in] : 設置SPI 輸入資料採樣時間點

SPI 主動模式下：

SSPCON1_SMP_MASTEREND : 在資料輸出時間的末端採樣輸入資料

SSPCON1_SMP_MASTERMID : 在資料輸出時間的中間採樣輸入資料

SPI 被動模式下：

SSPCON1_SMP_SLAVE : 被動模式下，使用者需要將SMP BIT置0

● 包含標頭檔

Driver/SPI.h

● 函數返回值

無

● 函數用法

/* 設置SPI 為主動模式且時鐘源為CPU_CK，有效狀態變為空間狀態時發送資料，及發送完成後為低電位，在時間中間採樣輸入資料 */

```
SPI_Open( SSPCON1_SSPM_CPUCK, SSPCON1_CKP_LOW,  
          SSPCON1_CKE_IDLE, SSPCON1_SMP_MASTERMID );
```

7.3.2. SPI_INT_Enable

● 函數

```
SPI_INT_Enable();
```

- 函數功能

使能 SPI 中斷功能，設置暫存器INTE2[2]=1。

- 輸入參數

無

- 包含標頭檔

Driver/SPI.h

- 函數返回值

無

- 函數用法

```
/* 使能 SPI 中斷功能 */  
SPI_INT_Enable();
```

7.3.3. SPI_INT_Disable

- 函數

SPI_INT_Disable();

- 函數功能

關閉 SPI 中斷功能，設置暫存器INTE2[2]=0。

- 輸入參數

無

- 包含標頭檔

Driver/SPI.h

- 函數返回值

無

- 函數用法

```
/* 關閉 SPI 中斷功能 */  
SPI_INT_Disable();
```

7.3.4. SPI_INT_IsFlag

- 函數

SPI_INT_IsFlag();

- 函數功能

讀取SPI 中斷請求旗標，設置暫存器INTF2[2]。

- 輸入參數

無

- 包含標頭檔

Driver/SPI.h

- 函數返回值

0x00 : SPI 沒有中斷請求
0x04 : SPI產生中斷請求

- 函數用法

```
/* 讀取 SPI 中斷請求旗標 */  
unsigned char flag;  
flag = SPI_INT_IsFlag(>>>2;
```

7.3.5. SPI_INT_ClearFlag

- 函數

```
SPI_INT_ClearFlag();
```

- 函數功能

清除 SPI 中斷請求旗標，設置暫存器INTF2[2]=0。

- 輸入參數

無

- 包含標頭檔

Driver/SPI.h

- 函數返回值

無

- 函數用法

```
/* 清除 SPI 中斷請求旗標 */  
SPI_INT_ClearFlag();
```

7.3.6. SPI_Enable

- 函數

```
SPI_Enable();
```

- 函數功能

開啓SPI 功能，設置暫存器SSPCON1[7]=1。

- 輸入參數

無

- 包含標頭檔

Driver/SPI.h

- 函數返回值

無

- 函數用法

```
/* 開啓SPI 功能 */  
SPI_Enable();
```

7.3.7. SPI_Disable

- 函數

```
SPI_Disable();
```

- 函數功能

關閉 SPI 功能，設置暫存器SSPCON1[7]=0。

● 輸入參數

無

● 包含標頭檔

Driver/SPI.h

● 函數返回值

無

● 函數用法

```
/* 關閉 SPI 功能 */
```

```
SPI_Disable();
```

7.3.8. SPI_CLKIdleConfig

● 函數

```
SPI_CLKIdleConfig(IdleCon);
```

● 函數功能

設置資料收發完整後SCK電位狀態，設置暫存器SSPCON1[6]。

● 輸入參數

IdleCon [in]：設置資料收發完整後SCK電位狀態

SSPCON1_CKP_HI : 時鐘源為高電位時為空閒狀態

SSPCON1_CKP_LOW : 時鐘源為低電位時為空閒狀態

● 包含標頭檔

Driver/SPI.h

● 函數返回值

無

● 函數用法

```
/* 設置 SPI 收發資料完整後SCK電位為高電位 */
```

```
SPI_CLKIdleConfig( SSPCON1_CKP_HI );
```

7.3.9. DrvSPI32_DisableRxInt

● 函數

```
SPI_CLKConfig(CLKCon);
```

● 函數功能

設置資料收發時SCK電位狀態，設置暫存器SSPCON1[5]。

● 輸入參數

CLKCon [in]：設置資料收發時SCK電位狀態

SSPCON1_CKE_IDLE : 當時鐘源從有效狀態變為空閒狀態時發送

SSPCON1_CKE_EFFECTIVE : 當時鐘源從空閒狀態變為有效狀態時發送

● 包含標頭檔

Driver/SPI.h

- 函數返回值

無

- 函數用法

/* 設置SCK從有效狀態變為空閒狀態時發送資料 */

SPI_CLKConfig(SSPCON1_CKE_IDLE);

7.3.10. SPI_MasterMode

- 函數

SPI_MasterMode(MModeCon);

- 函數功能

設置SPI 主動模式下輸入資料採樣時間點，設置暫存器SSPCON1[4] 。

- 輸入參數

MModeCon Div [in] : 設置SPI 輸入資料採樣時間點

SPI 主動模式下：

SSPCON1_SMP_MASTEREND : 在資料輸出時間的末端採樣輸入資料

SSPCON1_SMP_MASTERMID : 在資料輸出時間的中間採樣輸入資料

SPI 被動模式下：

SSPCON1_SMP_SLAVE : 被動模式下，使用者需要將SMP BIT置0

- 包含標頭檔

Driver/SPI.h

- 函數返回值

無

- 函數用法

/* 設置輸入資料採樣時間點為資料輸出時間的中間 */

SPI_MasterMode(SSPCON1_SMP_MASTERMID);

7.3.11. SPI_SlaveMode

- 函數

SPI_SlaveMode();

- 函數功能

設置SPI 被動模式下輸入資料採樣時間點，設置暫存器SSPCON1[4]=0 。

- 輸入參數

無

- 包含標頭檔

Driver/SPI.h

- 函數返回值

無

- 函數用法

/* 設置SPI 被動模式下輸入資料採樣時間點 */

```
SPI_SlaveMode();
```

7.3.12. SPI_Mode

- 函數

```
SPI_Mode(SSPM_Sel);
```

- 函數功能

設置 SPI 工作模式，讀取暫存器SSPCON1[1:0]。

- 輸入參數

SSPM_Sel [in] : SPI工作模式設置

SSPCON1_SSPM_LSCK : SPI主動模式，時鐘源為LS_CK

SSPCON1_SSPM_CPUCK : SPI主動模式，時鐘源為CPU_CK

SSPCON1_SSPM_SCK : SPI 3-線被動模式，時鐘源由SCK引腳輸入，SCE作為IO引腳使用

SSPCON1_SSPM_4WIRE : SPI 4-線被動模式，時鐘源由SCK引腳輸入，SCE作為使能控制引腳

- 包含標頭檔

Driver/SPI.h

- 函數返回值

無

- 函數用法

```
/* 設置SPI為主動模式，時鐘源為CPU_CK */
```

```
SPI_Mode( SSPCON1_SSPM_CPUCK );
```

7.3.13. SPI_BUYCheck

- 函數

```
SPI_BUYCheck();
```

- 函數功能

讀取SPI寫入衝突狀態旗標，設置暫存器SSPSTA[7]。

- 輸入參數

無

- 包含標頭檔

Driver/SPI.h

- 函數返回值

0x00 : 未發生衝突

0x80 : 發生寫入衝突，資料仍在發送

- 函數用法

```
/* 讀取SPI 寫入衝突狀態旗標 */
```

```
unsigned char flag ;
```

```
flag = SPI_BUYCheck();
```

7.3.14. SPI_BFCheck

- 函數

SPI_BFCheck();

- 函數功能

接收緩衝器滿旗標，讀取暫存器SSPSTA[0]。

- 輸入參數

無

- 包含標頭檔

Driver/SPI.h

- 函數返回值

0x00：接收未完整，SSPBUF為空

0x01：接收完成，SSPBUF已滿

- 函數用法

```
/* 讀取 SPI 接收緩衝器滿旗標 */
```

```
unsigned char flag;
```

```
flag = SPI_BFCheck();
```

7.3.15. SPI_ClearPOV

- 函數

SPI_ClearPOV();

- 函數功能

讀取SPI 接收溢出旗標，讀取暫存器SSPSTA[6]。

- 輸入參數

無

- 包含標頭檔

Driver/SPI.h

- 函數返回值

0x00：未發生溢出

0x40：發生溢出

- 函數用法

```
/* 讀取SPI 接收溢出旗標 */
```

```
unsigned char flag;
```

```
flag = SPI_ClearPOV();
```

8. 非同步串列通訊 UART

8.1. 函數簡介

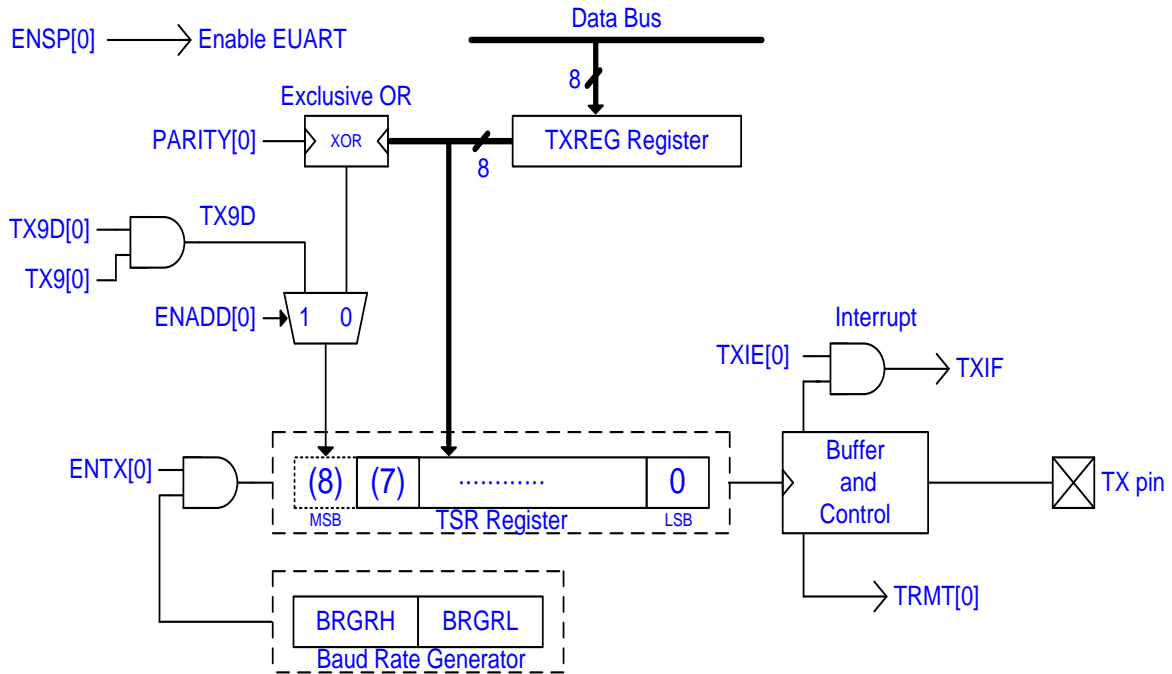
該部分函數描述對 UART 功能的控制，包含：

- UART 功能的啟動與關閉
- UART 功能的配置包括發送速率、時鐘源、資料格式等
- UART 資料的發送與接收
- UART 中斷向量控制
- UART 收發錯誤控制
- 包含 UART.h

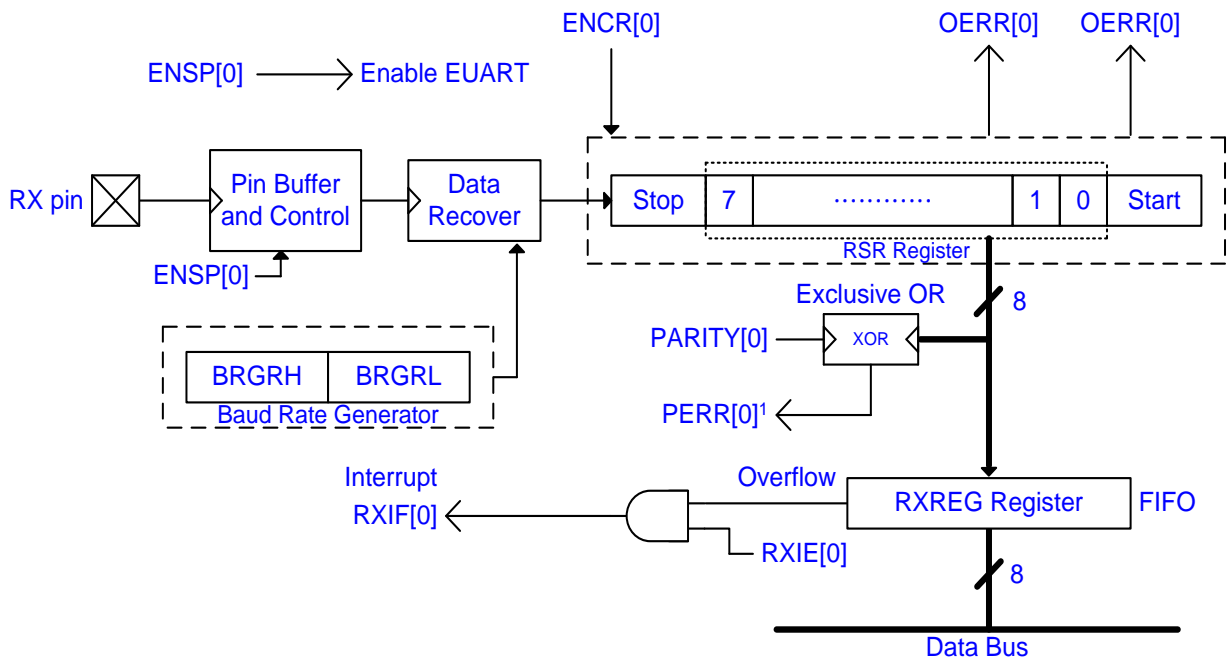
序號	函數名稱	功能描述
01	UART_Open	啟動UART及使能收發功能，設置串列傳輸速率值，設置資料校驗，設置UART通訊引腳，設置發送資料長度
02	UART_INT_TXEnable	使能UART發送中斷功能
03	UART_INT_TXIsEnable	讀取UART發送中斷使能控制位元狀態
04	UART_INT_TXDisable	關閉UART發送中斷功能
05	UART_INT_TXIsFlag	讀取UART發送中斷旗標狀態值
06	UART_INT_TXIsFlag	清除UART發送中斷旗標值
07	UART_INT_RCEnable	使能UART接收中斷功能
08	UART_INT_RCDisable	關閉UART接收中斷功能
09	UART_INT_RCIsFlag	讀取UART接收中斷請求旗標的值
10	UART_INT_RCClearFlag	清除UART接收中斷請求旗標
11	UART_Enable	使能UART功能
12	UART_Disable	關閉UART功能
13	UART_TXEnable	啟動UART發送功能
14	UART_TXDisable	關閉UART發送功能
15	UART_TX9Enable	啟動UART發送9位元資料的功能
16	UART_TX9Disable	關閉UART發送9位元資料功能
17	UART_WakeUpEnable	啟動接收自動喚醒功能
18	UART_WakeUpDisable	關閉接收自動喚醒功能
19	UART_DataReceiveEnable	使能UART接收功能
20	UART_DataReceiveDisable	關閉UART接收功能
21	UART_RC9Enable	使能UART接收9位元資料功能
22	UART_RC9Disable	關閉UART接收9位元資料功能
23	UART_AddrDetectionEnable	使能UART位地址檢測功能
24	UART_AddrDetectionDisable	關閉UART位地址檢測功能
25	UART_AutoBaudEnable	啟動UART自動串列傳輸速率功能
26	UART_AutoBaudDisable	關閉UART自動串列傳輸速率功能
27	UART_TXData9	設定UART發送9位元數據時的第9位的數值
28	UART_ParityCheck	設置UART的校驗模式
29	UART_RCData9	讀取接收9位元數據時，第9位的數值
30	UART_PERRIsFlag	讀取UART的資料同位元檢查結果旗標
31	UART_OERRIsFlag	讀取UART已接收到2筆資料未處理狀態旗標
32	UART_FERRIsFlag	讀取UART接收資料完整狀態旗標
33	UART_RCIDLIsFlag	讀取UART 的接收狀態旗標
34	UART_TRMTIsFlag	讀取UART發送移位暫存器(TSR)狀態旗標
35	UART_ABDVFIIsFlag	讀取UART自動串列傳輸速率溢出旗標

8.2. UART 模組方框圖

EUSART TRANSMIT BLOCK DIAGRAM



EUSART 8-BITS RECEIVE BLOCK DIAGRAM



¹Don't care PERR[0] state of 8-bits receive mode

8.3. 函數說明

8.3.1. UART_Open

- 函數

```
void UART_Open(unsigned int uBRGR,unsigned char uDataBits,unsigned char uParity);
```

- 函數功能

啓動UART及使能收發功能，設置串列傳輸速率值，設置資料校驗，設置UART通訊引腳，設置發送資料長度，設置暫存器URCON/BAUDCON/BRGRH/BRGRL/TRISC1/PT1PU/PT1M1。

- 輸入參數

uBRGR [in] : 設置UART通訊資料串列傳輸速率，輸入範圍0x00~0xFFFF

uDataBits [in] : UART數據長度

8 : 8位元數據

9 : 9位元數據

uParity [in] : 校驗模式，分別為無校驗/奇數同位檢查/偶校驗，設定值範圍

URCON_PARITY_Odd : 奇數同位檢查

URCON_PARITY_Even : 偶校驗

URCON_PARITY_None : 無校驗

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

```
/* HAO=2MHZ，設置UART串列傳輸速率為9600bp，8 位元資料，且無校驗，通訊口為PT1.4/PT1.3 */
```

```
UART_Open ( 0x33, 8 , URCON_PARITY_None );
```

8.3.2. UART_INT_TXEnable

- 函數

```
UART_INT_TXEnable();
```

- 函數功能

使能UART發送中斷功能，清零暫存器INTE2[7]=1。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

```
/* 使能UART發送中斷功能 */
```

```
UART_INT_TXEnable();
```

8.3.3. UART_INT_TXIsEnable

- 函數

```
UART_INT_TXIsEnable();
```

- 函數功能

讀取UART發送中斷使能控制位元狀態，讀取暫存器INTE2[7]。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

0x00：關閉UART發送中斷功能

0x80：使能UART發送中斷功能

- 函數用法

```
/* 讀取UART發送中斷使能控制位元狀態 */
```

```
unsigned char flag;
```

```
flag = UART_INT_TXIsEnable();
```

8.3.4. UART_INT_TXDisable

- 函數

```
UART_INT_TXDisable();
```

- 函數功能

關閉UART發送中斷功能，設置暫存器INTE2[7]=0。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

```
/* 關閉UART發送中斷功能 */
```

```
UART_INT_TXDisable();
```

8.3.5. UART_INT_TXIsFlag

- 函數

```
UART_INT_TXIsFlag();
```

- 函數功能

讀取UART發送中斷旗標狀態值，讀取暫存器INTF2[7]的值。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

0x80 : UART產生中斷請求

0x00 : UART未產生中斷請求

- 函數用法

```
/* 讀取UART發送中斷旗標 */
```

```
unsigned char flag ;
```

```
flag = UART_INT_TXIsFlag();
```

8.3.6. UART_INT_TXClearFlag

- 函數

```
UART_INT_TXIsFlag();
```

- 函數功能

清除UART發送中斷旗標值，設置暫存器INTF2[7]=0。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

```
/* 清除UART發送中斷旗標 */
```

```
UART_INT_TXIsFlag();
```

8.3.7. UART_INT_RCEnable

- 函數

```
UART_INT_RCEnable();
```

- 函數功能

使能UART接收中斷功能，設置暫存器INTE2[6]=1。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

```
/* 使能UART接收中斷功能 */
```

```
UART_INT_RCEnable();
```

8.3.8. UART_INT_RCDisable

- 函數

UART_INT_RCDisable();

- 函數功能

關閉UART接收中斷功能，設置暫存器INTE2[6]=0。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

```
/* 關閉UART接收中斷功能 */
```

```
UART_INT_RCDisable();
```

8.3.9. UART_INT_RCIsFlag

- 函數

UART_INT_RCIsFlag();

- 函數功能

讀取UART接收中斷請求旗標的值，讀取暫存器INTF2[6]。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

0x00 : UART無接收中斷請求

0x40 : UART有接收中斷請求

- 函數用法

```
/* 讀取UART的接收中斷旗標 */
```

```
unsigned char flag;
```

```
flag = UART_INT_RCIsFlag();
```

8.3.10. UART_INT_RCClearFlag

- 函數

UART_INT_RCClearFlag();

- 函數功能

清除UART接收中斷請求旗標，設置暫存器INTF2[6]=0。

- 輸入參數

uData [in] : 待發送的資料

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

/* 清除UART接收中斷請求旗標 */

UART_INT_RCClearFlag();

8.3.11. UART_Enable

- 函數

UART_Enable();

- 函數功能

使能UART功能，設置暫存器URCON[7]=1。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

/* 使能UART功能 */

UART_Enable();

8.3.12. UART_Disable

- 函數

UART_Disable();

- 函數功能

關閉UART功能，設置暫存器URCON[7]=0。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

/* 關閉UART功能 */

UART_Disable();

8.3.13. UART_TXEnable

- 函數

UART_TXEnable();

- 函數功能

啓動UART發送功能，設置暫存器URCON[6]=1。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

```
/* 啓動UART發送功能 */
```

```
UART_TXEnable();
```

8.3.14. UART_TXDisable

- 函數

UART_TXDisable();

- 函數功能

關閉UART發送功能，設置暫存器URCON[6]=0。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

```
/* 關閉UART發送功能 */
```

```
UART_TXDisable();
```

8.3.15. UART_TX9Enable

- 函數

UART_TX9Enable();

- 函數功能

啓動UART發送9位元資料的功能，設置暫存器URCON[5]=1。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

/* 啓動UART發送9位元資料功能 */

UART_TX9Enable();

8.3.16. UART_TX9Disable

- 函數

UART_TX9Disable();

- 函數功能

關閉UART發送9位元資料功能，設置暫存器URCON[5]=0。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

/* 關閉UART發送9位元資料功能 */

UART_TX9Disable();

8.3.17. UART_WakeUpEnable

- 函數

UART_WakeUpEnable();

- 函數功能

啓動接收自動喚醒功能，設置暫存器URCON[0]=1。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

/* 啓動接收自動喚醒功能 */

UART_WakeUpEnable();

8.3.18. UART_WakeUpDisable

- 函數

UART_WakeUpDisable();

- 函數功能

關閉接收自動喚醒功能，設置暫存器URCON[0]=0。

● 輸入參數

無

● 包含標頭檔

Drviver/UART.h

● 函數返回值

無

● 函數用法

```
/* 關閉接收自動喚醒功能 */  
UART_WakeUpDisable();
```

8.3.19. UART_DataReceiveEnable

● 函數

```
UART_DataReceiveEnable();
```

● 函數功能

使能UART接收功能，設置暫存器BAUDCON[3]=1。

● 輸入參數

無

● 包含標頭檔

Drviver/UART.h

● 函數返回值

無

● 函數用法

```
/* 使能UART接收功能 */  
UART_DataReceiveEnable();
```

8.3.20. UART_DataReceiveDisable

● 函數

```
UART_DataReceiveDisable();
```

● 函數功能

關閉UART接收功能，設置暫存器BAUDCON[3]=0。

● 輸入參數

無

● 包含標頭檔

Drviver/UART.h

● 函數返回值

無

● 函數用法

```
/* 關閉UART接收功能 */  
UART_DataReceiveDisable();
```

8.3.21. UART_RC9Enable

- 函數

UART_RC9Enable();

- 函數功能

使能UART接收9位元資料功能，設置暫存器BAUDCON[2]=1。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

```
/* 使能UART接收9位元資料功能 */
```

```
UART_RC9Enable();
```

8.3.22. UART_RC9Disable

- 函數

UART_RC9Disable();

- 函數功能

關閉UART接收9位元資料功能，設置暫存器BAUDCON[2]=0。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

```
/* 關閉UART接收9位元資料功能 */
```

```
UART_RC9Disable();
```

8.3.23. UART_AddrDetectionEnable

- 函數

UART_AddrDetectionEnable();

- 函數功能

使能UART位址檢測功能，讀取暫存器BAUDCON[1]=1。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

```
/* 使能UART位地址檢測功能 */  
UART_AddrDetectionEnable();
```

8.3.24. UART_AddrDetectionDisable

- 函數

```
UART_AddrDetectionDisable();
```

- 函數功能

關閉UART位地址檢測功能，讀取暫存器BAUDCON[1]=0。

- 輸入參數

無。

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

```
/* 關閉UART位地址檢測功能 */  
UART_AddrDetectionDisable();
```

8.3.25. UART_AutoBaudEnable

- 函數

```
UART_AutoBaudEnable();
```

- 函數功能

啟動UART自動串列傳輸速率功能，設置暫存器BAUDCON[0]=1。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

```
/* 啟動UART自動串列傳輸速率功能 */  
UART_AutoBaudEnable();
```

8.3.26. UART_AutoBaudDisable

- 函數

```
UART_AutoBaudDisable();
```

- 函數功能

關閉UART自動串列傳輸速率功能，設置暫存器BAUDCON[0]=0。

● 輸入參數

無

● 包含標頭檔

Drviver/UART.h

● 函數返回值

無

● 函數用法

```
/* 關閉UART自動串列傳輸速率功能 */  
UART_AutoBaudDisable();
```

8.3.27. UART_TXData9

● 函數

```
UART_TXData9(TX9DSel);
```

● 函數功能

設定UART發送9位元數據時的第9位的數值，設置暫存器URCON[4]。

● 輸入參數

TX9Dsel [in]：設定UART發送9位元數據時的第9位的數值

URCON_TX9D_1：第9位元數據為1

URCON_TX9D_0：第9位元數據為0

● 包含標頭檔

Drviver/UART.h

● 函數返回值

無

● 函數用法

```
/* 設置UART發送9位元資料時第九位元數值為1 */  
UART_TX9Enable();  
UART_TXData9( URCON_TX9D_1 );
```

8.3.28. UART_ParityCheck

● 函數

```
UART_ParityCheck(PARSel);
```

● 函數功能

設置UART的校驗模式，暫存器URCON[3]。

● 輸入參數

PARSel [in]：校驗模式，分別為無校驗/奇數同位檢查/偶校驗，設定值範圍：

URCON_PARITY_Odd：奇數同位檢查

URCON_PARITY_Even：偶校驗

URCON_PARITY_None：無校驗

- 包含標頭檔

Drviver/UART.h

- 函數返回值

無

- 函數用法

```
/* 設置UART為偶校驗 */
```

```
UART_ParityCheck( URCON_PARITY_Even );
```

8.3.29. UART_RCData9

- 函數

```
UART_RCData9();
```

- 函數功能

讀取接收9位元數據時，第9位的數值，讀取暫存器BAUDCON[2]。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

返回BAUDCON的值：

0x00 : RC9D為0

0x04 : RC9D為1

- 函數用法

```
/* 讀取接收到的第9位數值 */
```

```
unsigned char RC9D;
```

```
RC9D = UART_RCData9()>>2;
```

8.3.30. UART_PERRIsFlag

- 函數

```
UART_PERRIsFlag();
```

- 函數功能

讀取UART的資料同位元檢查結果旗標(PERR)，讀取暫存器URSTA[5]。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

0x00 : 接收同位檢查正確

0x20 : 接收同位元檢查錯誤

- 函數用法

```
/* 取UART的資料同位元檢查結果旗標(PERR) */  
unsigned char PERR_F;  
PERR_F = UART_PERRIsFlag();
```

8.3.31. UART_OERRIsFlag

- 函數

```
UART_OERRIsFlag();
```

- 函數功能

讀取UART已接收到2筆資料未處理狀態旗標(OERR)，讀取暫存器URSTA[3]。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

0x00：未發生

0x08：已發生

- 函數用法

```
/* 讀取UART已接收到2筆資料未處理狀態旗標(OERR) */  
unsigned char OERR_F;  
OERR_F = UART_OERRIsFlag();
```

8.3.32. UART_FERRIsFlag

- 函數

```
UART_FERRIsFlag()
```

- 函數功能

讀取UART接收資料完整狀態旗標(FERR)，讀取暫存器URSTA[4]。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

0x00：表示資料接收完整

0x10：表示資料接收不完整

- 函數用法

```
/* 讀取UART接收資料完整狀態旗標(FERR) */  
unsigned char FERR_F ;  
FERR_F = UART_FERRIsFlag();
```

8.3.33. UART_RCIDLIsFlag

- 函數

UART_RCIDLsFlag();

- 函數功能

讀取UART 的接收狀態旗標(RCIDL)，讀取暫存器URSTA[2]。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

0x00：不在接收狀態

0x04：在接收狀態

- 函數用法

```
/* 讀取UART 的接收狀態旗標(RCIDL) */
```

```
unsigned char flag;
```

```
flag = UART_RCIDLsFlag();
```

8.3.34. UART_TRMTIsFlag

- 函數

UART_TRMTIsFlag();

- 函數功能

讀取UART發送移位暫存器(TSR)狀態旗標(RTMT)，讀取暫存器URSTA[1]。

- 輸入參數

無

- 包含標頭檔

Drviver/UART.h

- 函數返回值

0x00：表示TSR暫存器有資料

0x02：表示TSR暫存器為空

- 函數用法

```
/* 讀取UART發送移位暫存器(TSR)狀態旗標(RTMT) */
```

```
unsigned char flag;
```

```
flag = UART_TRMTIsFlag();
```

8.3.35. UART_ABDOVFsFlag

- 函數

UART_ABDOVFsFlag();

- 函數功能

讀取UART自動串列傳輸速率溢出旗標(ABDOVF)，讀取暫存器URSTA[0]。

- 輸入參數

無

● 包含標頭檔

Drviver/UART.h

● 函數返回值

0x00 : 串列傳輸速率沒發生溢出

0x01 : 串列傳輸速率發生溢出

● 函數用法

/* 讀取UART自動串列傳輸速率溢出旗標(ABDOVF) */

unsigned char flag ;

flag = UART_ABDOVFIsFlag();

9. 增強型多功能比較器 ECPA

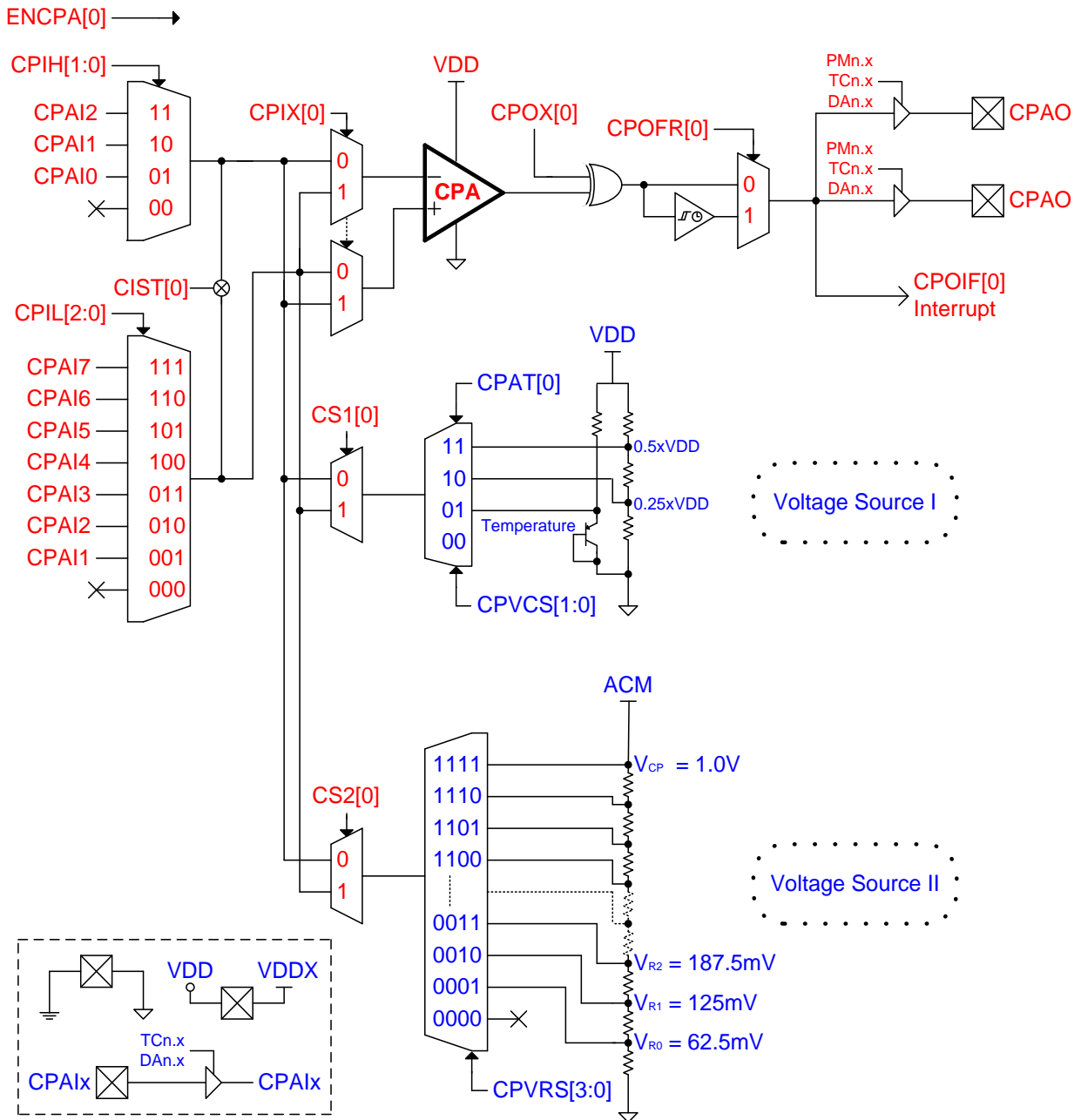
9.1. 函數簡介

該部分函數描述增強型比較器功能的控制，包含：

- ECPA 模組功能的啓動與關閉
- ECPA 配置控制
- ECPA 中斷向量控制
- 包含 ECPA.h

序號	函數名稱	功能描述
01	ECPA_Open	使能比較器功能，設置比較器信號輸入通道，設置比較器參考電壓及設置比較器信號輸出引腳
02	ECPA_INT_Enable	使能比較器中斷功能
03	ECPA_INT_Disable	關閉比較器中斷功能
04	ECPA_INT_IsFlag	讀取比較器中斷請求旗標
05	ECPA_INT_ClearFlag	清除比較器中斷請求旗標
06	ECPA_Enable	使能比較器功能
07	ECPA_Disable	關閉比較器功能
08	ECPA_InShort	設置比較器輸入通道短路器功能
9	ECPA_SignConverter	設置輸入通道交換器功能
10	ECPA_InChanH	設置比較器正向輸入通道
11	ECPA_InChanL	設置比較器負向輸入通道
12	ECPA_OutReverse	設置比較器輸出信號反相器功能
13	ECPA_SignProcessor	設置比較器輸出信號經過濾波器功能
14	ECPA_OutState	設置比較器使用參考電壓發生器I時，輸出狀態自動轉態
15	ECPA_Ref2CompIn1	設置參考電壓發生器I的輸出信號，輸出到正向端或負向端
16	ECPA_RefVolChan1	設置比較器參考電壓發生器 I 的輸出電壓
17	ECPA_Ref2CompIn2	設置參考電壓發生器II的輸出信號，輸出到正向端或負向端
18	ECPA_RefVolChan2	設置比較器參考電壓發生器II的輸出電壓

9.2. ECPA 模組方框圖



9.3. 函數說明

9.3.1. ECPA_Open

● 函數

```
void ECPA_Open(unsigned char cpil, unsigned char cpih, unsigned char cpvcs,  
               unsigned char cpvrx, unsigned char output);
```

● 函數功能

使能比較器功能，設置比較器信號輸入通道，設置比較器參考電壓及設置比較器信號輸出引腳，設置暫存器CPACN1/CPACN2/CPACN3。

● 輸入參數

cpih [in] : 比較器信號正向輸入通道設置

CPACN1_CPIH_CAPI2 : 輸入通道CAPI2

CPACN1_CPIH_CAPI1 : 輸入通道CAPI1

CPACN1_CPIH_CAPI0 : 輸入通道CAPI0

CPACN1_CPIH_HIZ : 高阻抗

cpil [in] : 比較器信號負向輸入通道設置

CPACN1_CPIL_CAPI7 : 輸入通道CAPI7

CPACN1_CPIL_CAPI6 : 輸入通道CAPI6

CPACN1_CPIL_CAPI5 : 輸入通道CAPI5

CPACN1_CPIL_CAPI4 : 輸入通道CAPI4

CPACN1_CPIL_CAPI3 : 輸入通道CAPI3

CPACN1_CPIL_CAPI2 : 輸入通道CAPI2

CPACN1_CPIL_CAPI1 : 輸入通道CAPI1

CPACN1_CPIL_HIZ : 高阻抗

cpvcs [in] : 設置比較器參考電壓發生器I的輸出電壓

CPACN2_CPVCS_VDDDIV2 : 0.5 * VDD

CPACN2_CPVCS_VDDDIV4 : 0.25 * VDD

CPACN2_CPVCS_REFTEMP : 參考溫度

CPACN2_CPVCS_HIZ : 高阻抗

cpvrx [in] : 設置比較器參考電壓發生器II的輸出電壓

CPACN3_CPVRX_1V000 : 輸出1000mV

CPACN3_CPVRX_0V9375 : 輸出0.9375mV

CPACN3_CPVRX_0V875 : 輸出0.875mV

CPACN3_CPVRX_0V8125 : 輸出0.8125mV

CPACN3_CPVRX_0V750 : 輸出0.750mV

CPACN3_CPVRX_0V6875 : 輸出0.6875mV

CPACN3_CPVRX_0V625 : 輸出0.625mV

CPACN3_CPVRX_0V5625 : 輸出0.5625mV

CPACN3_CPVRX_0V4375 : 輸出0.4375mV

CPACN3_CPVRX_0V375 : 輸出0.375mV
CPACN3_CPVRX_0V3125 : 輸出0.3125mV
CPACN3_CPVRX_0V250 : 輸出0.250mV
CPACN3_CPVRX_0V1875 : 輸出0.1875mV
CPACN3_CPVRX_0V125 : 輸出0.125mV
CPACN3_CPVRX_0V0625 : 輸出0.0625mV
CPACN3_CPVRX_HIZ : 高阻抗

output [in] : 設置比較器輸出信號引腳

0 : 沒有輸出
1 : PT2.7
2 : PT2.7
3 : PT2.6 & PT2.7

● 包含標頭檔

Driver/ECPA.h

● 函數返回值

無

● 函數用法

/* 使能比較器，設置輸入通道為CPIH=AI2,CPII=AI3，參考電壓發生器I輸出電壓為0.5*VDD，參考電壓發生器II輸出電壓為0.375mv，輸出引腳為PT2.7 */

```
ECPA_Open( CPACN1_CPIH_CAPI2, CPACN1_CPII_CAPI3, CPACN2_CPVCS_VDDDIV2,  
           CPACN3_CPVRX_0V375, 2 );
```

9.3.2. ECPA_INT_Enable

● 函數

```
ECPA_INT_Enable();
```

● 函數功能

使能比較器中斷功能，設置暫存器INTE2[3]=1。

● 輸入參數

無

● 包含標頭檔

Driver/ECPA.h

● 函數返回值

無

● 函數用法

/* 使能比較器中斷功能 */

```
ECPA_INT_Enable();
```

9.3.3. ECPA_INT_Disable

● 函數

```
ECPA_INT_Disable();
```

- 函數功能

關閉比較器中斷功能，設置暫存器INTE2[3]=0。

- 輸入參數

無

- 包含標頭檔

Driver/ECPA.h

- 函數返回值

無

- 函數用法

```
/* 關閉比較器中斷功能 */
```

```
ECPA_INT_Disable();
```

9.3.4. ECPA_INT_IsFlag

- 函數

```
ECPA_INT_IsFlag();
```

- 函數功能

讀取比較器中斷請求旗標，讀取暫存器INTF2[3]。

- 輸入參數

無

- 包含標頭檔

Driver/ECPA.h

- 函數返回值

0x00：比較器沒有產生中斷請求

0x08：比較器產生中斷請求

- 函數用法

```
/* 讀取比較器中斷請求旗標 */
```

```
unsigned char flag;
```

```
flag = ECPA_INT_IsFlag();
```

9.3.5. ECPA_INT_ClearFlag

- 函數

```
ECPA_INT_ClearFlag();
```

- 函數功能

清除比較器中斷請求旗標，設置暫存器INTF2[3]=0。

- 輸入參數

無

- 包含標頭檔

Driver/ECPA.h

- 函數返回值

無

- 函數用法

/* 清除比較器中斷請求旗標 */

```
ECPA_INT_ClearFlag();
```

9.3.6. ECPA_Enable

- 函數

```
ECPA_Enable();
```

- 函數功能

使能比較器功能，設置暫存器CPACN1[7]=1。

- 輸入參數

無

- 包含標頭檔

Driver/ECPA.h

- 函數返回值

無

- 函數用法

/* 開啓比較器功能 */

```
ECPA_Enable();
```

9.3.7. ECPA_Disable

- 函數

```
ECPA_Disable();
```

- 函數功能

關閉比較器功能，設置暫存器CPACN1[7]=0。

- 輸入參數

無

- 包含標頭檔

Driver/ECPA.h

- 函數返回值

無

- 函數用法

/* 關閉比較器功能 */

```
ECPA_Disable();
```

9.3.8. ECPA_InShort

- 函數

```
ECPA_InShort(ISTSel);
```

- 函數功能

設置比較器輸入通道短路器功能，設置暫存器CPACN1[6]。

- 輸入參數

ISTSel [in] 設置比較器輸入通道短路器功能

CPACN1_CPIST_SHORT : 輸入短路

CPACN1_CPIST_OPEN : 輸入不短路

- 包含標頭檔

Driver/ECPA.h

- 函數返回值

無

- 函數用法

/* 使能比較器輸入通道短路 */

ECPA_InShort(CPACN1_CPIST_SHORT);

9.3.9. ECPA_SignConverter

- 函數

ECPA_SignConverter(IXSel);

- 函數功能

設置輸入通道交換器功能，設置暫存器CPACN1[5]。

- 輸入參數

IXSel [in] : 設置輸入通道交換器功能

CPACN1_CPIX_SWITCH : 輸入通道交換

CPACN1_CPIX_NOTSWITCH : 輸入通道不交換

- 包含標頭檔

Driver/ECPA.h

- 函數返回值

無

- 函數用法

/* 設置比較器輸入信號通道不交換 */

ECPA_SignConverter(CPACN1_CPIX_NOTSWITCH);

9.3.10. ECPA_InChanH

- 函數

ECPA_InChanH(IHSel);

- 函數功能

設置比較器正向輸入通道，設置暫存器CPACN1[4:3]。

- 輸入參數

IHSel [in] : 比較器信號正向輸入通道設置

CPACN1_CPIH_CAPI2 : 輸入通道CAPI2

CPACN1_CPIH_CAPI1 : 輸入通道CAPI1
CPACN1_CPIH_CAPI0 : 輸入通道CAPI0
CPACN1_CPIH_HIZ : 高阻抗

- 包含標頭檔

Driver/ECPA.h

- 函數返回值

無

- 函數用法

/* 設置比較器正向輸入通道為AI2 */

ECPA_InChanH(CPACN1_CPIH_CAPI2);

9.3.11. ECPA_InChanL

- 函數

ECPA_InChanL(ILSel);

- 函數功能

設置比較器負向輸入通道，設置暫存器CPACN1[2:0]。

- 輸入參數

ILSel [in] : 比較器信號負向輸入通道設置

CPACN1_CPIL_CAPI7 : 輸入通道CAPI7

CPACN1_CPIL_CAPI6 : 輸入通道CAPI6

CPACN1_CPIL_CAPI5 : 輸入通道CAPI5

CPACN1_CPIL_CAPI4 : 輸入通道CAPI4

CPACN1_CPIL_CAPI3 : 輸入通道CAPI3

CPACN1_CPIL_CAPI2 : 輸入通道CAPI2

CPACN1_CPIL_CAPI1 : 輸入通道CAPI1

CPACN1_CPIL_HIZ : 高阻抗

- 包含標頭檔

Driver/ECPA.h

- 函數返回值

無

- 函數用法

/* 設置比較器負向輸入通道為AI3 */

ECPA_InChanL(CPACN1_CPIL_CAPI3);

9.3.12. ECPA_OutReverse

- 函數

ECPA_OutReverse(OXSel);

- 函數功能

設置比較器輸出信號反相器功能，設置暫存器CPACN2[6]。

- 輸入參數

OXSel [in] : 設置比較器輸出信號反相器功能

CPACN2_CPOX_REVERSE : 輸出反向

CPACN2_CPOX_NOTREVERSE : 輸出不反向

- 包含標頭檔

Driver/ECPA.h

- 函數返回值

無

- 函數用法

/* 設置比較器輸出信號反向 */

ECPA_OutReverse(CPACN2_CPOX_REVERSE);

9.3.13. ECPA_SignProcessor

- 函數

ECPA_SignProcessor(OFRSel);

- 函數功能

設置比較器輸出信號經過濾波器功能，讀取暫存器CPACN2[5]的值。

- 輸入參數

OFRSel [in] : 設置比較器輸出信號經過濾波功能

CPACN2_CPOFR_FILTER : 輸出信號經過濾波器

CPACN2_CPOFR_NOTFILTER : 輸出信號不經過濾波器

- 包含標頭檔

Driver/ECPA.h

- 函數返回值

無

- 函數用法

/* 設置比較器輸出信號不經過濾波器 */

ECPA_SignProcessor(CPACN2_CPOFR_NOTFILTER);

9.3.14. ECPA_OutState

- 函數

ECPA_OutState(ATSel);

- 函數功能

設置比較器使用參考電壓發生器I時，比較器輸出狀態是否自動轉態，設置暫存器CPACN2[3]。

- 輸入參數

ATSel [in] : 設置比較器使用參考電壓發生器I時，比較器輸出狀態是否自動狀態

CPACN2_CPAT_AUTO : 輸出狀態自動轉態

CPACN2_CPAT_NORMAL : 正常操作

- 包含標頭檔

Driver/ECPA.h

● 函數返回值

無

● 函數用法

/* 設置比較器在使用參考電壓發生器I時，輸出狀態自動轉態 */

ECPA_OutState(CPACN2_CPAT_AUTO);

9.3.15. ECPA_Ref2Compln1

● 函數

ECPA_Ref2Compln1(CS1Sel);

● 函數功能

設置參考電壓發生器I的輸出信號，輸出到正向端或負向端，設置暫存器CPACN2[4]。

● 輸入參數

CS1Sel [in]：設置參考電壓發生器I的輸出信號，輸出到正向端或負向端

當CPIX = 1時：

CPACN2_CS1_P：輸出至 '+' 端

CPACN2_CS1_N：輸出至 '-' 端

當CPIX = 0時：

CPACN2_CS1_P：輸出至 '-' 端

CPACN2_CS1_N：輸出至 '+' 端

● 包含標頭檔

Driver/ECPA.h

● 函數返回值

無

● 函數用法

/* CPIX=1，設置參考電壓輸出信號至 '+' 端 */

ECPA_SignConverter(CPACN1_CPIX_SWITCH);

ECPA_Ref2Compln1(CPACN2_CS1_P);

9.3.16. ECPA_RefVolChan1

● 函數

ECPA_RefVolChan1(VCSSel);

● 函數功能

設置比較器參考電壓發生器 I 的輸出電壓，設置暫存器CPACN2[2:1]。

● 輸入參數

VCSSel [in]：設置比較器參考電壓發生器 I 的輸出電壓

CPACN2_CPVCS_VDDDIV2 : 0.5 * VDD

CPACN2_CPVCS_VDDDIV4 : 0.25 * VDD

CPACN2_CPVCS_REFTEMP : 參考溫度

CPACN2_CPVCS_HIZ : 高阻抗

- 包含標頭檔

Driver/ECPA.h

- 函數返回值

無

- 函數用法

/* 設置比較器參考電壓發生器I的輸出電壓為0.5*VDD */

ECPA_RefVolChan1(CPACN2_CPVCS_VDDDIV2);

9.3.17. ECPA_Ref2CompIn2

- 函數

ECPA_Ref2CompIn2(CS2Sel);

- 函數功能

設置參考電壓發生器II的輸出信號，輸出到正向端或負向端，設置暫存器CPACN3[4]。

- 輸入參數

CS2Sel [in] : 設置參考電壓發生器I的輸出信號，輸出到正向端或負向端

當CPIX = 1時：

CPACN3_CS2_P : 輸出至 '+' 端

CPACN3_CS2_N : 輸出至 '-' 端

當CPIX = 0時：

CPACN3_CS2_P : 輸出至 '-' 端

CPACN3_CS2_N : 輸出至 '+' 端

- 包含標頭檔

Driver/ECPA.h

- 函數返回值

無

- 函數用法

/* CPIX=1，設置參考電壓發生器II輸出信號至 '+' 端 */

ECPA_SignConverter(CPACN1_CPIX_SWITCH);

ECPA_Ref2CompIn2(CPACN3_CS2_P);

9.3.18. ECPA_RefVolChan2

- 函數

ECPA_RefVolChan2(VRXSel);

- 函數功能

設置比較器參考電壓發生器II的輸出電壓，設置暫存器CPACN3[3:0]。

- 輸入參數

VRXSel [in] : 設置比較器參考電壓發生器II的輸出電壓

CPACN3_CPVRX_1V000 : 輸出1000mV

CPACN3_CPVRX_0V9375 : 輸出0.9375mV
CPACN3_CPVRX_0V875 : 輸出0.875mV
CPACN3_CPVRX_0V8125 : 輸出0.8125mV
CPACN3_CPVRX_0V750 : 輸出0.750mV
CPACN3_CPVRX_0V6875 : 輸出0.6875mV
CPACN3_CPVRX_0V625 : 輸出0.625mV
CPACN3_CPVRX_0V5625 : 輸出0.5625mV
CPACN3_CPVRX_0V4375 : 輸出0.4375mV
CPACN3_CPVRX_0V375 : 輸出0.375mV
CPACN3_CPVRX_0V3125 : 輸出0.3125mV
CPACN3_CPVRX_0V250 : 輸出0.250mV
CPACN3_CPVRX_0V1875 : 輸出0.1875mV
CPACN3_CPVRX_0V125 : 輸出0.125mV
CPACN3_CPVRX_0V0625 : 輸出0.0625mV
CPACN3_CPVRX_HIZ : 高阻抗

● 包含標頭檔

Driver/ECPA.h

● 函數返回值

無

● 函數用法

/* 設置參考電壓發生器 II 的輸出電壓為 0.375mV */

ECPA_RefVolChan2(CPACN3_CPVRX_0V375);

10. 低雜訊放大器 LNOP1/LNOP2

10.1. 功能簡介

該函數部分描述低雜訊放大器 LNOP1/LNOP2 功能的操作

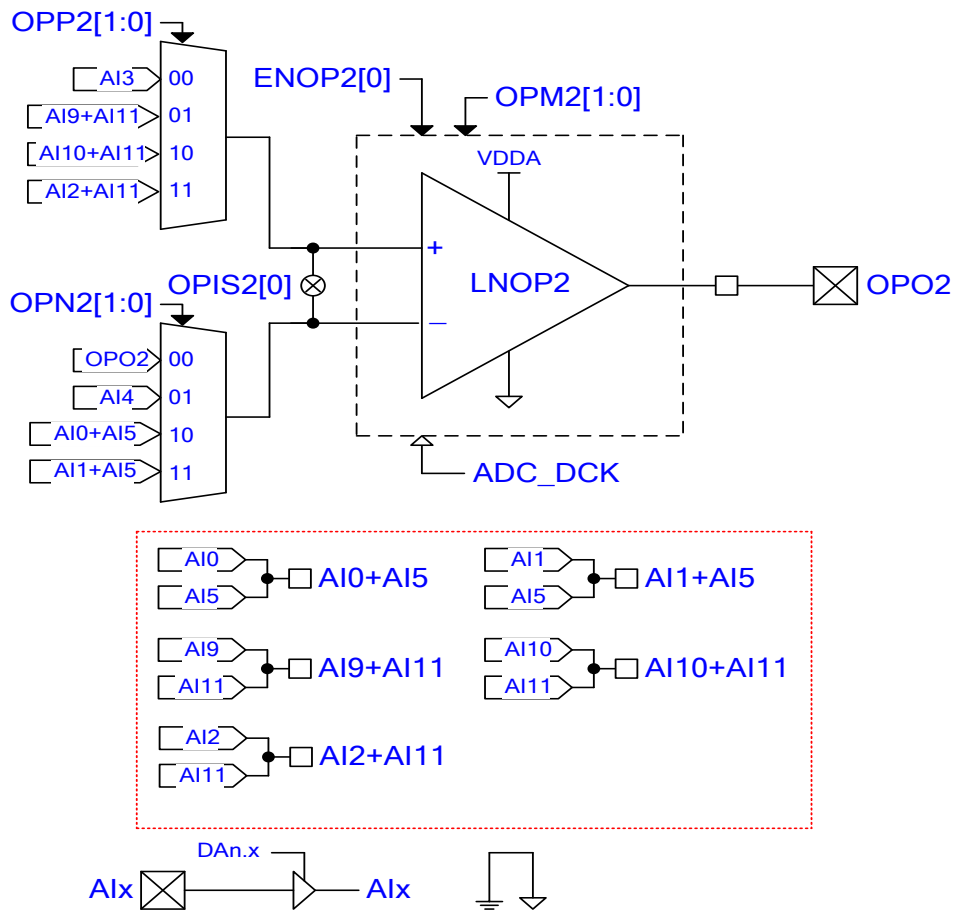
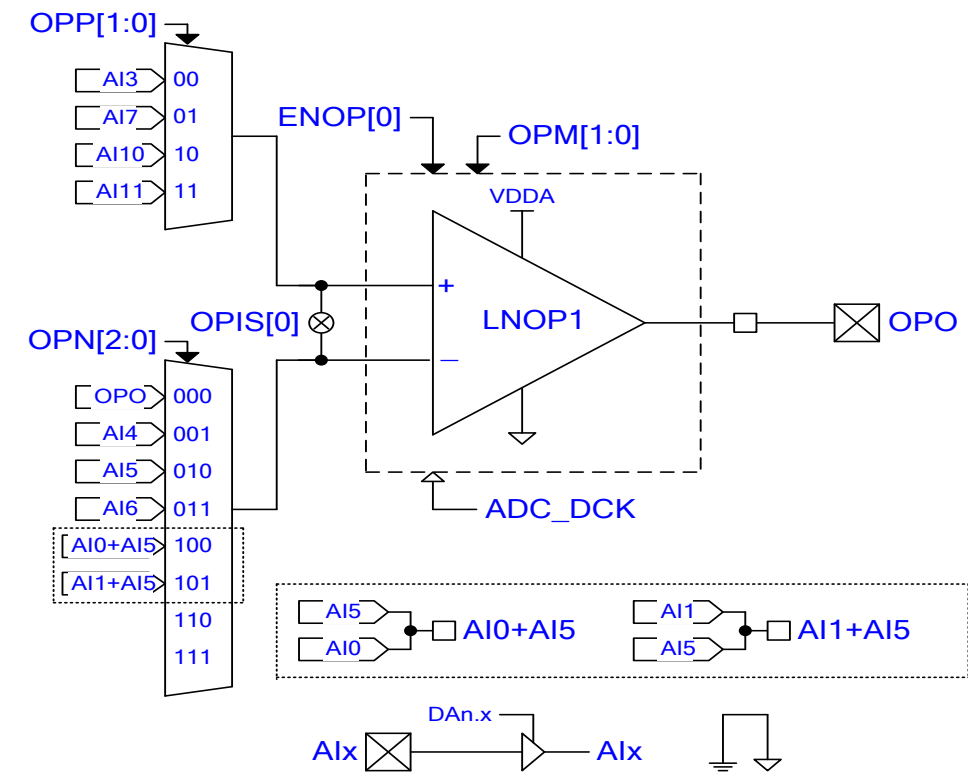
--LNOP1/LNOP1 功能的開關

--LNOP1/LNOP2 輸入埠及輸出的控制

--包含 LNOP1.h/LNOP2.h

序號	函數名稱	函數功能
01	LNOP1_Open	開啓低雜訊運算放大器(LNOP1)功能，設置LNOP1輸入端網路及chooper功能
02	LNOP1_Enable	開啓低雜訊運算放大器(LNOP1)功能
03	LNOP1_Disable	關閉低雜訊運算放大器(LNOP1)功能
04	LNOP1_OPMode	設置低雜訊運算放大器（LNOP1）輸入處理器模式
05	LNOP1_OPChanIn	設置低雜訊運算放大器（LNOP1）輸入端短路功能
06	LNOP1_OPIInputP	設置低雜訊運算放大器（LNOP1）正向輸入端通道
07	LNOP1_OPIInputN	設置低雜訊運算放大器（LNOP1）負向輸入端通道
08	LNOP2_Open	開啓低雜訊運算放大器(LNOP2)功能，設置LNOP2輸入端網路及chooper功能
09	LNOP2_Enable	開啓低雜訊運算放大器(LNOP2)功能
10	LNOP2_Disable	關閉低雜訊運算放大器(LNOP2)功能
11	LNOP2_OPMode	設置低雜訊運算放大器（LNOP2）輸入處理器模式
12	LNOP2_OPChanIn	設置低雜訊運算放大器（LNOP2）輸入端短路功能
13	LNOP2_OPIInputP	設置低雜訊運算放大器（LNOP2）正向輸入端通道
14	LNOP2_OPIInputN	設置低雜訊運算放大器（LNOP2）負向輸入端通道

10.2. LNOP1/LN OP2 模組方框圖



10.3. 函數說明

注意：使用LNOP1需要先開啓VDDA/ACM電壓！

10.3.1. LNOP1_Open

- 函數

```
void LNOP1_Open(unsigned char NInput, unsigned char PInput, unsigned char Chop);
```

- 函數功能

開啓低雜訊運算放大器(LNOP1)功能，設置LNOP1輸入端網路及chooper功能，設置暫存器OPCN1[7:0]。

- 輸入參數

NInput[in]：設置LNOP1負向輸入通道

OPCN1_OPN_AI1 : AI1—AI5，AI1與AI5短路，但不與AI0短路

OPCN1_OPN_AI0 : AI0—AI5，AI0與AI5短路，但不與AI1短路

OPCN1_OPN_AI6 : AI6

OPCN1_OPN_AI5 : AI5

OPCN1_OPN_AI4 : AI4

OPCN1_OPN_OPO : OPO

PInput[in]：設置LNOP1正向輸入通道

OPCN1_OPP_AI11 : AI11

OPCN1_OPP_AI10 : AI10

OPCN1_OPP_AI7 : AI7

OPCN1_OPP_AI3 : AI3

Chop [in]：設置LNOP1輸入處理器模式

OPCN1_OPM_REVERSE : 輸入的反向偏移量

OPCN1_OPM_FORWARD : 輸入的正向偏移量

OPCN1_OPM_CHOPPER128 : Chooper，Frequency as ADC_CK/128

OPCN1_OPM_CHOPPER64 : Chooper，Frequency as ADC_CK/64

- 包含標頭檔

Driver/LNOP1.h

- 函數返回值

無

- 函數用法

/* 設置LNOP1的輸入端為AI3—AI4，且為輸入正向偏移量模式 */

```
PWR_Open( PWRCN_VDDAX_2V4, PWRCN_ENACM_ENABLE, 0xFF ); //啓動VDDA/ACM
```

```
LNOP1_Open( OPCN1_OPN_AI4, OPCN1_OPP_AI3, OPCN1_OPM_REVERSE );
```

10.3.2. LNOP1_Enable

- 函數

```
LNOP1_Enable();
```

- 函數功能

開啓低雜訊運算放大器(LNOP1)功能，設置暫存器OPCN1[7]=1。

● 輸入參數

無

● 包含標頭檔

Driver/LNOP1.h

● 函數返回值

無

● 函數用法

```
/* 開啓低雜訊運算放大器(LNOP1) */
```

```
LNOP1_Enable();
```

10.3.3. LNOP1_Disable

● 函數

```
LNOP1_Disable();
```

● 函數功能

關閉低雜訊運算放大器(LNOP1)功能，設置暫存器OPCN1[7]=0。

● 輸入參數

無

● 包含標頭檔

Driver/LNOP1.h

● 函數返回值

無

● 函數用法

```
/* 關閉低雜訊運算放大器(LNOP1)功能 */
```

```
LNOP1_Disable();
```

10.3.4. LNOP1_OPMMode

● 函數

```
LNOP1_OPMMode(ModeSel);
```

● 函數功能

設置低雜訊運算放大器 (LNOP1) 輸入處理器模式，設置暫存器OPCN1[5:4]。

● 輸入參數

ModeSel [in] : 設置LNOP1輸入處理器模式

OPCN1_OPM_REVERSE : 輸入的反向偏移量

OPCN1_OPM_FORWARD : 輸入的正向偏移量

OPCN1_OPM_CHOPPER128 : Chooper, Frequency as ADC_CK/128

OPCN1_OPM_CHOPPER64 : Chooper, Frequency as ADC_CK/64

● 包含標頭檔

Driver/LNOP1.h

- 函數返回值

無

- 函數用法

/* 設置LNOP1的輸入處理為輸入正向偏移量 */

LNOP1_OPMode(OPCN1_OPM_FORWARD);

10.3.5. LNOP1_OPChanIn

- 函數

LNOP1_OPChanIn(OPISel);

- 函數功能

設置低雜訊運算放大器（LNOP1）輸入端短路功能，設置暫存器AINET1[0]。

- 輸入參數

OPISel [in] : 設置LNOP1的輸入端短路功能

AINET1_OPIS_ENABLE : 輸入端短路

AINET1_OPIS_DISABLE : 輸入端不短路

- 包含標頭檔

Driver/LNOP1.h , Driver/ADC.h

- 函數返回值

無

- 函數用法

/* 設置LNOP1輸入端短路 */

LNOP1_OPChanIn(AINET1_OPIS_ENABLE);

10.3.6. LNOP1_OPInputP

- 函數

LNOP1_OPInputP(OPPSel);

- 函數功能

設置低雜訊運算放大器（LNOP1）正向輸入端通道，設置暫存器OPCN1[4:3]。

- 輸入參數

OPPSel [in] : 設置LNOP1正向輸入端通道

OPCN1_OPP_AI11 : AI11

OPCN1_OPP_AI10 : AI10

OPCN1_OPP_AI7 : AI7

OPCN1_OPP_AI3 : AI3

- 包含標頭檔

Driver/LNOP1.h

- 函數返回值

無

- 函數用法

```
/* 設置LNOP1的正向輸入通道為AI3 */  
LNOP1_OPinPutP( OPCN1_OPP_AI3 );
```

10.3.7. LNOP1_OPinPutN

- 函數

```
LNOP1_OPinPutN(OPNSel);
```

- 函數功能

設置低雜訊運算放大器（LNOP1）的負向輸入端通道，設置暫存器OPCN1[2:0]。

- 輸入參數

OPNSel [in]：設置LNOP1的負向輸入端通道

OPCN1_OPN_AI1 : AI1—AI5，AI1與AI5短路，但不與AI0短路

OPCN1_OPN_AI0 : AI0—AI5，AI0與AI5短路，但不與AI1短路

OPCN1_OPN_AI6 : AI6

OPCN1_OPN_AI5 : AI5

OPCN1_OPN_AI4 : AI4

OPCN1_OPN_OPO : OPO

- 包含標頭檔

```
Driver/LNOP1.h
```

- 函數返回值

無

- 函數用法

```
/* 設置運算放大器 的負向輸入端通道為AI4 */
```

```
LNOP1_OPinPutN( OPCN1_OPN_AI4 );
```

10.3.8. LNOP2_Open

- 函數

```
void LNOP2_Open(unsigned char Ninput, unsigned char PInput, unsigned char Chop);
```

- 函數功能

開啓低雜訊運算放大器(LNOP2)功能，設置LNOP2輸入端網路及chooper功能，設置暫存器OPCN2[7:0]。

- 輸入參數

Ninput [in]：設置LNOP2負向輸入通道

OPCN2_OPN2_AI1 : AI1

OPCN2_OPN2_AI0 : AI0

OPCN2_OPN2_AI4 : AI4

OPCN2_OPN2_OPO2 : OPO2

Pinput [in]：設置LNOP2正向輸入通道

OPCN2_OPP2_AI2 : AI2

OPCN2_OPP2_AI10 : AI10

OPCN2_OPP2_AI9 : AI9

OPCN2_OPP2_AI3 : AI3

Chop [in] : 設置LNOP2輸入處理器模式

OPCN2_OPM2_REVERSE : 輸入的反向偏移量

OPCN2_OPM2_FORWARD : 輸入的正向偏移量

OPCN2_OPM2_CHOPPER128 : Chooper , Frequency as ADC_CK/128

OPCN2_OPM2_CHOPPER64 : Chooper , Frequency as ADC_CK/64

● 包含標頭檔

Driver/LNOP2.h

● 函數返回值

無

● 函數用法

/* 設置LNOP2的輸入端為AI3—AI4，且為輸入正向偏移量模式 */

PWR_Open(PWRCN_VDDAX_2V4, PWRCN_ENACM_ENABLE, 0xFF); //啓動VDDA/ACM

LNOP2_Open(OPCN2_OPN2_AI4, OPCN2_OPP2_AI3, OPCN2_OPM2_REVERSE);

10.3.9. LNOP2_Enable

● 函數

LNOP2_Enable();

● 函數功能

開啓低雜訊運算放大器(LNOP2)功能，設置暫存器OPCN2[7]=1。

● 輸入參數

無

● 包含標頭檔

Driver/LNOP2.h

● 函數返回值

無

● 函數用法

/* 開啓低雜訊運算放大器(LNOP2) */

LNOP2_Enable();

10.3.10. LNOP2_Disable

● 函數

LNOP2_Disable();

● 函數功能

關閉低雜訊運算放大器(LNOP2)功能，設置暫存器OPCN2[7]=0。

● 輸入參數

無

● 包含標頭檔

Driver/LNOP2.h

- 函數返回值

無

- 函數用法

/* 關閉低雜訊運算放大器(LNOP2)功能 */

LNOP2_Disable();

10.3.11. LNOP2_OPMODE

- 函數

LNOP2_OPMODE(ModeSel);

- 函數功能

設置低雜訊運算放大器（LNOP2）輸入處理器模式，設置暫存器OPCN2[5:4]。

- 輸入參數

ModeSel [in] : 設置LNOP2輸入處理器模式

OPCN2_OPM2_REVERSE : 輸入的反向偏移量

OPCN2_OPM2_FORWARD : 輸入的正向偏移量

OPCN2_OPM2_CHOPPER128 : Chooper, Frequency as ADC_CK/128

OPCN2_OPM2_CHOPPER64 : Chooper, Frequency as ADC_CK/64

- 包含標頭檔

Driver/LNOP2.h

- 函數返回值

無

- 函數用法

/* 設置LNOP2的輸入處理為輸入正向偏移量 */

LNOP2_OPMODE(OPCN2_OPM2_FORWARD);

10.3.12. LNOP2_OPChanIn

- 函數

LNOP2_OPChanIn(OPISel);

- 函數功能

設置低雜訊運算放大器（LNOP2）輸入端短路功能，設置暫存器OPCN2[4]。

- 輸入參數

OPISel[in] : 設置LNOP2的輸入端短路功能

OPCN2_OPIS2_ENABLE : 輸入端短路

OPCN2_OPIS2_DISABLE : 輸入端不短路

- 包含標頭檔

Driver/LNOP2.h

- 函數返回值

無

- 函數用法

```
/* 設置LNOP2輸入端短路 */  
LNOP2_OPChanIn( OPCN2_OPIS2_ENABLE );
```

10.3.13. LNOP2_OPIInputP

- 函數

```
LNOP2_OPIInputP(OPPSel);
```

- 函數功能

設置低雜訊運算放大器（LNOP2）正向輸入端通道，設置暫存器OPCN2[3:2]。

- 輸入參數

OPPSel [in]：設置LNOP2正向輸入端通道

OPCN2_OPP2_AI2 : AI2

OPCN2_OPP2_AI10 : AI10

OPCN2_OPP2_AI9 : AI9

OPCN2_OPP2_AI3 : AI3

- 包含標頭檔

Driver/LNOP2.h

- 函數返回值

無

- 函數用法

```
/* 設置LNOP2的正向輸入通道為AI3 */
```

```
LNOP2_OPIInputP( OPCN2_OPP2_AI3 );
```

10.3.14. LNOP2_OPIInputN

- 函數

```
LNOP2_OPIInputN(OPNSel);
```

- 函數功能

設置低雜訊運算放大器（LNOP2）的負向輸入端通道，設置暫存器OPCN2[1:0]。

- 輸入參數

OPNSel [in] 設置LNOP2的負向輸入端通道

OPCN2_OPN2_AI1 : AI1

OPCN2_OPN2_AI0 : AI0

OPCN2_OPN2_AI4 : AI4

OPCN2_OPN2_OPO2 : OPO2

- 包含標頭檔

Driver/LNOP1.h

- 函數返回值

無

- 函數用法

```
/* 設置運算放大器 的負向輸入端通道為AI4 */
```

LNOP2_OPIInputN(OPCN2_OPN2_AI4);

11. 電源管理 PMU

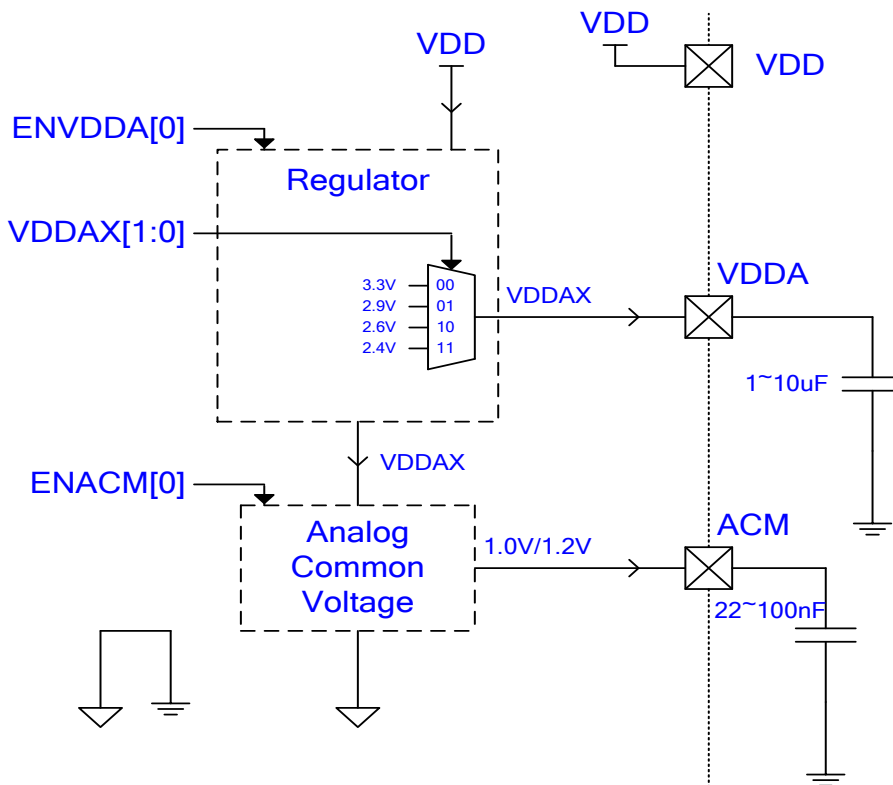
11.1. 函數簡介

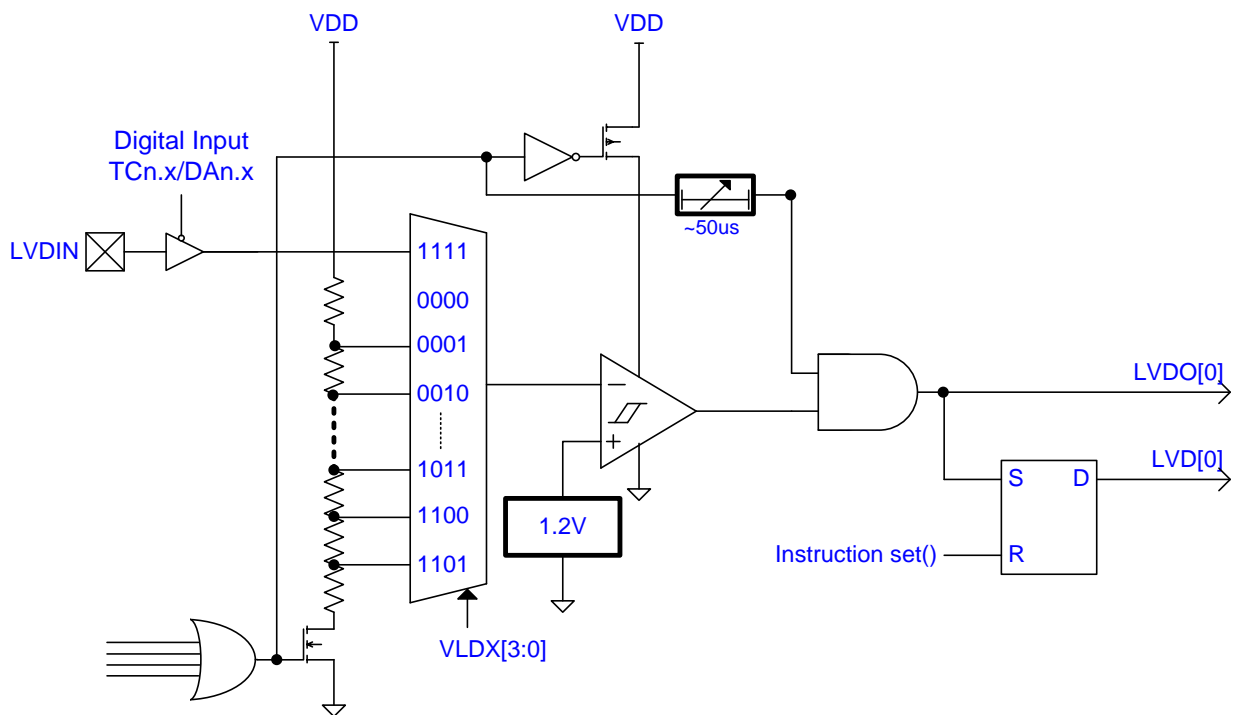
該部分函數描述電源管理系統的控制及低電壓檢測功能，包含：

- VDDA 電壓的控制
- ACM 電壓的控制
- 低電壓檢測功能控制
- 包含標頭檔 PWR.h/LVD.h

序號	函數名稱	功能描述
01	PWR_Open	使能內部ACM和VDDA，及設置VDDA輸出電壓值
02	PWR_ACMEnable	啓動內部模擬地ACM電壓
03	PWR_ACMDisable	關閉內部類比地ACM電壓
04	PWR_VDDAEnable	啓動內部LDO產生VDDA電壓輸出
05	PWR_VDDADisable	關閉內部LDO產生VDDA電壓功能
06	PWR_VDDASel	設置內部VDDA輸出電壓值
07	LVD_Open	使能低電壓檢測功能及設置低電壓檢測電壓點
08	LVD_FGClr	清除低電壓發生記錄旗標LVDFG
09	LVD_IsFlag	讀取低電壓檢測發生記錄旗標LVDFG
10	LVD_GetStatus	讀取低電壓檢測反應旗標LVD
11	LVD_GetLVDON	讀取低電壓檢測穩定旗標LVDON

11.2. PowerManage & LVD 模組方框圖





11.3. 函數說明

11.3.1. PWR_Open

- 函數

void PWR_Open(unsigned char VDDASel, unsigned char ACMEn, unsigned char DelayCnt);

- 函數功能

使能IC內部模擬地ACM及VDDA，及設置VDDA輸出電壓值，設置暫存器PWRCN[7:4]。

- 輸入參數

VDDASel [in] : 設置VDDA電壓

PWRCN_VDDAX_2V4 : VDDA輸出2.4V

PWRCN_VDDAX_2V6 : VDDA輸出2.6V

PWRCN_VDDAX_2V9 : VDDA輸出2.9V

PWRCN_VDDAX_3V3 : VDDA輸出3.3V

ACMEn [in] : 設置ACM電壓

PWRCN_ENACM_ENABLE : 啓動ACM電壓

PWRCN_ENACM_DISABLE : 關閉ACM電壓

DelayCnt [in] 設置VDDA電壓穩定時間，輸入範圍0x00~0xFF

- 包含標頭檔

Driver/PWR.h

- 函數返回值

無

- 函數用法

```
/* 設置VDDA =2.6V. 並啓動ACM電壓，VDDA穩定時間為0xFF */  
PWR_Open( PWRCN_VDDAX_2V6, PWRCN_ENACM_ENABLE, 0xFF );
```

11.3.2. PWR_ACMEnable

- 函數

```
PWR_ACMEnable();
```

- 函數功能

啓動內部模擬地ACM電壓，設置暫存器PWRCN[4]=1。

- 輸入參數

無

- 包含標頭檔

Driver/PWR.h

- 函數返回值

無

- 函數用法

```
/* 啓動ACM電壓 */
```

```
PWR_ACMEnable();
```

11.3.3. PWR_ACMDisable

- 函數

```
PWR_ACMDisable();
```

- 函數功能

關閉內部模擬地ACM電壓，設置暫存器PWRCN[4]=0。

- 輸入參數

無

- 包含標頭檔

Driver/PWR.h

- 函數返回值

無

- 函數用法

```
/* 關閉ACM電壓 */
```

```
PWR_ACMDisable();
```

11.3.4. PWR_VDDAEnable

- 函數

```
PWR_VDDAEnable();
```

- 函數功能

啓動內部VDDA電壓輸出，設置暫存器PWRCN[7]=1。

- 輸入參數

無

- 包含標頭檔

Driver/PWR.h

- 函數返回值

無

- 函數用法

/* 啓動VDDA電壓 */

PWR_VDDAEnable();

Delay(0xFF);

11.3.5. PWR_VDDADisable

- 函數

PWR_VDDADisable();

- 函數功能

關閉內部VDDA電壓功能，設置暫存器PWRCN[7]=0。

- 輸入參數

無

- 包含標頭檔

Driver/PWR.h

- 函數返回值

無

- 函數用法

/* 關閉VDDA電壓輸出 */

PWR_VDDADisable();

11.3.6. PWR_VDDASel

- 函數

PWR_VDDASel(VDDAVSel);

- 函數功能

設置內部VDDA輸出電壓值，設置暫存器PWRCN[6 :5]=0。

- 輸入參數

VDDASel [in] : 設置VDDA電壓

PWRCN_VDDAX_2V4 : VDDA輸出2.4V

PWRCN_VDDAX_2V6 : VDDA輸出2.6V

PWRCN_VDDAX_2V9 : VDDA輸出2.9V

PWRCN_VDDAX_3V3 : VDDA輸出3.3V

- 包含標頭檔

Driver/PWR.h

- 函數返回值

無

- 函數用法

```
/* 啟動VDDA輸出2.4V */
```

```
PWR_VDDASel( VDDAVSel );
```

```
PWE_VDDAEnable();
```

```
Delay( 0xFF );
```

11.3.7. LVD_Open

- 函數

```
LVD_Open(LVDSel);
```

- 函數功能

設置低電壓檢測電壓點，設置暫存器LVDCN[3 :0]。

- 輸入參數

LVDSel [in] : 設置電壓檢測點

LVDCN_VLDX_LVDIN : 待檢測電壓由PT1.2輸入，檢測電壓點位元1.2V

LVDCN_VLDX_3V31 : 檢測電壓點為3.31V

LVDCN_VLDX_3V21 : 檢測電壓點為3.21V

LVDCN_VLDX_3V11 : 檢測電壓點為3.11V

LVDCN_VLDX_3V01 : 檢測電壓點為3.01V

LVDCN_VLDX_2V91 : 檢測電壓點為2.91V

LVDCN_VLDX_2V81 : 檢測電壓點為2.81V

LVDCN_VLDX_2V71 : 檢測電壓點為2.71V

LVDCN_VLDX_2V61 : 檢測電壓點為2.61V

LVDCN_VLDX_2V51 : 檢測電壓點為2.51V

LVDCN_VLDX_2V41 : 檢測電壓點為2.41V

LVDCN_VLDX_2V31 : 檢測電壓點為2.31V

LVDCN_VLDX_2V21 : 檢測電壓點為2.21V

LVDCN_VLDX_2V11 : 檢測電壓點為2.11V

LVDCN_VLDX_2V01 : 檢測電壓點為2.01V

LVDCN_VLDX_LVDOFF : 關閉低電壓檢測功能

- 包含標頭檔

```
Driver/LVD.h
```

- 函數返回值

無

- 函數用法

```
/* 設置LVD檢測電壓點為2.61V */
```

```
LVD_Open( LVDCN_VLDX_2V61 );
```

11.3.8. LVD_FGClr

- 函數

LVD_FGClr();

- 函數功能

清除低電壓發生記錄旗標LVDFG，設置暫存器LVDCN[6]=0。

- 輸入參數

無

- 包含標頭檔

Driver/LVD.h

- 函數返回值

無

- 函數用法

/* 清除低電壓發生記錄旗標 */

LVD_FGClr();

11.3.9. LVD_IsFlag

- 函數

LVD_IsFlag();

- 函數功能

讀取低電壓檢測發生記錄旗標LVDFG，讀取暫存器LVDCN[6]。

- 輸入參數

無

- 包含標頭檔

Driver/LVD.h

- 函數返回值

0x00：電壓檢測未發生過

0x40：電壓檢測已發生過

- 函數用法

/* 讀取低電壓檢測發生記錄旗標 */

unsigned char flag;

flag = LVD_IsFlag();

11.3.10. LVD_GetStatus

- 函數

LVD_GetStatus();

- 函數功能

讀取低電壓檢測反應旗標LVD，讀取暫存器LVD[5]。

- 輸入參數

無

- 包含標頭檔

Driver/LVD.h

- 函數返回值

0x00 : 未低電壓

0x20 : 低電壓

- 函數用法

/* 讀取低電壓反應旗標 */

unsigned char flag ;

flag = LVD_GetStatus();

11.3.11. LVD_GetLVDON

- 函數

LVD_GetLVDON();

- 函數功能

讀取低電壓檢測穩定旗標LVDON，讀取暫存器LVD[4]。

- 輸入參數

無

- 包含標頭檔

Driver/LVD.h

- 函數返回值

0x00 : 未穩定

0x10 : 穩定

- 函數用法

/* 讀取低電壓檢測穩定旗標LVDON */

unsigned char flag ;

flag = LVD_GetLVDON();

12. 捕捉/比較器(CCP)

12.1. 函數功能簡介

該部分函數介紹捕捉/比較器 (CCP) 功能的設置

--CCP 功能的啟動與關閉

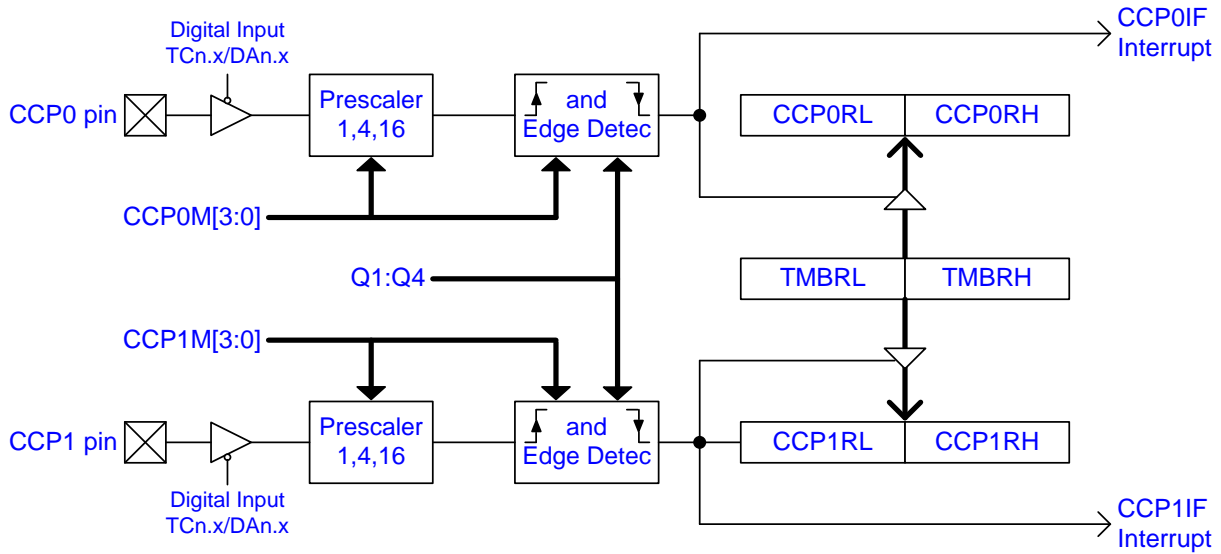
--CCP 工作模式及輸入埠的設置

--CCP 中斷功能控制

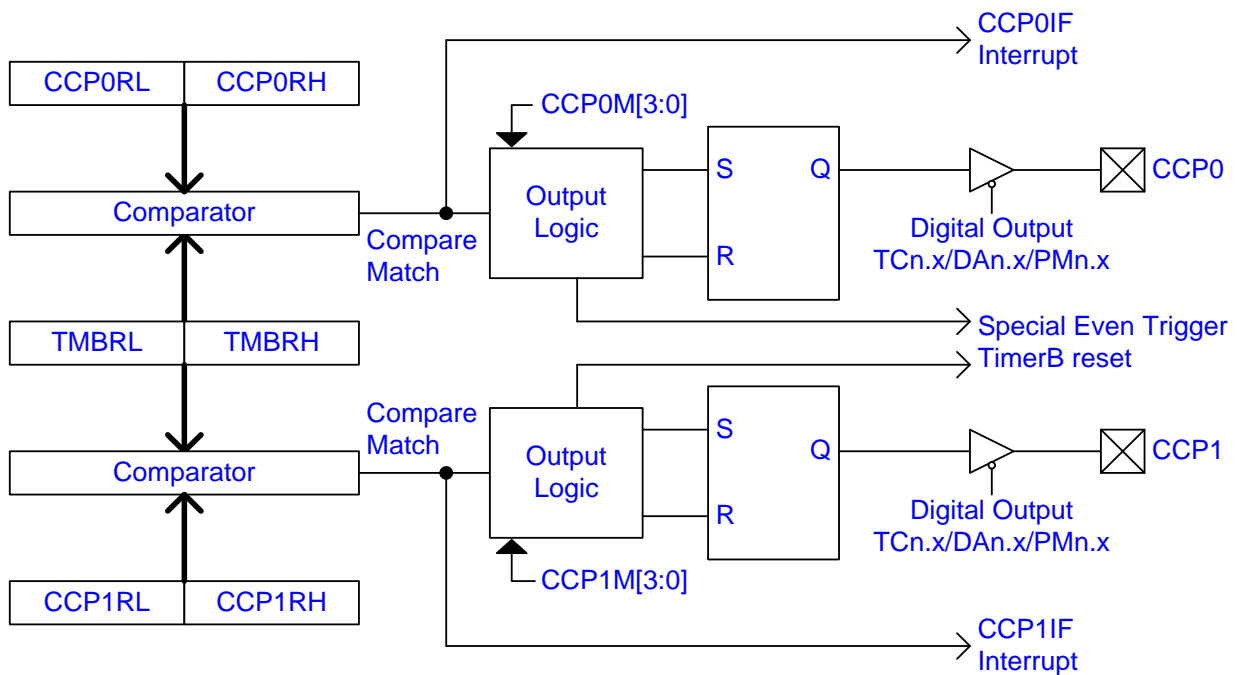
--包含 CCP.h

序號	函數名稱	功能描述
01	CCP_Open	設置捕捉/比較器的工作模式
02	CCP_SetData0	設置比較暫存器CCP0R計數值
03	CCP_SetData1	設置比較暫存器CCP1R計數值
04	CCP_GetData0	讀取捕捉器的計數器CCP0R的值
05	CCP_GetData1	讀取捕捉器的計數器CCP1R的值
06	CCP_INT0_Enable	使能捕捉/比較器CCP0中斷功能
07	CCP_INT0_Disable	關閉捕捉/比較器CCP0中斷功能
08	CCP_INT0_IsFlag	讀取捕捉/比較器CCP0的中斷請求旗標
09	CCP_INT0_ClearFlag	清除捕捉/比較器CCP0的中斷請求旗標
10	CCP_INT1_Enable	使能捕捉/比較器CCP1中斷功能
11	CCP_INT1_Disable	關閉捕捉/比較器CCP1中斷功能
12	CCP_INT1_IsFlag	讀取捕捉/比較器CCP1的中斷請求旗標
13	CCP_INT1_ClearFlag	清除捕捉/比較器CCP1的中斷請求旗標
14	CCP_CCP1Mode	設置捕捉/比較器CCP1工作模式
15	CCP_CCP0Mode	設置捕捉/比較器CCP0工作模式

12.2. 捕捉/比較器模組方框圖



(a) 捕捉器功能方框圖



(b) 比較器功能方框圖

12.3. 函數說明

12.3.1. CCP_Open

● 函數

```
void CCP_Open(unsigned char ccp0m, unsigned char ccp1m);
```

● 函數功能

設置捕捉/比較器的工作模式，設置暫存器CCPCN。

● 輸入參數

ccp0m [in] : DAC 正向參考輸入端選擇

CCPCN_CCP0M_CLRTMB : 比較模式，事件成立CCP0IF置1並清零TMB計數值

CCPCN_CCP0M_NOCCP : 比較模式，事件成立CCP0IF置1，不送信號至CCP0引腳

CCPCN_CCP0M_LOWCCP : 比較模式，CCP0引腳初始為High，事件成立CCP0IF=1，CCP0為Low

CCPCN_CCP0M_HICCP : 比較模式，CCP0引腳初始為Low，事件成立CCP0IF=1，CCP0為High

CCPCN_CCP0M_RISE16 : 捕捉模式，每16個上升沿捕捉一次，事件成立CCP0IF=1

CCPCN_CCP0M_RISE4 : 捕捉模式，每4個上升沿捕捉一次，事件成立CCP0IF=1

CCPCN_CCP0M_RISE1 : 捕捉模式，每1個上升沿捕捉一次，事件成立CCP0IF=1

CCPCN_CCP0M_FALL1 : 捕捉模式，每1個下降沿捕捉一次，事件成立CCP0IF=1

CCPCN_CCP0M_REVCCP : 比較模式，事件成立CCP0IF置1，CCP0引腳輸出電位反相

CCPCN_CCP0M_CLOSE : 關閉捕捉/比較器功能

ccp1m [in] : DAC 負向參考輸入端選擇

CCPCN_CCP1M_CLRTMB : 比較模式，事件成立CCP1IF置1並清零TMB計數值

CCPCN_CCP1M_NOCCP : 比較模式，事件成立CCP1IF置1，不送信號至CCP1引腳

CCPCN_CCP1M_LOWCCP : 比較模式，CCP1引腳初始為High，事件成立CCP1IF=1，CCP1為Low

CCPCN_CCP1M_HICCP : 比較模式，CCP1引腳初始為Low，事件成立CCP1IF=1，CCP1為High

CCPCN_CCP1M_RISE16 : 捕捉模式，每16個上升沿捕捉一次，事件成立CCP1IF=1

CCPCN_CCP1M_RISE4 : 捕捉模式，每4個上升沿捕捉一次，事件成立CCP1IF=1

CCPCN_CCP1M_RISE1 : 捕捉模式，每1個上升沿捕捉一次，事件成立CCP1IF=1

CCPCN_CCP1M_FALL1 : 捕捉模式，每1個下降沿捕捉一次，事件成立CCP1IF=1

CCPCN_CCP1M_REVCCP : 比較模式，事件成立CCP1IF置1，CCP1引腳輸出電位反相

CCPCN_CCP1M_CLOSE : 關閉捕捉/比較器功能

● 包含標頭檔

Driver/CCP.h

● 函數返回值

無

● 函數用法

/* 設置CCP0為捕捉模式，且16個上升沿捕捉，CCP1為捕捉模式，且16個上升沿捕捉 */

```
CCP_Open( CCPCN_CCP0M_RISE16, CCPCN_CCP1M_RISE16 );
```

12.3.2. CCP_SetData0

- 函數

```
void CCP_SetData0(unsigned int CCPdata);
```

- 函數功能

設置比較暫存器CCP0R計數值，設置暫存器CCP0RH/CCP0RL。

- 輸入參數

CCPdata [in]：設置比較暫存器CCP0R的值，範圍0x0~0xFFFF

- 包含標頭檔

Driver/CCP.h

- 函數返回值

無

- 函數用法

```
/* 設置比較暫存器值CCP0R為0X7F7F */
```

```
CCP_SetData0( 0x7F7F );
```

12.3.3. CCP_SetData1

- 函數

```
void CCP_SetData1(unsigned int CCPdata);
```

- 函數功能

設置比較暫存器CCP1R計數值，設置暫存器CCP1RH/CCP1RL。

- 輸入參數

CCPdata [in]：設置比較暫存器CCP1R的值，範圍0x0~0xFFFF

- 包含標頭檔

Driver/CCP.h

- 函數返回值

無

- 函數用法

```
/* 設置比較暫存器值CCP1R為0X7F7F */
```

```
CCP_SetData1( 0x7F7F );
```

12.3.4. CCP_GetData0

- 函數

```
int CCP_GetData0(void);
```

- 函數功能

讀取捕捉器的計數器CCP0R的值，讀取暫存器CCP0RH/CCP0RL。

- 輸入參數

無

- 包含標頭檔

Driver/CCP.h

- 函數返回值

回饋CCP0R:CCP0RL的值

● 函數用法

```
/* 讀取捕捉器的計數值CCP0R/CCP0RL */
```

```
int data;
```

```
data = CCP_GetData0();
```

12.3.5. CCP_GetData1

● 函數

```
int CCP_GetData1(void);
```

● 函數功能

讀取捕捉器的計數器CCP1R的值，讀取暫存器CCP1RH/CCP1RL。

● 輸入參數

無

● 包含標頭檔

Driver/CCP.h

● 函數返回值

回饋CCP1R:CCP1RL的值

● 函數用法

```
/* 讀取捕捉器的計數值CCP1R/CCP1RL */
```

```
int data;
```

```
data = CCP_GetData1();
```

12.3.6. CCP_INT0_Enable

● 函數

```
CCP_INT0_Enable();
```

● 函數功能

使能捕捉/比較器CCP0中斷功能，設置暫存器INTE2[0]=1。

● 輸入參數

無

● 包含標頭檔

Driver/CCP.h

● 函數返回值

無

● 函數用法

```
/* 使能捕捉/比較器CCP0中斷功能 */
```

```
CCP_INT0_Enable();
```

12.3.7. CCP_INT0_Disable

● 函數

CCP_INT0_Disable();

● 函數功能

關閉捕捉/比較器CCP0中斷功能，設置暫存器INTE2[0]=0。

● 輸入參數

無

● 包含標頭檔

Driver/CCP.h

● 函數返回值

無

● 函數用法

```
/* 關閉捕捉/比較器CCP0中斷功能 */  
CCP_INT0_Disable();
```

12.3.8. CCP_INT0_IsFlag

● 函數

CCP_INT0_IsFlag();

● 函數功能

讀取捕捉/比較器CCP0的中斷請求旗標，讀取暫存器INTF2[0]。

● 輸入參數

無

● 包含標頭檔

Driver/CCP.h

● 函數返回值

0x00：捕捉/比較器CCP0未產生中斷請求
0x01：捕捉/比較器CCP0產生中斷請求

● 函數用法

```
/* 讀取捕捉/比較器CCP0的中斷請求旗標 */  
unsigned char flag;  
flag = CCP_INT0_IsFlag();
```

12.3.9. CCP_INT0_ClearFlag

● 函數

CCP_INT0_ClearFlag();

● 函數功能

清除捕捉/比較器CCP0的中斷請求旗標，設置暫存器INTF2[0]=0。

● 輸入參數

無

● 包含標頭檔

Driver/CCP.h

- 函數返回值

無

- 函數用法

/* 清除捕捉/比較器CCP0的中斷請求旗標 */

CCP_INT0_ClearFlag();

12.3.10. CCP_INT1_Enable

- 函數

CCP_INT1_Enable();

- 函數功能

使能捕捉/比較器CCP1中斷功能，設置暫存器INTE2[1]=1。

- 輸入參數

無

- 包含標頭檔

Driver/CCP.h

- 函數返回值

無

- 函數用法

/* 使能捕捉/比較器CCP1中斷功能 */

CCP_INT1_Enable();

12.3.11. CCP_INT1_Disable

- 函數

CCP_INT1_Disable();

- 函數功能

關閉捕捉/比較器CCP1中斷功能，設置暫存器INTE2[1]=0。

- 輸入參數

無

- 包含標頭檔

Driver/CCP.h

- 函數返回值

無

- 函數用法

/* 關閉捕捉/比較器CCP1中斷功能 */

CCP_INT1_Disable();

12.3.12. CCP_INT1_IsFlag

- 函數

CCP_INT1_IsFlag();

- 函數功能

讀取捕捉/比較器CCP1的中斷請求旗標，讀取暫存器INTF2[1]。

- 輸入參數

無

- 包含標頭檔

Driver/CCP.h

- 函數返回值

0x00：捕捉/比較器CCP1未產生中斷請求

0x02：捕捉/比較器CCP1產生中斷請求

- 函數用法

```
/* 讀取捕捉/比較器CCP1的中斷請求旗標 */
```

```
unsigned char flag;
```

```
flag = CCP_INT1_IsFlag();
```

12.3.13. CCP_INT1_IsFlag

- 函數

```
CCP_INT1_ClearFlag();
```

- 函數功能

清除捕捉/比較器CCP1的中斷請求旗標，設置暫存器INTF2[1]=0。

- 輸入參數

無

- 包含標頭檔

Driver/CCP.h

- 函數返回值

無

- 函數用法

```
/* 清除捕捉/比較器CCP1的中斷請求旗標 */
```

```
CCP_INT1_ClearFlag();
```

12.3.14. CCP_CCP1Mode

- 函數

```
CCP_CCP1Mode(CCP1Sel);
```

- 函數功能

設置捕捉/比較器CCP1工作模式，設置暫存器CCPCN[7:4]。

- 輸入參數

CCP1Sel [in]：DAC 負向參考輸入端選擇。

CCPCN_CCP1M_CLRTMB：比較模式，事件成立CCP1IF置1並清零TMB計數值

CCPCN_CCP1M_NOCCP：比較模式，事件成立CCP1IF置1，不送信號至CCP1引腳

CCPCN_CCP1M_LOWCCP：比較模式，CCP1引腳初始為High，事件成立CCP1IF=1，CCP1為Low

CCPCN_CCP1M_HICCP : 比較模式，CCP1引腳初始為Low，事件成立CCP1IF=1，CCP1為High
CCPCN_CCP1M_RISE16 : 捕捉模式，每16個上升沿捕捉一次，事件成立CCP1IF=1
CCPCN_CCP1M_RISE4 : 捕捉模式，每4個上升沿捕捉一次，事件成立CCP1IF=1
CCPCN_CCP1M_RISE1 : 捕捉模式，每1個上升沿捕捉一次，事件成立CCP1IF=1
CCPCN_CCP1M_FALL1 : 捕捉模式，每1個下降沿捕捉一次，事件成立CCP1IF=1
CCPCN_CCP1M_REVCCP : 比較模式，事件成立CCP1IF置1，CCP1引腳輸出電位反相
CCPCN_CCP1M_CLOSE : 關閉捕捉/比較器功能

● 包含標頭檔

Driver/CCP.h

● 函數返回值

無

● 函數用法

/* 設置捕捉/比較器CCP1工作模式為16個上升捕捉 */

CCP_CCP1Mode(CCPCN_CCP1M_RISE16);

12.3.15. CCP_CCP0Mode

● 函數

CCP_CCP0Mode(CCP0Sel);

● 函數功能

設置捕捉/比較器CCP0工作模式，設置暫存器CCPCN[3:0]。

● 輸入參數

CCP0Sel [in] : DAC 負向參考輸入端選擇

CCPCN_CCP0M_CLRTMB : 比較模式，事件成立CCP0IF置1並清零TMB計數值
CCPCN_CCP0M_NOCCP : 比較模式，事件成立CCP0IF置1，不送信號至CCP0引腳
CCPCN_CCP0M_LOWCCP : 比較模式，CCP0引腳初始為High，事件成立CCP0IF=1，CCP0為Low
CCPCN_CCP0M_HICCP : 比較模式，CCP0引腳初始為Low，事件成立CCP0IF=1，CCP0為High
CCPCN_CCP0M_RISE16 : 捕捉模式，每16個上升沿捕捉一次，事件成立CCP0IF=1
CCPCN_CCP0M_RISE4 : 捕捉模式，每4個上升沿捕捉一次，事件成立CCP0IF=1
CCPCN_CCP0M_RISE1 : 捕捉模式，每1個上升沿捕捉一次，事件成立CCP0IF=1
CCPCN_CCP0M_FALL1 : 捕捉模式，每1個下降沿捕捉一次，事件成立CCP0IF=1
CCPCN_CCP0M_REVCCP : 比較模式，事件成立CCP0IF置1，CCP0引腳輸出電位反相
CCPCN_CCP0M_CLOSE : 關閉捕捉/比較器功能

● 包含標頭檔

Driver/CCP.h

● 函數返回值

無

● 函數用法

/* 設置捕捉/比較器CCP0工作模式為16個上升捕捉 */

CCP_CCP0Mode (CCPCN_CCP0M_RISE16);

13. LCD 顯示驅動器

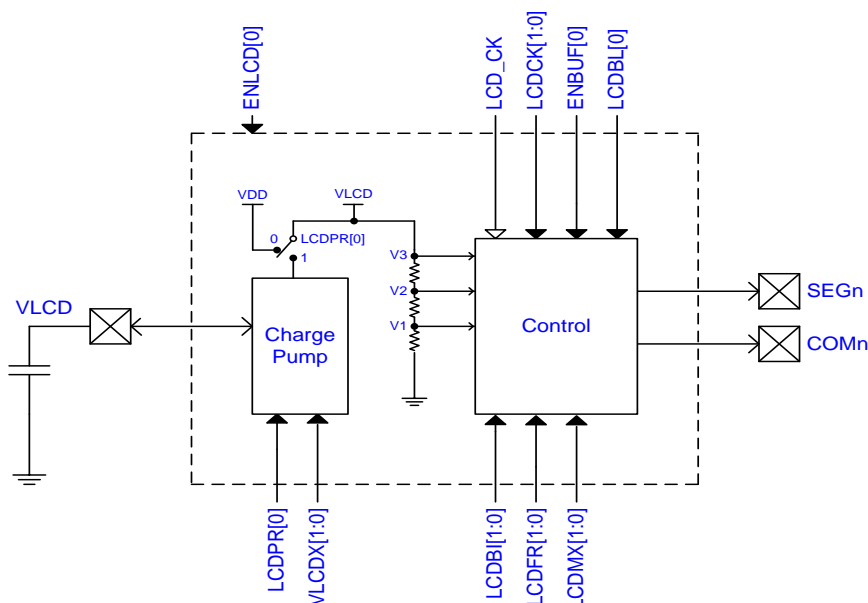
13.1. 函數簡介

該部分函數描述 LCD 驅動器相關設置

- LCD 驅動器的時鐘頻率設置
- LCD 驅動器的偏置電壓的設置及 duty 設置
- LCD 驅動器顯示功能設置
- LCD 驅動器顯示資料的寫入及輸出緩衝器設置
- 包含 LCD.h

序號	函數名稱	功能描述
01	LCD_Open	使能LCD功能；設置LCD工作頻率，設置LCD偏置電壓，設置LCD驅動電壓，設置LCD波形驅動模式及使能LCD輸出緩衝器
02	LCD_Disable	關閉LCD功能
03	LCD_Enable	使能LCD功能
04	LCD_OutBufferEnable	使能LCD輸出緩衝器
05	LCD_OutBufferDisable	關閉LCD輸出緩衝器
06	LCD_DisplayOn	設置LCD顯示全亮
07	LCD_DisplayOff	設置LCD顯示全滅
08	LCD_CLKSel	設置LCD工作頻率
09	LCD_ChargePumpConfig	LCD的驅動電壓VLCD電壓源設置
10	LCD_ChargePumpSelect	VLCD輸出電壓點設置
11	LCD_BiasInput	LCD波形偏壓設置
12	LCD_DutyMode	LCD驅動波形設置
13	LCD_WriteData	向LCD顯示暫存器寫入顯示值
14	LCD_ReadData	讀取LCD顯示暫存器的顯示值

13.2. LCD 驅動器功能方框圖



13.3. 函數說明

13.3.1. LCD_Open

● 函數

```
void LCD_Open(unsigned char lcdpr, unsigned char ckdiv, unsigned char vlcdx,  
              unsigned char lcdmx, unsigned char lcdbi);
```

● 函數功能

使能LCD功能，設置LCD工作頻率，設置LCD偏置電壓，設置LCD驅動電壓，設置LCD波形驅動模式及使能LCD輸出緩衝器，設置暫存器MCKCN3[7:5]/LCDCN1/LCDCN2。

● 輸入參數

lcdpr [in] : LCD驅動電壓VLCD電壓源設置

LCDCN1_LCDPR_INTERNAL : VLCD電壓源由內部產生

LCDCN1_LCDPR_EXTERNAL : VLCD電壓源由外部供應

vlcdx [in] : 設置內部產生的VLCD電壓值

LCDCN1_VLCDX_2V55 : VLCD=2.55V

LCDCN1_VLCDX_2V8 : VLCD=2.80V

LCDCN1_VLCDX_3V05 : VLCD=3.05V

LCDCN1_VLCDX_3V3 : VLCD=3.30V

ckdiv [in] : LCD工作頻率源設置

MCKCN3_LCDS_PERACKDIV128 : LCDS_CK=PERA_CK/128

MCKCN3_LCDS_PERACKDIV64 : LCDS_CK=PERA_CK/64

MCKCN3_LCDS_PERACKDIV32 : LCDS_CK=PERA_CK/32

MCKCN3_LCDS_PERACKDIV16 : LCDS_CK=PERA_CK/16

MCKCN3_LCDS_PERACKDIV8 : LCDS_CK=PERA_CK/8

MCKCN3_LCDS_PERACKDIV4 : LCDS_CK=PERA_CK/4

MCKCN3_LCDS_PERACKDIV2 : LCDS_CK=PERA_CK/2

MCKCN3_LCDS_PERACKDIV1 : LCDS_CK=PERA_CK/1

lcdmx [in] : LCD驅動波形設置

LCDCN2_LCDMX_Static : 固定狀態 (COM0)

LCDCN2_LCDMX_duty2 : 1/2 duty, (COM0, COM1), COM3=SEG1, COM2=SEG0

LCDCN2_LCDMX_duty3 : 1/3 duty, (COM0, COM1, COM2), COM3=SEG1

LCDCN2_LCDMX_duty4 : 1/4 duty, (COM0, COM1, COM2, COM3)

lcdbi [in] : LCD波形偏壓設置

LCDCN1_LCDBI_Used : 未使用

LCDCN1_LCDBI_BIAS : 1/3偏壓

LCDCN1_LCDBI_Reserved : 保留

LCDCN1_LCDBI_Static : 靜態操作

● 包含標頭檔

Drviver/LCD.h, Drviver/CLK.h

- 函數返回值

無

- 函數用法

/* 設置LCD工作頻率為PERA_CK/4,VLCD=3.05V, 1/3BIAS,1/4duty */

```
LCD_Open( LCDCN1_LCDPR_INTERNAL, MCKCN3_LCDS_PERACKDIV4, LCDCN1_VLCDX_3V05,  
          LCDCN2_LCDMX_duty4, LCDCN1_LCDBI_BIAS );
```

13.3.2. LCD_Disable

- 函數

```
LCD_Disable();
```

- 函數功能

關閉LCD功能，設置暫存器LCDCN1[7]=0。

- 輸入參數

無

- 包含標頭檔

Drviver/LCD.h

- 函數返回值

無

- 函數用法

/* 關閉LCD功能 */

```
LCD_Disable();
```

13.3.3. LCD_Enable

- 函數

```
LCD_Enable();
```

- 函數功能

使能LCD功能，設置暫存器LCDCN1[7]=1。

- 輸入參數

無

- 包含標頭檔

Drviver/LCD.h

- 函數返回值

無

- 函數用法

/* 使能LCD功能 */

```
LCD_Enable();
```

13.3.4. LCD_OutBufferEnable

- 函數

```
LCD_OutBufferEnable();
```

- 函數功能

使能LCD輸出緩衝器，設置暫存器LCDCN1[3]=1。

- 輸入參數

無

- 包含標頭檔

Drviver/LCD.h

- 函數返回值

無

- 函數用法

```
/* 使能LCD輸出緩衝器 */
```

```
LCD_OutBufferEnable();
```

13.3.5. LCD_OutBufferDisable

- 函數

```
LCD_OutBufferDisable();
```

- 函數功能

關閉LCD輸出緩衝器，設置暫存器LCDCN1[3]=0。

- 輸入參數

無

- 包含標頭檔

Drviver/LCD.h

- 函數返回值

無

- 函數用法

```
/* 關閉LCD輸出緩衝器 */
```

```
LCD_OutBufferDisable();
```

13.3.6. LCD_DisplayOn

- 函數

```
LCD_DisplayOn();
```

- 函數功能

設置LCD顯示全亮，設置暫存器LCDCN2[7]=0。

- 輸入參數

無

- 包含標頭檔

Drviver/LCD.h

- 函數返回值

無

- 函數用法

```
/* 設置LCD顯示全亮 */
```

```
LCD_DisplayOn();
```

13.3.7. LCD_DisplayOff

- 函數

```
LCD_DisplayOff();
```

- 函數功能

設置LCD顯示全滅，設置暫存器LCDCN2[7]=1。

- 輸入參數

無

- 包含標頭檔

Drviver/LCD.h

- 函數返回值

無

- 函數用法

```
/* 設置LCD顯示全滅 */
```

```
LCD_DisplayOff();
```

13.3.8. LCD_CLKSel

- 函數

```
LCD_CLKSel(ClkSel);
```

- 函數功能

設置LCD工作頻率，設置暫存器MCKCN3[7:5]。

- 輸入參數

ClkSel [in] : LCD工作頻率設置

MCKCN3_LCDS_PERACKDIV128 : LCDS_CK=PERA_CK/128

MCKCN3_LCDS_PERACKDIV64 : LCDS_CK=PERA_CK/64

MCKCN3_LCDS_PERACKDIV32 : LCDS_CK=PERA_CK/32

MCKCN3_LCDS_PERACKDIV16 : LCDS_CK=PERA_CK/16

MCKCN3_LCDS_PERACKDIV8 : LCDS_CK=PERA_CK/8

MCKCN3_LCDS_PERACKDIV4 : LCDS_CK=PERA_CK/4

MCKCN3_LCDS_PERACKDIV2 : LCDS_CK=PERA_CK/2

MCKCN3_LCDS_PERACKDIV1 : LCDS_CK=PERA_CK/1

- 包含標頭檔

Drviver/LCD.h

- 函數返回值

無

- 函數用法


```
/* 設置LCD工作頻率為PERA_CK/4 */  
LCD_CLKSel( MCKCN3_LCDS_PERACKDIV4 );
```

13.3.9. LCD_ChargePumpConfig

- 函數

```
LCD_ChargePumpConfig(PRSel);
```

- 函數功能

LCD的驅動電壓VLCD電壓源設置，設置暫存器LCDCN1[6]。

- 輸入參數

PRSel [in] : LCD驅動電壓VLCD電壓源設置

LCDCN1_LCDPR_INTERNAL : VLCD電壓源由內部產生

LCDCN1_LCDPR_EXTERNAL : VLCD電壓源由外部供應

- 包含標頭檔

Drviver/LCD.h

- 函數返回值

無

- 函數用法

```
/* 設置VLCD電壓源為內部產生 */
```

```
LCD_ChargePumpConfig( LCDCN1_LCDPR_INTERNAL );
```

13.3.10. LCD_ChargePumpSelect

- 函數

```
LCD_ChargePumpSelect(VLCDSel);
```

- 函數功能

VLCD輸出電壓點設置，設置暫存器LCDCN1[5:4]。

- 輸入參數

PRSel [in] : 設置內部產生的VLCD電壓值

LCDCN1_VLCDX_2V55 : VLCD=2.55V

LCDCN1_VLCDX_2V8 : VLCD=2.80V

LCDCN1_VLCDX_3V05 : VLCD=3.05V

LCDCN1_VLCDX_3V3 : VLCD=3.30V

- 包含標頭檔

Drviver/LCD.h

- 函數返回值

無

- 函數用法

```
/* 設置VLCD電壓值為3.05V */
```

```
LCD_ChargePumpSelect( LCDCN1_VLCDX_3V05 );
```

13.3.11. LCD_BiasInput

- 函數

LCD_BiasInput(BISel);

- 函數功能

LCD波形偏壓設置，設置暫存器LCDCN1[2:1]。

- 輸入參數

BISel [in] : LCD波形偏壓設置

LCDCN1_LCDBI_Unused : 未使用

LCDCN1_LCDBI_BIAS : 1/3偏壓

LCDCN1_LCDBI_Reserved : 保留

LCDCN1_LCDBI_Static : 靜態操作

- 包含標頭檔

Drviver/LCD.h

- 函數返回值

無

- 函數用法

/* 設置LCD波形偏壓為1/3 BIAS */

LCD_BiasInput(LCDCN1_LCDBI_BIAS);

13.3.12. LCD_DutyMode

- 函數

LCD_DutyMode(LCDMX);

- 函數功能

LCD驅動波形設置，設置暫存器LCDCN2[6:5]。

- 輸入參數

LCDMX [in] : LCD驅動波形設置

LCDCN2_LCDMX_Static : 固定狀態 (COM0)

LCDCN2_LCDMX_duty2 : 1/2 duty,(COM0 , COM1) , COM3=SEG1 , COM2=SEG0

LCDCN2_LCDMX_duty3 : 1/3 duty, (COM0 , COM1 , COM2) ,COM3=SEG1

LCDCN2_LCDMX_duty4 : 1/4 duty, (COM0 , COM1 , COM2 , COM3)

- 包含標頭檔

Drviver/LCD.h

- 函數返回值

無

- 函數用法

/* 設置LCD驅動波形為1/4 duty */

LCD_DutyMode(LCDCN2_LCDMX_duty4);

13.3.13. LCD_WriteData

- 函數

```
LCD_WriteData(uAddr,uData);
```

- 函數功能

向LCD顯示暫存器寫入顯示值，設置暫存器

LCD0/LCD1/LCD2/LCD3/LCD4/LCD5/LCD6/LCD7/LCD8/LCD9/LCD10
LCD11/LCD12/LCD13/LCD14/LCD15/LCD16/LCD17/LCD18/LCD19 ◦

- 輸入參數

uAddr [in] : 顯示暫存器位址

LCD0/LCD1/LCD2/LCD3/LCD4/LCD5/LCD6/LCD7/LCD8/LCD9/LCD10
LCD11/LCD12/LCD13/LCD14/LCD15/LCD16/LCD17/LCD18/LCD19

uData [in] 顯示內容，0x00~0xFF

- 包含標頭檔

Drviver/LCD.h

- 函數返回值

無

- 函數用法

```
/* 往LCD0寫入0XAA */
```

```
LCD_WriteData( LCD0 , 0XAA );
```

13.3.14. LCD_ReadData

- 函數

```
LCD_ReadData(uAddr);
```

- 函數功能

讀取LCD顯示暫存器的顯示值，設置暫存器

LCD0/LCD1/LCD2/LCD3/LCD4/LCD5/LCD6/LCD7/LCD8/LCD9/LCD10
LCD11/LCD12/LCD13/LCD14/LCD15/LCD16/LCD17/LCD18/LCD19 ◦

- 輸入參數

uAddr [in] : 顯示暫存器位址

LCD0/LCD1/LCD2/LCD3/LCD4/LCD5/LCD6/LCD7/LCD8/LCD9/LCD10
LCD11/LCD12/LCD13/LCD14/LCD15/LCD16/LCD17/LCD18/LCD19

- 包含標頭檔

Drviver/LCD.h

- 函數返回值

無

- 函數用法

```
/* 讀取LCD0顯示值 */
```

```
unsigned char lcd_buf;
```

```
lcd_buf = LCD_ReadData( LCD0 );
```

14. Library

14.1. Library File

HY11P Driver C Library source code 在軟體安裝目錄下的 HY11P CIDE\Driver\HY11 。

15. Revision History

Version	Page	Revision Summary	The Date Of Revision
V01	ALL	First edition	2017/11/10

16. C Library Change List

Date	舊版本 Queries List		新版本改善	
	版本	Bug List	版本	改善
2017-11-13	V01	無	V01	無