HYCON TECHNOLOGY

HY16F18 系列

C 函式程式庫手冊

目錄

1 導讀

1.1 C 函式程式庫簡介

本檔用於描述HYCON HY16F18系列C 函式程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 函式程式庫直接調用開發替換寄存器操作來有效的提高整個產品的開發效率。

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/

2 系統控制

2.1 函數簡介

該部分函數描述晶片工作系統控制，包含：

--工作模式（休眠模式（sleep）、待機模式（Idle）、等待模式（Waite mode））的控制

--晶片工作狀態標誌位元控制

序號	函數名稱	功能描述
01	SYS_SleepFlagRead	讀取休眠標誌位元
02	SYS_SleepFlagClear	清除休眠標誌位元
03	SYS_WdogFlagRead	讀取看門狗標誌位元
04	SYS_WdogFlagClear	清除看門狗標誌位元
05	SYS_ResetFlagRead	讀取重定標誌位元
06	SYS_ResetFlagClear	清除重定標誌位元
07	SYS_BOR_FlagRead	讀取低電壓重定(BOR) 標誌位元
08	SYS_BOR_FlagClear	清除低電壓重定(BOR) 標誌位元
09	SYS_EnableGIE	使能全域中斷GIE且開啟對應的中斷向量
10	SYS_DisableGIE	關閉全域中斷GIE
11	SYS_LowPower	設置IC低功耗工作模式
12	SYS_INTPriority	設置對應中斷向量的中斷優先權級別

2.2 函數說明

2.2.1 SYS_SleepFlagRead

- **函數**

  unsigned int SYS_SleepFlagRead (void);

- **函數功能**

  讀取休眠標誌位元（Sleep flag）的值;

  讀取寄存器0x40104[3]的值。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/System.h

- **函數返回值**

  0：正常工作模式

  1：晶片進入休眠模式

- **函數用法**

```
/* 讀取休眠標誌位元 */
unsigned char temp_flag;     temp_flag=SYS_SleepFlagRead();
```

2.2.2 SYS_SleepFlagClear

- **函數**

  void SYS_SleepFlagClear(void);

- **函數功能**

  清零休眠標誌位元；

  清零寄存器0x40104[3]的值。.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/System.h

- **函數返回值**

  無

- **函數用法**

```
/* 清零sleep flag. */
SYS_SleepFlagClear();    //set 0x40104[3]=0
```

2.2.3 SYS_WdogFlagRead

- **函數**

unsigned int SYS_WdogFlagRead (void);

- **函數功能**

讀取看門狗(WDT)標誌位元的值；

讀取寄存器0x40104[2]的值。

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/System.h

- **函數返回值**

0： 正常

1： 看門狗(WDT)已經觸發

- **函數用法**

```
/* 讀取看門狗(WDT)標誌位元. */
unsigned char flag;   flag=SYS_WdogFlagRead();
```

2.2.4 SYS_WdogFlagClear

- **函數**

void SYS_WdogFlagClear(void);

- **函數功能**

清零看門狗(WDT)標誌位元；

清零寄存器0x40104[2]的值。

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/System.h

- **函數返回值**

無

- **函數用法**

```
/* 清零看門狗(WDT)的標誌位元 */
SYS_WdogFlagClear();   //0x40104[2]=0
```

2.2.5 SYS_ResetFlagRead

- **函數**

unsigned int SYS_ResetFlagRead (void);

- **函數功能**

讀取重定標誌位元的值；

讀取寄存器0x40104[1]的值。

- **輸入參數**

無

- **包含標頭檔**

  Peripheral_lib/System.h

- **函數返回值**

  0： 正常

  1 : Reset PIN 外部復位已經觸發

- **函數用法**

```
/* 讀取重置標誌位元. */
unsigned char flag;   flag=SYS_ResetFlagRead();
```

2.2.6 SYS_ResetFlagClear

- **函數**

```
void SYS_ResetFlagClear(void);
```

- **函數功能**

  清零重定標誌位元的值；

  清零寄存器0x40104[1]的值。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/System.h

- **函數返回值**

  無

- **函數用法**

```
/* 清零重定標誌位元 */
SYS_ResetFlagClear();   //0x40104[1]=0
```

2.2.7 SYS_BOR_FlagRead

- **函數**

```
unsigned int SYS_BOR_FlagRead (void);
```

- **函數功能**

  讀取低電壓重定(BOR)標誌位元的值；

  讀取寄存器0x40104[0]的值。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/System.h

- **函數返回值**

  0： 正常

  1： 低電壓重定(BOR) 功能已觸發

- **函數用法**

```
/* 讀取低電壓重定(BOR)標誌位元 . */
unsigned char flag;   flag=SYS_BOR_FlagRead();
```

2.2.8 SYS_BOR_FlagClear

- **函數**

  void SYS_BOR_FlagClear(void);

- **函數功能**

  清零低電壓重定(BOR)標誌位元；

  清零寄存器0x40104[0]的值.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/System.h

- **函數返回值**

  無

- **函數用法**

```
/* 清零低電壓重定(BOR)  標誌位元 */
SYS_BOR_FlagClear();   //0x40104[0]=0
```

2.2.9 SYS_EnableGIE

- **函數**

  unsigned int SYS_EnableGIE (unsigned int uPriority,unsigned short intvector);

- **函數功能**

  使能全域中斷(GIE),使能對應中斷向量並設置相應優先權級別的中斷可以進行中斷嵌套回應，優先順序別高的先回應，中斷向量優先權級別可以通過函數SYS_INTPriority()設置.

- **輸入參數**

  uPriority [in]： 設定允許開啟中斷向量的優先權級別，設置範圍是0~4

  0: 不允許任何優先權級別的中斷向量回應

  1: 只允許優先權級別被SYS_INTPriority()函數設定為最高級別的中斷向量回應.

  2: 只允許優先權級別被SYS_INTPriority()函數設定為最高級別、次高級別的中斷向量回應 .

  3: 只允許優先權級別被SYS_INTPriority()函數設定為最高級別、次高級別、低級別的中斷向量回應

  4: 只允許先權級別被SYS_INTPriority()函數設定為最高級別、次高級別、低級別、最低級別的中斷向量回應

  intvector[in]選擇中斷向量[HW5:HW4:HW3:HW2:HW1:HW0]；輸入範圍為0~0x3F，每一位值對應一個中斷向量使能位HW5~HW0，只有對應位為1，才能使能對應的中斷向量；

  invector=[HW5:HW4:HW3:HW2:HW1:HW0]

  使能HW0/HW3/HW5，則invectior=0x29（101001B）;

  使能所有中斷向量，則invector=0x3F（111111B）;

  不開啟任何中斷向量，則invector=0x00（000000B）

- **包含標頭檔**

Peripheral_lib/System.h

- **函數返回值**

0：設置成功

1：設置失敗

- **函數用法**

/* 使能全域中斷GIE，並允許優先權級別為0, 1, 2,3的中斷向量回應,使能中斷向量HW0~HW5 */

SYS_EnableGIE(4,0x3F);

2.2.10 SYS_DisableGIE

- **函數**

void SYS_DisableGIE (void);

- **函數功能**

關閉全域中斷使能（GIE）。

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/System.h

- **函數返回值**

無

- **函數用法**

/* 關閉全域中斷使能(GIE). */

SYS_DisableGIE();

2.2.11 SYS_LowPower

- **函數**

unsigned char SYS_LowPower(unsigned char umode)

- **函數功能**

設置並啟動低功耗工作模式；開啟低功耗模式前需要開啟任何一個中斷向量。且需要切換為低頻晶震源。

設置寄存器0x40104[4]。

- **輸入參數**

umode[in]：輸入範圍0~2

0：休眠模式（Sleep mode）

1：待機模式（Idle mode）

2：等待模式（Waite mode）

- **包含標頭檔**

Peripheral_lib/System.h

- **函數返回值**

0：設置成功

1：設置失敗

- **函數用法**

```
/* 啟動休眠工作模式*/
DrvGPIO_Open(E_PT1,0xFF,E_IO_IntEnable);   // 開啟PT1 外部中斷向量
SYS_EnableGIE(4, 0x3F);                    //開啟全域中斷控制
DrvCLOCK_SelectMCUClock(1,0);              //切換頻率源為低頻
SYS_LowPower(0);                           //進入休眠工作模式
```

2.2.12 SYS_INTPriority

- **函數**

unsigned char SYS_INTPriority(unsigned short intvector,unsigned short upriority);

- **函數功能**

設置對應中斷向量的優先權級別，優先權級別為0~3,且0為最高級別。

注意：使用前，必須關閉所有的中斷使能，才能修改中斷優先權級別。

- **輸入參數**

intvector[in] 中斷向量選擇，輸入範圍為0~5，分別對應HW0~HW5;

upriority[in]：設置開啟中斷向量的優先權級別，設置範圍是0~3

0: 優先權級別為最高級別

1: 優先權級別為次高級別

2: 優先權級別為低級級別.

3: 優先權級別為最低級別

在設置中斷優先權級別都為同一級別時，中斷回應的的先後順序為：

HW0 > HW1 > HW2 > …> HW5

- **包含標頭檔**

Peripheral_lib/System.h

- **函數返回值**

0：設置成功

1：設置失敗

- **函數用法**

```
/* 設置中斷向量0優先權級別為 1 * /
SYS_INTPriority(0,1);
```

3 晶片時鐘源 CLOCK

3.1 函數簡介

函數描述 CPU 及其他功能模組的時鐘源操作，包含：

--內部高速及低速晶振的控制

--外部高速及低速晶振的控制

--CPU 時鐘源的切換

序號	函數名稱	功能描述
01	DrvCLOCK_EnableHighOSC	開啟高頻晶震
02	DrvCLOCK_CloseEHOSC	關閉外部高頻晶震
03	DrvCLOCK_CloseIHOSC	關閉內部高頻晶震
04	DrvCLOCK_SelectIHOSC	設置內部高頻晶震HAO的頻率
05	DrvCLOCK_EnableLowOSC	開啟低頻晶震
06	DrvCLOCK_CloseELOSC	關閉外部低頻晶震
07	DrvCLOCK_SelectMCUClock	設置MCU時鐘
08	DrvCLOCK_TrimHAO	內部高頻晶震校正
09	DrvCLOCK_CalibrateHAO	按照出廠時HAO校正參數進行HAO頻率校正
10	DrvCLOCK_SelectOHS_HS	外部高頻晶振模式(HSXT)選擇
11	DrvCLOCK_EnableENHAO	開啟內部高頻震盪器
12	DrvCLOCK_SelectIHOSC_CalHAO	設置內部高頻晶震HAO的頻率並且按照晶片出廠校正值進行HAO頻率校正

3.2 內部定義常量

E_CLOCK_SOURCE

識別字	數值	功能意義
E_INTERNAL	0x0	內部
E_EXTERNAL	0x1	外部

E_TRIM_FREQUEN

識別字	數值	功能意義
TRIM_HAO2MHZ	0x0	校正HAO 2MHZ頻率
TRIM_HAO4MHZ	0x1	校正HAO 4MHZ頻率
TRIM_HAO10MHZ	0x2	校正HAO 10MHZ頻率
TRIM_HAO16MHZ	0x3	校正HAO 16MHZ頻率

3.3 函數說明

3.3.1 DrvCLOCK_EnableHighOSC

- **函數**

  unsigned int DrvCLOCK_EnableHighOSC(E_CLOCK_SOURCE uSource, unsigned int delay)

- **函數功能**

  開啟高速晶震，並選擇CPU時鐘來源為外部晶震(HSXT)或者內部晶震(HSRC);

  設定等待晶震達到穩定所需時間;

  若CPU時鐘源選擇外部晶震，則寄存器0x40300[5]=1 , 0x40300[1]=1;

  若CPU時鐘源選擇為內部晶震，則寄存器0x40300[5]=0, 0x40300[0]=1;

- **輸入參數**

  uSource[in] : CPU時鐘源選擇. 設定範圍：0~1

  0: 內部晶震

  1: 外部晶震

delay[in]： 設置等待晶震達到穩定所需時間. 設定範圍：1~0xFFFFFFFF

需要注意當前CPU的運算速度CPU_CLK；晶震達到穩定時間t=(1/CPU_CLK)*4000*delay；

函數內部已經有4000次運算速度的迴圈，參數delay是倍數設置，設置參數時需要參考當前運算速度及需要啟動的晶震頻率。

外部晶震(EXT OSC)達到穩定需要的時間t :

4MHZ/8MHZ 約30ms

內部晶震（HAO）達到穩定需要的時間t :

2MHZ 約1.0ms

4MHZ 約0.5ms

10MHZ 約0.2ms

- **包含標頭檔**

  Peripheral_lib/DrvCLOCK.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  // 開啟外部高速晶震,當前CPU_CK=10MHZ/2, 開啟外部4MHZ, 設定延時t=40ms=(1/(10MHZ/2))*4000*50

  DrvCLOCK_EnableHighOSC(E_EXTERNAL,50);

3.3.2 DrvCLOCK_CloseEHOSC

- **函數**

  void DrvCLOCK_CloseEHOSC()

- **函數功能**

關閉外部高速晶震; 注意：若被關閉的晶震當前是作為CPU時鐘源，必須先切換為有效時鐘源才能關閉該晶震。

寄存器0x40300[1]=0;

- **函數輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvCLOCK.h

- **函數返回值**

無

- **函數用法**

```
/* 開啟設定內部晶震作為CPU時鐘源, 關閉外部高速晶震 */
DrvCLOCK_EnableHighOSC(E_INTERNAL,50); //開啟內部高速晶震
DrvCLOCK_CloseEHOSC();                 //關閉外部高速晶震
```

3.3.3 DrvCLOCK_CloseIHOSC

- **函數**

void DrvCLOCK_CloseIHOSC()

- **函數功能**

關閉內部高速晶震(HAO)；但是前提必須先將CPU時鐘源切換到有效的時鐘源。

寄存器0x40300[0]=0;

- **函數輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvCLOCK.h

- **函數返回值**

無

- **函數用法**

```
/* 開啟外部高速晶震作為CPU時鐘源, 關閉內部高速晶震*/
DrvCLOCK_EnableHighOSC(E_EXTERNAL,50);  //開啟外部高速時鐘源
DrvCLOCK_CloseIHOSC(); //關閉內部高速晶震
```

3.3.4 DrvCLOCK_SelectIHOSC

- **函數**

unsigned int DrvCLOCK_SelectIHOSC(uMode)

- **函數功能**

設置內部晶震HAO的頻率模式;

設置寄存器0x40300[4:3]。

- **輸入參數**

uMode [in]：HAO頻率值，有效輸入範圍 ：0~2

0 ：2MHz, 0x40300[4:3]=00b

1 ：4MHz, 0x40300[4:3]=01b

2 ：10MHz, 0x40300[4:3]=10b

3 ：Rsv

- **包含標頭檔**

Peripheral_lib/DrvCLOCK.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/* 設置內部高速晶震(HAO)=4MHz*/

DrvCLOCK_SelectIHOSC(1);

3.3.5 DrvCLOCK_EnableLowOSC

- **函數**

unsigned int DrvCLOCK_EnableLowOSC(E_CLOCK_SOURCE uSource，unint delay)

- **函數功能**

開啟低速晶震頻率，並設置CPU時鐘源是內部晶震(LSRC)或者外部晶震(LSXT)及設置等待晶震達到穩定所需時間；

寄存器0x40300[6]=1. 0x40300[2]=1

- **輸入參數**

uSource [in]： 輸入範圍 0~1

0: 內部晶震LSRC

1: 外部晶震LSXT

Delay[in]： 等待晶震的時間設置，需要參考當前的運算速度來設置

設定範圍：0x0~0xffffffff

外部低速晶振LSXT穩定時間：32768HZ 約1.3s

內部低速晶振LSRC穩定時間: 約510us

- **包含標頭檔**

Peripheral_lib/DrvCLOCK.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/* 開啟外部低速晶振LSXT並設置穩定時間為130000*/

DrvCLOCK_EnableLowOSC(E_EXTERNAL,130000);

3.3.6 DrvCLOCK_CloseELOSC

- **函數**

  void DrvCLOCK_CloseELOSC()

- **函數功能**

  關閉外部低速晶振;但是需要先喚醒內部晶振(LPO)並切換至內部晶振(LPO)。

  寄存器0x40300[2]=0。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvCLOCK.h

- **函數返回值**

  無

- **函數用法**

  /* 先開啟內部低速晶振，再關閉外部低速晶振 */

  DrvCLOCK_EnableLowOSC(E_INTERNAL,130000); //開啟內部低速晶振

  DrvCLOCK_CloseELOSC(); //關閉外部低速晶震

3.3.7 DrvCLOCK_SelectMCUClock

- **函數**

  unsigned int DrvCLOCK_SelectMCUClock(uSource,uDiv)

- **函數功能**

  設置CPU的時鐘源為高速頻率(HS_CK)或低速頻率(LS_CK)，設置時鐘分頻數值；

  設置寄存器0x40308[0]與0x40308[1]。

- **輸入參數**

  uSource [in] : CPU時鐘源選擇

  0： 高速頻率(HS_CK)

  1： 低速頻率(LS_CK)

  uDiv [in] : CPU時鐘源分頻設置

  0 : ÷1

  1 : ÷2

- **包含標頭檔**

  Peripheral_lib/DrvCLOCK.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 設置CPU時鐘源為高速頻率(HS_CK)並且2分頻 */

  DrvCLOCK_SelectMCUClock(0,1); //設置MCU的高速頻率源為HS_CK，且設置2分頻

3.3.8 DrvCLOCK_TrimHAO

- **函數**

  unsigned int DrvCLOCK_TrimHAO(uTrim)

- **函數功能**

  校正內部晶震(HAO);

  設置寄存器0x40304[7:0]的值。

- **輸入參數**

  uTrim[in]:頻率校正值，輸入範圍 ：0~0xff

- **包含標頭檔**

  Peripheral_lib/DrvCLOCK.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /*寫入0x80在效正內部晶振控制暫存器 */

  DrvCLOCK_TrimHAO(0x80); // 設置 0x40304[7:0]=0x80

3.3.9 DrvCLOCK_CalibrateHAO

- **函數**

  void DrvCLOCK_CalibrateHAO(short int uMHZ)

- **函數功能**

  按照晶片出廠時HAO校正值來校正內部晶震(HAO);使用時注意要與選定的HAO頻率對應；

  設置寄存器0x40304[7:0]的值。

- **輸入參數**

  uMHZ [in]： 待校正值的HAO頻率模式選擇

  0：校正 2MHZ；1：校正 4MHZ； 2：校正 10MHZ；3：Rsv；

- **包含標頭檔**

  Peripheral_lib/DrvCLOCK.h

- **函數返回值**

  無

- **函數用法**

  /*校正內部晶震(HAO)4MHZ */

  DrvCLOCK_SelectIHOSC(1); //setting HAO=4MHZ;

  DrvCLOCK_CalibrateHAO(1); //calibrate 4MHZ；

3.3.10 DrvCLOCK_SelectOHS_HS

- **函數**

  unsigned int DrvCLOCK_SelectOHS_HS(unsigned int uMode)

- **函數功能**

  外部高頻晶振模式(HSXT)選擇, HSXT可選擇是大於4MHZ, 或是小於4MHZ；

  設置寄存器0x40300[7]的值。

- **輸入參數**

  uMode [in]： 外部高頻晶振(HSXT)模式選擇. 輸入範圍 ：0~1

  0：HSXT<4MHz；1：HSXT>4MHz；

- **包含標頭檔**

  Peripheral_lib/DrvCLOCK.h

- **函數返回值**

  0： 設置成功

  1： 設置失敗

- **函數用法**

```
/*選擇外部晶震(HSXT)>4MHZ */
DrvCLOCK_SelectOHS_HS(1);   //Select HSXT > 4MHZ;
```

3.3.11 DrvCLOCK_EnableENHAO

- **函數**

  void DrvCLOCK_EnableENHAO (void)

- **函數功能**

  開啟內部高速震盪器

  設置寄存器0x40300[0]=1

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvCLOCK.h

- **函數返回值**

  無

- **函數用法**

```
/*開啟內部高速震盪器 */
DrvCLOCK_EnableENHAO();   //ENHAO=1b
```

3.3.12 DrvCLOCK_SelectIHOSC_CalHAO

- **函數**

  unsigned int DrvCLOCK_SelectIHOSC_CalHAO(unsigned int uMode)

- **函數功能**

設置內部晶震HAO的頻率模式, 並且按照晶片出廠時HAO校正值來校正內部晶震(HAO)

設置寄存器0x40300[4:3]、0x40304[7:0]的值。

- **輸入參數**

uMode [in] : HAO頻率值, 輸入範圍 : 0~2

0 : 2MHz, 0x40300[4:3]=00b

1 : 4MHz, 0x40300[4:3]=01b

2 : 10MHz, 0x40300[4:3]=10b

- **包含標頭檔**

Peripheral_lib/DrvCLOCK.h

- **函數返回值**

0 : 設置成功 1 : 設置失敗

- **函數用法**

```
/*選擇HAO=4MHz, 並且校正內部晶震(HAO)4MHZ=4MHz <2%誤差 */
DrvCLOCK_SelectIHOSC_CalHAO(1);
```

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、啟動控制、中斷控制

--定時計數器 A(Timer A)的配置控制、啟動控制、定時中斷控制

--定時計數器 B(Timer B)的配置控制、啟動控制、定時控制及 PWM 模式控制

--定時計數器 C(Timer C)的配置控制及 Capture 的控制

序號	函數名稱	功能描述
01	DrvWDT_Open	開啟看門狗(WDT)
02	DrvWDT_CounterRead	讀取看門狗計數值
03	DrvWDT_ClearWDT	清零看門狗計數值
04	DrvWDT_ResetEnable	WDT 中斷工作模式選擇為 Reset Mode
05	DrvTMA_Open	開啟定時計數器(Timer A)
06	DrvTMA_Close	關閉定時計數器(Timer A)
07	DrvTMA_CounterRead	讀取定時計數器(Timer A)計數值
08	DrvTMA_ClearTMA	清零定時計數器(Timer A)計數值
09	DrvTIMER_EnableInt	開啟計時器中斷 TA/TB/TC/WDT.
10	DrvTIMER_DisableInt	關閉計時器中斷 TA/TB/TC/WDT
11	DrvTIMER_GetIntFlag	讀取插斷要求標誌位元
12	DrvTIMER_ClearIntFlag	清除插斷要求標誌位元
13	DrvTMB_Open	開啟定時計數器(Timer B)
14	DrvTMB_Clk_Source	設置定時計數器(Timer B/C)時鐘源
15	DrvTMB_Clk_Disable	關閉定時計數器(Timer B/C)時鐘源
16	DrvTMB_ClearTMB	清零定時計數器(Timer B)計數值
17	DrvTMB_CounterRead	讀取定時計數器(Timer B)計數值
18	DrvTMB_Close	關閉定時計數器(Timer B)
19	DrvPWM0_Open	開啟 PWM 功能及 PWM0 模式
20	DrvPWM1_Open	開啟 PWM 功能及 PWM1 模式
21	DrvPWM_CountCondition	設置 PWM 占空比設置
22	DrvPWM0_Close	關閉 PWM0 模式
23	DrvPWM1_Close	關閉 PWM1 模式
24	DrvCAPTURE1_Open	開啟捕捉比較器 1
25	DrvCAPTURE2_Open	開啟捕捉比較器 2

26	DrvCAPTURE1_Read	讀取捕捉比較器 1 計數值
27	DrvCAPTURE2_Read	讀取捕捉比較器 2 計數值
28	DrvCAPTURE_IPort	設置捕捉比較器信號輸入 IO 口

4.2 內部定義常量

E_WDT_MODE

識別字	設定值	功能意義
E_IRQ	0x0	IRQ mode
E_RST	0x1	RESET mode

E_WDT_PRE_SCALER

識別字	設定值	功能意義
E_PRE_SCALER_D2	0x0	WDT_CK / 2
E_PRE_SCALER_D8	0x1	WDT_CK / 8
E_PRE_SCALER_D32	0x2	WDT_CK / 32
E_PRE_SCALER_D128	0x3	WDT_CK / 128
E_PRE_SCALER_D512	0x4	WDT_CK / 512
E_PRE_SCALER_D2048	0x5	WDT_CK / 2048
E_PRE_SCALER_D8192	0x6	WDT_CK / 8192
E_PRE_SCALER_D32768	0x7	WDT_CK / 32768

E_TIMER_CHANNEL

識別字	設定值	功能意義
E_TMA	0x0	計時器 A
E_TMB	0x1	計時器 B
E_TMC	0x2	計時器 C
E_WDT	0x3	看門狗 WDT

E_TMB_MODE

識別字	設定值	功能意義
E_TMB_MODE0	0x0	16-bit 遞增計數器TBR[15:0]，步長為1；
E_TMB_MODE1	0x1	16-bit 遞增/遞減計數器TBR[15:0]，步長為1；
E_TMB_MODE2	0x2	兩個8-bit的遞增計數器TBR[15:8]/TBR[7:0]，兩個計數器獨立同時計數，步長為1。
E_TMB_MODE3	0x3	兩個8-bit遞增計數器TBR[15:8]/TBR[7:0],，計數器步長為1，計數器TBR[7:0]計數溢出後計數器TBR[15:0]才會自動加1。

E_TRIGGER_SOURCE

識別字	設定值	功能意義
E_TMB_NORMAL	0x0	總是啟用（Always Enable）連續計數方式
E_TMB_CMP_HIGH	0x1	比較器(CMP)高電位觸發

E_TMB_OP_HIGH	0x2	運放(OP)高電位觸發
E_TMB_GPIO_HIGH	0x3	Timer C的輸出CPI1 高電位觸發

E_DRVTIMER_CLOCK_SOURCE

識別字	數值	函數功能
E_HS_CK	0	TMA 時鐘源為HS_CK
E_LS_CK	1	TMA 時鐘源為 LS_CK

E_CAPTURE_SOURCE

識別字	數值	函數功能
E_TMC_CMPO	0x0	比較器輸出
E_TMC_OPOD	0x1	運算放大器數位輸出
E_TMC_LSCK	0x2	低頻時鐘源
E_TMC_TCI0	0x3	捕捉比較器1 I/O輸入
E_TMC_TCI1	0x0	捕捉比較器2 I/O輸入
E_TMC_ASTC0	0X1	捕捉比較器2 的輸入源與捕捉比較器1一致

4.3 函數說明

4.3.1 DrvWDT_Open

- **函數**

  uint32_t DrvWDT_Open (E_WDT_MODE eMode , E_WDT_PRE_SCALER eWDTpreScaler)

- **函數功能**

  使能看門狗(WDT)，設置看門狗(WDT)工作模式，設置時鐘分頻來設定計數溢出值；

  設置寄存器0x40108[2:0] / 0x40108[6] / 0x40108[4]=1。

- **輸入參數**

  eMode[in]： 看門狗工作模式選擇

  0： 定時中斷模式

  1： 重定模式

  eWDTpreScaler[in]： 看門狗時鐘源分頻設置

  0 : WDT_CK / 2

  1 : WDT_CK / 8

  2 : WDT_CK / 32

  3 : WDT_CK / 128

  4 : WDT_CK / 512

  5 : WDT_CK / 2048

  6: WDT_CK / 8192

  7: WDT_CK / 32768

- **包含標頭檔**

  Peripheral_lib/DrvTIMER.h

- **函數返回值**

  0：設置成功

  1：設置失敗

- **函數用法**

  /* 設置看門狗(WDT)為 IRQ mode 及 CLK / 32 */

  DrvWDT_Open(E_IRQ , E_PRE_SCALER_D32);

4.3.2 DrvWDT_CounterRead

- **函數**

  uint32_t DrvWDT_CounterRead (void)

- **函數功能**

  讀取看門狗(WDT)計數寄存器的值；讀取寄存器 0x40108 [30:16] 。

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvTIMER.h

- **函數返回值**

看門狗計數值.

- **函數用法**

/* 讀取看門狗(WDT)計數寄存器的值 */

unsigned int data；data=DrvWDT_CounterRead();

4.3.3 DrvWDT_ClearWDT

- **函數**

void DrvWDT_ClearWDT (void)

- **函數功能**

清零看門狗(WDT)計數寄存器值，設置寄存器0X40108[5]=1，且清零後該位自動變為0 。

寄存器0x40108[30:16]清除為0。

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvTIMER.h

- **函數返回值**

無

- **函數用法**

/* 清零看門狗 */

DrvWDT_ClearWDT();

4.3.4 DrvWDT_ResetEnable

- **函數**

void DrvWDT_ResetEnable(void)

- **函數功能**

WDT中斷工作模式選擇為Reset Mode，設置寄存器0x40108[6]=1b。

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvTIMER.h

- **函數返回值**

無

- **函數用法**

```
/* WDT中斷工作模式選擇為Reset Mode */
DrvWDT_ResetEnable();
```

4.3.5 DrvTMA_Open

- **函數**

  unsigned int DrvTMA_Open (eTMAOV, E_DRVTIMER_CLOCK_SOURCE uclk)

- **函數功能**

  使能定時計數器A(Timer A)，設置定時計數器A(Timer A)時鐘源，設定計數溢出值；

  設置寄存器0x40C00[5]=1, 0x40C00[3:0], 0x40308[3], 0x40308[2]。

- **輸入參數**

  eTMAOV [in]： 定時計數器A(Timer A) 計數溢出值設置：.

  0 : taclk/2

  1 : taclk/4

  2 : taclk/8

  3 : taclk/16

  4 : taclk/32

  5 : taclk/64

  6 : taclk/128

  7 : taclk/256

  8 : taclk/512

  9 : taclk/1024

  10 : taclk/2048

  11 : taclk/4096

  12 : taclk/8192

  13 : taclk/16384

  14 : taclk/32768

  15: taclk/65536

  uclk[in]： 定時計數器A(Timer A)時鐘源設置.

  0: HS_CK

  1: LS_CK

- **包含標頭檔**

  Peripheral_lib/DrvTIMER.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

```
/* 使能定時計數器A(Timer A)，且計數溢出值為taclk/8. */
DrvTMA_Open(2, 0);
```

4.3.6 DrvTMA_Close

- **函數**

  void DrvTMA_Close (void)

- **函數功能**

  關閉定時計數器A(Timer A);

  設置寄存器0x40C00[5]=0。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvTIMER.h

- **函數返回值**

  無

- **函數用法**

  /* 關閉定時計數器A(Timer A) */

  DrvTMA_Close();

4.3.7 DrvTMA_CounterRead

- **函數**

  unsigned int DrvTMA_CounterRead (void)

- **函數功能**

  讀取定時計數器A(Timer A)計數寄存器的值TAR;

  讀取寄存器0x40C00[15:0] 。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvTIMER.h

- **函數返回值**

  定時計數器A(Timer A)計數值.

- **函數用法**

  /*讀取定時計數器A(Timer A)計數值 */

  unsigned short tcounter; tcounter=DrvTMA_CounterRead();

4.3.8 DrvTMA_ClearTMA

- **函數**

  void DrvTMA_ClearTMA (void)

- **函數功能**

  清零定時計數器A(Timer A)計數寄存器TAR;

  設置寄存器0x40C00[4]=1，清零完成後自動置0。寄存器0x40C00[15:0]清除為0

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvTIMER.h

- **函數返回值**

  無

- **函數用法**

  /* 清零定時計數器A(Timer A)計數寄存器 */

  DrvTMA_ClearTMA();

4.3.9 DrvTIMER_EnableInt

- **函數**

  unsigned int DrvTIMER_EnableInt (E_TIMER_CHANNEL ch)

- **函數功能**

  使能WDT/Timer A/Timer B/Timer C 中斷功能；

  設置寄存器0x40004[20:16]的對應中斷使能位=1。

- **輸入參數**

  ch [in] ：中斷源設置. 輸入範圍 ：0~4

  0：定時計數器 A　　1：定時計數器B　2：定時計數器C的C0中斷

  3：定時計數器C的C1中斷　　4：看門狗

- **包含標頭檔**

  Peripheral_lib/DrvTIMER.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /*使能定時計數器A(Timer A) 中斷 */

  DrvTIMER_EnableInt(E_TMA);

4.3.10 DrvTIMER_DisableInt

- **函數**

  unsigned int DrvTIMER_DisableInt (E_TIMER_CHANNEL ch)

- **函數功能**

  關閉WDT/Timer A/Timer B/Timer C 中斷功能；

  設置寄存器0x40004[20:16]的對應模組中斷使能位元=0。

- **輸入參數**

  ch [in]：中斷源選擇. 輸入範圍 ：0~4

  0：定時計數器 A　　1：定時計數器B　2：定時計數器C的C0中斷

  3：定時計數器C的C1中斷　　4：看門狗

- **包含標頭檔**

  Peripheral_lib/DrvTIMER.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 關閉定時計數器A(Timer A) 中斷向量*/

  DrvTIMER_DisableInt(E_TMA);

4.3.11 DrvTIMER_GetIntFlag

- **函數**

  unsigned int DrvTIMER_GetIntFlag (E_TIMER_CHANNEL ch)

- **函數功能**

  讀取WDT/Timer A/Timer B/Timer C中斷要求標誌位元;

  讀取寄存器0x40004[4:0]對應模組中斷要求標誌位元。

- **輸入參數**

  ch [in]：中斷源選擇. 輸入範圍 ：0~4

  0：定時計數器 A　　1：定時計數器B　2：定時計數器C的C0中斷

  3：定時計數器C的C1中斷　　4：看門狗

- **包含標頭檔**

  Peripheral_lib/DrvTIMER.h

- **函數返回值**

  0: 無中斷要求

  1: 有中斷要求

- **函數用法**

  /*讀取定時計數器A(Timer A) 中斷要求標誌位元*/

  unsigned char flag ; flag=DrvTIMER_GetIntFlag(E_TMA);

4.3.12 DrvTIMER_ClearIntFlag

- **函數**

  unsigned int DrvTIMER_ClearIntFlag (E_TIMER_CHANNEL ch)

- **函數功能**

  清除WDT/Timer A/Timer B/Timer C 中斷要求標誌位元；

  設置寄存器0x40004[4:0]對應模組功能中斷標誌位元=0。

- **輸入參數**

ch [in]：中斷源設置. 輸入範圍 ：0~4

0：定時計數器 A　　1：定時計數器B　2：定時計數器C的C0中斷

3：定時計數器C的C1中斷　　4：看門狗

- **包含標頭檔**

Peripheral_lib/DrvTIMER.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/* 清除定時計數器A(Timer A) 中斷要求標誌位元*/

DrvTIMER_ClearIntFlag(E_TMA);

4.3.13 DrvTMB_Open

- **函數**

unsigned int DrvTMB_Open (E_TMB_MODE eTMBmode, E_TRIGGER_SOURCE eTriSource, eTMBOV)

- **函數功能**

使能定時計數器B(Timer B)，設置定時計數器B(Timer B)寄存器計數模式，設置定時計數器B(Timer B)計數觸發源，設置定時計數器B(Timer B)計數溢出值；支援比較器、捕捉、計數和定時功能；

設置寄存器0x40C0C[15:0], 0x40C04[3:0], 0x40C04[5]=1b。

- **輸入參數**

eTMBmode [in]： 代表定時計數器B(Timer B)計數模式.

0: 計數寄存器(TBR)是遞增計數模式，在每一個TBCLK的上升沿加1.當TBR >TBC0時，在下一個TBCLK的上升沿TBR變為0且TMBIF被置1，然後TBR又重新遞增計數。

1: 計數寄存器(TBR)是遞增遞減計數模式;作為遞增模式,每一個TBCLK上升沿TBR自動加1,當TBR=TBC0時,TBR 變為遞減模式,且 TBR 在每一個 TBCLK 上升沿自動減 1,當 TBR 遞減為 0 時,中斷標誌位元(TMBIF)被置 1，然後 TBR 又重新開始遞增計數.

2: 計數寄存器(TBR)分為兩個 8-bitPWM 模式，兩個獨立的遞增計數器，兩個計數器都是在 TBCLK 的上升沿自動加 1；當 TBR[15:8]=TBC0[15:8]時 TBR[15:0]=0，TBR[7:0]=TBC0[7:0]時，TBR[7:0]=0；當 TBR[15:8]=TBC0[15:8]時，在 TBCLK 的下一個上升沿時 TBR[15:8]=0，但 TMBIF 保持為 0,；在 TBR[7:0]=TBC0[7:0]時，在 TBCLK 的下一個上升沿 TBR[7：0]=0，且中斷標誌位元(TMBIF)被置 1。

3: 計數寄存器(TBR)作為步進模式。TBR分為兩個8-bit遞增計數器，在TBCLK的每個上升沿自動加1，TBR[15:8]計數上限受控於TBC0[15:8]，TBR[7:0]計數上限受控於TBC0[7:0]；當TBR[7:0]=TBC0[7:0]時，在TBCLK的下一個上升沿時TBR[7:0]=0，且TBR[15:8]自動加1，中斷標誌位元TMBIF被置1。

eTriSource [in]： 表示Timer B 計數觸發源選擇.

0: 總是啟用（Always Enable ）連續計數方式

1: 比較器(CMP)高電位觸發

2: 運算放大器(OP)數位輸出高電位觸發

3: 定時計數器 C(Timer C) 輸出高電位觸發（CPI1)

eTMAOV [in]：計數溢出值設置，設定範圍是0~0xffff

- **包含標頭檔**

Peripheral_lib/DrvTIMER.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/* 開啟定時計數器B(TMB)，設置模式1，計數值為Taclk/32，OP 高電位觸發 */

DrvTMB_Open(E_TMB_MODE0, E_TMB_OP_HIGH,4);

4.3.14 DrvTMB_Clk_Source

- **函數**

unsigned int DrvTMBC_Clk_Source (E_DRVTIMER_CLOCK_SOURCE uclk, uPerScale)

- **函數功能**

定時計數器B/C 時鐘源設置，及時鐘源分頻設置；

設置寄存器 0x40308[7:6], 0x40308[5:4]

- **輸入參數**

uclk[in]：定時計數器 B/C 時鐘源

0: HS_CK

1: LS_CK

uPerScale[in]：定時計數器B/C 時鐘分頻設置

0: ÷1

1: ÷2

2: ÷4

3: ÷8

- **包含標頭檔**

Peripheral_lib/DrvTIMER.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/* 設置定時計數器B 時鐘源為HS_CK, 分頻為 2. */

DrvTMBC_Clk_Source(0,1);

4.3.15 DrvTMB_Clk_Disable

- **函數**

viod DrvTMBC_Clk_Disable (viod)

- **函數功能**

  關閉定時計數器B/C 時鐘源;

  設置寄存器0x40308[6]=0。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvTIMER.h

- **函數返回值**

  無

- **函數用法**

  /* 關閉定時計數器B/C 時鐘源*/

  DrvTMBC_Clk_Disable();

4.3.16 DrvTMB_ClearTMB

- **函數**

  void DrvTMB_ClearTMB (void)

- **函數功能**

  清除定時計數器B(Timer B)的計數寄存器;

  設置寄存器0x40C04[4]=1,清零後該位自動置0, 寄存器0x40C08[15:0]清除為0。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvTIMER.h

- **函數返回值**

  無

- **函數用法**

  /* 清除定時計數器B(Timer B)的計數寄存器. */

  DrvTMB_ClearTMB();

4.3.17 DrvTMB_CounterRead

- **函數**

  unsigned int DrvTMB_CounterRead (void)

- **函數功能**

  讀取定時計數器B(Timer B)的計數寄存器的值；

  讀取寄存器0x40C08[15:0]。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvTIMER.h

- **函數返回值**

  Timer B 計數值

- **函數用法**

  /* 讀取定時計數器B(Timer B)的計數值 */

  unsigned short Tcounter; Tcounter=DrvTMB_CounterRead();

4.3.18 DrvTMB_Close

- **函數**

  void DrvTMB_Close (void)

- **函數功能**

  關閉定時計數器B(Timer B)的功能；設置寄存器0x40C04[5]=0。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvTIMER.h

- **函數返回值**

  無

- **函數用法**

  /*關閉定時計數器B(Timer B)*/

  DrvTMB_Close();

4.3.19 DrvPWM0_Open

- **函數**

  unsigned int DrvPWM0_Open (uPWM_Mode , uInv, uOuputPin)

- **函數功能**

  使能PWM0功能，及設置PWM0的工作模式、輸出波形反相設置與輸出IO設置；

  設置寄存器0x40C04[18:16] / 0x40C04[19] / 0x40840[4:2] / 0x40840[0]=1b 。

- **輸入參數**

  uPWM_Mode [in] : PWM 工作模式設置

  0: PWM A 1: PWM B

  2: PWM C 3: PWM D

  4 : RSV 5 : PWM F

  6 : PWM G 7 : PWM G

  uInv[in] : PWM輸出PWM波形相位控制

  0 : 輸出波形反相

  1 : 輸出波形正常

  uOuputPin[in] : PWM輸出IO 設置

  0 : Port 1.0 =PWMO1, Port 1.1 =PWMO2

  1 : Port 1.2 =PWMO1, Port 1.3 =PWMO2

2 : Port 1.4 =PWMO1, Port 1.5 =PWMO2

3 : Port 1.6 =PWMO1, Port 1.7 =PWMO2

4 : Port 2.0 =PWMO1, Port 2.1 =PWMO2

5 : Port 2.2 =PWMO1, Port 2.3 =PWMO2

6 : Port 2.4 =PWMO1, Port 2.5 =PWMO2

7 : Port 2.6 =PWMO1, Port 2.7 =PWMO2

- **包含標頭檔**

Peripheral_lib/DrvTIMER.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

```
/*使能PWM0且工作模式是PWMA ，反相輸出，PT1.0輸出 */
DrvPWM0_Open(0, 0, 0);
```

4.3.20 DrvPWM1_Open

- **函數**

unsigned int DrvPWM1_Open (uPWM_Mode , uInv, uOuputPin)

- **函數功能**

使能PWM1功能，及設置PWM1的工作模式、輸出波形反相設置及輸出IO設置；

設置寄存器0x40C04[23:20]/ 0x40840[4:2] / 0x40840[1]=1b 。

- **輸入參數**

uPWM_Mode [in] : PWM 工作模式設置

0: PWM A 1: PWM B

2: PWM C 3: PWM D

4 : RSV 5 : PWM F

6 : PWM G 7 : PWM G

uInv[in] : PWM 輸出波形相位控制

0： 輸出波形反相

1： 輸出波形正常

uOuputPin[in] PWM輸出IO口控制

0 : Port 1.0 =PWMO1, Port 1.1 =PWMO2

1 : Port 1.2 =PWMO1, Port 1.3 =PWMO2

2 : Port 1.4 =PWMO1, Port 1.5 =PWMO2

3 : Port 1.6 =PWMO1, Port 1.7 =PWMO2

4 : Port 2.0 =PWMO1, Port 2.1 =PWMO2

5 : Port 2.2 =PWMO1, Port 2.3 =PWMO2

6 : Port 2.4 =PWMO1, Port 2.5 =PWMO2

7 : Port 2.6 =PWMO1, Port 2.7 =PWMO2

- **包含標頭檔**

Peripheral_lib/DrvTIMER.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

```
/*使能PWM1，且設置工作模式是PWMA ，反相輸出，PT1.1輸出 */
DrvPWM1_Open(0, 0, 0);
```

4.3.21 DrvPWM_CountCondition

- **函數**

void DrvPWM_CountCondition (uTBC2 , uTBC1)

- **函數功能**

PWM0/PWM1占空比設置，寫入計數寄存器(TBC2, TBC1);

設置寄存器0x40C10[15:0](TBC1) / 0x40C10[31:16](TBC2)

- **輸入參數**

uTBC1 [in] ：PWM0占空比設置

TBC1 設定範圍0~0xFFFF

uTBC2 [in] ：PWM1占空比設置

TBC2 設定範圍0~0xFFFF

- **包含標頭檔**

Peripheral_lib/DrvTIMER.h

- **函數返回值**

無

- **函數用法**

```
/* 設置TBC1, TBC2 值為0x4000 */
DrvPWM_CountCondition(0x4000,0x4000);
```

4.3.22 DrvPWM0_Close

- **函數**

void DrvPWM0_Close (void)

- **函數功能**

關閉PWM0輸出;

設置寄存器0x40840[0]=0.

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvTIMER.h

- **函數返回值**

  無

- **函數用法**

  /*PWM0輸出關閉 */

  DrvPWM0_Close();

4.3.23 DrvPWM1_Close

- **函數**

  void DrvPWM1_Close (void)

- **函數功能**

  關閉PWM1輸出；

  設置寄存器0x40840[1]=0

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvTIMER.h

- **函數返回值**

  無

- **函數用法**

  /*PWM1輸出關閉*/

  DrvPWM1_Close();

4.3.24 DrvCAPTURE1_Open

- **函數**

  unsigned int DrvCapture1_Open (CAPTURE_SOURCE uChannel , uDivider, uEdge)

- **函數功能**

  開啟信號捕捉比較器Capture1，捕捉比較器輸入信號源設置、信號除頻器設置及捕捉信號觸發邊沿設置.

  設置寄存器0x40C14[21:20] / 0x40C14[19:16] / 0x40C14[1] / 0x40C14[0]=1。

- **輸入參數**

  uChannel [in] ： 捕捉器Capture1輸入信號源設置. 輸入範圍 :0~3

  0： 比較器輸出(CMPO)

  1： 運算放大器輸出(OPOD)

  2： 低速時鐘源(LS_CK)

  3： IO口輸入(TCI1)

  uDivider [in] ： 輸入信號除頻設置. 輸入範圍 :0~15

  0: ÷1　　8: ÷256

  1: ÷2　　9: ÷512

  2: ÷4　　10: ÷1024

3: ÷8 11: ÷2048

4: ÷16 12: ÷4096

5: ÷32 13: ÷8192

6: ÷64 14: ÷16384

7: ÷128 15: ÷32768

uEdge [in]：捕捉信號觸發邊沿設置

0： 上升沿觸發

1： 下降沿觸發

- **包含標頭檔**

Peripheral_lib/DrvTIMER.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/* 使能捕捉器capture1，輸入信號選擇TCI1，信號除頻為2048，上升沿觸發模式 */

DrvCapture1_Open(3, 11, 0);

4.3.25 DrvCAPTURE2_Open

- **函數**

unsigned int DrvCapture2_Open (CAPTURE_SOURCE uChannel, uEdge)

- **函數功能**

使能信號捕捉比較器Capture2，設置捕捉信號輸入源及捕捉信號觸發邊沿.

設置寄存器0x40C14[22] / 0x40C14[2] / 0x40C14[0]=1。

- **輸入參數**

uChannel [in] : Capture 2 捕捉信號輸入源設置

0: 信號輸入源 TCI2 來自 GPIO

1: 與 Capture1 一樣的信號輸入源

uEdge [in]： 捕捉信號觸發邊沿設置

0: 上升沿觸發

1: 下降沿觸發

- **包含標頭檔**

Peripheral_lib/DrvTIMER.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/* 使能捕捉器capture2，輸入信號選擇與Capture1 一樣的信號輸入源 ,上升沿觸發模式 */

DrvCapture2_Open(1, 0);

4.3.26 DrvCAPTURE1_Read

- **函數**

  unsigned int DrvCapture1_Read (void)

- **函數功能**

  讀取捕捉比較器Capture1的計數值;

  讀取寄存器0x40C18[15:0]值

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvTIMER.h

- **函數返回值**

  Capture1 計數值TCR0(0~0xffff)

- **函數用法**

  /* 讀取捕捉比較器Capture1 計數值*/

  unsigned short tcounter; tcounter=DrvCapture1_Read();

4.3.27 DrvCAPTURE2_Read

- **函數**

  unsigned int DrvCapture2_Read (void)

- **函數功能**

  讀取捕捉比較器Capture2 計數值;

  讀取寄存器0x40C18[31:16]值

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvTIMER.h

- **函數返回值**

  Capture2 計數值TCR1(0~0xffff)

- **函數用法**

  /* 讀取捕捉比較器Capture2 計數值 */

  unsigned short tcounter; tcounter=DrvCapture2_Read();

4.3.28 DrvCAPTURE_IPort

- **函數**

  unsigned int DrvCapture_Iport (uInputPin)

- **函數功能**

  設置信號捕捉比較器的信號輸入IO口;

設置寄存器0x40840[7:5]。

- **輸入參數**

  uInputPin[in] :

  0 : Port 1.0 =TCI1, Port 1.1 =TCI2

  1 : Port 1.2 =TCI1, Port 1.3 =TCI2

  2 : Port 1.4 =TCI1, Port 1.5 =TCI2

  3 : Port 1.6 =TCI1, Port 1.7 =TCI2

  4 : Port 2.0 =TCI1, Port 2.1 =TCI2

  5 : Port 2.2 =TCI1, Port 2.3 =TCI2

  6 : Port 2.4 =TCI1, Port 2.5 =TCI2

  7 : Port 2.6 =TCI1, Port 2.7 =TCI2

- **包含標頭檔**

  Peripheral_lib/DrvTIMER.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

```
/*  捕捉比較器Capture輸入IO設置Port 1.6=TCI1, Port1.7=TCI2 */
DrvCapture_Iport(3);
```

5 晶片 IO 口 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 口的工作模式控制

--GPIO 口的上拉控制

--GPIO 口的外部中斷功能控制

--GPIO 口狀態及採樣頻率控制

序號	函數名稱	功能描述
01	DrvGPIO_Open	設置GPIO port 操作模式
02	DrvGPIO_SetBit	設置GPIO pin 為1
03	DrvGPIO_ClrBit	設置GPIO pin 為0.
04	DrvGPIO_GetBit	獲取GPIO pin的值
05	DrvGPIO_SetPortBits	設置GPIO port 值
06	DrvGPIO_ClrPortBits	清除輸出GPIO port 的值
07	DrvGPIO_GetPortBits	獲取輸入GPIO port 的值
08	DrvGPIO_IntTrigger	設置GPIO 口中斷觸發源模式
09	DrvGPIO_ClkGenerator	設置GPIO 採樣時鐘源
10	DrvGPIO_ClearIntFlag	清除外部中斷標誌位元
11	DrvGPIO_GetIntFlag	讀取外部中斷標誌位元
12	DrvGPIO_Close	關閉GPIO 任何引腳的工作模式
13	DrvGPIO_EnableAnalogPin	關閉IO數位功能，開啟類比功能
14	DrvGPIO_PT1_EnableINPUT	使能PT1口對應引腳輸入模式功能
15	DrvGPIO_PT1_DisableINPUT	關閉PT1口對應引腳輸入模式功能
16	DrvGPIO_PT1_EnablePullHigh	使能PT1口對應引腳上拉電阻功能
17	DrvGPIO_PT1_DisablePullHigh	關閉PT1口對應引腳上拉電阻功能
18	DrvGPIO_PT1_EnableOUTPUT	使能PT1口對應引腳輸出模式功能
19	DrvGPIO_PT1_DisableOUTPUT	關閉PT1口對應引腳輸出模式功能
20	DrvGPIO_PT1_EnableINT	使能PT1口對應引腳外部中斷功能
21	DrvGPIO_PT1_DisableINT	關閉PT1口對應引腳外部中斷功能
22	DrvGPIO_PT1_IntTriggerPorts	設置PT1外部中斷觸發邊沿
23	DrvGPIO_PT1_IntTriggerBit	設置PT1口的某一位IO pin的外部中斷觸發沿
24	DrvGPIO_PT1_GetIntFlag	清除PT1外部插斷要求標誌位元
25	DrvGPIO_PT1_ClearIntFlag	讀取PT1外部插斷要求標誌位元
26	DrvGPIO_PT1_GetPortBits	讀取PT1口輸入狀態值
27	DrvGPIO_PT1_SetPortBits	設置PT1口對應引腳輸出1

28	DrvGPIO_PT1_ClrPortBits	設置PT1口對應引腳輸出0
29	DrvGPIO_PT2_EnableINPUT	使能PT2口對應引腳輸入模式功能
30	DrvGPIO_PT2_DisableINPUT	關閉PT2口對應引腳輸入模式功能
31	DrvGPIO_PT2_EnablePullHigh	使能PT2口對應引腳上拉電阻功能
32	DrvGPIO_PT2_DisablePullHigh	關閉PT2口對應引腳上拉電阻功能
33	DrvGPIO_PT2_EnableOUTPUT	使能PT2口對應引腳輸出模式功能
34	DrvGPIO_PT2_DisableOUTPUT	關閉PT2口對應引腳輸出模式功能
35	DrvGPIO_PT2_EnableINT	使能PT2口對應引腳外部中斷功能
36	DrvGPIO_PT2_DisableINT	關閉PT2口對應引腳外部中斷功能
37	DrvGPIO_PT2_IntTriggerPorts	設置PT2外部中斷觸發邊沿
38	DrvGPIO_PT2_IntTriggerBit	設置PT2口的某一位IO pin的外部中斷觸發沿
39	DrvGPIO_PT2_GetIntFlag	清除PT2外部插斷要求標誌位元
40	DrvGPIO_PT2_ClearIntFlag	讀取PT2外部插斷要求標誌位元
41	DrvGPIO_PT2_GetPortBits	讀取PT2口輸入狀態值
42	DrvGPIO_PT2_SetPortBits	設置PT2口對應引腳輸出1
43	DrvGPIO_PT2_ClrPortBits	設置PT2口對應引腳輸出0
44	DrvGPIO_PT3_EnableINPUT	使能PT3口對應引腳輸入模式功能
45	DrvGPIO_PT3_DisableINPUT	關閉PT3口對應引腳輸入模式功能
46	DrvGPIO_PT3_EnablePullHigh	使能PT3口對應引腳上拉電阻功能
47	DrvGPIO_PT3_DisablePullHigh	關閉PT3口對應引腳上拉電阻功能
48	DrvGPIO_PT3_EnableOUTPUT	使能PT3口對應引腳輸出模式功能
49	DrvGPIO_PT3_DisableOUTPUT	關閉PT3口對應引腳輸出模式功能
50	DrvGPIO_PT3_GetPortBits	讀取PT3口輸入狀態值
51	DrvGPIO_PT3_SetPortBits	設置PT3口對應引腳輸出1
52	DrvGPIO_PT3_ClrPortBits	設置PT3口對應引腳輸出0

5.2 內部定義常量

E_DRVGPIO_PORT

識別字	數值	功能意義
E_PT0	0	定義PT0
E_PT1	1	定義PT1
E_PT2	2	定義PT2
E_PT3	3	定義PT3
E_PT4	4	定義PT4

E_DRVGPIO_IO

識別字	數值	功能意義
E_IO_INPIT	0	設置GPIO作為輸入模式
E_IO_OUTPUT	1	設置GPIO作為輸出模式
E_IO_PullHigh	2	使能上拉電阻功能
E_IO_ IntEnable	3	使能外部中斷功能

E_DRVGPIO_IntTriMethod

識別字	數值	功能意義
E_DisableGPIOInt	0	關閉GPIO中斷功能
E_P_Edge	1	上升沿觸發
E_N_Edge	2	下降沿觸發
E_Chang_Level	3	電平變化觸發
E_LLTri	4	低電平觸發
E_LHTri	5	高電平觸發
E_LLTri	6	低低電平觸發
E_LHTri	7	高電平觸發

E_DRVGPIO_CLOCK_SOURCE

識別字	數值	功能意義
E_HS_CK	0	設置IO 採樣時鐘源為HS_CK
E_LS_CK	1	設置IO 採樣時鐘源為LS_CK

5.3 函數說明

5.3.1 DrvGPIO_Open

- **函數**

  int32_t DrvGPIO_Open (E_DRVGPIO_PORT port, int32_t i32Bit, E_DRVGPIO_IO mode)

- **函數功能**

  設置GPIO PT1~PT3任何一位IO引腳的工作模式，可選工作模式有輸入/輸出/外部中斷/電阻上拉。

  設置PT1寄存器0x40800[23:16] / 0x40800[7:0] / 0x40804[23:16] / 0x40010[23:16]

  PT2寄存器 0x40810[23:16] / 0x40810[7:0] / 0x40814[23:16] / 0x40014[23:16]

  PT3寄存器 0x40820[23:16] / 0x40820[7:0] / 0x40824[23:16] / 0x40010[23:16]

- **輸入參數**

  port [in]： 代表GPIO port. 它的值可以是1~4.

  1：PT1 2：PT2

  3：PT3 4：Reserved.

  i32Bit [in] ： 代表 GPIO 任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳工作模式，為0時不做設置；

  設置值範圍是 0~255.

  mode [in] ： 代表GPIO 每一位IO 口的工作模式,. 設置值範圍 ：0~3

  0：輸入模式 1：輸出模式

  2：內部上拉 3：使能外部中斷

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 設置PT1.0 作為輸出模式及 PT1.1 作為輸入模式*/

  DrvGPIO_Open(E_PT1, 0x01, E_IO_OUTPUT); //PT1.0打開輸出模式

  DrvGPIO_Open(E_PT1, 0x02, E_IO_INPUT);//PT1.1打開輸入模式

5.3.2 DrvGPIO_SetBit

- **函數**

  unsigned int DrvGPIO_SetBit (E_DRVGPIO_PORT uport, unsigned int i32Bit)

- **函數功能**

  設置PT1~PT3對應的IO 口輸出1.

  設置GPIO的輸出狀態寄存器0x40804[7:0]/0x40814[7:0]/0x40824[7:0]

- **輸入參數**

  uport [in]： 代表GPIO port. 設定值範圍是1~4

  1：PT1　2：PT2

  3：PT3　4：Rsv.

  i32Bit [in]： 代表GPIO的每一位IO 口，設定值範圍是0~7.

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

```
/* 設置PT1.0 作為輸出模式*/
DrvGPIO_Open(E_PT1, 1, E_IO_OUTPUT);
/* 設定PT1.0輸出1 */
DrvGPIO_SetBit(E_PT1, 0);
```

5.3.3 DrvGPIO_ClrBit

- **函數**

  unsigned int DrvGPIO_ClrBit (E_DRVGPIO_PORT uport, unsigned int i32Bit)

- **函數功能**

  設置PT1~PT3對應任何一位的IO口輸出狀態為 0.

  清零GPIO的輸出狀態寄存器0x40804[7:0] / 0x40814[7:0] / 0x40824[7:0]

- **輸入參數**

  uport [in]： 代表GPIO port. 它的設置值範圍是1~4.

  1：PT1　2：PT2

  3：PT3　4：Rsv.

  i32Bit [in]： 代表GPIO的每一位IO 口，設定值範圍是0~7.

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  0：設置成功

  0xff000000：設置失敗

- **函數用法**

```
/* 設定PT1.0輸出0 */
DrvGPIO_ClrBit(E_PT1, 0);
```

5.3.4 DrvGPIO_GetBit

- **函數**

uint8_t DrvGPIO_GetBit (E_DRVGPIO_PORT port, uint8_t u32Bit)

- **函數功能**

讀取GPIO PT1~PT3任何一位IO 口的輸入狀態值.

讀取GPIO輸入狀態寄存器0x40808[7:0]/0x40818[7:0]/0x40828[7:0]

- **輸入參數**

port [in]： 代表GPIO port. 它的設定值範圍是1~4.

1：PT1　2：PT2

3：PT3　4：Rsv.

u32Bit [in]： 代表GPIO port任何一位IO 口，它的設定值範圍是 0、1、2、3、4、5、6、7.

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

0/1：IO pin的輸入狀態

0xff000000：讀取失敗

- **函數用法**

```
uint32_t  i32Bit數值;
/* 設置PT1.1 作為輸入模式，並讀取PT1.1的輸入狀態值*/
DrvGPIO_Open(E_PT1, 1, E_IO_INPUT);
i32Bit數值 = DrvGPIO_GetBit(E_PT1, 1);
if (u32Bit數值 == 1)
{
   printf(“PT1-1 pin status is high.\n”); }
else
{
   printf(“PT1-1 pin status is low.\n”); }
```

5.3.5 DrvGPIO_SetPortBits

- **函數**

unsigned int DrvGPIO_SetPortBits (E_DRVGPIO_PORT uport, unsigned int ui32Data)

- **函數功能**

設置GPIO PT1~PT3 對應IO口的輸出狀態.

設置GPIO的輸出狀態寄存器0x40804[7:0]/0x40814[7:0]/0x40824[7:0]

- **輸入參數**

uport [in]： 代表GPIO port. 設定值範圍是1~4.

1：PT1　2：PT2

3：PT3　4：Rsv.

i32Data [in]： 設置對應位IO口，對應位為1則會被置1，對應位為0則會被置0，設定值範圍0~0xFF.

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/* 設定PT1.1、PT1.4為1，所以設定參數0x12 */

DrvGPIO_SetPortBits(E_PT1, 0x12);

5.3.6 DrvGPIO_ClrPortBits

- **函數**

unsigned int DrvGPIO_ClrPortBits (E_DRVGPIO_PORT uport, unsigned int ui32Data)

- **函數功能**

清除GPIO PT1~PT3 對應位IO口輸出狀態值.

清零GPIO的輸出狀態寄存器0x40804[7:0] / 0x40814[7:0] / 0x40824[7:0]

- **輸入參數**

uport [in]： 代表GPIO port，設定值範圍是1~4.

1：PT1　2：PT2

3：PT3　4：Rsv.

i32Data [in]： 代表對應位的IO口，對應位為1才會被置0，設定範圍是0~0xFF.

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/* 清除PT1.1/PT1.4的輸出為0，設定輸入參數 0x12 */

DrvGPIO_ClrPortBits(E_PT1, 0x12);

5.3.7 DrvGPIO_GetPortBits

- **函數**

uint32_t DrvGPIO_GetPortBits (E_DRVGPIO_PORT port)

- **函數功能**

  讀取GPIO PT1~PT3輸入狀態值. 讀取GPIO的輸出狀態寄存器0x40808[7 :0] / 0x40818[7 :0] / 0x40828[7 :0]

- **輸入參數**

  port [in]： 代表GPIO port，設定值範圍是1~4.

  1：PT1　2：PT2

  3：PT3　4：Rsv.

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  0 ~ 0xFF：讀取成功，待讀取GPIO PORT的輸入狀態值:

  0xff000000：讀取失敗

- **函數用法**

  /*讀取PT1的輸入狀態值*/

  uint32_t　i32Port; i32Port = DrvGPIO_GetPortBits(E_PT1);

5.3.8 DrvGPIO_IntTrigger

- **函數**

  int32_t DrvGPIO_IntTrigger (E_DRVGPIO_PORT port, uint32_t u32Bit, E_DRVGPIO_TriMethod mode)

- **函數功能**

  使能GPIO PT1~PT2的外部中斷觸發沿並設置外部中斷的觸發沿模式.

  設置GPIO寄存器0x4080C[31:0]/0x4081C[31:0]

- **輸入參數**

  port [in]： 代表GPIO port.設定值範圍是1~2.

  1：PT1　2：PT2

  u32Bit [in]： 代表GPIO port的每一位IO口，對應位為1表示該位IO被設置，輸入0x0時，觸發功能無效

  設置值範圍 ：0x00~0xFF

  mode [in] : IO口的中斷觸發模式選擇，設定值範圍是0~7

  0：關閉IO 外部中斷觸發　1：上升沿觸發

  2：下降沿觸發　3：電平變化觸發

  4：低電平觸發　5：高電平觸發

  6：低電平觸發　7：高電平觸發.

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 設置PT1.0中斷觸發模式為下降沿觸發*/

  DrvGPIO_ClkGenerator(0,1);　//設置IO 採樣頻率

DrvGPIO_Open(E_PT1, 0x01, E_IO_ IntEnable); //使能IO外部中斷

DrvGPIO_IntTrigger(E_PT1, 0x01, E_N_Edge); //設置中斷觸發模式

5.3.9 DrvGPIO_ClkGenerator

- **函數**

uint32_t DrvGPIO_ClkGenerator (E_DRVGPIO_CLK_SOURCE uClk, uint32_t uDivider)

- **函數功能**

設置IO採樣頻率時鐘源及時鐘分頻值.

設置寄存器0x4030C[20:16]

- **輸入參數**

uClk [in] : 代表GPIO採樣頻率時鐘源. 設置值範圍 : 0~1.

0：高速時鐘源（HS_CK）

1：低速時鐘源（LS_CK）

uDivider [in] : IO 口的時鐘源分頻值. 設置值範圍 : 0~15.

0: off	8: ÷128
1: ÷1	9: ÷256
2: ÷2	10: ÷512
3: ÷4	11: ÷1024
4: ÷8	12: ÷2048
5: ÷16	13: ÷4096
6: ÷32	14: ÷8192
7: ÷64	15: ÷16384

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/* 設置IO 採樣頻率時鐘源為高速時鐘(HS_CK)及clk/2 .*/

DrvGPIO_ClkGenerator(E_HS_CK, 2); //0x4030C[20]=0b,[19:16]=0010

5.3.10 DrvGPIO_ClearIntFlag

- **函數**

unsigned int DrvGPIO_ClearIntFlag (E_DRVGPIO_PORT port, uint32_t u32Bit)

- **函數功能**

清除GPIO PT1~PT2外部中斷標誌位元；

清零中斷寄存器0x40010[7 :0] / 0x40014[7 :0]。

- **輸入參數**

port [in] ：代表GPIO，設定範圍是1~2

1：PT1　2：PT2

u32Bit [in] ：

代表GPIO port的每一位IO，對應位為1的才會被清零，設定值範圍是0x00~0xFF;

設定值的每一位對應一位IO pin，對應位為1的IO 口的標誌位元就被清零。

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

GPIO 外部中斷標誌位元的當前狀態值

- **函數用法**

```
/* 清零 PT1.2 interrupt flag */
DrvGPIO_ClearIntFlag(E_PT1, 0x04);
/*清零 PT1.3 interrupt flag*/
DrvGPIO_ClearIntFlag(E_PT1,0x08);
```

5.3.11 DrvGPIO_GetIntFlag

- **函數**

unsigned int DrvGPIO_GetIntFlag(E_DRVGPIO_PORT port)

- **函數功能**

讀取對應GPIO PT1~PT2的中斷標誌位元，返回寄存器的值，返回值的對應位為1表示該位的IO 口發生中斷，若為0,則表示沒有中斷產生.

讀取中斷寄存器0x40010[7 :0] / 0x40014[7 :0]的值。

- **輸入參數**

port [in]：代表GPIO port.設定值範圍值是1~2

1：PT1　2：PT2

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

返回值是 GPIO 的中斷標誌位元值: 0 ~ 0Xff

- **函數用法**

```
 /* 讀取 PT1 外部中斷標誌位元 */
unsigned char　flag；flag=DrvGPIO_GetIntFlag(E_PT1);
```

5.3.12 DrvGPIO_Close

- **函數**

int32_t DrvGPIO_Close (E_DRVGPIO_PORT port, int32_t i32Bit, E_DRVGPIO_IO mode)

- **函數功能**

關閉GPIO PT1~PT3任何一位IO引腳的工作模式，可選工作模式有輸入/輸出/外部中斷/電阻上拉。

設置PT1寄存器0x40800[23:16] / 0x40800[7:0] / 0x40804[23:16] / 0x40010[23:16]

PT2寄存器0x40810[23:16] / 0x40810[7:0] / 0x40814[23:16] / 0x40014[23:16]

PT3寄存器0x40820[23:16] / 0x40820[7:0] / 0x40824[23:16]

- **輸入參數**

port [in]：代表GPIO port. 它的值可以是1~4.

1：PT1　2：PT2

3：PT3　4：Rsv.

i32Bit [in] ：代表 GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳工作模式，為0時不做設置；

設置值範圍是 0~255.

mode [in] ：代表GPIO 每一位IO 口的工作模式, 設置值範圍 ：0~3.

0：輸入模式　1：輸出模式

2：內部上拉　3：使能外部中斷

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

```
/* 關閉PT1.0 作為輸出模式及 PT1.1 作為輸入模式*/
DrvGPIO_Close(E_PT1, 0x01, E_IO_OUTPUT); //關閉PT1.0打開輸出模式
DrvGPIO_Close(E_PT1, 0x02, E_IO_INPUT);  // 關閉PT1.1打開輸入模式
```

5.3.13 DrvGPIO_EnableAnalogPin

- **函數**

unsigned char DrvGPIO_EnableAnalogPin(short port,unsigned int i32Bit)

- **函數功能**

關閉GPIO任何一位IO引腳的數位工作模式，如輸入/輸出/外部中斷/電阻上拉/中斷觸發沿，開啟IO類比工作模式。

設置PT1寄存器 0x40800[23:16] / 0x40800[7:0] / 0x40804[23:16] /0x4080C[23:0]/ 0x40010[23:16]

PT2寄存器 0x40810[23:16] / 0x40810[7:0] / 0x40814[23:16] /0x4081C[23:0]/ 0x40014[23:16]

PT3寄存器 0x40820[23:16] / 0x40820[7:0] / 0x40824[23:16]

- **輸入參數**

port [in]：代表GPIO port. 設定值範圍是1~3.

1：PT1　2：PT2

3：PT3

u32Bit [in]：代表 GPIO 任何一位IO口引腳，對應位的值為1表示關閉對應IO引腳工作模式，為0時不做設置；

設置值範圍是 0~0xFF.

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  0：設置成功

  1：設置失敗

- **函數用法**

```
/* 關閉PT3.1/PT3.3/PT3.5/PT3.7數位功能*/
DrvGPIO_Open(E_PT3,0xAA,E_IO_INPUT);
DrvGPIO_Open(E_PT3,0x55,E_IO_OUTPUT);
DrvGPIO_Open(E_PT3,0xAA,E_IO_PullHigh);
DrvGPIO_IntTrigger(E_PT3,0xAA,E_N_Edge);
DrvGPIO_EnableAnalogPin(E_PT3,0xAA);
```

5.3.14 DrvGPIO_PT1_EnableINPUT

- **函數**

  void DrvGPIO_PT1_EnableINPUT(short int ubit)

- **函數功能**

  使能GPIO PT1任何一位IO引腳的輸入模式

  設置PT1寄存器0x40804[23:16]

- **輸入參數**

  ubit [in]：代表 GPIO 任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳輸入模式，為0時不做設置；

  設置值範圍是 0~0xff ;

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

```
/* 設置PT1.0/PT1.1 作為輸入模式*/
DrvGPIO_PT1_EnableINPUT(0x01| 0x02); //PT1.0/PT1.1打開輸入模式
```

5.3.15 DrvGPIO_PT1_DisableINPUT

- **函數**

  void DrvGPIO_PT1_DisableINPUT(short int ubit)

- **函數功能**

  關閉GPIO PT1任何一位IO引腳的輸入模式

  設置PT1寄存器0x40804[23:16]

- **輸入參數**

  ubit [in] :代表 GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳輸入模式，為0時不做設置；設置值範圍是 0~0xff ;

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

  /* 關閉PT1.0/PT1.1 作為輸入模式*/

  DrvGPIO_PT1_DisableINPUT(0x01|0x02); //PT1.0/PT1.1關閉輸入模式

5.3.16 DrvGPIO_PT1_EnablePullHigh

- **函數**

  void DrvGPIO_PT1_EnablePullHigh(short int ubit)

- **函數功能**

  使能GPIO PT1任何一位IO引腳的上拉電阻

  設置PT1寄存器0x40800[23:16]

- **輸入參數**

  ubit [in] ： 代表 GPIO 任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳上拉電阻，為0時不做設置；設置值範圍是 0~0xff ;

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

  /* 使能PT1.0/PT1.1 上拉電阻*/

  DrvGPIO_PT1_EnablePullHigh(0x01|0x02); //PT1.0/PT1.1打開上拉電阻

5.3.17 DrvGPIO_PT1_DisablePullHigh

- **函數**

  void DrvGPIO_PT1_DisablePullHigh(short int ubit)

- **函數功能**

  關閉GPIO PT1任何一位IO引腳的上拉電阻

  設置PT1寄存器0x40800[23:16]

- **輸入參數**

  ubit [in]： 代表 GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳上拉電阻，為0時不做設置；設置值範圍是 0~0xff ;

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

  /* 關閉PT1.0/PT1.1 上拉電阻*/

  DrvGPIO_PT1_ DisablePullHigh (0x01|0x02); //PT1.0/PT1.1關閉上拉電阻

5.3.18 DrvGPIO_PT1_EnableOUTPUT

- **函數**

  void DrvGPIO_PT1_EnableOUTPUT(short int ubit)

- **函數功能**

  使能GPIO PT1任何一位IO引腳的輸出模式

  設置PT1寄存器0x40800[7:0]

- **輸入參數**

  ubit [in]：代表 GPIO 任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳輸出模式，為0時不做設置；

  設置值範圍是 0~0xff ;

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

  /* 使能PT1.0/PT1.1 輸出模式*/

  DrvGPIO_PT1_EnableOUTPUT(0x01|0x02); //PT1.0/PT1.1打開輸出模式

5.3.19 DrvGPIO_PT1_DisableOUTPUT

- **函數**

  void DrvGPIO_PT1_DisableOUTPUT(short int ubit)

- **函數功能**

  關閉GPIO PT1任何一位IO引腳的輸出模式

  設置PT1寄存器0x40800[7:0]

- **輸入參數**

  ubit [in]：代表 GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳輸出模式，為0時不做設置；

  設置值範圍是 0~0xff ;

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

```
/* 關閉PT1.0/PT1.1 輸出模式*/
DrvGPIO_PT1_DisableOUTPUT(0x01|0x02); //PT1.0/PT1.1關閉輸出模式
```

5.3.20 DrvGPIO_PT1_EnableINT

- **函數**

void DrvGPIO_PT1_EnableINT(short int ubit)

- **函數功能**

使能GPIO PT1任何一位IO引腳的外部中斷功能。

設置PT1寄存器0x40010[23:16]

- **輸入參數**

ubit [in]：代表 GPIO 任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳外部中斷功能，為0時不做設置；設置值範圍是 0~0xFF；

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

無

- **函數用法**

```
/* 使能PT1.0/PT1.1 外部中斷功能*/
DrvGPIO_PT1_EnableINT(0x01|0x02); //PT1.0/PT1.1打開外部中斷功能
```

5.3.21 DrvGPIO_PT1_DisableINT

- **函數**

void DrvGPIO_PT1_DisableINT(short int ubit)

- **函數功能**

關閉GPIO PT1任何一位IO引腳的外部中斷功能。

設置PT1寄存器0x40010[23:16]

- **輸入參數**

ubit [in]：代表 GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳外部中斷功能，為0時不做設置；設置值範圍是 0~0xFF；

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

無

- **函數用法**

```
/* 關閉PT1.0/PT1.1 外部中斷功能*/
DrvGPIO_PT1_DisableINT(0x01|0x02); //PT1.0/PT1.1關閉外部中斷功能
```

5.3.22 DrvGPIO_PT1_IntTriggerPorts

- **函數**

  void DrvGPIO_PT1_IntTriggerPorts(uint32_t i32Bit, uint32_t mode)

- **函數功能**

  使能GPIO PT1的外部中斷觸發沿並設置外部中斷的觸發沿模式.

  設置GPIO寄存器0x4080C[31:0]

- **輸入參數**

  u32Bit [in] :

  代表GPIO port的每一位IO 口，對應位為1表示該位IO被設置，輸入0x0時，觸發功能無效，設定值是 0~0xFF.

  mode [in] : IO口的中斷觸發模式選擇，設定值範圍是0~7

  0：關閉IO 外部中斷觸發　　1：上升沿觸發　　2：下降沿觸發　　3：電平變化觸發

  4：低電平觸發　　5：高電平觸發　　6：低電平觸發　　7：高電平觸發.

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

  /* 設置PT1.0中斷觸發模式為下降沿觸發*/

  DrvGPIO_ClkGenerator(0,1); //設置IO 採樣頻率

  DrvGPIO_PT1_EnableINT(0x1); //使能IO外部中斷

  DrvGPIO_PT1_IntTriggerPorts(0x1, E_N_Edge); //設置中斷觸發模式

5.3.23 DrvGPIO_PT1_IntTriggerBit

- **函數**

  void DrvGPIO_PT1_IntTriggerBit(uint32_t i32Bit, uint32_t mode)

- **函數功能**

  使能GPIO PT1被選中引腳的外部中斷觸發沿並設置外部中斷的觸發沿模式.

  設置GPIO寄存器0x4080C[31:0]

- **輸入參數**

  u32Bit [in]： 輸入範圍為0~7，代表GPIO port的bit7~bit0，選中的引腳才被設置.

  mode [in] : IO口的中斷觸發模式選擇，設定值範圍是0~7

  0：關閉IO 外部中斷觸發　　1：上升沿觸發　　2：下降沿觸發　　3：電平變化觸發

  4：低電平觸發　　5：高電平觸發　　6：低電平觸發　　7：高電平觸發.

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

```
/* 設置PT1.0中斷觸發模式為下降沿觸發*/
DrvGPIO_ClkGenerator(0,1); //設置IO 採樣頻率
DrvGPIO_PT1_EnableINT(0x1);  //使能IO外部中斷
DrvGPIO_PT1_IntTriggerPorts(0x1, E_N_Edge);    //設置中斷觸發模式
```

5.3.24 DrvGPIO_PT1_GetIntFlag

- **函數**

unsigned char DrvGPIO_PT1_GetIntFlag(void)

- **函數功能**

讀取對應GPIO PT1的中斷標誌位元，返回寄存器的值，返回值的對應位為1表示該位的IO 口發生中斷，若為0,則表示沒有中斷產生.

讀取中斷寄存器0x40010[7 :0]的值。

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

返回值是 PT1 的中斷標誌位元值: 0 ~ 0xff

- **函數用法**

```
 /* 讀取 PT1 外部中斷標誌位元 */
unsigned char   flag；flag=DrvGPIO_PT1_GetIntFlag();
```

5.3.25 DrvGPIO_PT1_ClearIntFlag

- **函數**

void DrvGPIO_PT1_ClearIntFlag(short int uint32)

- **函數功能**

清除GPIO PT1外部中斷標誌位元；

清零中斷寄存器0x40010[7 :0] 。

- **輸入參數**

u32Bit [in] :

代表GPIO port的每一位IO，對應位為1的才會被清零，設定值範圍是0x00~0xFF;

設定值的每一位對應一位IO pin，對應位為1的IO 口的標誌位元就被清零。

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

無

- **函數用法**

```
/* 清零 PT1.2 interrupt flag */
DrvGPIO_PT1_ClearIntFlag(0x04);
/*清零 PT1.3 interrupt flag*/
DrvGPIO_PT1_ClearIntFlag(0x08);
```

5.3.26 DrvGPIO_PT1_GetPortBits

- **函數**

  unsigned char DrvGPIO_PT1_GetPortBits (void)

- **函數功能**

  讀取GPIO PT1輸入狀態值. 讀取GPIO的輸入狀態寄存器0x40808[7 :0]

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  0 ~ 0xFF：待讀取GPIO PORT的輸入狀態值:

- **函數用法**

  /*讀取PT1的輸入狀態值*/

  uint32_t i32Port; i32Port = DrvGPIO_PT1_GetPortBits();

5.3.27 DrvGPIO_PT1_SetPortBits

- **函數**

  void DrvGPIO_PT1_SetPortBits (unsigned char ui32Data)

- **函數功能**

  設置GPIO PT1對應IO口的輸出狀態.

  設置GPIO的輸出狀態寄存器0x40804[7:0]

- **輸入參數**

  i32Data [in]： 設定值範圍0~0xFF.bit7~bit0對應每一位IO PIN，對應位為1則會被置1，對應位為0則會被置0

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

  /* 設定PT1.2、PT1.4為1，所以設定參數0x14 */

  DrvGPIO_PT1_SetPortBits(0x14);

5.3.28 DrvGPIO_PT1_ClrPortBits

- **函數**

  void DrvGPIO_PT1_ClrPortBits (unsigned int ui32Data)

- **函數功能**

  清除GPIO PT1 對應位IO口輸出狀態值.

  清零GPIO的輸出狀態寄存器0x40804[7:0]

- **輸入參數**

  i32Data [in]： 設定範圍是0~0xFF. bit7~bit0對應每一位IO PIN，對應位為1輸出才被置0，

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

  /* 清除PT1.1/PT1.4的輸出為0，設定輸入參數 0x12 */

  DrvGPIO_PT1_ClrPortBits(0x12);

5.3.29 DrvGPIO_PT2_EnableINPUT

- **函數**

  void DrvGPIO_PT2_EnableINPUT(short int ubit)

- **函數功能**

  使能GPIO PT2任何一位IO引腳的輸入模式

  設置PT2寄存器0x40814[23:16]

- **輸入參數**

  ubit [in]： 代表 GPIO 任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳輸入模式，為0時不做設置；

  設置值範圍是 0~0xff ;

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

  /* 設置PT2.0/PT2.1 作為輸入模式*/

  DrvGPIO_PT2_EnableINPUT(0x01|0x02); //PT2.0/PT2.1打開輸入模式

5.3.30 DrvGPIO_PT2_DisableINPUT

- **函數**

  void DrvGPIO_PT2_DisableINPUT(short int ubit)

- **函數功能**

  關閉GPIO PT2任何一位IO引腳的輸入模式

  設置PT2寄存器0x40814[23:16]

- **輸入參數**

ubit [in]：代表 GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳輸入模式，為0時不做設置；設置值範圍是 0~0xff；

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

無

- **函數用法**

```
/* 關閉PT2.0/PT2.1 作為輸入模式*/
DrvGPIO_PT2_DisableINPUT(0x01|0x02); //PT2.0/PT2.1關閉輸入模式
```

5.3.31 DrvGPIO_PT2_EnablePullHigh

- **函數**

void DrvGPIO_PT2_EnablePullHigh(short int ubit)

- **函數功能**

使能GPIO PT2任何一位IO引腳的上拉電阻

設置PT2寄存器0x40810[23:16]

- **輸入參數**

ubit [in]：代表 GPIO 任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳上拉電阻，為0時不做設置；設置值範圍是 0~0xff；

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

無

- **函數用法**

```
/* 使能PT2.0/PT2.1 上拉電阻*/
DrvGPIO_PT2_EnablePullHigh(0x01| 0x02); //PT2.0/PT2.1打開上拉電阻
```

5.3.32 DrvGPIO_PT2_DisablePullHigh

- **函數**

void DrvGPIO_PT2_DisablePullHigh(short int ubit)

- **函數功能**

關閉GPIO PT2任何一位IO引腳的上拉電阻

設置PT2寄存器0x40810[23:16]

- **輸入參數**

ubit [in]：代表 GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳上拉電阻，為0時不做設置；設置值範圍是 0~0xff；

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

  /* 關閉PT2.0/PT2.1 上拉電阻*/

  DrvGPIO_PT2_DisablePullHigh(0x01|0x02); //PT2.0/PT2.1關閉上拉電阻

5.3.33 DrvGPIO_PT2_EnableOUTPUT

- **函數**

  void DrvGPIO_PT2_EnableOUTPUT(short int ubit)

- **函數功能**

  使能GPIO PT2任何一位IO引腳的輸出模式

  設置PT2寄存器0x40810[7:0]

- **輸入參數**

  ubit [in]：代表 GPIO 任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳輸出模式，為0時不做設置；

  設置值範圍是 0~0xff ;

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

  /* 使能PT2.0/PT2.1 輸出模式*/

  DrvGPIO_PT2_EnableOUTPUT(0x01|0x02); //PT2.0/PT2.1打開輸出模式

5.3.34 DrvGPIO_PT2_DisableOUTPUT

- **函數**

  void DrvGPIO_PT2_DisableOUTPUT(short int ubit)

- **函數功能**

  關閉GPIO PT2任何一位IO引腳的輸出模式

  設置PT2寄存器0x40810[7:0]

- **輸入參數**

  ubit [in]：代表 GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳輸出模式，為0時不做設置；

  設置值範圍是 0~0xff ;

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

```
/* 關閉PT2.0/PT2.1 輸出模式*/
DrvGPIO_PT2_DisableOUTPUT(0x01|0x02); //PT2.0/PT2.1關閉輸出模式
```

5.3.35 DrvGPIO_PT2_EnableINT

- **函數**

void DrvGPIO_PT2_EnableINT(short int ubit)

- **函數功能**

使能GPIO PT2任何一位IO引腳的外部中斷功能。

設置PT2寄存器0x40014[23:16]

- **輸入參數**

ubit [in] :代表 GPIO 任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳外部中斷功能，為0時不做設置；設置值範圍是 0~0xFF ;

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

無

- **函數用法**

```
/* 使能PT2.0/PT2.1 外部中斷功能*/
DrvGPIO_PT2_EnableINT(0x01|0x02); //PT2.0/PT2.1打開外部中斷功能
```

5.3.36 DrvGPIO_PT2_DisableINT

- **函數**

void DrvGPIO_PT2_DisableINT(short int ubit)

- **函數功能**

關閉GPIO PT2任何一位IO引腳的外部中斷功能。

設置PT2寄存器0x40014[23:16]

- **輸入參數**

ubit [in]： 代表 GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳外部中斷功能，為0時不做設置；設置值範圍是 0~0xFF ;

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

無

- **函數用法**

```
/* 關閉PT2.0/PT2.1 外部中斷功能*/
DrvGPIO_PT2_DisableINT(0x01|0x02); //PT2.0/PT2.1關閉外部中斷功能
```

5.3.37 DrvGPIO_PT2_IntTriggerPorts

- **函數**

  void DrvGPIO_PT2_IntTriggerPorts(uint32_t i32Bit, uint32_t mode)

- **函數功能**

  使能GPIO PT2的外部中斷觸發沿並設置外部中斷的觸發沿模式.

  設置GPIO寄存器0x4081C[31:0]

- **輸入參數**

  u32Bit [in] :

  代表GPIO port的每一位IO 口，對應位為1表示該位IO被設置，輸入0x0時，觸發功能無效，設定值是 0~0xFF.

  mode [in] : IO口的中斷觸發模式選擇，設定值範圍是0~7

  0：關閉IO 外部中斷觸發　　1：上升沿觸發　　2：下降沿觸發　　3：電平變化觸發

  4：低電平觸發　　5：高電平觸發　　6：低電平觸發　　7：高電平觸發.

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

```
/* 設置PT2.0中斷觸發模式為下降沿觸發*/
DrvGPIO_ClkGenerator(0,1); //設置IO 採樣頻率
DrvGPIO_PT2_EnableINT(0x1);   //使能IO外部中斷
DrvGPIO_PT2_IntTriggerPorts(0x1, E_N_Edge);    //設置中斷觸發模式
```

5.3.38 DrvGPIO_PT2_IntTriggerBit

- **函數**

  void DrvGPIO_PT2_IntTriggerBit(uint32_t i32Bit, uint32_t mode)

- **函數功能**

  使能GPIO PT2被選中引腳的外部中斷觸發沿並設置外部中斷的觸發沿模式.

  設置GPIO寄存器0x4081C[31:0]

- **輸入參數**

  u32Bit [in]： 輸入範圍為0~7，代表GPIO port的bit7~bit0，選中的引腳才被設置.

  mode [in] : IO口的中斷觸發模式選擇，設定值範圍是0~7

  0：關閉IO 外部中斷觸發　　1：上升沿觸發　　2：下降沿觸發　　3：電平變化觸發

  4：低電平觸發　　5：高電平觸發　　6：低電平觸發　　7：高電平觸發.

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

```
/* 設置PT2.0中斷觸發模式為下降沿觸發*/
DrvGPIO_ClkGenerator(0,1); //設置IO 採樣頻率
DrvGPIO_PT2_EnableINT(0x1);  //使能IO外部中斷
DrvGPIO_PT2_IntTriggerPorts(0x1, E_N_Edge);    //設置中斷觸發模式
```

5.3.39 DrvGPIO_PT2_GetIntFlag

- **函數**

unsigned char DrvGPIO_PT2_GetIntFlag(void)

- **函數功能**

讀取對應GPIO PT2的中斷標誌位元，返回寄存器的值，返回值的對應位為1表示該位的IO 口發生中斷，若為0,則表示沒有中斷產生.

讀取中斷寄存器0x40014[7 :0]的值。

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

返回值是 PT2 的中斷標誌位元值: 0 ~ 0Xff

- **函數用法**

```
 /* 讀取 PT2 外部中斷標誌位元 */
unsigned char   flag；flag=DrvGPIO_PT2_GetIntFlag();
```

5.3.40 DrvGPIO_PT2_ClearIntFlag

- **函數**

void DrvGPIO_PT2_ClearIntFlag(short int uint32)

- **函數功能**

清除GPIO PT2外部中斷標誌位元；

清零中斷寄存器0X40014[7 :0] 。

- **輸入參數**

u32Bit [in] :

代表GPIO port的每一位IO，對應位為1的才會被清零，設定值範圍是0x00~0xFF;

設定值的每一位對應一位IO pin，對應位為1的IO 口的標誌位元就被清零。

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

無

- **函數用法**

```
/* 清零 PT2.2 interrupt flag */
DrvGPIO_PT2_ClearIntFlag(0x04);
/*清零 PT2.3 interrupt flag*/
DrvGPIO_PT2_ClearIntFlag(0x08);
```

5.3.41 DrvGPIO_PT2_GetPortBits

- **函數**

unsigned char DrvGPIO_PT2_GetPortBits (void)

- **函數功能**

讀取GPIO PT2輸入狀態值. 讀取GPIO的輸入狀態寄存器0x40818[7 :0]

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

0 ~ 0xFF：待讀取GPIO PORT的輸入狀態值:

- **函數用法**

```
/*讀取PT2的輸入狀態值*/
uint32_t    i32Port; i32Port = DrvGPIO_PT2_GetPortBits();
```

5.3.42 DrvGPIO_PT2_SetPortBits

- **函數**

void DrvGPIO_PT2_SetPortBits (unsigned char ui32Data)

- **函數功能**

設置GPIO PT2對應IO口的輸出狀態.

設置GPIO的輸出狀態寄存器0x40814[7:0]

- **輸入參數**

i32Data [in]： 設定值範圍0~0xFF.bit7~bit0對應每一位IO PIN，對應位為1則會被置1，對應位為0則會被置0

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

無

- **函數用法**

```
/* 設定PT2.2、PT2.4為1，所以設定參數0x14 */
DrvGPIO_PT2_SetPortBits(0x14);
```

5.3.43 DrvGPIO_PT2_ClrPortBits

- **函數**

  void DrvGPIO_PT2_ClrPortBits (unsigned int ui32Data)

- **函數功能**

  清除GPIO PT2 對應位IO口輸出狀態值.

  清零GPIO的輸出狀態寄存器0x40814[7:0]

- **輸入參數**

  i32Data [in]： 設定範圍是0~0xFF. bit7~bit0對應每一位IO PIN，對應位為1輸出才被置0，

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

  /* 清除PT2.1/PT2.4的輸出為0，設定輸入參數 0x12 */

  DrvGPIO_PT2_ClrPortBits(0x12);

5.3.44 DrvGPIO_PT3_EnableINPUT

- **函數**

  void DrvGPIO_PT3_EnableINPUT(short int ubit)

- **函數功能**

  使能GPIO PT3任何一位IO引腳的輸入模式

  設置PT3寄存器0x40824[23:16]

- **輸入參數**

  ubit [in]： 代表 GPIO 任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳輸入模式，為0時不做設置；

  設置值範圍是 0~0xff ;

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

  /* 設置PT3.0/PT3.1 作為輸入模式*/

  DrvGPIO_PT3_EnableINPUT (0x01|0x02); //PT3.0/PT3.1打開輸入模式

5.3.45 DrvGPIO_PT3_DisableINPUT

- **函數**

  void DrvGPIO_PT3_DisableINPUT(short int ubit)

- **函數功能**

關閉GPIO PT3任何一位IO引腳的輸入模式

設置PT3寄存器0x40824[23:16]

- **輸入參數**

ubit [in]：代表 GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳輸入模式，為0時不做設置；

設置值範圍是 0~0xff ;

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

無

- **函數用法**

```
/* 關閉PT3.0/PT3.1 作為輸入模式*/
DrvGPIO_PT3_DisableINPUT(0x01|0x02); //PT3.0/PT3.1關閉輸入模式
```

5.3.46 DrvGPIO_PT3_EnablePullHigh

- **函數**

void DrvGPIO_PT3_EnablePullHigh(short int ubit)

- **函數功能**

使能GPIO PT3任何一位IO引腳的上拉電阻

設置PT3寄存器0x40820[23:16]

- **輸入參數**

ubit [in]：代表 GPIO 任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳上拉電阻，為0時不做設置；

設置值範圍是 0~0xff ;

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

無

- **函數用法**

```
/* 使能PT3.0/PT3.1 上拉電阻*/
DrvGPIO_PT3_EnablePullHigh(0x01|0x02); //PT3.0/PT3.1打開上拉電阻
```

5.3.47 DrvGPIO_PT3_DisablePullHigh

- **函數**

void DrvGPIO_PT3_DisablePullHigh(short int ubit)

- **函數功能**

關閉GPIO PT3任何一位IO引腳的上拉電阻

設置PT3寄存器0x40820[23:16]

- **輸入參數**

ubit [in]：代表 GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳上拉電阻，為0時不做設置；設置值範圍是 0~0xff；

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

無

- **函數用法**

```
/* 關閉PT3.0/PT3.1 上拉電阻*/
DrvGPIO_PT3_DisablePullHigh(0x01|0x02); //PT3.0/PT3.1關閉上拉電阻
```

5.3.48 DrvGPIO_PT3_EnableOUTPUT

- **函數**

void DrvGPIO_PT3_EnableOUTPUT(short int ubit)

- **函數功能**

使能GPIO PT3任何一位IO引腳的輸出模式

設置PT3寄存器0x40820[7:0]

- **輸入參數**

ubit [in]：代表 GPIO 任何一位IO 口引腳，對應位的值為1表示打開對應IO引腳輸出模式，為0時不做設置；設置值範圍是 0~0xff；

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

無

- **函數用法**

```
/* 使能PT3.0/PT3.1 輸出模式*/
DrvGPIO_PT3_EnableOUTPUT(0x01|0x02); //PT3.0/PT3.1打開輸出模式
```

5.3.49 DrvGPIO_PT3_DisableOUTPUT

- **函數**

void DrvGPIO_PT3_DisableOUTPUT(short int ubit)

- **函數功能**

關閉GPIO PT3任何一位IO引腳的輸出模式

設置PT3寄存器0x40820[7:0]

- **輸入參數**

ubit [in]：代表 GPIO 任何一位IO 口引腳，對應位的值為1表示關閉對應IO引腳輸出模式，為0時不做設置；設置值範圍是 0~0xff；

- **包含標頭檔**

Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

  /* 關閉PT3.0/PT3.1 輸出模式*/

  DrvGPIO_PT3_DisableOUTPUT(0x01|0x02); //PT3.0/PT3.1關閉輸出模式

5.3.50 DrvGPIO_PT3_GetPortBits

- **函數**

  unsigned char DrvGPIO_PT3_GetPortBits (void)

- **函數功能**

  讀取GPIO PT3輸入狀態值. 讀取GPIO的輸入狀態寄存器0x40828[7 :0]

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  0 ~ 0xFF：待讀取GPIO PORT的輸入狀態值:

- **函數用法**

  /*讀取PT3的輸入狀態值*/

  uint32_t i32Port; i32Port = DrvGPIO_PT3_GetPortBits();

5.3.51 DrvGPIO_PT3_SetPortBits

- **函數**

  void DrvGPIO_PT3_SetPortBits (unsigned char ui32Data)

- **函數功能**

  設置GPIO PT3對應IO口的輸出狀態.

  設置GPIO的輸出狀態寄存器0x40824[7:0]

- **輸入參數**

  i32Data [in]： 設定值範圍0~0xFF.bit7~bit0對應每一位IO PIN，對應位為1則會被置1，對應位為0則會被置0

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

  /* 設定PT3.2、PT3.4為1，所以設定參數0x14 */

  DrvGPIO_PT3_SetPortBits(0x14);

5.3.52 DrvGPIO_PT3_ClrPortBits

- **函數**

  void DrvGPIO_PT3_ClrPortBits (unsigned int ui32Data)

- **函數功能**

  清除GPIO PT3 對應位IO口輸出狀態值.

  清零GPIO的輸出狀態寄存器0x40824[7:0]

- **輸入參數**

  i32Data [in]： 設定範圍是0~0xFF. bit7~bit0對應每一位IO PIN，對應位為1輸出才被置0，

- **包含標頭檔**

  Peripheral_lib/DrvGPIO.h

- **函數返回值**

  無

- **函數用法**

  /* 清除PT3.1/PT3.4的輸出為0，設定輸入參數 0x12 */

  DrvGPIO_PT3_ClrPortBits(0x12);

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

序號	函數名稱	功能描述
01	DrvADC_PInputChannel	ADC正端信號輸入源設置
02	DrvADC_NInputChannel	ADC負端信號輸入源設置
03	DrvADC_SetADCInputChannel	ADC正負端信號輸入源設置
04	DrvADC_InputSwitch	ADC輸入端短路開關控制
05	DrvADC_RefInputShort	ADC參考電壓輸入端短路開關控制
06	DrvADC_SetPGA	ADC輸入信號放大倍數PGA設置
07	DrvADC_ADGain	ADC輸入信號放大倍數ADGain設置
08	DrvADC_Gain	ADC輸入信號整體放大倍數設置
09	DrvADC_DCoffset	ADC輸入零點平移(DC offset)設置
10	DrvADC_RefVoltage	ADC參考電壓輸入設置
11	DrvADC_FullRefRange	ADC參考電壓放大倍數設置
12	DrvADC_OSR	ADC轉換輸出率OSR設置
13	DrvADC_ClkEnable	開啟ADC時鐘源
14	DrvADC_ClkDisable	關閉ADC時鐘源
15	DrvADC_FastChopper	ADC Fast-chopper模式設置
16	DrvADC_CombFilter	梳狀濾波器開啟控制
17	DrvADC_EnableInt	ADC中斷開啟
18	DrvADC_DisableInt	ADC中斷關閉
19	DrvADC_ReadIntFlag	讀取ADC中斷標誌位元
20	DrvADC_ClearIntFlag	清除ADC中斷標誌位元
21	DrvADC_Enable	開啟ADC
22	DrvADC_Disable	關閉ADC
23	DrvADC_GetConversionData	讀取ADC的A/D轉換值

6.2 內部定義常量

E_ADC_INPUT_CHANNEL

識別字	數值	函數功能
ADC_Input_AIO0	0	信號輸入端
ADC_Input_AIO1	1	信號輸入端
ADC_Input_AIO2	2	信號輸入端
ADC_Input_AIO3	3	信號輸入端
REFO_I	4	信號輸入端
OPOI	5	信號輸入端
TSP0	6	信號輸入端
TSP1	7	信號輸入端
DAOI	8	信號輸入端
VDDA/VSSA	9	信號輸入端

E_ADC_REFV

識別字	數值	函數功能
External	0	外部輸入源
Internal	1	使能緩衝器並使用內部源

E_ADC_PGA & E_ADC_ADGN

識別字	數值	函數功能	識別字	數值	函數功能
ADC_PGA_Disable	0	Disable PGA	ADC_ADGN_1	0	ADGN=1
ADC_PGA_8	1	PGA=8	ADC_ADGN_2	1	ADGN=2
ADC_PGA_16	3	PGA=16	ADC_ADGN_RESER	2	Reserve
ADC_PGA_32	7	PGA=32	ADC_ADGN_4	3	ADGN=4

E_ADC_SIGNAL_SHORT

識別字	數值	函數功能
OPEN	0	ADC信號輸入短路開關斷開
SHORT	1	ADC信號輸入短路開關閉合

E_ADC_VRPS_REF_VOLTAGE

識別字	數值	函數功能
VDDA	0	參考電壓正端輸入為 VDDA
AIO2	1	參考電壓正端輸入為 AIO2
AIO4	2	參考電壓正端輸入為 AIO4
REF_BUFFER_OUT	3	參考電壓正端輸入為 REFO_I

E_ADC_VRNS_REF_VOLTAGE

識別字	數值	函數功能
VSSA	0	參考電壓負端輸入為VSSA
AIO3	1	參考電壓負端輸入為 AIO3
AIO5	2	參考電壓負端輸入為 AIO5
REF_BUFFER_OUT	3	參考電壓負端輸入為 REFO_I

6.3 函數說明

6.3.1 DrvADC_PInputChannel

- **函數**

  unsigned int DrvADC_PInputChannel (E_ADC_INPUT_Channel uINP);

- **函數功能**

  設置ADC 輸入信號正向輸入端；

  設置寄存器0x41104[7:4].

- **輸入參數**

  uINP [in]：代表ADC的正向輸入埠選擇，設定值範圍0~9

0：AIO0, 1：AIO1,

2：AIO2, 3：AIO3,

4：REFO_I, 5：OPOI,

6：TPSP0, 7：TPSP1,

8： DAOI, 9：VDDA;

此處TPS0= TPSP0, TPS1=TPSP1;

- **包含標頭檔**

  Peripheral_lib/DrvADC.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 設定ADC正向輸入端為AIO0*/

  DrvADC_PInputChannel(ADC_Input_AIO0);

6.3.2 DrvADC_NInputChannel

- **函數**

  unsigned int DrvADC_NInputChannel (E_ADC_INPUT_Channel uINN);

- **函數功能**

  設置ADC 輸入信號負向輸入端，設置寄存器0x41104[3:0].

- **輸入參數**

  uINN [in]：代表ADC負端輸入選擇埠, 設定範圍值0~9.

  0：AIO0, 1：AIO1,

  2：AIO2, 3：AIO3,

  4：REFO_I, 5：OPOI,

  6：TPSN0, 7：TPSN1,

8：DAOI,　　9：VSSA

此處TPS0=TPSN0,TPS1=TPSN1；

- **包含標頭檔**

Peripheral_lib/DrvADC.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

```
/*  設定ADC負向輸入端為AIO1*/
DrvADC_NInputChannel(ADC_Input_AIO1);
```

6.3.3 DrvADC_SetADCInputChannel

- **函數**

```
unsigned int DrvADC_SetADCInputChannel (
                    E_ADC_INPUT_Channel uINP,
                    E_ADC_INPUT_Channel uINN    );
```

- **函數功能**

設置ADC輸入信號的正向、負向輸入埠，設置寄存器0x41104[7:4] / 0x41104[3:0].

- **輸入參數**

uINP [in]：代表ADC的正向輸入選擇埠，設定值範圍0~9

0：AIO0,　　1：AIO1,

2：AIO2,　　3：AIO3,

4：REFO_I,　　5：OPOI,

6：TPSP0,　7：TPSP1,

8：　DAOI,　　9：VDDA;

uINN [in]：代表ADC負端輸入選擇埠，設定範圍值0~9.

0：AIO0,　　1：AIO1,

2：AIO2,　　3：AIO3,

4：REFO_I,　　5：OPOI,

6：TPSN0,　7：TPSN1,

8：DAOI,　　9：VSS。

在C函式程式庫中正向與負向輸入使用同一組代表符：

{ADC_Input_AIO0, ADC_Input_AIO1, ADC_Input_AIO2,ADC_Input_AIO3,

REFO_I,OPOI,TPS0,TPS1,DAOI,VDDA_VSS}.

- **包含標頭檔**

Peripheral_lib/DrvADC.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/* 設定ADC正向輸入端為AIO0，負向輸入端為AIO1*/

DrvADC_SetADCInputChannel(ADC_Input_AIO0, ADC_Input_AIO1);

6.3.4 DrvADC_InputSwitch

- **函數**

unsigned int DrvADC_InputSwitch (uVISHR)

- **函數功能**

ADC信號輸入端短路開關控制.

設置寄存器0x41100[21]

- **輸入參數**

uVISHR[in] : ADC信號輸入端短路開關控制. 設定值範圍 : 0~1

0: 短路開關斷開

1: 短路開關閉合

- **包含標頭檔**

Peripheral_lib/DrvADC.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/* ADC 輸入端短路開關閉合*/

DrvADC_InputSwitch(1);

6.3.5 DrvADC_RefInputShort

- **函數**

unsigned int DrvADC_RefInputShort (E_ADC_SIGNAL_SHORT uVrshr);

- **函數功能**

ADC參考電壓輸入端短路開關控制.

設置寄存器0x41100[20].

- **輸入參數**

uVrshr [in] : ADC參考電壓輸入端短路開關控制. 設定值範圍 : 0~1

0 : ADC 參考電壓輸入端短路開關斷開

1 : ADC 參考電壓輸入端短路開關閉合

- **包含標頭檔**

Peripheral_lib/DrvADC.h

- **函數返回值**

0：設置失敗

其他：設置失敗

- **函數用法**

/* 設置ADC 參考電壓輸入端短路開關閉合 */

DrvADC_RefInputShort(SHORT);

6.3.6 DrvADC_SetPGA

- **函數**

unsigned int DrvADC_SetPGA (E_ADC_PGA uPGA);

- **函數功能**

配置ADC 輸入信號內部放大倍數控制器PGA;

設置寄存器0x41104[18:16].

- **輸入參數**

uPGA [in]： 代表ADC內部放大倍數控制器PGA. 設定值範圍 ：0~7

0: 放大倍數為 1

1: 放大倍數為 8

2: 不使用

3: 放大倍數為 16

4: 不使用

5: 不使用

6: 不使用

7: 放大倍數為 32

- **包含標頭檔**

Peripheral_lib/DrvADC.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/* 設置PGA=8 */

DrvADC_SetPGA(ADC_Gain_8);

6.3.7 DrvADC_ADGain

- **函數**

unsigned int DrvADC_ADGain (uADgain);

- **函數功能**

配置ADC 輸入信號內部放大倍數控制器ADGIN；

設置寄存器0x41104[21:20].

- **輸入參數**

  uADgain[in]： 代表ADC 內部放大倍數控制器ADGN. 有效設定值為 ：0, 1, 3

  0: 放大倍數為 1

  1: 放大倍數為 2

  3: 放大倍數為 4

- **包含標頭檔**

  Peripheral_lib/DrvADC.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /*設置ADGN=2 */

  DrvADC_ADGain(1);

6.3.8 DrvADC_Gain

- **函數**

  unsigned int DrvADC_Gain (E_ADC_PGA uPGA ,uADgain);

- **函數功能**

  配置ADC 輸入信號內部放大倍數控制器PGA及ADGN

  設置寄存器0x41104[18:16]/ 0x41104[21:20].

- **輸入參數**

  uPGA [in]： 代表ADC 放大倍數控制器PGA. 設定值範圍 ：0~7

  0: 放大倍數為 1

  1: 放大倍數為 8

  2: 不使用

  3: 放大倍數為 16

  4: 不使用

  5: 不使用

  6: 不使用

  7: 放大倍數為 32

  uADgain [in]： 代表ADC 放大倍數器 ADGN. 有效設定值為 ：0, 1, 3

  0: 放大倍數為 1

  1: 放大倍數為 2

  3: 放大倍數為 4

- **包含標頭檔**

  Peripheral_lib/DrvADC.h

- **函數返回值**

  0：設置成功

其他：設置失敗

- **函數用法**

/* 設置ADC放大倍數PGA*ADGN=32*4=128 */

DrvADC_Gain(7,3);

6.3.9 DrvADC_DCoffset

- **函數**

unsigned int DrvADC_DCoffset (uDCoffset);

- **函數功能**

設置ADC 輸入信號的零點平移(DC offset);設置寄存器0x41104[27:24]。

- **輸入參數**

uDCoffice [in]： 代表ADC 零點平移DCSET，VREF=REFP-REFN。設定值範圍 ：0~15

0 ： 0 VREF

1 ： +1/8 VREF

2 ： +1/4 VREF

3 ： +3/8 VREF

4 ： +1/2 VREF

5 ： +5/8 VREF

6 ： +3/4 VREF

7 ： +7/8 VREF

8 ： 0 VREF

9 ： -1/8 VREF

10 ： -1/4 VREF

11 ： -3/8 VREF

12 ： -1/2 VREF

13 ： -5/8 VREF

14 ： -3/4 VREF

15 ： -7/8 VREF

- **包含標頭檔**

Peripheral_lib/DrvADC.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/* 設置零點平移(DCSET)為+1/8 VREF. */

DrvADC_DCoffset(1);

6.3.10 DrvADC_RefVoltage

- **函數**

```
unsigned int DrvADC_RefVoltage (
E_ADC_VRPS_REF_VOLTAGE uVrps,
E_ADC_VRNS_REF_VOLTAGE uVrns
 );
```

- **函數功能**

設置ADC 參考電壓輸入埠,參考電壓(VREF)=VRPS-VRNS；

設置寄存器0x41100[19:18] 及 0x41100[17:16].

- **輸入參數**

uVrps [in]：代表ADC 參考電壓正向輸入端VRPS. 設定值範圍 ：0~3

0： 參考電壓正向輸入來自 VDDA

1： 參考電壓正向輸入來自 AIO2

2： 參考電壓正向輸入來自 AIO4

3： 參考電壓正向輸入來自 REFO_I

uVrns [in]：代表ADC 參考電壓的負向輸入端VRNS. 設定值範圍 ：0~3

0： 參考電壓負向輸入來自 VSSA

1： 參考電壓負向輸入來自 AIO3

2： 參考電壓負向輸入來自 AIO5

3： 參考電壓負向輸入來自 REFO_I

- **包含標頭檔**

Peripheral_lib/DrvADC.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

```
/* 設置ADC參考輸入電壓(VRPS=AIO2, VRNS=AIO3) */
DrvADC_RefVoltage(AIO2, AIO3);
```

6.3.11 DrvADC_FullRefRange

- **函數**

```
unsigned int DrvADC_FullRefRange(uFullRange);
```

- **函數功能**

設置ADC 輸入參考電壓(VREF)的放大倍數;設置寄存器0x41104[19]。

- **輸入參數**

uFullRange[in]：設置 ADC 輸入參考電壓(VREF)的放大數，VREF=VRPS-VRNS. 設定值範圍 ：0~1

0: 1　　輸入參考電壓VREF*1

1: 1/2　輸入參考電壓VREF*1/2

- **包含標頭檔**

  Peripheral_lib/DrvADC.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /*設置ADC輸入參考電壓VREF*1 */

  DrvADC_FullRefRange(0);

6.3.12 DrvADC_OSR

- **函數**

  unsigned int DrvADC_OSR (uADCOSR);

- **函數功能**

  配置ADC 轉換值的輸出頻率(OSR)，設置寄存器0x41100[5:2]。

- **輸入參數**

  uADCOSR[in]： 表示ADC 轉換值輸出率(OSR)除頻器設置(以下輸出率是以時鐘源為327680HZ計算). 設定值範圍 ：0~10

  0 ： ÷32768，資料輸出率是10sps

  1 ： ÷16384，資料輸出率是20sps

  2 ： ÷8192，資料輸出率是40sps

  3 ： ÷4096，資料輸出率是80sps

  4 ： ÷2048，資料輸出率是160sps

  5 ： ÷1024，資料輸出率是320sps

  6 ： ÷512，資料輸出率是640sps

  7 ： ÷256，資料輸出率是1280sps

  8 ： ÷128 ，資料輸出率是2560sps

  9 ： ÷64，資料輸出率是5120sps

  10 ：÷32，資料輸出率是10240sps

- **包含標頭檔**

  Peripheral_lib/DrvADC.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 設置輸出率(OSR)為8192/40sps */

  DrvADC_OSR(2);

6.3.13 DrvADC_ClkEnable

- **函數**

  unsigned int DrvADC_ClkEnable(uADCD, uClkPH);

- **函數功能**

  使能ADC 時鐘源，並設置時鐘源分頻值和ADC 時鐘源相位調整；

  設置寄存器0x4030C[7:4].

- **輸入參數**

  uADCD[in]： 表示ADC 時鐘源分頻器. 設定值範圍 ：0~3

  0 ： ÷6

  1 ： ÷12

  2 ： ÷30

  3 ： ÷60

  uClkPH[in] : ADC 時鐘源相位調整設置. 設定值範圍 ：0~1

  0 ： ADC clock 上升沿在CPU Clock的低電平.

  1 ： ADC clock 上升沿在CPU Clock的高電平

- **包含標頭檔**

  Peripheral_lib/DrvADC.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /*設置ADC 頻率分頻÷12，且ADC時鐘的上升沿為CPU時鐘的高電平 */

  DrvADC_ClkEnable(1,1);

6.3.14 DrvADC_ClkDisable

- **函數**

  void DrvADC_ClkDisable(void);

- **函數功能**

  關閉ADC 時鐘源；設置寄存器0x4030C[6]=0.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvADC.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 關閉ADC時鐘源 */

DrvADC_ClkDisable();

6.3.15 DrvADC_FastChopper

- **函數**

  unsigned int DrvADC_FastChopper(uADFDR);

- **函數功能**

  快速chopper 模式控制.,設置寄存器0x41100[6]. 設定值範圍 ：0~1

- **輸入參數**

  uADFDR[in]： 快速chopper模式控制.

  0： 正常chopper模式，頻率等於 ADCLK/128

  1： 快速chopper模式，頻率等於 ADCLK/32

- **包含標頭檔**

  Peripheral_lib/DrvADC.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 設置 ADC正常chopper模式. */

  DrvADC_FastChopper(0); //設置正常chopper模式，頻率等於ADCLK/128

6.3.16 DrvADC_CombFilter

- **函數**

  unsigned int DrvADC_CombFilter(uCFRST);

- **函數功能**

  梳狀濾波器使能控制，設置該位可以自動丟棄前3筆無效ADC資料；設置寄存器0x41100[1]。

- **輸入參數**

  uCFRST[in]： 梳狀濾波器使能控制. 設定值範圍 ：0~1

  0: 復位(RESET)

  1: 開啟(ON)

- **包含標頭檔**

  Peripheral_lib/DrvADC.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 使能梳狀濾波器並配置自動丟棄前3筆無效數據. */

  DrvADC_CombFilter(0); //濾波器復位

```
DrvADC_CombFilter(1);     //使能濾波器
```

6.3.17 DrvADC_EnableInt

- **函數**

  void DrvADC_EnableInt (void)

- **函數功能**

  開啟ADC中斷向量，ADC為中斷向量HW2；設置寄存器0x40008[16]=1.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvADC.h

- **函數返回值**

  無

- **函數用法**

```
/* 使能ADC 中斷 */
DrvADC_EnableInt();
```

6.3.18 DrvADC_DisableInt

- **函數**

  void DrvADC_DisableInt (void)

- **函數功能**

  關閉ADC 中斷向量；設置寄存器0x40008[16]=0.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvADC.h

- **函數返回值**

  無

- **函數用法**

```
/* 關閉ADC中斷向量 */
DrvADC_DisableInt();
```

6.3.19 DrvADC_ReadIntFlag

- **函數**

  unsigned int DrvADC_ReadIntFlag (void)

- **函數功能**

讀取ADC中斷標誌位元(ADCIF).讀取寄存器0x40008[0]值

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvADC.h

- **函數返回值**

  0：中斷標誌位元值是0，表示無中斷產生

  1：中斷標誌位元值是1，表示有中斷產生

  >1: 無效返回值

- **函數用法**

```
/* 讀取ADC 中斷標誌位元*/
flag=DrvADC_ReadIntFlag();  //讀取ADC中斷要求標誌位元
```

6.3.20 DrvADC_ClearIntFlag

- **函數**

  void DrvADC_ClearIntFlag (void)

- **函數功能**

  清除ADC中斷標誌位元(ADCIF).清零寄存器0x40008[0]=0.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvADC.h

- **函數返回值**

  無

- **函數用法**

```
/* 清除ADC中斷標誌位元*/
DrvADC_ClearIntFlag();    //清除ADC中斷標誌位元
```

6.3.21 DrvADC_Enable

- **函數**

  void DrvADC_Enable(void)

- **函數功能**

  開啟ADC功能；設置寄存器0x41100[0]=1.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvADC.h

- **函數返回值**

  無

- **函數用法**

  /* 使能ADC */

  DrvADC_Enable();

6.3.22 DrvADC_Disable

- **函數**

  void DrvADC_Disable(void)

- **函數功能**

  關閉ADC功能；設置寄存器0x41100[0]=0

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvADC.h

- **函數返回值**

  無

- **函數用法**

  /* 關閉ADC功能 */

  DrvADC_Disable();

6.3.23 DrvADC_GetConversionData

- **函數**

  int DrvADC_GetConversionData (void);

- **函數功能**

  讀取A/D 轉換值，資料是帶符號的.讀取寄存器0x41108[31 :0].

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvADC.h

- **函數返回值**

  返回A/D轉換值。.

- **函數用法**

  /*讀取ADC 轉換值*/

  int adc_data ;

  adc_data=DrvADC_GetConversionDate();

7 SPI32 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 功能的開啟控制

--SPI 的工作模式及參數的配置

--SPI 的中斷向量的控制

--SPI 狀態控制

--SPI 的資料的收發

序號	函數名稱	功能描述
01	DrvSPI32_Open	開啟SPI 功能
02	DrvSPI32_Close	關閉SPI 功能
03	DrvSPI32_IsBusy	查詢SPI 匯流排繁忙狀態
04	DrvSPI32_SetClockFreq	配置SPI 時鐘頻率
05	DrvSPI32_IsRxBufferFull	查詢接收暫存器狀態位元
06	DrvSPI32_IsTxBufferFull	查詢發送暫存器狀態位元
07	DrvSPI32_EnableRxInt	開啟SPI 接收中斷功能
08	DrvSPI32_EnableTxInt	開啟SPI 發送中斷功能
09	DrvSPI32_DisableRxInt	關閉SPI 接收中斷功能
10	DrvSPI32_DisableTxInt	關閉SPI 發送中斷功能
11	DrvSPI32_GetRxIntFlag	讀取SPI 接收中斷標誌位元
12	DrvSPI32_GetTxIntFlag	讀取SPI 發送中斷標誌位元
13	DrvSPI32_ClrIntRxFlag	清除SPI 接收中斷標誌位元
14	DrvSPI32_ClrIntTxFlag	清除SPI 發送中斷標誌位元
15	DrvSPI32_Read	讀取SPI 接收暫存器的資料
16	DrvSPI32_Write	寫入資料至發送暫存器
17	DrvSPI32_Enable	開啟SPI 功能
18	DrvSPI32_BitLength	設置數據的長度
19	DrvSPI32_GetDCFlag	讀取SPI 資料丟失狀態位元
20	DrvSPI32_IsABFlag	讀取SPI 接收到的資料長度小的狀態
21	DrvSPI32_IsOVFlag	檢查SPI 接收到資料是否過長
22	DrvSPI32_IsRxFlag	檢查接收暫存器資料是否更新

23	DrvSPI32_SetEndian	設定資料發送是從MSB或LSB開始發送
24	DrvSPI32_SetCSO	SPI 時序源極性選擇位設置
25	DrvSPI32_DisableIO	關閉IO口複用為SPI通訊口的功能
26	DrvSPI32_EnableIO	開啟及選擇IO口複用為SPI通訊口功能

7.2 內部定義常量

E_DRVSPI_MODE

識別字	數值	函數功能
E_DRVSPI_MASTER1	0	4-wire 主動模式
E_DRVSPI_MASTER2	1	3-wire 主動模式
E_DRVSPI_MASTER3	2	TI 方式主動模式
E_DRVSPI_SLAVE1	3	4-wire 被動模式
E_DRVSPI_SLAVE2	4	3-wire 被動模式
E_DRVSPI_SLAVE3	5	TI 方式被動模式

E_DRVSPI_TRANS_TYPE

識別字	數值	函數功能
E_DRVSPI_TYPE0	0	SPI 發送模式0
E_DRVSPI_TYPE1	1	SPI 發送模式1
E_DRVSPI_TYPE2	2	SPI 發送模式2
E_DRVSPI_TYPE3	3	SPI 發送模式3

E_DRVSPI_ENDIAN

識別字	數值	函數功能
E_DRVSPI_LSB_FIRST	1	從低8bit(LSB)開始發送
E_DRVSPI_MSB_FIRST	0	從高8bit(MSB)開始發送

E_DRVSPI_CS

識別字	數值	函數功能
E_DRVSPI_CSLow	0	CSO low
E_DRVSPI_CSHigh	1	CSO high

7.3 函數說明

7.3.1 DrvSPI32_Open

- 函數

```
unsigned int DrvSPI32_Open(
E_DRVSPI_MODE uMode,
E_DRVSPI_TRANS_TYPE uType,
uOuputPin,
uClkDiv
  );
```

- 函數功能

函數開啟SPI功能，設置SPI工作是主動模式或者被動模式，設置SPI匯流排時序及通訊IO;
設置寄存器
0x4030C[2:0],0x4030C[3]=1b, 0x40844[4]=1b, 0x40844[7:5],0x40F00[3:0],0x40f04[16:17]
uMode : 0x40f00[0]=1b, 0x40f00[1]=xb, 0x40f04[16:17]=0xb. uMode : 0~5
uType : 0x40f00[3:2]=xxb. uType : 0~3
uOuputPin : 0x40844[4]=1b, 0x40844[7:5]=xxxb. uOuputPin : 0~7
uClkDiv : 0x4030C[2:0]=xxxb, 0x4030C[3]=1b. uClkDiv : 0~7

- 輸入參數

uMode [in] ：工作模式設置，設置範圍0~5
0： 4-wire通訊介面的主動模式.
1： 3-wire通訊介面的主動模式.
2： TI 模式介面的主動模式.
3： 4-wire通訊介面的被動模式.
4： 3-wire通訊介面的被動模式.
5： TI 模式介面的被動模式.
uType [in]： 傳輸類型，如通訊匯流排時序，設置範圍是0~3.
0: 抓取資料在第一個時鐘沿，時鐘源低電平為空閒狀態.(CPHA=0 CPOL=0)
1: 抓取資料在第一個時鐘沿，時鐘源高電平為空閒狀態.(CPHA=0 CPOL=1)
2: 抓取資料在第二個時鐘沿，時鐘源低電平為空閒狀態.(CPHA=1 CPOL=0)
3: 抓取資料在第二個時鐘沿，時鐘源高電平為空閒狀態.(CPHA=1 CPOL=1)
uOuputPin[in]：SPI通訊IO 口設置, 設置範圍是 ：0~3
0 : Port1.0 =CS, Port1.1 =CK, Port1.2 = DI, Port1.3 =DO
1 : Port1.4 =CS, Port1.5 =CK, Port1.6 = DI, Port1.7 =DO
2 : Port2.0 =CS, Port2.1 =CK, Port2.2 = DI, Port2.3 =DO
3 : Port2.4 =CS, Port2.5 =CK, Port2.6 = DI, Port2.7 =DO
uClkDiv[in]：SPI時鐘源分頻器設置, 設置範圍是 ：0~7

0 : ÷1

1 : ÷2

2 : ÷4

3 : ÷8

4 : ÷32

5 : ÷128

6 : ÷512

7 : ÷2048

- **包含標頭檔**

Peripheral_lib/DrvSPI32.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/*使能SPI主動模式，時鐘源分頻CLOCK/512，設置傳送類型1，通訊IO 口設置: Port1.4 =CS, Port1.5 =CK, Port1.6 = DI, Port1.7 =DO*/

DrvSPI32_Open(E_DRVSPI_MASTER1, E_DRVSPI_TYPE1, 1,6);

7.3.2 DrvSPI32_Close

- **函數**

void DrvSPI32_Close (void);

- **函數功能**

關閉SPI功能，關閉SPI的時鐘、IO等功能；

設置寄存器0x40F00[0]=0, 0x4030C[3]=0,0x40844[4]=0 .

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvSPI32.h

- **函數返回值**

無

- **函數用法**

/* 關閉SPI */

DrvSPI32_Close();

7.3.3 DrvSPI32_IsBusy

- **函數**

unsigned int DrvSPI32_IsBusy(void);

- **函數功能**

  查詢SPI匯流排上是否繁忙狀態.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  1：SPI匯流排繁忙

  0：SPI匯流排空閒.

- **函數用法**

```
/* 檢查匯流排繁忙狀態*/
unsigned char   flag;
flag=DrvSPI32_IsBusy (); //read 0x40f00[19]
```

7.3.4 DrvSPI32_SetClockFreq

- **函數**

  unsigned int DrvSPI32_SetClockFreq(unsigned int uCPUDV, unsigned int uTMRDV);

- **函數功能**

  配置MCU時鐘分頻器及SPI時鐘分頻器且使能SPI時鐘源，主動模式下輸出時鐘頻率可程式設計設置;

  設置寄存器0x40308[1], 0x4030C[2:0].

- **輸入參數**

  eCPUDV[in] : MCU 時鐘分頻器設置. 設定值範圍 ：0~1

  0 : ÷1

  1 : ÷2

  eTMRDV[in] : SPI 時鐘分頻器設置. 設定值範圍 ：0~7

  0 : ÷1

  1 : ÷2

  2 : ÷4

  3 : ÷8

  4 : ÷32

  5 : ÷128

  6 : ÷512

  7 : ÷2048

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  無

- **函數用法**

```
/* SPI  頻率是APCK/512 */
DrvSPI32_SetClockFreq(1, 6);
```

7.3.5 DrvSPI32_IsRxBufferFull

- **函數**

  unsigned int DrvSPI32_IsRxBufferFull(void);

- **函數功能**

  查詢接收暫存器滿狀態位元(RXBF)（只用於資料接收）; 設置寄存器0x40F00[16]。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  1: 接收已完成，接收暫存器已滿.

  0: 接收未完成，接收暫存器為空.

- **函數用法**

```
/*  查詢接收暫存器狀態*/
unsigned char    flag;
flag = DrvSPI32_IsRxBufferFull() ;
```

7.3.6 DrvSPI32_IsTxBufferFull

- **函數**

  unsigned int DrvSPI32_IsTxBufferFull(void);

- **函數功能**

  查詢發送暫存器滿的狀態(TXBF)（只用於資料發送）設置寄存器0x40F00[17]。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  1: 發送未完成，發送暫存器還有資料.

  0: 發送已完成，發送暫存器為空.

- **函數用法**

  /* 查詢發送暫存器的狀態 */

  unsigned char flag; flag =DrvSPI32_IsTxBufferFull();

7.3.7 DrvSPI32_EnableRxInt

- **函數**

  void DrvSPI32_EnableRxInt(void);

- **函數功能**

  使能SPI接收中斷，屬於中斷向量HW0；設置寄存器0x40000[16]=1.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  無

- **函數用法**

  /* SPI 接收中斷使能*/

  DrvSPI32_EnableRxInt();

7.3.8 DrvSPI32_EnableTxInt

- **函數**

  void DrvSPI32_EnableTxInt(void);

- **函數功能**

  使能SPI 發送中斷，屬於中斷向量HW0；設置寄存器0x40000[17]=1。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  無

- **函數用法**

  /*SPI 發送中斷使能*/

  DrvSPI32_EnableTxInt();

7.3.9 DrvSPI32_DisableRxInt

- **函數**

  void DrvSPI32_DisableRxInt(void);

- **函數功能**

  關閉SPI 接收中斷功能，設置寄存器0x40000[16]=0 ..

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  無

- **函數用法**

  /*關閉SPI 接收中斷 */

  DrvSPI32_DisableRxInt();

7.3.10 DrvSPI32_DisableTxInt

- **函數**

  void DrvSPI32_DisableTxInt(void);

- **函數功能**

  關閉SPI 發送中斷，設置寄存器0x40000[17]=0 .

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  無

- **函數用法**

  /* 關閉SPI 發送中斷 */

  DrvSPI32_DisableTxInt();

7.3.11 DrvSPI32_GetRxIntFlag

- **函數**

  unsigned int DrvSPI32_GetRxIntFlag ();

- **函數功能**

  讀取SPI 接收中斷要求標誌位元(SRXIF) ；讀取寄存器0x40000[0]。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  1: 中斷標誌位元為1，有中斷要求

  0: 中斷標誌位元為0，無中斷要求

- **函數用法**

  /*讀取SPI接收中斷要求標誌位元*/

  unsigned char flag；flag=DrvSPI32_GetRxIntFlag();

7.3.12 DrvSPI32_GetTxIntFlag

- **函數**

  unsigned int DrvSPI32_GetTxIntFlag ();

- **函數功能**

  讀取SPI 發送中斷要求標誌位元(STXIF)；讀取寄存器0x40000[1] 。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  1: 中斷標誌位元為1，有中斷要求

  0: 中斷標誌位元為0，無中斷要求

- **函數用法**

  /* 讀取SPI 發送中斷要求標誌位元.*/

  unsigned char flag；flag=DrvSPI32_GetTxIntFlag();

7.3.13 DrvSPI32_ClrIntRxFlag

- **函數**

  void DrvSPI32_ClrIntRxFlag ();

- **函數功能**

  清除SPI 接收中斷要求標誌位元(SRXIF)；設置寄存器0x40000[0]=0.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  無

- **函數用法**

  /*清除SPI 接收中斷要求標誌位元*/

  DrvSPI32_ClrIntRxFlag();

7.3.14 DrvSPI32_ClrIntTxFlag

- **函數**

  void DrvSPI32_ClrIntTxFlag ();

- **函數功能**

  清除SPI 發送中斷要求標誌位元（STXIF）,設置寄存器0x40000[1]=0 .

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  無

- **函數用法**

  /* 清除SPI發送中斷要求標誌位元*/

  DrvSPI32_ClrIntTxFlag();

7.3.15 DrvSPI32_Read

- **函數**

  unsigned int DrvSPI32_Read();

- **函數功能**

  讀取SPI 資料接收暫存器；讀取寄存器0x40F08[31:0] 。 .

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  返回值是SPI 接收暫存器的值

- **函數用法**

  /* 讀取接收暫存器的值 */

  /*資料接收方式LSB First 8bit 數據*/

  unsigned int data；data=DrvSPI32_Read()>>24;

  /*資料接收方式MSB 8bit 數據*/

  unsigned int data；data=DrvSPI32_Read();

7.3.16 DrvSPI32_Write

- **函數**

  void DrvSPI32_Write (unsigned int uData);

- **函數功能**

  寫入待發送資料至發送暫存器並發送；寫入寄存器0x40FC[31:0].

- **輸入參數**

  uData [in]： 待發送資料：0~0xFFFFFFFF。

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  無

- **函數用法**

  /*資料傳送方式MSB First 8bit 發送0x55*/

  DrvSPI32_Write(0x55<<24);

  /*資料傳送方式LSB First 8bit 發送0x55*/

  DrvSPI32_Write(0x55);

7.3.17 DrvSPI32_Enable

- **函數**

  void DrvSPI32_Enable (void);

- **函數功能**

  使能SPI 功能；設置寄存器0x40F00[0]=1 .

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  無

- **函數用法**

  /* 開啟SPI */

  DrvSPI32_Enable();

7.3.18 DrvSPI32_BitLength

- **函數**

  void DrvSPI32_BitLength (unsigned int uData);

- **函數功能**

  設置SPI 發送資料的長度；設置寄存器0x40F04[4:0] .

- **輸入參數**

uData[in]： 設置SPI 發送資料的長度，設定值範圍是 ：0x04~0x20

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  無

- **函數用法**

  /* 設定SPI發送資料的長度為8bit*/

  DrvSPI32_BitLength(8);

7.3.19 DrvSPI32_GetDCFlag

- **函數**

  unsigned int DrvSPI32_GetDCFlag(void);

- **函數功能**

  讀取SPI 資料丟失狀態位元(DCF) ，讀取寄存器0x40F00[18]。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  0: 正常.

  1: 接收暫存器已滿，讀取接收暫存器可以清零該位.

- **函數用法**

  /* 讀取資料丟失狀態位元 DCF */

  unsigned char flag；flag=DrvSPI32_GetDCFlag();

7.3.20 DrvSPI32_IsABFlag

- **函數**

  unsigned int DrvSPI32_IsABFlag(void);

- **函數功能**

  讀取SPI 接收到的資料長度是否缺少的狀態位元(ABF)；讀取寄存器0x40F00[20]的值 .

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  0: 正常.

  1: SPI接收到的資料長度比設置的資料長度少

- **函數用法**

  /* 讀取資料長度標誌位元ABF */

  unsigned char flag；flag=DrvSPI32_IsABFlag();

7.3.21 DrvSPI32_IsOVFlag

- **函數**

  unsigned int DrvSPI32_IsOVFlag(void);

- **函數功能**

  讀取接收到的資料長度是否比設定值長的狀態位元(VOF)，讀取寄存器0x40F00[21]的值 .

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  0: 正常

  1: SPI接收到的資料長度比設定的資料長度大

- **函數用法**

  /*讀取接收到資料長度過長標誌位元OVF*/

  unsigned char flag；flag=DrvSPI32_IsOVFlag();

7.3.22 DrvSPI32_IsRxFlag

- **函數**

  unsigned int DrvSPI32_IsRxFlag(void);

- **函數功能**

  讀取SPI 資料接收暫存器的資料更新標誌位元(RXF)，確定是否讀取接收暫存器；讀取寄存器0x40F00[22] ..

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  0: 正常.

  1: SPI 接收暫存器有資料在更新，不能讀取接收暫存器。

- **函數用法**

  /*讀取SPI 接收暫存器資料更新標誌位元RxF */

  unsigned char flag；flag=DrvSPI32_IsRxFlag();

7.3.23 DrvSPI32_SetEndian

- **函數**

  void DrvSPI32_SetEndian(E_DRVSPI_ENDIAN eEndian);

- **函數功能**

  設置SPI 是從高8位還是低8位元資料開始發送；設置寄存器0x40F04[18] .

- **輸入參數**

  eEndian [in]： 輸入範圍 ：0~1

  1：低8位(LSB) 開始發送

  0：高8位(MSB) 開始發送

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  無

- **函數用法**

  /*設置SPI 從低 8位元資料開始發送 */

  DrvSPI32_ SetEndian(E_DRVSPI_LSB_FIRST);

7.3.24 DrvSPI32_SetCSO

- **函數**

  void DrvSPI32_SetCSO(E_DRVSPI_CS eCS);

- **函數功能**

  SPI 時序源極性選擇位設置，設置寄存器0x40F04[20] .

  注意：該函數是將舊的函數DrvSPI32_SetCS(E_DRVSPI_CS eCS);的名稱修改，但是功能新舊函數是一致的；新函數名稱明確指出函數是操作CSO位元。

  舊函數DrvSPI32_SetCS(E_DRVSPI_CS eCS);依然運行有效。

- **輸入參數**

  eCS [in]：輸入範圍 ：0~1

  0：時序源低電平有效（CSO low）

  1：時序源高電平有效（CSO high）

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  無

- **函數用法**

```
/* 設置低電平有效 */
DrvSPI32_SetCSO(E_DRVSPI_CSLow);
```

7.3.25 DrvSPI32_DisableIO

- **函數**

  void DrvSPI32_DisableIO(void);

- **函數功能**

  關閉 SPI 通訊口，設置寄存器0x40844[4]=0; .

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  無

- **函數用法**

```
/*關閉SPI 通訊口 */
DrvSPI32_DisableIO();
```

7.3.26 DrvSPI32_EnableIO

- **函數**

  unsigned char DrvSPI32_EnableIO(uint32_t uOuputPin);

- **函數功能**

  開啟SPI 通訊口，設置寄存器0x40844[7:5] / 0x40844[4]=1; .

- **輸入參數**

  uOuputPin[in]：SPI通訊IO 口設置. 輸入範圍 : 0~3

  0 : Port1.0 =CS, Port1.1 =CK, Port1.2 = DI, Port1.3 =DO

  1 : Port1.4 =CS, Port1.5 =CK, Port1.6 = DI, Port1.7 =DO

  2 : Port2.0 =CS, Port2.1 =CK, Port2.2 = DI, Port2.3 =DO

  3 : Port2.4 =CS, Port2.5 =CK, Port2.6 = DI, Port2.7 =DO

- **包含標頭檔**

  Peripheral_lib/DrvSPI32.h

- **函數返回值**

  0：設置成功

  1：設置失敗

- **函數用法**

```
/*開啟SPI 通訊口並選擇PT1.0~PT1.3*/
DrvSPI32_EnableIO(0);
```

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

序號	函數名稱	功能描述
01	DrvUART_Open	開啟UART 功能並配置相關參數
02	DrvUART_Close	關閉UART 功能
03	DrvUART_EnableInt	使能UART 中斷向量
04	DrvUART_GetTxFlag	讀取TX中斷標誌位元
05	DrvUART_GetRxFlag	讀取RX中斷標誌位元
06	DrvUART_ClrTxFlag	清除TX中斷標誌位元
07	DrvUART_ClrRxFlag	清除RX中斷標誌位元
08	DrvUART_Read	接收到8位元資料
09	DrvUART_Read9Bit	接收到9位元資料
10	DrvUART_Write	寫入資料並發送
11	DrvUART_EnableWakeUp	開啟喚醒功能
12	DrvUART_GetPERR	讀取UART校驗錯誤狀態位元
13	DrvUART_GetFERR	讀取UART幀錯誤狀態為
14	DrvUART_GetOERR	讀取UART溢出錯誤狀態位元
15	DrvUART_GetABDOVF	讀取UART自動串列傳輸速率翻轉狀態檢測位元
16	DrvUART_Enable_AutoBaudrate	開啟UART自動串列傳輸速率功能
17	DrvUART_Disable_AutoBaudrate	關閉UART自動串列傳輸速率功能
18	DrvUART_CheckTRMT	讀取UART發送寄存器狀態位元
19	DrvUART_ClkEnable	開啟UART時鐘源並設置時鐘源
20	DrvUART_ClkDisable	關閉UART時鐘源
21	DrvUART_Enable	開啟UART
22	DrvUART_ConfigIO	設置IO複用為UART通訊口並選擇IO口

8.2 內部定義常量

E_DATABITS_SETTINGS

識別字	數值	函數功能
DRVUART_DATABITS_8	0x0	數據長度為8 bits
DRVUART_DATABITS_9	0x1	數據長度為9 bits.

E_PARITY_SETTINGS

識別字	數值	函數功能
DRVUART_PARITY_NONE	0x0	無同位校驗
DRVUART_PARITY_ODD	0x1	使能奇同位校驗
DRVUART_PARITY_EVEN	0x2	使能偶同位校驗

E_BAUD_RATE_SETTINGS

識別字	數值	函數功能
B1200	0x0	Baud rate=1200
B2400	0x1	Baud rate=2400
B4800	0x2	Baud rate=4800
B9600	0x3	Baud rate=9600
B14400	0x4	Baud rate=14400
B19200	0x5	Baud rate=19200
B38400	0x6	Baud rate=38400

E_UART_ERROR_MESSAGE

識別字	數值	函數功能
E_UART_ERR_CLOCK	0x2	時鐘源輸入錯誤
E_UART_ERR_BAUDRATE	0x3	串列傳輸速率輸入錯誤
E_UART_ERR_PARITY	0x4	校驗方式輸入錯誤
E_UART_ERR_DATABIT	0x5	資料長度輸入錯誤
E_UART_ERR_OUTPIN	0x6	輸出 IO 設置輸入錯誤

8.3 函數說明

8.3.1 DrvUART_Open

- **函數**

```
unsigned int DrvUART_Open (
                        unsigned int    uClock
E_RAUD_RATE_SETTINGS uBaudRate ,
E_PARITY_SETTINGS uParity,
E_DATABITS_SETTINGS uDataBits,
uOuputPin
);
```

- **函數功能**

設置UART的工作頻率源（除了晶震源為外部晶震(HSXT)或內部晶震(HSRC), UART除頻設置也會影響到實際UART的工作頻率源）並根據寫入的串列傳輸速率值自動計算出串列傳輸速率寄存器0x40E0C[4:0] / 0x40E10[7:0] 的值；設置UART的資料校驗模式、資料的位元數及TX/RX的通訊用IO口。

設置寄存器0x40E00[7]=1, 0x40E00[6]=1, 0x40E00[5] / 0x40E00[3]；

寄存器0x40E0C[3:0]/0x40E10[7:0]；設置IO口寄存器0x40844[3:0].

- **輸入參數**

uClock[in]： 設置UART工作頻率源, 輸入值為URCK 的頻率大小, URCK是由高速晶振頻率(外部高速HSXT或者內部高速頻率HSRC) 經過UACD[3:0]分頻得到, 若UACD=1，則URCK=HSXT(或HSRC), 若UACD=2，則URCK=HSXT/2(或HSRC/2) 依此類推, 以kHZ作為單位計算; 輸入範圍 ：1000~20000

uBaudRate[in] : UART通訊資料串列傳輸速率

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：

0： 無校驗

1： 偶校驗

2： 奇數同位檢查

uDataBits[in]： 資料位元數設置，設定範圍是：

0 : 8 bit 數據.

1 : 9 bit 數據.

uOuputPin[in]：通訊線TX/RX IO口設置

0 : Port 1.0 =TX, Port 1.1 =RX

1 : Port 1.2 =TX, Port 1.3 =RX

2 : Port 1.4 =TX, Port 1.5 =RX

3 : Port 1.6 =TX, Port 1.7 =RX

4 : Port 2.0 =TX, Port 2.1 =RX

5 : Port 2.2 =TX, Port 2.3 =RX

6 : Port 2.4 =TX, Port 2.5 =RX

7 : Port 2.6 =TX, Port 2.7 =RX

- **包含標頭檔**

Peripheral_lib/DrvUART.h

- **函數返回值**

0：設置成功.

2：時鐘設置錯誤

3：串列傳輸速率設置錯誤

4：校驗位元設置錯誤

5：資料長度設置錯誤

6：通訊IO設置錯誤

- **函數用法**

/* 設置UART baud rate 115200bps, 8 位元數據 ，且無校驗，通訊口為PT1.4/PT1.5*/

DrvUART_Open(4000,115200, DRVUART_PARITY_NONE ,DRVUART_DATABITS_8,2);

Note： 因為 UART 工作頻率源為 4MHz，所以輸入頻率為 4000，單位為 kHz.

8.3.2 DrvUART_Close

- **函數**

void DrvUART_Close (void);

- **函數功能**

關閉UART 功能；清零寄存器0x40E00[7]=0.;

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvUART.h

- **函數返回值**

無

- **函數用法**

/* 關閉UART */

DrvUART_Close();

8.3.3 DrvUART_EnableInt

- **函數**

unsigned int DrvUART_EnableInt(unsigned int uTXIE, unsigned int uRXIE);

- **函數功能**

UART的發送(TX)或接收(RX)中斷向量控制. UART屬於中斷向量HW0;設置寄存器0x40000[19:18]。

- **輸入參數**

uTXIE [in] : UART 發送(TX) 中斷控制

0： 關閉中斷

1： 使能中斷

uRXIE [in] : UART 接收(RX) 中斷控制

0： 關閉中斷

1： 使能中斷

- **包含標頭檔**

Peripheral_lib/DrvUART.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/* 使能發送及接收中斷 */

DrvUART_EnableInt(1,1);

8.3.4 DrvUART_GetTxFlag

- **函數**

unsigned int DrvUART_GetTxFlag (void);

- **函數功能**

讀取發送中斷標誌位元(UTXIF)值，讀取寄存器0x40000[3]的值。

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvUART.h

- **函數返回值**

1: 有中斷產生

0: 無中斷產生

- **函數用法**

/* 讀取發送中斷標誌位元. */

DrvUART_GetTxFlag();

8.3.5 DrvUART_GetRxFlag

- **函數**

unsigned int DrvUART_GetRxFlag (void);

- **函數功能**

讀取接收中斷標誌位元URXIF值，讀取寄存器0x40000[2]的值。

- **輸入參數**

無

- **包含標頭檔**

  Peripheral_lib/DrvUART.h

- **函數返回值**

  1：有中斷要求

  0：無中斷要求

- **函數用法**

  /* 讀取接收中斷標誌位元. */

  unsigned char flag；flag=DrvUART_GetRxFlag();

8.3.6 DrvUART_ClrTxFlag

- **函數**

  void DrvUART_ClrTxFlag (void);

- **函數功能**

  清除發送中斷標誌位元UTXIF值，清零寄存器0x40000[3]

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvUART.h

- **函數返回值**

  無

- **函數用法**

  /* 清除發送中斷標誌位元. */

  DrvUART_ClrTxFlag();

8.3.7 DrvUART_ClrRxFlag

- **函數**

  void DrvUART_ClrRxFlag (void);

- **函數功能**

  清除接收中斷標誌位元URXIF值，清零寄存器0x40000[2]

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvUART.h

- **函數返回值**

  無

- **函數用法**

  /* 清除接收中斷標誌位元. */

  DrvUART_ClrRxFlag();

8.3.8 DrvUART_Read

- **函數**

  unsigned int DrvUART_Read(void);

- **函數功能**

  UART接收8位元資料，讀取寄存器0x40E18[7:0]的值

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvUART.h

- **函數返回值**

  返回接收暫存器的值.

- **函數用法**

  /* UART接收8位元資料. */

  unsined int rx_data；rx_data=DrvUART_Read();

8.3.9 DrvUART_Read9Bit

- **函數**

  unsigned int DrvUART_Read9Bit(void);

- **函數功能**

  接收9位元資料

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvUART.h

- **函數返回值**

  接收到的9Bits資料

- **函數用法**

  /* 接收9位元資料*/

  unsigned int data; data=DrvUART_Read9Bit();

8.3.10 DrvUART_Write

- **函數**

  void DrvUART_Write(unsigned int uData);

- **函數功能**

  寫入數值至發送暫存器(TXREG)並等待發送，寫入待發送的值至寄存器0x40E14[7:0]。

- **輸入參數**

  uData [in]

  待發送的8bit資料

- **包含標頭檔**

  Peripheral_lib/DrvUART.h

- **函數返回值**

  無

- **函數用法**

  /*發送0x55 */

  DrvUART_Write(0x55);

8.3.11 DrvUART_EnableWakeUp

- **函數**

  void DrvUART_EnableWakeUp(void);

- **函數功能**

  使能UART的喚醒功能，同樣啟動接收喚醒功能只要接收中斷打開；

  設置寄存器0x40E00[0]=1。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvUART.h

- **函數返回值**

  無

- **函數用法**

  /* 使能UART喚醒功能 */

  DrvUART_EnableWakeUp();

8.3.12 DrvUART_GetPERR

- **函數**

  unsigned int DrvUART_GetPERR(void);

- **函數功能**

  讀取校驗錯誤標誌位元(PERR)，讀取寄存器0x40E04[5]的值。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvUART.h

- **函數返回值**

1： 有校驗錯誤

0： 無校驗錯誤

- **函數用法**

```
/* 讀取校驗錯誤標誌位元. */
unsigned char flag; flag=DrvUART_GetPERR();
```

8.3.13 DrvUART_GetFERR

- **函數**

```
unsigned int DrvUART_GetFERR(void);
```

- **函數功能**

讀取幀錯誤標誌位元(FERR)，讀取寄存器0x40E04[4]的值。

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvUART.h

- **函數返回值**

1： 有幀錯誤

0： 無幀錯誤

- **函數用法**

```
/* 讀取幀錯誤標誌位元. */
unsigned char flag ; flag=DrvUART_GetFERR();
```

8.3.14 DrvUART_GetOERR

- **函數**

```
unsigned int DrvUART_GetOERR(void);
```

- **函數功能**

讀取溢出錯誤標誌位元(OERR)，讀取寄存器0x40E04[3]的值。

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvUART.h

- **函數返回值**

1： 有溢出錯誤

0： 無溢出錯誤

- **函數用法**

```
/* 讀取溢出錯誤標誌位元. */
unsigned char flag ; flag=DrvUART_GetOERR();
```

8.3.15 DrvUART_GetABDOVF

- **函數**

  unsigned int DrvUART_GetABDOVF(void);

- **函數功能**

  讀取自動串列傳輸速率發生器翻轉狀態檢測標誌位元(ABDOVF)，讀取寄存器0x40E04[0]值。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvUART.h

- **函數返回值**

  1： 在自動串列傳輸速率檢測模式下串列傳輸速率發生器發生翻轉

  0： 沒有串列傳輸速率發生器發生翻轉

- **函數用法**

  /* 讀取串列傳輸速率發生器翻轉標誌位元ABDOVF. */

  unsigned char flag ; flag=DrvUART_GetABDOVF();

8.3.16 DrvUART_Enable_AutoBaudrate

- **函數**

  void DrvUART_Enable_AutoBaudrate ();

- **函數功能**

  使能UART 自動串列傳輸速率功能，設置寄存器0x40E08[0]=1.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvUART.h

- **函數返回值**

  無

- **函數用法**

  /* 使能UART自動串列傳輸速率功能 */

  DrvUART_Enable_AutoBaudrate();

8.3.17 DrvUART_Disable_AutoBaudrate

- **函數**

  void DrvUART_Disable_AutoBaudrate ();

- **函數功能**

  關閉UART 自動串列傳輸速率功能，設置寄存器0x40E08[0]=0。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvUART.h

- **函數返回值**

  無

- **函數用法**

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

8.3.18 DrvUART_CheckTRMT

- **函數**

  Unsigned int DrvUART_CheckTRMT(void)

- **函數功能**

  讀取UART發送狀態位元(TRMT)，讀取寄存器0x40E04[1]值

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvUART.h

- **函數返回值**

  返回發送狀態位元TRMT的值;

- **函數用法**

  /* 讀取發送狀態位元值並實現查詢方式發送資料*/
  DrvUART_Write(data) ;
  While(DrvUART_CheckTRMT()) ;//等待TRMT=0
  While(!DrvUART_CheckTRMT()) ;//等待TRMT=1

8.3.19 DrvUART_ClkEnable

- **函數**

  unsigned int DrvUART_ClkEnable(unsigned int uclk,unsigned int uprescale) ;

- **函數功能**

  使能UART的時鐘源並選擇時鐘源及設置時鐘源的分頻值

  設置寄存器0x40308[21 :16] 。

- **輸入參數**

  uclk[in] : EUART 時鐘源設置

  0： 外部晶震高速時鐘

  1： 內部晶震高速時鐘

  uprescale[in]：時鐘源分頻器

0000	EUART CLOCK SOURCE/1	1000	EUART CLOCK SOURCE/256
0001	EUART CLOCK SOURCE/2	1001	EUART CLOCK SOURCE/512
0010	EUART CLOCK SOURCE/4	1010	EUART CLOCK SOURCE/1024
0011	EUART CLOCK SOURCE/8	1011	EUART CLOCK SOURCE/2048
0100	EUART CLOCK SOURCE/16	1100	EUART CLOCK SOURCE/4096
0101	EUART CLOCK SOURCE/32	1101	EUART CLOCK SOURCE/8192
0110	EUART CLOCK SOURCE/64	1110	EUART CLOCK SOURCE/16384
0111	EUART CLOCK SOURCE/128	1111	EUART CLOCK SOURCE/32768

- **包含標頭檔**

  Peripheral_lib/DrvUART.h

- **函數返回值**

  0 ：設置成功

  1 ：設置失敗

- **函數用法**

```
/* 設置UART時鐘源為外部時鐘且分頻clk/1 */
DrvUART_ClkEnable(0,0);
```

8.3.20 DrvUART_ClkDisable

- **函數**

  Void DrvUART_ClkDisable(void) ;

- **函數功能**

  關閉UART時鐘源,設置寄存器0x40308[20]=0。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvUART.h

- **函數返回值**

無

- **函數用法**

/*關閉UART 時鐘源*/

DrvUART_ClkDisable();

8.3.21 DrvUART_Enable

- **函數**

Void DrvUART_Enable(void) ;

- **函數功能**

使能UART功能 ,設置寄存器0x40E00[7:6]=11b

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvUART.h

- **函數返回值**

無

- **函數用法**

/*使能UART功能*/

DrvUART_Enable();

8.3.22 DrvUART_ConfigIO

- **函數**

unsigned char DrvUART_ConfigIO(unsigned char ioen,unsigned int uOuputPin) ;

- **函數功能**

設置IO口複用為UART通訊口，及選擇IO口 ,設置寄存器0x40844[3 :0]。

- **輸入參數**

ioen[in] :IO 口複用功能使能控制

0：關閉IO 複用功能

1：開啟IO 複用功能

uoutputPin[in] :選擇通訊IO 口

0 : Port 1.0 =TX, Port 1.1 =RX

1 : Port 1.2 =TX, Port 1.3 =RX

2 : Port 1.4 =TX, Port 1.5 =RX

3 : Port 1.6 =TX, Port 1.7 =RX

4 : Port 2.0 =TX, Port 2.1 =RX

5 : Port 2.2 =TX, Port 2.3 =RX

6 : Port 2.4 =TX, Port 2.5 =RX

7 : Port 2.6 =TX, Port 2.7 =RX

- **包含標頭檔**

  Peripheral_lib/DrvUART.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

```
/*開啟IO複用為UART通訊口，並選擇PT2.0/PT2.1*/
DrvUART_ConfigIO(1,4);
```

9 多功能比較器 CMP

9.1 函數簡介

該部分函數描述CMP 功能的控制，包含：

--CMP 模組功能的啟動與關閉

--CMP 配置控制

--CMP 中斷向量控制

--CMP 的 NON-OVER LAP 功能控制

序號	函數名稱	功能描述
01	DrvCMP_Open	開啟比較器CMP
02	DrvCMP_Close	關閉比較器CMP
03	DrvCMP_Enable	開啟比較器CMP
04	DrvCMP_PInput	設置CMP的正端輸入源
05	DrvCMP_NInput	設置CMP的負端輸入源
06	DrvCMP_InputSwitch	CMP的輸入短路開關控制
07	DrvCMP_OutputFilter	CMP數位濾波控制
08	DrvCMP_OutputPinEnable	開啟CMP數位輸出功能
09	DrvCMP_OutputPinDisable	關閉CMP數位輸出功能
10	DrvCMP_OutputInverse	設置CMP數位輸出反相控制
11	DrvCMP_EnableInt	開啟CMP中斷
12	DrvCMP_DisableInt	關閉CMP中斷
13	DrvCMP_ReadIntFlag	讀取CMP中斷標誌位元
14	DrvCMP_ClearIntFlag	清除CMP中斷標誌位元
15	DrvCMP_Input	比較器CMP信號輸入正負端設置
16	DrvCMP_RLO_Ctrl	CMP內部電阻分壓(RLO)網路設置
17	DrvCMP_RLO_refv	CMP內部電阻分壓參考電壓設置
18	DrvCMP_EnableNonOverlap	開啟CMP的non-overlap功能
19	DrvCMP_DisableNonOverlap	關閉CMP的non-overlap功能
20	DrvCMP_ReadData	讀取CMP數位輸出狀態

9.2 內部定義常量

E_NON_OVERLAP_PIN

識別字	數值	函數功能
E_CL1	0x0	設置PT1.2 作為正端輸入源
E_CL2	0x1	設置PT1.3 作為正端輸入源
E_CL3	0x2	設置PT3.1 作為正端輸入源
E_CL4	0x3	設置PT3.2 作為正端輸入源

9.3 函數說明

9.3.1 DrvCMP_Open

- **函數**

  unsigned int DrvCMP_Open (uPowerMode , uCPDF, uCPOR)

- **函數功能**

  使能比較器功能，設置低功耗模式，設置比較器2us 延時濾波功能控制及比較器數位輸出相位控制。

  設置寄存器0x41804[7:6] / 0x41804[1:0]

- **輸入參數**

  uPowerMode[in]：比較器功耗控制：.

  0：低功耗模式

  1：正常模式

  uCPDF [in]：比較器輸出濾波控制

  0：不經過delitch濾波

  1：經過2us延時濾波

  uCPOR [in]：比較器數位輸出相位控制

  0：正常輸出

  1：反相輸出

- **包含標頭檔**

  Peripheral_lib/DrvCMP.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /*使能比較器，設置低功耗，2us濾波，正常輸出 */

  DrvCMP_Open(0,1,0);

9.3.2 DrvCMP_Close

- **函數**

  void DrvCMP_Close (void)

- **函數功能**

  關閉比較器功能，設置寄存器0x41804[0]=0.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvCMP.h

- **函數返回值**

  無

- **函數用法**

  /* 關閉比較器功能*/

  DrvCMP_Close();

9.3.3 DrvCMP_Enable

- **函數**

  void DrvCMP_Enable (void)

- **函數功能**

  使能比較器功能,設置寄存器0x41804[0]=1.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvCMP.h

- **函數返回值**

  無

- **函數用法**

  /* 使能比較器功能 */

  DrvCMP_Enable();

9.3.4 DrvCMP_PInput

- **函數**

  unsigned int DrvCMP_PInput (uCPPS)

- **函數功能**

  設置比較器正向輸入端. 設置寄存器0x41800[7:6].

- **輸入參數**

  uCPPS[in]：比較器正向輸入端選擇：

  0 : CH1

  1 : CH2

  2 : CH3

  3 : V12(V12=1.2V)

- **包含標頭檔**

  Peripheral_lib/DrvCMP.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 設置比較器正向輸入端為CH1. */

  DrvCMP_PInput(0);

9.3.5 DrvCMP_NInput

- **函數**

  unsigned int DrvCMP_NInput (uCPNS)

- **函數功能**

  設置比較器負向輸入端，設置寄存器0x41800[5:4]

- **輸入參數**

  uOPNS[in]：比較器負向輸入端選擇：

  0 : CH1

  1 : CH2

  2 : CH3

  3 : RLO

- **包含標頭檔**

  Peripheral_lib/DrvCMP.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

```
/* 比較器負向輸入端為CH1. */
DrvCMP_NInput(0);
```

9.3.6 DrvCMP_InputSwitch

- **函數**

  unsigned int DrvCMP_InputSwitch (uInputSwitch)

- **函數功能**

  比較器輸入端短路開關控制，設置寄存器0x41804[5]

- **輸入參數**

  uInputSwitch[in]：輸入端短路開關控制.

  0：短路開關斷開

  1：短路開關閉合

- **包含標頭檔**

  Peripheral_lib/DrvCMP.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

```
/* 比較器輸入端短路開關閉合*/
```

DrvCMP_InputSwitch(1);

9.3.7 DrvCMP_OutputFilter

- **函數**

  unsigned int DrvCMP_OutputFilter(uFilter)

- **函數功能**

  比較器數位輸出濾波控制，設置寄存器0x41804[1]

- **輸入參數**

  uFilter[in] : 2us延時濾波設置

  0： 不經過濾波

  1： 經過2us濾波

- **包含標頭檔**

  Peripheral_lib/DrvCMP.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 比較器輸出經過2us延時濾波. */

  DrvCMP_OutputFilter(1);

9.3.8 DrvCMP_OutputPinEnable

- **函數**

  unsigned int DrvCMP_OutputPinEnable (E_OUTPUT_PIN uPin)

- **函數功能**

  使能比較器數位輸出IO 口，設置寄存器0x40840[16]=1。

- **輸入參數**

  uPin [in] : 0 : PT1.7

- **包含標頭檔**

  Peripheral_lib/DrvCMP.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 使能CMP 數位輸出IO 口*/

  DrvCMP_OutputPinEnable(0);

9.3.9 DrvCMP_OutputPinDisable

- **函數**

  void DrvCMP_OutputPinDisable (void)

- **函數功能**

  關閉比較器數位輸出IO 口，設置寄存器0x40840[16]=0。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvCMP.h

- **函數返回值**

  無

- **函數用法**

  /* 關閉比較器數位輸出IO 口*/

  DrvCMP_OutputPinDisable();

9.3.10 DrvCMP_OutputInverse

- **函數**

  unsigned int DrvCMP_OutputInverse(uInv)

- **函數功能**

  比較器數位輸出相位控制，設置寄存器0x41804[7] .

- **輸入參數**

  uInv [in] :

  0： 正常輸出

  1： 反相輸出

- **包含標頭檔**

  Peripheral_lib/DrvCMP.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 比較器輸出反相*/

  DrvCMP_OutputInverse(1);

9.3.11 DrvCMP_EnableInt

- **函數**

  void DrvCMP_EnableInt (void)

- **函數功能**

  使能比較器中斷，比較器中斷屬於中斷向量HW3，設置寄存器0x40008[17]=1。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvCMP.h

- **函數返回值**

  無

- **函數用法**

  /* 使能比較器中斷*/

  DrvCMP_EnableInt();

9.3.12 DrvCMP_DisableInt

- **函數**

  void DrvCMP_DisableInt (void)

- **函數功能**

  關閉比較器中斷功能, 設置寄存器0x40008[17]=0 .

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvCMP.h

- **函數返回值**

  無

- **函數用法**

  /* 關閉比較器中斷功能*/

  DrvCMP_DisableInt();

9.3.13 DrvCMP_ReadIntFlag

- **函數**

  unsigned int DrvCMP_ReadIntFlag (void)

- **函數功能**

  讀取比較器中斷要求標誌位元CPOIF，讀取寄存器0x40008[1]的值。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvCMP.h

- **函數返回值**

  0： 中斷標誌位元為0，標明無CMP中斷要求

  1： 中斷標誌位元為1，標明有CMP中斷要求

  >1: 無效函數返回值

- **函數用法**

  /* 讀取比較器中斷標誌位元 */

  unsigned char flag；flag=DrvCMP_ReadIntFlag();

9.3.14 DrvCMP_ClearIntFlag

- **函數**

  void DrvCMP_ClearIntFlag (void)

- **函數功能**

  清除比較器中斷標誌位元CPOIF ，清零寄存器0x40008[1] =0。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvCMP.h

- **函數返回值**

  無

- **函數用法**

  /* 清除比較器中斷標誌位元*/

  DrvCMP_ClearIntFlag();

9.3.15 DrvCMP_Input

- **函數**

  unsigned int DrvCMP_Input (uPositiveInput, uNegativeInput,uInputSwitch)

- **函數功能**

  設置比較器正向、負向輸入端並設置輸入端短路開關，

  設置寄存器0x41800[7:6] / 0x41800[5:4] / 0x41804[5] .

- **輸入參數**

  uPositiveInput[in] : CMP 正向輸入端選擇：.

  0 : CH1

  1 : CH2

  2 : CH3

  3 : V12

  uNegativeInput[in] : CMP 負向輸入端選擇：

  0 : CH1

  1 : CH2

  2 : CH3

  3 : RLO

  uInputSwitch[in]： 輸入短路開關控制：.

  0： 短路開關斷開

  1： 短路開關閉合

- **包含標頭檔**

  Peripheral_lib/DrvCMP.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 設置CPPS=CH1,CPNS=CH3, 及輸入端短路開關斷開 */

  DrvCMP_Input(0,2,0);

9.3.16 DrvCMP_RLO_Ctrl

- **函數**

unsigned int DrvCMP_RLO_Ctrl (uCPDA ,uCPDM)

- **函數功能**

使能比較器內部自帶電阻分壓功能RLO，並設置分壓值，設置寄存器0x41804[23:20] / 0x41804[19:16]值。

- **輸入參數**

uCPDA[in]：比較器內部分壓階梯電阻比例控制.

0 : 0　　8 : 8/16 (CPRLH – CPRLL)

1 : 1/16 (CPRLH – CPRLL)　　9 : 9/16 (CPRLH – CPRLL)

2 : 2/16 (CPRLH – CPRLL)　　10 :10/16 (CPRLH – CPRLL)

3 : 3/16 (CPRLH – CPRLL)　　11 :11/16 (CPRLH – CPRLL)

4 : 4/16 (CPRLH – CPRLL)　　12 :12/16 (CPRLH – CPRLL)

5 : 5/16 (CPRLH – CPRLL)　　13 :13/16 (CPRLH – CPRLL)

6 : 6/16 (CPRLH – CPRLL)　　14 :14/16 (CPRLH – CPRLL)

7 : 7/16 (CPRLH – CPRLL)　　15 : 15/16 (CPRLH – CPRLL)

uCPDM [3:0]　比較器輸出遲滯控制器，位元控制操作模式；在遲滯模式下，uCPDA [3:0]對應位的值與 CMPO 的值一樣;

uCPDM [3]　0：關閉

1：開啟

uCPDM [2]　0：關閉

1：開啟

uCPDM [1]　0：關閉

1：開啟

uCPDM [0]　0：關閉

1：開啟

- **包含標頭檔**

Peripheral_lib/DrvCMP.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

```
/* 設置CPDM=0101,CPDA=0011 */
DrvCMP_RLO_Ctrl(0x03,0x05);
```

9.3.17 DrvCMP_RLO_refv

- **函數**

  unsigned int DrvCMP_RLO_refV (uCPRH, uCPRLS)

- 函數功能

  比較器內部自帶電阻分壓功能的輸入參考電壓源的設置及電阻低階短路開關控制；

  設置寄存器0x41800[2:1] / 0x41800[3]

- **輸入參數**

  uCPRH[in]： 電阻分壓功能的參考電壓源輸入選擇：

  0： 關閉（高阻態）

  1 : ChargePump 輸入（CP_I）

  2 : VDD3V(系統電壓)

  3 : VDD18 (1.8V數位電壓)

  uCPRLS[in]： 電阻低階短路開關控制：.

  0： 短路開關斷開

  1： 短路開關閉合

- **包含標頭檔**

  Peripheral_lib/DrvCMP.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 電阻參考電壓源為VDD18,短路開關閉合 */

  DrvCMP_RLO_refV(3,1);

9.3.18 DrvCMP_EnableNonOverlap

- **函數**

  unsigned int DrvCMP_EnableNonOverlap (E_NON_OVERLAP_PIN uInput)

- **函數功能**

  使能比較器的non-overlap自動切換功能，並設置參考輸入通道CLx.

  設置寄存器0x41804[4:2]

- **輸入參數**

  uInput[in]： 比較器自動切換功能正向參考輸入端選擇：

  0 : CL1

  1 : CL2

  2 : CL3

  3 : CL4

- **包含標頭檔**

  Peripheral_lib/DrvCMP.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/* 使能比較器non-overlap自動切換功能並設置CL1 作為正向參考輸入端*/

DrvCMP_EnableNonOverlap(0)

9.3.19 DrvCMP_DisableNonOverlap

- **函數**

void DrvCMP_DisableNonOverlap (void)

- **函數功能**

關閉比較器non-overlap 自動切換功能，設置寄存器0x41804[4] =0 。

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvCMP.h

- **函數返回值**

無

- **函數用法**

/* 關閉比較器non-overlap自動切換功能 */

DrvCMP_DisableNonOverlap();

9.3.20 DrvCMP_ReadData

- **函數**

unsigned int DrvCMP_ReadData (void)

- **函數功能**

讀取比較器數位輸出狀態值CMPO，讀取寄存器0x41800[16]值

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvCMP.h

- **函數返回值**

0: 負端輸入 > 正端輸入

1: 正端輸入 > 負端輸入

- **函數用法**

/* 讀取CMP 輸出狀態值*/

unsigned char flag；flag=DrvCMP_ReadData();

10 低雜訊運算放大器 OPAMP

10.1 功能簡介

該函數部分描述低雜訊運算放大器 OPAMP 功能的操作

--OPAMP 功能的開關

--OPAMP 輸入埠及輸出埠的控制

--OPAMP 中斷的設置及開關

--OPAMP 輸出信號的反相輸出及濾波控制

序號	函數名稱	函數功能
01	DrvOP_Open	開啟OPAMP功能
02	DrvOP_Close	關閉OPAMP功能
03	DrvOP_PInput	OPAMP正端輸入設置
04	DrvOP_NInput	OPAMP負端輸入設置
05	DrvOP_OPOoutEnable	開啟OPAMP輸出
06	DrvOP_OPOoutDisable	關閉OPAMP輸出
07	DrvOP_OuputFilter	OPAMP數位濾波輸出設置
08	DrvOP_OutputPinEnable	開啟OPAMP 數位輸出IO pin.
09	DrvOP_OutputPinDisable	關閉OPAMP 數位輸出IO pin
10	DrvOP_OutputInverse	OPAMP數位輸出反相設置
11	DrvOP_OutputWithCHPCK	CHPCK多功能器設置
12	DrvOP_EnableInt	開啟OPAMP中斷
13	DrvOP_DisableInt	關閉OPAMP中斷
14	DrvOP_ReadIntFlag	讀取OPAMP中斷標誌位元
15	DrvOP_ClearIntFlag	清除OPAMP中斷標誌位元
16	DrvOP_Feedback	OPAMP回饋電路設置
17	DrvOP_OPDEN	OPAMP數位輸出功能控制

10.2 內部定義常量

E_OUTPUT_PIN

識別字	數值	函數功能
E_OPO1	0x0	PT3.0作為 OPAMP 數位輸出IO口
E_OPO2	0x1	PT3.1作為 OPAMP 數位輸出IO口

E_OPN_PPIN

識別字	數值	功能定義
E_OPP_AIO2	0x1	AIO2 作為OPAMP輸入口
E_OPP_AIO4	0x2	AIO4 作為OPAMP輸入口
E_OPP_DAOI	0x4	DAOI作為OPAMP輸入口
E_OPP_REFO_I	0x8	REFO_I作為OPAMP輸入口
E_OPN_AIO3	0x1	AIO3 作為OPAMP輸入口
E_OPN_AIO5	0x2	AIO5 作為OPAMP輸入口
E_OPN_DAOI	0x4	DAOI 作為OPAMP輸入口
E_OPN_OPOI	0x8	OPOI 作為OPAMP輸入口
E_OPN_OPO	0x10	OPO 作為OPAMP輸入口
E_OPN_OPC	0x20	OPC 作為OPAMP輸入口

10.3 函數說明

10.3.1 DrvOP_Open

- **函數**

  void DrvOP_Open (void)

- **函數功能**

  開啟運算放大器(OPAMP)功能；設置寄存器0x41900[0]=1.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvOP.h

- **函數返回值**

  無

- **函數用法**

  /* 使能運算放大器(OPAMP) */

  DrvOP_Open();

10.3.2 DrvOP_Close

- **函數**

  void DrvOP_Close (void)

- **函數功能**

  關閉運算放大器(OPAMP)功能 ,設置寄存器0x41900[0]=0。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvOP.h

- **函數返回值**

  無

- **函數用法**

  /* 關閉運算放大器(OPAMP) */

  DrvOP_Close();

10.3.3 DrvOP_PInput

- **函數**

  unsigned int DrvOP_PInput (uOPPS)

- **函數功能**

運算放大器(OPAMP)的正向輸入端設置，

設置寄存器0x41904[19:16]即OPPS[3:0]，每一位對應一路通道，且可以同時設置多路通道有效。

- **輸入參數**

uOPPS [3:0]　運算放大器OPAMP 正向輸入端選擇，輸入範圍0x00~0x0f，為0時關閉所有通道。

uOPPS[3]：運算放大器(OPAMP)正向輸入通道3

0：關閉通道，高阻抗狀態

1：開啟通道 REFO_I

uOPPS[2]：運算放大器(OPAMP)正向輸入通道2

0：關閉通道，高阻抗狀態

1：開啟通道 DAOI

uOPPS[1]：運算放大器(OPAMP)正向輸入通道1

0：關閉通道，高阻抗狀態

1：開啟通道 AIO4

uOPPS[0]：運算放大器(OPAMP)正向輸入通道0

0：關閉通道，高阻抗狀態

1：開啟通道 AIO2

- **包含標頭檔**

Peripheral_lib/DrvOP.h

- **函數返回值**

0：設置成功;　　其他：設置失敗

- **函數用法**

```
/* 設置運算放大器(OPAMP)正向輸入端AIO2與AIO4. */
DrvOP_PInput(0x1|0x2);
```

10.3.4 DrvOP_NInput

- **函數**

unsigned int DrvOP_NInput (uOPNS)

- **函數功能**

運算放大器(OPAMP)負向端輸入埠選擇;

設置寄存器0x41904[5:0]即OPNS[5:0]，且每一位對應一路輸入通道，且可以同時設置多路通道有效。

- **輸入參數**

uOPNS[5:0]：運算放大器(OPAMP)負向輸入端選擇，輸入範圍0x00~0x3f,為0時關閉所有通道。

uOPNS[5]：運算放大器(OPAMP)負向輸入通道 5

0：關閉通道，高阻抗狀態

1：開啟通道 OPC: 內部連接10pF 電容

uOPNS[4]：運算放大器(OPAMP)負向輸入通道　4

0：關閉通道，高阻抗狀態

1：開啟通道 OPO: 運算放大器OPAMP 內部輸出

uOPNS[3]：運算放大器(OPAMP)負向輸入通道 3

0：關閉通道，高阻抗狀態

1：開啟通道 OPOI: 運算放大器OPAMP 外部輸出

uOPNS[2]：運算放大器(OPAMP)負向輸入通道 2

0：關閉通道，高阻抗狀態

1：開啟通道 DAOI DAC的輸出

uOPNS[1]：運算放大器(OPAMP)負向輸入通道 1

0：關閉通道，高阻抗狀態

1：開啟通道 AI5

uOPNS[0]：運算放大器(OPAMP)負向輸入通道 0

0：關閉通道，高阻抗狀態

1：開啟通道 AI3

- **包含標頭檔**

  Peripheral_lib/DrvOP.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

```
/*開啟運算放大器(OPAMP)負向輸入端通道AIO3與AIO5. */
DrvOP_NInput(0x1|0x2);
```

10.3.5 DrvOP_OPOoutEnable

- **函數**

```
void DrvOP_OPOoutEnable(void)
```

- **函數功能**

  打開運算放大器(OPAMP)模擬輸出功能，寄存器0x41900[1]=1。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvOP.h

- **函數返回值**

  無

- **函數用法**

```
/* 打開運算放大器(OPAMP)模式輸出*/
DrvOP_OPOoutEnable();
```

10.3.6 DrvOP_OPOoutDisable

- **函數**

void DrvOP_OPOoutDisable(void)

- **函數功能**

運算放大器(OPAMP) 模擬輸出功能關閉，寄存器0x41900[1]=0.

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvOP.h

- **函數返回值**

無

- **函數用法**

/* 關閉運算放大器(OPAMP)模擬輸出 */

DrvOP_OPOoutDisable();

10.3.7 DrvOP_OuputFilter

- **函數**

unsigned int DrvOP_OuputFilter(uFilter)

- **函數功能**

運算放大器(OPAMP)數位輸出濾波控制，控制數字輸出是否經過2s延時濾波;

設置寄存器0x41900[3]。

- **輸入參數**

uFilter[in]

0： 無濾波

1： 經過2us delay 濾波

- **包含標頭檔**

Peripheral_lib/DrvOP.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/*設置運算放大器(OPAMP) 2us 延時濾波. */

DrvOP_OuputFilter(1);

10.3.8 DrvOP_OutputPinEnable

- **函數**

unsigned int DrvOP_OutputPinEnable (E_OUTPUT_PIN uPin)

- **函數功能**

使能運算放大器(OPAMP)數位輸出埠，並選擇OPAMP輸出IO口；

寄存器0x41900[2]=1,且設置寄存器0x40840[19:18]。

- **輸入參數**

uPin [in]

0 : PT3.0

1 : PT3.1

- **包含標頭檔**

Peripheral_lib/DrvOP.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/* 使能運算放大器(OPAMP)數位輸出，IO= PT3.0*/

DrvOP_OutputPinEnable(0);

10.3.9 DrvOP_OutputPinDisable

- **函數**

void DrvOP_OutputPinDisable (void)

- **函數功能**

關閉運算放大器(OPAMP)數位輸出功能及OPAMP輸出IO口功能；

寄存器0x41900[2]=0, 寄存器0x40840[18]=0。.

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvOP.h

- **函數返回值**

無

- **函數用法**

/* 關閉運算放大器(OPAMP)數位輸出*/

DrvOP_OutputPinDisable();

10.3.10 DrvOP_OutputInverse

- **函數**

unsigned int DrvOP_OutputInverse(uInv)

- **函數功能**

運算放大器(OPAMP)數位輸出信號輸出反相控制，設置寄存器0x41900[5].

- **輸入參數**

  uInv [in]

  0： 正常輸出

  1： 輸出反相

- **包含標頭檔**

  Peripheral_lib/DrvOP.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

```
/* 使能運算放大器(OPAMP)數位輸出反相*/
DrvOP_OutputInverse(1);
```

10.3.11 DrvOP_OutputWithCHPCK

- **函數**

  unsigned int DrvOP_OutputWithCHPCK (uCHPCK)

- **函數功能**

  運算放大器(OPAMP)數位輸出信號OPO1/OPO2 是否經過CHPCK多功能器設置.

  設置寄存器0x41900[6],且在使能該功能前需要打開ADC clock，才能正確啟動該功能。

  該函數的舊名稱是：DrvOP_OutputWithCPCLK()，其功能一樣。

- **輸入參數**

  uCHPCK [in]

  0： 不帶CHPCK多功能器, OPO1/OPO2 輸出等於 OPOD

  1： 帶CHPCK多功能器r, OPO1/OPO2 輸出一個基於CHPCK的高頻信號

- **包含標頭檔**

  Peripheral_lib/DrvOP.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

```
/* 設置運算放大器(OPAMP)數位輸出帶CHPCK */
DrvADC_ClkEnable(0,0);     //開啟ADC 時鐘源
DrvOP_OutputWithCHPCK(1);  //使能CHPCK 多功能器
```

10.3.12 DrvOP_EnableInt

- **函數**

  void DrvOP_EnableInt (void)

- **函數功能**

  使能運算放大器(OPAMP)中斷向量，運算放大器(OPAMP)處於中斷向量HW3；

  設置寄存器0x4000c[16]=1.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvOP.h

- **函數返回值**

  無

- **函數用法**

  /* 使能運算放大器(OPAMP)中斷 */

DrvOP_EnableInt();

10.3.13 DrvOP_DisableInt

- **函數**

  void DrvOP_DisableInt (void)

- **函數功能**

  關閉運算放大器(OPAMP)中斷向量 ,清零寄存器0x4000c[16]=0;

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvOP.h

- **函數返回值**

  無

- **函數用法**

  /* 關閉運算放大器(OPAMP)中斷 */

  DrvOP_DisableInt();

10.3.14 DrvOP_ReadIntFlag

- **函數**

  unsigned int DrvOP_ReadIntFlag (void)

- **函數功能**

  讀取運算放大器(OPAMP)中斷標誌位元OPOIF ；讀取寄存器0x4000c[0]的值。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvOP.h

- **函數返回值**

  0：運算放大器(OPAMP)中斷標誌位元為0，無中斷要求

  1：運算放大器(OPAMP)中斷標誌位元為1，有中斷要求

  >1: 無效返回值

- **函數用法**

  /* 讀取運算放大器(OPAMP)中斷標誌位元 */

  unsigned char flag；flag=DrvOP_ReadIntFlag();

10.3.15 DrvOP_ClearIntFlag

- **函數**

  void DrvOP_ClearIntFlag (void)

- **函數功能**

  清除運算放大器(OPAMP)中斷要求標誌位元OPOIF．清零寄存器0x4000c[0] =0.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvOP.h

- **函數返回值**

  無

- **函數用法**

  /* 清除運算放大器(OPAMP)中斷要求標誌位元 */

  DrvOP_ClearIntFlag();

10.3.16 DrvOP_Feedback

- **函數**

  unsigned int DrvOP_Feedback(uFeedback)

- **函數功能**

  運算放大器(OPAMP)回饋電路或採樣電容連接設置，設置寄存器0x41900[4]。

- **輸入參數**

  uFeedback [in]

  0：電容作為集成電容器，下端連接至 OPOI

  1：電容作為採樣電容，下端連接至 VSSA

- **包含標頭檔**

  Peripheral_lib/DrvOP.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/*運算放大器(OPAMP)回饋電容連接到 OPOI */

DrvOP_Feedback(0);

10.3.17 DrvOP_OPDEN

- **函數**

unsigned char DrvOP_OPDEN(uOPDEN)

- **函數功能**

OPAMP數位輸出功能控制，寫入寄存器0x41900[2]。

- **輸入參數**

uOPDEN [in]：OPAMP數位輸出功能控制, 輸入範圍: 0~1

0： 關閉

1： 開啟

- **包含標頭檔**

Peripheral_lib/ DrvOP.h

- **函數返回值**

0：設置成功

1：設置失敗

- **函數用法**

/* 開啟OPAMP數位輸出功能控制*/

DrvOP_OPDEN(1);

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，包含：

--VDDA 電壓的控制

--帶隙(BANDGAP)參考電壓的控制

--Charge Pump升壓電路控制

--REFO 電壓的控制

--Low Power模式及ADC analog ground的控制

序號	函數名稱	功能描述
01	DrvPMU_VDDA_Voltage	VDDA電壓值設置
02	DrvPMU_VDDA_LDO_Ctrl	VDDA LDO 使能控制
03	DrvPMU_BandgapEnable	帶隙參考電壓開啟控制
04	DrvPMU_BandgapDisable	帶隙參考電壓關閉控制
05	DrvPMU_ChargePumpEnable	Charge Pump升壓電路開啟控制
06	DrvPMU_ChargePumpDisable	Charge Pump升壓電路關閉控制
07	DrvPMU_REFO_Enable	類比參考電壓REFO開啟控制
08	DrvPMU_REFO_Disable	類比參考電壓REFO關閉控制
09	DrvPMU_AnalogGround	ADC模擬共地端控制
10	DrvPMU_LDO_LowPower	VDD LDO 低功耗模式控制

11.2 內部定義常量

E_VDDA_OUTPUT_VOLTAGE

識別字	數值	函數功能
E_VDDA2_4	0x0	設置VDDA=2.4V
E_VDDA2_7	0x1	設置VDDA=2.7V
E_VDDA3_0	0x2	設置VDDA=3.0V
E_VDDA3_3	0x3	設置VDDA=3.3V

E_VDDA_LDO_ENABLE_CONTROL

識別字	數值	函數功能
E_HighZ	0x0	設置VDDA=0v
E_ VDD3V	0x1	設置VDDA=VDD3V
E_ PullDown	0x2	設置VDDA=0v
E_LDO	0x3	設置VDDA=2.4~3.3V可調

11.3 函數說明

11.3.1 DrvPMU_VDDA_Voltage

- **函數**

  unsigned int DrvPMU_VDDA_Voltage(E_VDDA_OUTPUT_VOLTAGE uVoltage)

- **函數功能**

  設置VDDA輸出電壓值，設置寄存器0x40400[19:18].

- **輸入參數**

  uVoltage [in] :VDDA電壓選擇. 輸入範圍 : 0~3

  0 : 2.4V

  1 : 2.7V

  2 : 3.0V

  3 : 3.3V

- **包含標頭檔**

  Peripheral_lib/DrvPMU.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 設置VDDA =2.7V. */

  DrvPMU_VDDA_Voltage(E_VDDA2_7);

11.3.2 DrvPMU_VDDA_LDO_Ctrl

- **函數**

  unsigned int DrvPMU_VDDA_LDO_Ctrl(E_VDDA_LDO_ENABLE_CONTROL uCtrl)

- **函數功能**

  設置VDDA穩壓電壓輸入源；該功能影響到VDDA輸出電壓，所以配合VDDA設置一起使用；

  設置寄存器0x40400[17:16] .

- **輸入參數**

  uCtrl [in] : VDDA穩壓電壓輸入源選擇. 輸入範圍 : 0~3

  0： 高阻態（High Z）,VDDA=0

  1 : VDD3V，VDDA=VDD3V

  2： 弱下拉(Weak pull down)，VDDA=0

  3： 可調穩壓(LDO),此模式VDDA才能可調。

- **包含標頭檔**

Peripheral_lib/DrvPMU.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

```
/* 設置VDDA LDO 使能.且設置VDDA=2.7V*/
DrvPMU_VDDA_LDO_Ctrl(E_LDO);
DrvPMU_VDDA_Voltage(E_VDDA2_7);
```

11.3.3 DrvPMU_BandgapEnable

- **函數**

void DrvPMU_BandgapEnable(void)

- **函數功能**

使能帶隙(Bandgap)參考電壓，設置寄存器0x40400[4] =1 .

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvPMU.h

- **函數返回值**

無

- **函數用法**

```
/* 使能帶隙參考電壓*/
DrvPMU_BandgapEnable();
```

11.3.4 DrvPMU_BandgapDisable

- **函數**

void DrvPMU_BandgapDisable(void)

- **函數功能**

關閉帶隙(Bandgap)參考電壓功能，設置寄存器0x40400[4]=0 .

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvPMU.h

- **函數返回值**

無

- **函數用法**

```
/* 關閉帶隙參考電壓. */
DrvPMU_BandgapDisable();
```

11.3.5 DrvPMU_ChargePumpEnable

- **函數**

  void DrvPMU_ChargePumpEnable(void)

- **函數功能**

  使能ChargePump升壓電路，設置寄存器0x40400[2] =1 .

  注意：Charge Pump升壓電路工作需要ADC clock 作為時鐘源，因此使用時要打開ADC 時鐘源。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvPMU.h

- **函數返回值**

  無

- **函數用法**

```
/* 開啟charge pump升壓功能*/
DrvADC_ClkEnable(0,0) ;                 //開啟ADC 時鐘源
DrvPMU_ChargePumpEnable();     //開啟ChargePump升壓功能
```

11.3.6 DrvPMU_ChargePumpDisable

- **函數**

  void DrvPMU_ChargePumpDisable (void)

- **函數功能**

  關閉Charge pump升壓電路，設置寄存器0x40400[2]= 0 .

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvPMU.h

- **函數返回值**

  無

- **函數用法**

```
/*關閉charge pump升壓功能. */
DrvPMU_ChargePumpDisable();
```

11.3.7 DrvPMU_REFO_Enable

- **函數**

  void DrvPMU_REFO_Enable(void)

- **函數功能**

  類比參考電壓REFO使能控制，輸出1.2v電壓,但是需要先開啟帶隙參考電壓；設置寄存器0x40400[1] =1 .

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvPMU.h

- **函數返回值**

  無

- **函數用法**

```
/*  使能類比參考電壓REFO. */
DrvPMU_BandgapEnable() ;   //開啟帶隙參考電壓
DrvPMU_REFO_Enable();      //開啟模擬參考電壓REFO
```

11.3.8 DrvPMU_REFO_Disable

- **函數**

  void DrvPMU_REFO_Disable(void)

- **函數功能**

  模擬參考電壓REFO關閉控制，關閉1.2v電壓輸出；設置寄存器0x40400[1] =0 .

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvPMU.h

- **函數返回值**

  無

- **函數用法**

```
/*關閉類比參考電壓REFO. */
DrvPMU_REFO_Disable();
```

11.3.9 DrvPMU_AnalogGround

- **函數**

unsigned int DrvPMU_AnalogGround(uAG)

- **函數功能**

ADC模擬地輸入源選擇，設置寄存器0x40400[3] 。

- **輸入參數**

uAG [in]

0： 外部

1： 使能buffer及使用內部（配合ADC一起使用）

- **包含標頭檔**

Peripheral_lib/DrvPMU.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

```
/* 設置ADC 模擬地輸入來源外部 */
DrvPMU_AnalogGround(0);
```

11.3.10 DrvPMU_LDO_LowPower

- **函數**

unsigned int DrvPMU_LDO_LowPower(uLP)

- **函數功能**

VDD LDO 低功耗控制，設置寄存器0x40400[0]

- **輸入參數**

uLP [in]

0： 正常功耗（從sleep模式喚醒後需要設置0）

1： 低功耗

- **包含標頭檔**

Peripheral_lib/DrvPMU.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

```
/* 使能 LDO 低功耗模式 */
DrvPMU_LDO_LowPower(1);
```

12 數模轉換器 DAC

12.1 函數功能簡介

該部分函數介紹 DAC 功能的設置

--DAC 功能的啟動與關閉

--DAC 輸入埠的設置

--DAC 的輸出口設置

--DAC 輸出電壓的設置

序號	函數名稱	功能描述
01	DrvDAC_Open	開啟DAC並配置相關參數
02	DrvDAC_Close	關閉DAC及DAC IO口輸出功能
03	DrvDAC_Enable	開啟DAC
04	DrvDAC_Disable	關閉DAC
05	DrvDAC_EnableOutput	開啟DAC輸出
06	DrvDAC_DisableOutput	關閉DAC輸出
07	DrvDAC_PInput	DAC正端參考輸入設置
08	DrvDAC_NInput	DAC負端參考輸入設置
09	DrvDAC_DABIT	D/A轉換輸出電壓比例設置
10	DrvDAC_SetoutputIO	DAC輸出IO 口設置

12.2 內部定義常量

E_DAC_INPUT

識別字	數值	函數功能
E_DAC_PVDD3V	0x0	正端信號輸入端
E_ DAC_PVDDA	0x1	正端信號輸入端
E_ DAC_PREFO_I	0x2	正端信號輸入端
E_DAC_POPO	0x3	正端信號輸入端
E_ DAC_PAIO6	0x4	正端信號輸入端
E_ DAC_NVSSA	0x0	負端信號輸入端
E_ DAC_NREFO_I	0x1	負端信號輸入端
E_ DAC_NOPO	0x2	負端信號輸入端
E_ DAC_NAIO7	0x3	負端信號輸入端

12.3 函數說明

12.3.1 DrvDAC_Open

- **函數**

  unsigned int DrvDAC_Open(E_DAC_INPUT uPinput ,E_DAC_INPUT uNinput, uDAO)

- **函數功能**

  使能DAC及設置DAC正向、負向參考電壓輸入埠，並且設置DAC輸出分壓的初始比例值；

  設置寄存器0x41700[5:0] 及寄存器0x41704[7:0].

  注意：AIO6的設置需要通過設置ADC 寄存器0x41104[28]BIT DA 來啟動與關閉！

- **輸入參數**

  uPinput[in] : DAC 正向參考輸入端選擇：

  0 : VDD3V

  1 : VDDA

  2 : REFO_I

  3 : OPO

  4：AIO6 （需要通過設置ADC寄存器0x41104[28]BIT DA來選擇）

  uNinput[in] : DAC 負向參考輸入端選擇：.

  0 : VSSA

  1 : REFO_I

  2 : OPO

  3 : AIO7

  uDAO [in] : DAO[7:0] 輸出電壓值的分壓比例設置DAO/255. 輸入範圍 ：0~255

- **包含標頭檔**

  Peripheral_lib/DrvDAC.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 使能DAC, 設定正向輸入端為AIO6, 負向輸入端為VSSA ,DAO=5 */

  DrvDAC_Open(E_DAC_AIO6, E_DAC_VSSA ,5);

12.3.2 DrvDAC_Close

- **函數**

  void DrvDAC_Close(void)

- **函數功能**

關閉DAC及關閉DAC電壓輸出，設置寄存器0x41700[1:0]=00b;

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvDAC.h

- **函數返回值**

  無

- **函數用法**

  /* 關閉 DAC及輸出功能*/

  DrvDAC_Close();

12.3.3 DrvDAC_Enable

- **函數**

  void DrvDAC_Enable(void)

- **函數功能**

  開啟DAC ，設置寄存器0x41700[0]=1.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvDAC.h

- **函數返回值**

  無

- **函數用法**

  /* 開啟 DAC */

  DrvDAC_Enable();

12.3.4 DrvDAC_Disable

- **函數**

  void DrvDAC_Disable(void)

- **函數功能**

  關閉DAC，設置寄存器0x41700[0]=0;

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvDAC.h

- **函數返回值**

  無

- **函數用法**

  /* 關閉 DAC */

  DrvDAC_Disable();

12.3.5 DrvDAC_EnableOutput

- **函數**

  void DrvDAC_EnableOutput(void)

- **函數功能**

  使能DAC 輸出，設置寄存器0x41700[1]=1;

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvDAC.h

- **函數返回值**

  無

- **函數用法**

  /* 使能DAC輸出 */

  DrvDAC_EnableOutput();

12.3.6 DrvDAC_DisableOutput

- **函數**

  void DrvDAC_DisableOutput (void)

- **函數功能**

  關閉DAC 輸出，設置寄存器0x41700[1] =0.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvDAC.h

- **函數返回值**

  無

- **函數用法**

  /*關閉 DAC 輸出 */

  DrvDAC_DisableOutput();

12.3.7 DrvDAC_PInput

- **函數**

  unsigned int DrvDAC_Pinput(E_DAC_INPUT uPinput)

- **函數功能**

  DAC正向輸入端選擇設置，設置寄存器0x41700[5:4]，及ADC寄存器0x41104[28]

- **輸入參數**

  uPinput [in] : DAC 正向輸入端選擇. 輸入範圍 : 0~4

  0 : VDD3V

  1 : VDDA

  2 : REFO_I

  3 : OPO

  4 : AIO6

- **包含標頭檔**

  Peripheral_lib/DrvDAC.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  ```
  /* 設置DAC正向輸入端為VDD3V. */
  DrvDAC_Pinput(E_DAC_VDD3V);
  ```

12.3.8 DrvDAC_NInput

- **函數**

  unsigned int DrvDAC_Ninput(E_DAC_INPUT uNinput)

- **函數功能**

  DAC 負向輸入端選擇設置，設置寄存器0x41700[3:2].

- **輸入參數**

  uNinput [in] : DAC 負向輸入端選擇. 輸入範圍 : 0~3

  0 : VSSA

  1 : REFO_I

  2 : OPO

  3 : AIO7

- **包含標頭檔**

  Peripheral_lib/DrvDAC.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 設定 DAC 負向輸入端為VSSA. */

  DrvDAC_Ninput(E_DAC_VSSA);

12.3.9 DrvDAC_DABIT

- **函數**

  unsigned int DrvDAC_DABIT(uDABIT)

- **函數功能**

  DAO[7:0] 輸出電壓值的分壓比例設置即DAO/255，值寫入寄存器0x41704[7:0]。

- **輸入參數**

  uDABIT [in]：寫入DAO[7:0]，D/A轉換輸出電壓比例值設置uDABIT/255. 輸入範圍: 0~255

- **包含標頭檔**

  Peripheral_lib/DrvDAC.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* DAO [7:0] =5 */

  DrvDAC_DABIT(5);

12.3.10 DrvDAC_SetoutputIO

- **函數**

  unsigned int DrvDAC_SetoutputIO(unsigned int uio)

- **函數功能**

  設置DAC輸出IO口,設置PT3寄存器0x40828[17]

- **輸入參數**

  uio [in] :

  0： 關閉PT3.1作為DAC 輸出口的功能

  1： 開啟PT3.1作為DAC 輸出口的功能

- **包含標頭檔**

  Peripheral_lib/DrvDAC.h

- **函數返回值**

  0：設置成功

  1：設置失敗

- **函數用法**

  /* 使能PT3.1作為DAC輸出 IO*/

DrvDAC_SetoutputIO(1);

13 即時時鐘 RTC

13.1 函數簡介

函數描述對 RTC 系統的控制包含：

--RTC 時鐘源的設置

--RTC 的啟動與關閉

--RTC 的時間格式設置及當前時間的寫入與讀取操作

--RTC 的鬧鐘功能的設置

序號	函數名稱	功能描述
01	DrvRTC_SetFrequencyCompensation	設置 RTC 頻率補償值
02	DrvRTC_WriteEnable	寫入寄存器解鎖碼，解鎖寄存器寫入操作
03	DrvRTC_WriteDisable	清除寄存器解鎖碼，寄存器無法寫入
04	DrvRTC_ClockSource	設置RTC的時鐘源為內部或外部
05	DrvRTC_AlarmEnable	開啟RTC鬧鐘功能
06	DrvRTC_AlarmDisable	關閉RTC鬧鐘功能
07	DrvRTC_PeriodicTimeEnable	開啟RTC定時喚醒功能及設置定時喚醒時間
08	DrvRTC_PeriodicTimeDisable	關閉RTC定時喚醒功能
09	DrvRTC_Enable	開啟RTC功能
10	DrvRTC_Disable	關閉RTC功能
11	DrvRTC_HourFormat	設置時間格式為12小時制或24小時制
12	DrvRTC_ReadState	讀取RTC的狀態位元
13	DrvRTC_ClearState	清除RTC的狀態位元
14	DrvRTC_EnableInt	開啟RTC中斷功能
15	DrvRTC_DisableInt	關閉RTC中斷功能
16	DrvRTC_ReadIntFlag	讀取RTC中斷標誌位元
17	DrvRTC_ClearIntFlag	清除RTC中斷標誌位元
18	DrvRTC_Write	設定’當前/鬧鐘’的時間和日期
19	DrvRTC_Read	讀取’當前/鬧鐘’的時間和日期
20	DrvRTC_ClkConfig	RTC時鐘源的設置控制
21	DrvRTC_EnableWUEn	開啟RTC喚醒功能
22	DrvRTC_DisableWUEn	關閉RTC喚醒功能

13.2 內部定義常量

E_DRVRTC_CLOCK_SOURCE

識別字	數值	功能意義
E_EXT_CK	0	RTC時鐘源由外部低頻時鐘提供
E_INT_CK	1	RTC時鐘源有內部低頻時鐘提供

E_DRVRTC_ TICK

識別字	數值	功能意義
E_DRVRTC_1_128_SEC	0	定時喚醒除頻 1/128
E_DRVRTC_1_64_SEC	1	定時喚醒除頻 1/64
E_DRVRTC_1_32_SEC	2	定時喚醒除頻 1/32
E_DRVRTC_1_16_SEC	3	定時喚醒除頻 1/16
E_DRVRTC_1_8_SEC	4	定時喚醒除頻 1/8
E_DRVRTC_1_4_SEC	5	定時喚醒除頻 1/4
E_DRVRTC_1_2_SEC	6	定時喚醒除頻 1/2
E_DRVRTC_1_SEC	7	定時喚醒除頻 1

E_DRVRTC_HOUR_FORMAT

識別字	數值	功能意義
E_DRVRTC_HOUR_12	1	12小時制
E_DRVRTC_HOUR_24	0	24小時制

E_DRVRTC_TIME_SELECT

識別字	數值	功能意義
DRVRTC_CURRENT_TIME	0	選擇'當前時間'選項
DRVRTC_ALARM_TIME	1	選擇'鬧鐘時間'選項

E_DRVRTC_FLAG

識別字	數值	功能意義
E_DRVRTC_ALARM_FLAG	0	鬧鐘標誌位元
E_DRVRTC_PERIODIC_FLAG	1	定時時間標誌位元
E_DRVRTC_CLEAR_ALL	2	鬧鐘標誌位元和定時時間標誌位元

13.3 函數說明

注意：需要先使能 RTC clock，再寫入解鎖碼(對寄存器 0X41A00[23:20]寫 0110b)，然後才能正確寫入寄存器

13.3.1 DrvRTC_SetFrequencyCompensation

- **函數**

  unsigned int DrvRTC_SetFrequencyCompensation(

  unsigned int uFrequencyCom);

- **函數功能**

  設置RTC時鐘頻率補償值，設置寄存器0x41a04[22:16]。

- **輸入參數**

  uFrequencyCom [in]： 設置RTC時鐘頻率補償值，設定範圍是 0~0x7f

  0111111 : +126 ppm

  0111110 : +124 ppm

  | :

  0000001 : +2 ppm

  0000000 : +0 ppm

  1000000 : - 0 ppm

  1000001 : - 2 ppm

  | :

  1111110 : -124 ppm

  1111111 : -126 ppm

- **包含標頭檔**

  Peripheral_lib/DrvRTC.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /*設置頻率補償為 -2 PPM */

  DrvRTC_ SetFrequencyCompensation(0x41);

13.3.2 DrvRTC_WriteEnable

- **函數**

  void DrvRTC_WriteEnable(void);

- **函數功能**

  寫入解鎖碼恢復RTC寄存器寫入操作.，對寄存器0x41A00[23 :20]寫入0110b

注意必須要寫入解鎖碼才能對寄存器進行寫入！

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvRTC.h

- **函數返回值**

  無

- **函數用法**

```
/* RTC寄存器解鎖，解鎖後才能寫入操作 */
DrvRTC_WriteEnable();
```

13.3.3 DrvRTC_WriteDisable

- **函數**

```
void DrvRTC_WriteDisable(void);
```

- **函數功能**

  清除RTC解鎖碼，重新鎖住寄存器不允許寫入，對寄存器0x41A00[23:20]寫入0000b

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvRTC.h

- **函數返回值**

  無

- **函數用法**

```
/* 鎖住RTC寄存器，不允許寫入操作*/
DrvRTC_WriteDisable();
```

13.3.4 DrvRTC_ClockSource

- **函數**

```
unsigned int DrvRTC_ClockSource(
E_DRVRTC_CLOCK_SOURCE uClockSource
);
```

- **函數功能**

  設定RTC時鐘源為內部或外部低速時鐘. 設置寄存器0x41A00[1]。

- **輸入參數**

  uClockSource [in]

0：外部低頻時鐘源

1：內部低頻時鐘源

- **包含標頭檔**

Peripheral_lib/DrvRTC.h

- **函數返回值**

CKS ：擁有防誤操作保護

0：外部低頻時鐘

1：內部低頻時鐘

- **函數用法**

```
/* 設置RTC時鐘源來自外部低頻時鐘 */
DrvRTC_ClockSource(E_EXT_CK);
```

13.3.5 DrvRTC_AlarmEnable

- **函數**

void DrvRTC_AlarmEnable (void);

- **函數功能**

使能鬧鐘(Alarm)功能；設置寄存器0x41A00[3]=1.

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvRTC.h

- **函數返回值**

無

- **函數用法**

```
/* 使能 RTC 鬧鐘(Alarm)功能*/
DrvRTC_AlarmEnable();
```

13.3.6 DrvRTC_AlarmDisable

- **函數**

void DrvRTC_AlarmDisable (void);

- **函數功能**

關閉鬧鐘(Alarm)功能；設置寄存器0x41A00[3]=0.

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvRTC.h

- **函數返回值**

無

- **函數用法**

/*關閉鬧鐘(Alarm)功能*/

DrvRTC_AlarmDisable();

13.3.7 DrvRTC_PeriodicTimeEnable

- **函數**

unsigned int DrvRTC_PeriodicTimeEnable (E_DRVRTC_TICK uPeriodicTimer);

- **函數功能**

使能定時喚醒(Periodic Time)功能並設置定時喚醒的時間；

設置寄存器0x41A04[2:0] 及 寄存器0x41A00[5]=1,0x41A00[4]=1.

- **輸入參數**

uPeriodicTimer[in]：定時喚醒除頻器設置

0: 1/128

1: 1/64

2: 1/32

3: 1/16

4: 1/8

5: 1/4

6: 1/2

7: 1

- **包含標頭檔**

Peripheral_lib/DrvRTC.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

/* 使能RTC定時喚醒功能，設置定時喚醒時間為1/16second*/

DrvRTC_PeriodicTimeEnable(3);

13.3.8 DrvRTC_PeriodicTimeDisable

- **函數**

void DrvRTC_PeriodicTimeDisable (void);

- **函數功能**

關閉定時喚醒(Periodic Time)功能，設置寄存器0x41A00[5]=0 / 0x41A00[4]=0。

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvRTC.h

- **函數返回值**

無

- **函數用法**

```
/* 關閉定時喚醒(Periodic time)功能*/
DrvRTC_PeriodicTimeDisable();
```

13.3.9 DrvRTC_Enable

- **函數**

```
void DrvRTC_Enable (void);
```

- **函數功能**

使能RTC 功能，設置寄存器0x41A00[0]=1。

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvRTC.h

- **函數返回值**

無

- **函數用法**

```
/* 使能 RTC 功能*/
DrvRTC_Enable();
```

13.3.10 DrvRTC_Disable

- **函數**

```
void DrvRTC_Disable (void);
```

- **函數功能**

關閉RTC功能，設置寄存器0x41A00[0]=0.

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvRTC.h

- **函數返回值**

無

- **函數用法**

```
/* 關閉RTC功能 */
DrvRTC_Disable();
```

13.3.11 DrvRTC_HourFormat

- **函數**

  unsigned int DrvRTC_HourFormat(E_DRVRTC_HOUR_FORMAT uHourFormat);

- **函數功能**

  設置小時格式為12小時制或24小時制，設置寄存器0x41A00[2]。

- **輸入參數**

  uHourFormat[in]： 小時格式設置

  0 : 24小時制

  1 : 12小時制

- **包含標頭檔**

  Peripheral_lib/DrvRTC.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

```
/* 設置12小時制 */
DrvRTC_DrvRTC_HourFormat(1);
```

13.3.12 DrvRTC_ReadState

- **函數**

  unsigned int DrvRTC_ReadState(void);

- **函數功能**

  讀取RTC狀態標誌位元，讀取寄存器0x41A00[19:16]值

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvRTC.h

- **函數返回值**

  返回值有效範圍是0x0~0xf，每一位元值對應一個標誌位元狀態值

  Bit 0 : RTC 鬧鐘標誌位元TAF

  Bit 1 : RTC 喚醒功能標誌位元WUF

  Bit 2 : RTC 定時喚醒標誌位元PTF

Bit 3 : RTC 閏年標誌位元LPYF

- **函數用法**

/* 查詢RTC鬧鐘狀態位元 */

Sample code 1 :

```
If (DrvRTC_ReadState()&0x1)
        //RTC alarm triggered
else
        // RTC Wakeup triggered
```

Sample code 2 :

```
flag = DrvRTC_ReadState();
```

13.3.13 DrvRTC_ClearState

- **函數**

unsigned int DrvRTC_ClearState(E_DRVRTC_FLAG uFlag);

- **函數功能**

清除RTC狀態標誌位元，清零寄存器0x41A00[19:16]值

- **輸入參數**

uFlag[in] 待清除狀態位元選擇

0： 清除鬧鐘標誌位元TAF

1： 清除定時喚醒標誌位元PTF

2: 清除鬧鐘（TAF）和定時（PTF）標誌位元

- **包含標頭檔**

Peripheral_lib/DrvRTC.h

- **函數返回值**

0：設置成功

其他：設置失敗

- **函數用法**

```
/* 清除TAF/ PTF標誌位元 */
DrvRTC_ClearState(2);
```

13.3.14 DrvRTC_EnableInt

- **函數**

void DrvRTC_EnableInt(void)

- **函數功能**

使能RTC中斷。每次進入中斷都需要清除定時喚醒標誌位元（PTF）下次才能正常進入中斷；

設置寄存器0x40004[21]=1.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvRTC.h

- **函數返回值**

  無

- **函數用法**

  /* 使能RTC中斷 */

  DrvRTC_EnableInt();

13.3.15 DrvRTC_DisableInt

- **函數**

  void DrvRTC_DisableInt(void)

- **函數功能**

  關閉RTC 中斷, 設置寄存器0x40004[21]=0.

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvRTC.h

- **函數返回值**

  無

- **函數用法**

  /* 關閉RTC中斷 */

  DrvRTC_DisableInt();

13.3.16 DrvRTC_ReadIntFlag

- **函數**

  unsigned int DrvRTC_ReadIntFlag(void)

- **函數功能**

  讀取RTC中斷要求標誌位元RTCIF，讀取寄存器0x40004[5]的值。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvRTC.h

- **函數返回值**

  0： 無中斷要求

  1： 有中斷要求

- **函數用法**

/* 讀取RTC中斷標誌位元*/

unsigned char flag；flag=DrvRTC_ReadIntFlag();

13.3.17 DrvRTC_ClearIntFlag

- **函數**

  void DrvRTC_ClearIntFlag(void)

- **函數功能**

  清除RTC中斷要求標誌位元RTCIF； 寄存器0x40004[5]=0

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvRTC.h

- **函數返回值**

  無

- **函數用法**

  /* 清除RTC中斷要求標誌位元 */

  DrvRTC_ClearIntFlag();

13.3.18 DrvRTC_Write

- **函數**

  unsigned int DrvRTC_Write (

  E_DRVRTC_TIME_SELECT eTime, S_DRVRTC_TIME_DATA_T *sPt);

- **函數功能**

  設置RTC的當前（或鬧鐘）的時間和日期;

  設置寄存器0x41A08/0x41A0C/0x41A10/0x41A14/0x41A18/0x41A1C

- **輸入參數**

  eTime [in]： 指向‘當前時間/日期’或‘鬧鐘時間/日期’.

  0: 當前時間/日期

  1: 鬧鐘時間/日期

  *sPt [in]： 表示要設置的時間/日期，該變數為一個結構體，設置內容為：

  u8cClockDisplay DRVRTC_CLOCK_12 / DRVRTC_CLOCK_24

  u8cAmPm DRVRTC_AM / DRVRTC_PM

  u32cSecond Second 數值

  u32cMinute Minute 數值

  u32cHour Hour 數值(12小時制的輸入範圍：0~11；24小時制輸入範圍：0~23)

  u32cDayOfWeek Day of week

  u32cDay Day 數值

u32cMonth Month 數值

u32Year Year 數值

- **包含標頭檔**

Peripheral_lib/DrvRTC.h

- **函數返回值**

0：設置成功

其他：設置失敗.

- **函數用法**

```
/* 更新當前時間‘秒數’為0 */
S_DRVRTC_TIME_DATA_T sCurTime;
DrvRTC_Read(DRVRTC_ALARM_TIME, &sCurTime);
sCurTime.u32cSecond = 0;
DrvRTC_Write(DRVRTC_ALARM_TIME, &sCurTime);
```

13.3.19 DrvRTC_Read

- **函數**

```
unsigned int DrvRTC_Read (
E_DRVRTC_TIME_SELECT eTime,
S_DRVRTC_TIME_DATA_T *sPt );
```

- **函數功能**

讀取RTC的‘當前時間/日期’或‘鬧鐘的時間/日期’的資料

讀取寄存器0x41A08/0x41A0C/0x41A10/0x41A14/0x41A18/0x41A1C

- **輸入參數**

eTime [in]：指向‘當前時間/日期’或‘鬧鐘的時間/日期’選項：

0：當前時間/日期(Current time)

1：鬧鐘時間/日期(Alarm time)

*sPt [in]：表示要設置的時間/日期，該變數為一個結構體，設置內容為：

u8cClockDisplay DRVRTC_CLOCK_12 / DRVRTC_CLOCK_24

u8cAmPm DRVRTC_AM / DRVRTC_PM

u32cSecond Second 數值

u32cMinute Minute 數值

u32cHour Hour 數值

u32cDayOfWeek Day of week

u32cDay Day 數值

u32cMonth Month 數值

u32Year Year 數值

- **包含標頭檔**

Peripheral_lib/DrvRTC.h

- **函數返回值**

  0：設置成功

  其他：設置失敗.

- **函數用法**

```
/* 獲取當前的時間和日期 */
S_DRVRTC_TIME_DATA_T sCurTime;
DrvRTC_Read(DRVRTC_CURRENT_TIME, &sCurTime);
```

13.3.20 DrvRTC_ClkConfig

- **函數**

```
unsigned char DrvRTC_ClkConfig(unsigned char uclken);
```

- **函數功能**

  RTC 時鐘源使能控制 設置寄存器0x40308[23] 。

- **輸入參數**

  uClockSource [in]

  0： 關閉RTC的時鐘源

  1： 開啟RTC的時鐘源

- **包含標頭檔**

  Peripheral_lib/DrvRTC.h

- **函數返回值**

  1： 設置失敗

  0： 設置成功

- **函數用法**

```
/*使能RTC 時鐘源*/
DrvRTC_ClkConfig(1);
```

13.3.21 DrvRTC_EnableWUEn

- **函數**

```
void DrvRTC_EnableWUEn (void)
```

- **函數功能**

  開啟RTC喚醒功能, WUEn=1b

  設置寄存器0x41A00[4]=1

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvRTC.h

- **函數返回值**

無

- **函數用法**

/* 開啟RTC喚醒功能 */

DrvRTC_EnableWUEn();

13.3.22 DrvRTC_DisableWUEn

- **函數**

void DrvRTC_DisableWUEn (void)

- **函數功能**

關閉RTC喚醒功能, WUEn=0b

設置寄存器0x41A00[4]=0

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvRTC.h

- **函數返回值**

無

- **函數用法**

/* 關閉RTC喚醒功能 */

DrvRTC_DisableWUEn();

14 IIC 串列通訊 I2C

14.1 函數簡介

該部分函數描述 IIC 通訊模組的控制，包含：

--IIC 啟動控制

--IIC 工作模式控制

--IIC 通訊及收發資料配置控制

--IIC 中斷向量控制

序號	函數名稱	功能描述
01	DrvI2C_Open	開啟 I2C 及配置 I2C 匯流排時鐘
02	DrvI2C_Close	關閉I2C
03	DrvI2C_SlaveSet	開啟I2C從機模式，設置從機位址及GC使能控制
04	DrvI2C_SetIOPin	開啟並設置I2C通訊IO 口
05	DrvI2C_WriteData	發送1byte資料
06	DrvI2C_Write3ByteData	發送3byte資料
07	DrvI2C_ReadData	讀取接收暫存器的資料
08	DrvI2C_Ctrl	設置I2C控制位：STA、STO、AA、SI，控制起始信號、停止信號及應答信號
09	DrvI2C_EnableInt	開啟I2C中斷功能
10	DrvI2C_DisableInt	關閉I2C中斷功能
11	DrvI2C_ReadIntFlag	讀取I2C插斷要求標誌位元
12	DrvI2C_ClearIntFlag	清除I2C插斷要求標誌位元
13	DrvI2C_ClearEIRQ	清除錯誤中斷標誌位元
14	DrvI2C_ClearIRQ	清除I2C器件準備好標誌位元
15	DrvI2C_GetStatusFlag	讀取I2C狀態位元
16	DrvI2C_TimeOutEnable	開啟超時重定功能，設置時鐘分頻及超時控制值
17	DrvI2C_TimeOutDisable	關閉超時復位功能
18	DrvI2C_STSP	設置I2C發送起始信號或停止信號
19	DrvI2C_MGetACK	查詢從機回饋的應答信號ACK/NACK
20	DrvI2C_DisableIOPin	關閉IO口複用作為I2C通訊口功能
21	DrvI2C_EnableSEn	開啟I2C Slave模式功能
22	DrvI2C_DisableSEn	關閉I2C Slave模式功能
23	DrvI2C_EnableI2CEn	開啟I2C

14.2 內部定義常量

E_DRVI2C_Status

識別字	數值	功能意義
E_DRVI2C_ARBITRATION_FLAG	0	仲裁漏失標誌位元
E_DRVI2C_GENERAL_CALL_FLAG	1	全呼標誌位元
E_DRVI2C_ACKNOWLEDGE_FLAG	2	應答信號狀態標誌位元
E_DRVI2C_DATA_FIELD_FLAG	3	資料標誌位元
E_DRVI2C_RW_STATE_FLAG	4	讀/寫狀態標誌位元
E_DRVI2C_RS_FLAG	5	接收停止或重新開始標誌位元
E_DRVI2C_SLAVE_ACTIVE_FLAG	6	從機模式有效標誌位元
E_DRVI2C_MASTER_ACTIVE_FLAG	7	主機模式有效標誌位元

E_DRVI2C_TIMEOUT_PRESCALE

識別字	數值	功能意義
E_DRVI2C_I2CLK_DIV_1	0	I2C CLK/1
E_DRVI2C_I2CLK_DIV_2	1	I2C CLK/2
E_DRVI2C_I2CLK_DIV_4	2	I2C CLK/4
E_DRVI2C_I2CLK_DIV_8	3	I2C CLK/8
E_DRVI2C_I2CLK_DIV_16	4	I2C CLK/16
E_DRVI2C_I2CLK_DIV_32	5	I2C CLK/32
E_DRVI2C_I2CLK_DIV_64	6	I2C CLK/64
E_DRVI2C_I2CLK_DIV_128	7	I2C CLK/128

E_DRVI2C_TIMEOUT_LIMIT

識別字	數值	功能意義
E_DRVI2C_CLKPSX1	0	1 * CLKps Cycle
E_DRVI2C_CLKPSX2	1	2 * CLKps Cycle
E_DRVI2C_CLKPSX3	2	3 * CLKps Cycle
E_DRVI2C_CLKPSX4	3	4 * CLKps Cycle
E_DRVI2C_CLKPSX5	4	5 * CLKps Cycle
E_DRVI2C_CLKPSX6	5	6 * CLKps Cycle
E_DRVI2C_CLKPSX7	6	7 * CLKps Cycle
E_DRVI2C_CLKPSX8	7	8 * CLKps Cycle
E_DRVI2C_CLKPSX9	8	9 * CLKps Cycle
E_DRVI2C_CLKPSX10	9	10 * CLKps Cycle
E_DRVI2C_CLKPSX11	10	11 * CLKps Cycle
E_DRVI2C_CLKPSX12	11	12 * CLKps Cycle
E_DRVI2C_CLKPSX13	12	13 * CLKps Cycle
E_DRVI2C_CLKPSX14	13	14 * CLKps Cycle
E_DRVI2C_CLKPSX15	14	15 * CLKps Cycle
E_DRVI2C_CLKPSX16	15	16 * CLKps Cycle

E_DRVI2C_INTERRUPT

識別字	數值	功能意義
E_DRVI2C_INT	1	開啟I2C 中斷功能

E_DRVI2C_ERROR_INT	2	開啟I2C 錯誤中斷功能
E_DRVI2C_INT_ALL	3	開啟I2C 中斷及錯誤中斷功能

E_DRVI2C_SLAVE_BIT

識別字	數值	功能意義
E_DRVI2C_SLAVE_7BIT	0	從機7bit 位址碼模式
E_DRVI2C_SLAVE_10BIT	1	從機10bit 位址碼模式

14.3 函數說明

注意：只有使能 I2C 後才能對 I2C 其他寄存器設置。

14.3.1 DrvI2C_Open

- **函數**

  unsigned int DrvI2C_Open (uint32_t u32CRG);

- **函數功能**

  使能I2C功能，並設置I2C匯流排串列傳輸速率；

  設置寄存器0x41000[0]=1,串列傳輸速率寫入寄存器0x41008[23:16] .

  注意：只有使能I2C後才能對I2C其他寄存器設置。

- **輸入參數**

  u32CRG [in]

  匯流排串列傳輸速率設置值CRG，設置範圍 0~0xff.

  資料匯流排串列傳輸速率 = (I2CLK/(4*(CRG+1)))

- **包含標頭檔**

  Peripheral_lib/DrvI2C.h

- **函數返回值**

  0：設置成功

  其他：設置失敗

- **函數用法**

  /* 使能I2C，設置CRG =100 */

  DrvI2C_Open(100);

14.3.2 DrvI2C_Close

- **函數**

  void DrvI2C_Close (void);

- **函數功能**

  關閉I2C功能.，設置寄存器0x41000[0]=0。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvI2C.h

- **函數返回值**

  無

- **函數用法**

  /* 關閉I2C */

DrvI2C_Close();

14.3.3 DrvI2C_SlaveSet

- **函數**

unsigned int DrvI2C_SlaveSet(
uint32_t uSlaveAddr,
E_DRVI2C_SLAVE_BIT uAddrBit,
uint8_t uSlave3Byte,
uint8_t GC_Flag);

- **函數功能**

使能I2C從機模式，並設置從機位址碼及位址碼模式，從機3byte資料發送模式設置，全呼模式GC設置。;
設置寄存器0x41004[7] /0x41004[5] /0x41000[2] /0x4100C[7:0]

- **輸入參數**

uSlaveAddr[in]： 從機地址碼
7bit :0~0x7f，主要輸入值為偶數，如0x00/0x02/0x0C等。
10bit: 0~0x3ff，主要輸入值為偶數，即保持bit[0]為0，如0x30C/0x3CC等。
uAddrBit[in]： 從機位址碼模式
0: 從機地址碼為7bit
1: 從機地址碼為10bit
uSlave3Byte[in]：從機3byte資料發送模式
0: 正常資料發送模式
1: 從機發送3byte資料模式
GC_Flag[in] 全呼模式設置
0: 正常呼叫模式
1: 使能全線廣播模式

- **包含標頭檔**

Peripheral_lib/DrvI2C.h

- **函數返回值**

0：設置成功
其他：設置失敗

- **函數用法**

/* 使能從機模式，從機位址碼為0x30，正常資料模式，正常呼叫模式*/
DrvI2C_SlaveSet(0x30,0,0,0);

14.3.4 DrvI2C_SetIOPin

- **函數**

unsigned char DrvI2C_SetIOPin(unsigned int upin);

- **函數功能**

  設置I2C通訊IO口，設置寄存器0x40844[19 :16] 。

- **輸入參數**

  upin[in]： 通訊IO選擇

  0 SCL=PT1.0;SDA=PT1.1

  1 SCL=PT1.2;SDA=PT1.3

  2 SCL=PT1.4;SDA=PT1.5

  3 SCL=PT1.6;SDA=PT1.7

  4 SCL=PT2.0;SDA=PT2.1

  5 SCL=PT2.2;SDA=PT2.3

  6 SCL=PT2.4;SDA=PT2.5

  7 SCL=PT2.6;SDA=PT2.7

- **包含標頭檔**

  Peripheral_lib/DrvI2C.h

- **函數返回值**

  無

- **函數用法**

  /* 使能通訊口 PT1.0 and PT1.1 */

  DrvI2C_SetIOPin(0);

14.3.5 DrvI2C_WriteData

- **函數**

  void DrvI2C_WriteData(uint8_t uData);

- **函數功能**

  寫入待發送的資料，寫入寄存器0x41014[7:0].

- **輸入參數**

  uData [IN]： 待發送的資料, 1Byte 資料, 輸入範圍 : 0~0xFF

- **包含標頭檔**

  Peripheral_lib/DrvI2C.h

- **函數返回值**

  無

- **函數用法**

  /*寫入資料0x55至發送暫存器 */

  DrvI2C_WriteData(0x55);

14.3.6 DrvI2C_Write3ByteData

- **函數**

  void DrvI2C_Write3ByteData(uint8_t uData1,uData2,uData3);

- **函數功能**

  寫入3byte待發送資料，連續發送3byte.

- **輸入參數**

  uData1, uData2, uData3 [IN] 待發送的資料, 1Byte 數據, 輸入範圍 : 0~0xFF

- **包含標頭檔**

  Peripheral_lib/DrvI2C.h

- **函數返回值**

  無

- **函數用法**

  /* 寫入3byte 數據0x11 0x22 0x33 準備發送 */

  DrvI2C_Write3ByteData(0x11,0x22,0x33);

14.3.7 DrvI2C_ReadData

- **函數**

  unsigned char DrvI2C_ReadData(void);

- **函數功能**

  讀取接收暫存器接收到的資料並控制主機返回應答或非應答信號.

  讀取寄存器0x41010[7:0]的值。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvI2C.h

- **函數返回值**

  1Byte 暫存器接收到的資料

- **函數用法**

  /* 讀取數據*/

  unsigned int data; data=DrvI2C_ReadData();

14.3.8 DrvI2C_Ctrl

- **函數**

  void DrvI2C_Ctrl(uint8_t start, uint8_t stop, uint8_t intFlag, uint8_t ack);

- **函數功能**

  設置I2C控制位包括STA, STO, AA, SI，控制start信號、stop信號、I2C器件準備狀態標誌位元及應答信號。

  設置寄存器0x41004[3:0]

- **輸入參數**

start [in]：起始信號控制位元

1：I2C產生start信號

0：I2C正常空閒。

stop [in]：停止信號控制位元

1：I2C生產停止信號

0：I2C正常空閒。

intFlag [in]　I2C器件狀態控制標誌位元

1：產生中斷，接收到到9個clock後器件回應，此時SCL會被器件強行拉低，直到IRQFlag清零後釋放SCL

0：正常，寫入0，將會清除I2C器件狀態控制旗標，使I2C往一個狀態執行

ack [in]

1：ACK應答回復

0：未回復ACK或回復NACK信號

- **包含標頭檔**

Peripheral_lib/DrvI2C.h

- **函數返回值**

無

- **函數用法**

DrvI2C_Ctrl(0, 0, 1, 0); /* 清除I2C器件狀態控制標誌位元IRQFlag */

DrvI2C_Ctrl(1, 0, 0, 0); /* 設置I2C 發送起始信號 */

14.3.9 DrvI2C_EnableInt

- **函數**

void DrvI2C_EnableInt(E_DRVI2C_INTERRUPT uINT)

- **函數功能**

使能I2C中斷，包括收發中斷及錯誤中斷，屬於中斷向量HW0.

設置寄存器0x40000[21:20]

- **輸入參數**

uINT[IN] :中斷項選擇

0 : I2C 收發中斷使能

1 : I2C 錯誤中斷使能

2 : I2C 收發中斷及錯誤中斷使能

- **包含標頭檔**

Peripheral_lib/DrvI2C.h

- **函數返回值**

無

- **函數用法**

/*I2C 收發中斷使能*/

DrvI2C_EnableInt(1);

14.3.10 DrvI2C_DisableInt

- **函數**

  void DrvI2C_DisableInt(E_DRVI2C_INTERRUPT uINT)

- **函數功能**

  關閉I2C中斷控制，包括收發中斷和錯誤中斷；

  清零寄存器0x40000[21:20]

- **輸入參數**

  uINT[IN]： 中斷項選擇

  0 : 關閉I2C收發中斷

  1 : 關閉I2C錯誤中斷

  2： 關閉I2C收發中斷及錯誤中斷

- **包含標頭檔**

  Peripheral_lib/DrvI2C.h

- **函數返回值**

  無

- **函數用法**

  ```
  /* 關閉I2C收發中斷 */
  DrvI2C_DisableInt(1);
  ```

14.3.11 DrvI2C_ReadIntFlag

- **函數**

  E_DRVI2C_INTERRUPT DrvI2C_ReadIntFlag(void)

- **函數功能**

  讀取I2C中斷標誌位元I2CEIF/I2CIF. 讀取寄存器0x40000[5:4]的值

- **輸入參數**

  None

- **包含標頭檔**

  Peripheral_lib/DrvI2C.h

- **函數返回值**

  0 : I2C 無中斷要求

  1 : I2C 收發中斷要求標誌位元I2CIF

  2 : I2C 錯誤中斷要求標誌位元I2CEIF

  3 : I2C 有收發中斷要求標誌位元I2CIF和錯誤中斷要求標誌位元I2CEIF

- **函數用法**

  ```
  /* Read the I2C Interrupt flag */
  uint32_t temp;
  temp=DrvRTC_ReadIntFlag();
  ```

14.3.12 DrvI2C_ClearIntFlag

- **函數**

  void DrvI2C_ClearIntFlag(E_DRVI2C_INTERRUPT uINT)

- **函數功能**

  清除I2C中斷標誌位元I2CEIF/I2CIF. 清除寄存器0x40000[5:4]的值

- **輸入參數**

  uINT[IN]

  0： 清除I2C中斷標誌位元I2CIF

  1：清除I2C中斷標誌位元I2CEIF

  2： 清除I2C中斷標誌位元I2CEIF/I2CIF

- **包含標頭檔**

  Peripheral_lib/DrvI2C.h

- **函數返回值**

  無

- **函數用法**

  /*清除I2C中斷標誌位元I2CIF */

  DrvI2C_ClearIntFlag(0);

14.3.13 DrvI2C_ClearEIRQ

- **函數**

  void DrvI2C_ClearEIRQ(void)

- **函數功能**

  清除錯誤中斷旗標EIRQFlag，只有先清零超時標誌位元(TOFLAG)才能清零EIRQFlag，寫入0就可清零；只有清除EIRQFlag才能清除中斷要求標誌位元(I2CEIF)。

  設置寄存器0x41004[4]=0。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvI2C.h

- **函數返回值**

  無

- **函數用法**

  /*清除錯誤中斷旗標EIRQ */

  DrvI2C_ClearEIRQ();

14.3.14 DrvI2C_ClearIRQ

- **函數**

  void DrvI2C_ClearIRQ(void)

- **函數功能**

  清除I2C器件狀態控制標誌位元IRQFlag，IRQFlag從1變為0，使I2C執行下一個狀態。

  清零寄存器0x41004[1]=0 。

  無論作為主機模式還是從機模式，只要接收到9個clock後，IRQFlag就被置1，此時SCL就會被拉低直到IRQFlag被清零，SCL得到釋放，器件才能執行下一個狀態動作。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvI2C.h

- **函數返回值**

  無

- **函數用法**

  /*清除I2C器件準備狀態標誌位元IRQFlag */

  DrvI2C_ClearIRQ();

14.3.15 DrvI2C_GetStatusFlag

- **函數**

  unsigned char DrvI2C_GetStatusFlag(void)

- **函數功能**

  獲取I2C狀態標誌位元，返回一個8位元的資料，讀取寄存器0x41004[23:16]的值 。

- **輸入參數**

- **包含標頭檔**

  Peripheral_lib/DrvI2C.h

- **函數返回值**

  返回值對應位表示的專案設置

  Bit 0： 仲裁漏失標誌而為 ARB

  Bit 1: General Call Flag(GC)

  Bit 2: ACK應答標誌位元（A/NA）

  Bit 3: 資料標誌位元（DF）

  Bit 4: 讀寫狀態標誌位元（R/W）

  Bit 5: 接收停止或重新開始標誌（RX P/SR）

  Bit 6: 從機模式有效標誌位元（SAct）

  Bit 7: 主機模式有效標誌位元(MAct)

  ARB: uStatus=0

0：正常

1：仲裁漏失

GC: uStatus=1

0：正常I

1：當前全呼模式

A/NA: uStatus=2

0 : NACK已經發送或接收.

1 : ACK已經接收或發送.

DF: uStatus=3

0：正常

1：資料被發送或接收.

R/W: uStatus=4

0：寫命令已被發送或接收

1：讀命令已被發送或接收.

RX P/Sr: uStatus=5

0：正常

1：接收停止或重新開始信號已被發送或接收.

SAct：uStatus=6

0：從機模式無效

1 :從機模式有效

MAct：uStatus=7

0：主機模式無效

1：主機模式有效

- **函數用法**

/* 讀取應答信號標誌位元 /

DrvRTC_GetStatusFlag(2); //讀取應答信號ACK的狀態標誌位元

14.3.16 DrvI2C_TimeOutEnable

- **函數**

unsigned char DrvI2C_TimeOutEnable(

E_DRVI2C_TIMEOUT_PRESCALE uPreScale,

E_DRVI2C_TIMEOUT_LIMIT uTimeOutLimit

);

- **函數功能**

使能超時復位功能，設置超時功能時鐘分頻及超時時間，

設置寄存器0x41000[1]=1，及寄存器0x41008[6:0] 。

- **輸入參數**

uPreScale[in]: 時鐘分頻

0 I2C CLK/1

1 I2C CLK/2

2 I2C CLK/4

3 I2C CLK/8

4 I2C CLK/16

5 I2C CLK/32

6 I2C CLK/64

7 I2C CLK/128

uTimeOutLimit [in]： 超時時間設置

0 1 * CLKps Cycle

1 2 * CLKps Cycle

2 3 * CLKps Cycle

3 4 * CLKps Cycle

4 5 * CLKps Cycle

5 6 * CLKps Cycle

6 7 * CLKps Cycle

7 8 * CLKps Cycle

8 9 * CLKps Cycle

9 10 * CLKps Cycle

10 11 * CLKps Cycle

11 12 * CLKps Cycle

12 13 * CLKps Cycle

13 14 * CLKps Cycle

14 15 * CLKps Cycle

15 16 * CLKps Cycle

- **包含標頭檔**

  Peripheral_lib/DrvI2C.h

- **函數返回值**

  0：設置成功

  0xff：設置失敗

- **函數用法**

```
/*開啟超時重定功能，設置頻率分頻器/ 32 ，及超時限制15 * CLKps Cycle */
DrvI2C_TimeOutEnable(5,14);
```

14.3.17 DrvI2C_TimeOutDisable

- **函數**

void DrvI2C_TimeOutDisable(void)

- **函數功能**

  關閉超時復位功能，設置寄存器0x41000[1] =0 。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvI2C.h

- **函數返回值**

  無

- **函數用法**

  /* 關閉超時復位功能 */

  DrvI2C_TimeOutDisable();

14.3.18 DrvI2C_STSP

- **函數**

  void DrvI2C_STSP(unsigned char usignal);

- **函數功能**

  啟動I2C的起始信號或停止信號，設置寄存器0x41004[3:2] 。

- **輸入參數**

  usignal[in]： 信號模式設置

  0 啟動起始信號

  1 啟動停止信號

- **包含標頭檔**

  Peripheral_lib/DrvI2C.h

- **函數返回值**

  無

- **函數用法**

  /* 啟動起始信號*/

  DrvI2C_STSP(0);

14.3.19 DrvI2C_MGetACK

- **函數**

  unsigned char DrvI2C_MGetACK(unsigned int utime);

- **函數功能**

  在設定的時間內查詢從機回饋的應答信號，設置寄存器0x41004[1]

- **輸入參數**

  utimel[in] :設定的查詢時間0~0xffff

- **包含標頭檔**

  Peripheral_lib/DrvI2C.h

- **函數返回值**

  0 查詢到應答信號ACK標明發送成功

  1 查詢到非應答信號NACK，標明發送失敗

- **函數用法**

  /* check the ACK during the 0xffff time*/

  Err_flag=DrvI2C_MGetACK(0xffff);

14.3.20 DrvI2C_DisableIOPin

- **函數**

  void DrvI2C_DisableIOPin(void)

- **函數功能**

  關閉I2C通訊口功能，設置寄存器0x40844[16]=0

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvI2C.h

- **函數返回值**

  無

- **函數用法**

  /*關閉I2C的通訊IO口*/

  DrvI2C_DisableIOPin();

14.3.21 DrvI2C_EnableSEn

- **函數**

  void DrvI2C_EnableSEn (void);

- **函數功能**

  開啟I2C Slave模式功能.，設置寄存器0x41004[7]=1。

- **輸入參數**

  無

- **包含標頭檔**

  Peripheral_lib/DrvI2C.h

- **函數返回值**

  無

- **函數用法**

  /* 開啟I2C Slave模式功能 */

  DrvI2C_EnableSEn();

14.3.22 DrvI2C_DisableSEn

- **函數**

  void DrvI2C_DisableSEn (void);

- **函數功能**

  關閉I2C Slave模式功能.，設置寄存器0x41004[7]=0。

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvI2C.h

- **函數返回值**

無

- **函數用法**

/* 關閉I2C Slave模式功能 */

DrvI2C_DisableSEn();

14.3.23 DrvI2C_EnableI2CEn

- **函數**

void DrvI2C_EnableI2CEn (void);

- **函數功能**

開啟I2C功能.，設置寄存器0x41000[0]=1。

- **輸入參數**

無

- **包含標頭檔**

Peripheral_lib/DrvI2C.h

- **函數返回值**

無

- **函數用法**

/* 開啟I2C功能 */

DrvI2C_EnableI2CEn();

15 Flash 讀寫

15.1 函數簡介

該部分函數描述晶片 Flash 區域的讀寫操作，包含：

--Flash 的資料寫入與讀取

--Flash 的字/頁寫入、字/頁的讀取；

--Flash 的數據擦除

序號	函數名稱	功能描述
01	DrvFlash_Burn_Word	寫入一個字資料到flash
02	ROM_BurnPage	寫入連續一頁的32個字資料到flash
03	ReadWord	讀取flash的一個字的資料
04	ReadPage	讀取flash連續一頁32個字的資料
05	PageErase	擦除flash一頁連續的資料
06	SectorErase	擦除一個sector的資料

15.2 函數說明

注意1：在執行Flash燒錄與讀取程式指令之前，必須先執行SYS_DisableGIE()；關閉全域使能中斷，這可以避免程式運行異常的行為發生.

注意2：執行Flash燒錄指令，必須確保晶片工作電壓VDD3V高於2.7V，如果晶片電壓VDD3V低於2.7V，則可能會發生燒錄錯誤行為.

15.2.1 DrvFlash_Burn_Word

- **函數**

int DrvFlash_Burn_Word(unsigned int addr,unsigned int DelayTime,unsigned int data);

- **函數功能**

寫入一個word的資料至Flash的對應位址。

注意 ：使用該函數做位址資料寫入功能之後，請再使用Read功能回讀數據做驗證動作確保資料以正確被寫入.

- **輸入參數**

addr[in]：待寫入資料的位址

16位的位址，Flash空間是從0x90000開始，所以該位址值只需寫16位值，輸入範圍0~0xffff，且位址的間隔步長為4；如要想0x9A880寫入一個word資料，函數參數只需填寫0xA880值就可以。

Delay time[in]：燒錄延時

data [in]：帶寫入的資料，輸入範圍0~0xffffffff

- **包含標頭檔**

Peripheral_lib/Drvflash.h

- **函數返回值**

0 設置成功

- **函數用法**

```
/* 寫入0xFF05到flash的0x90880位址 */
DrvFlash_Burn_Word(0x0880, 0x2000, 0xFF05);
```

注意 ：執行Flash燒錄指令，必須確保晶片工作電壓VDD3V高於2.7V，如果晶片電壓VDD3V低於2.7V，則可能會發生燒錄錯誤行為.

15.2.2 ROM_BurnPage

- **函數**

int ROM_BurnPage(unsigned int addr,unsigned int DelayTime,int* data);

- **函數功能**

一次性寫入一頁32個word的資料到Flash對應的連續位址。

注意: 使用該函數做位址資料寫入功能之後，請再使用Read功能回讀數據做驗證動作確保資料以正確被寫入.

- **輸入參數**

addr[in]：待寫入資料的首位址，16位的位址，Flash空間是從0x90000開始，所以該位址值只需寫16位值，輸入範圍0~0xffff，且位址的間隔步長為128（32*4），且不能進行跨頁寫入，因為每個page只有128byte，所以位址只能為0xuu00或0xuu80；如要想從0x9A880開始，函數參數只需填寫0xA880值就可以。

Delay time[in]：燒錄延時

data [in]：待寫入的資料，屬於指標類型，資料的長度為32word，每個資料的輸入範圍0~0xffffffff

- **包含標頭檔**

Peripheral_lib/Drvflash.h

- **函數返回值**

0xFF 設置成功

- **函數用法**

/* 從位址0x90880開始連續一次寫入32word的資料 */

int *A[32]={0};

ROM_BurnPage(0x0880, 0x2000, A);

注意 ：執行Flash燒錄指令, 必須確保晶片工作電壓VDD3V高於2.7V, 如果晶片電壓VDD3V低於2.7V, 則可能會發生燒錄錯誤行為.

15.2.3 ReadWord

- **函數**

int ReadWord(unsigned int addr);

- **函數功能**

讀取Flash對應位址的一個word的資料。

- **輸入參數**

addr[in]：待讀取資料的位址

16位的位址，Flash空間是從0x90000開始，所以該位址值只需寫16位值，輸入範圍0~0xffff，且位址的間隔步長為4；如要想0x9A880讀取一個word資料，函數參數只需填寫0xA880值就可以。

- **包含標頭檔**

Peripheral_lib/Drvflash.h

- **函數返回值**

返回一個word的資料

- **函數用法**

/* 讀取Flash的0x90880位址的資料 */

Int flag; flag= ReadWord(0x0880);

15.2.4 ReadPage

- **函數**

int ReadPage(unsigned int addr, int* data);

- **函數功能**

一次性連續讀取flash對應連續位址的32個word的數據

- **輸入參數**

addr[in]：待讀取資料的首位址，16位的地址，Flash空間是從0x90000開始，所以該位址值只需寫16位值，輸入範圍0~0xffff，且地址的間隔步長為128（32*4），且不能進行跨頁讀取，因為每個page只有128byte，所以位址只能為0xuu00或0xuu80；如要想從0x9A880開始，函數參數只需填寫0xA880值就可以。

data [in]：用於保存讀取到的資料，屬於指標類型，資料的長度為32word，每個資料的輸入範圍0~0xffffffff

- **包含標頭檔**

Peripheral_lib/DrvFlash.h

- **函數返回值**

0xFF 設置成功

- **函數用法**

```
/* 讀取flash以0x90880位址開始的連續32個word的資料 */
int *A[32]={0};
ReadPage(0x0880, A);
```

15.2.5 PageErase

- **函數**

```
int PageErase(unsigned int addr,unsigned int DelayTime);
```

- **函數功能**

一次性清除flash中對應連續位址的32個word的資料

- **輸入參數**

addr[in]：待清除資料的首位址，16位的位址，Flash空間是從0x90000開始，所以該位址值只需寫16位值，輸入範圍0~0xffff，且位址的間隔步長為128（32*4），且不能進行跨頁寫入，因為每個page只有128byte，所以位址只能為0xuu00或0xuu80；如要想從0x9A880開始，函數參數只需填寫0xA880值就可以。

Delay time[in]：延時時間

- **包含標頭檔**

Peripheral_lib/DrvFlash.h

- **函數返回值**

0xFF 設置成功

- **函數用法**

```
/*清除從0x90880位址開始連續32個word資料 */
PageErase(0x0880, 0x2000);
```

15.2.6 SectorErase

- **函數**

```
int SectorErase(unsigned int addr,unsigned int DelayTime);
```

- **函數功能**

清除flash對應位址的一個sector的資料

- **輸入參數**

addr[in]：待清除資料的首位址，16位的位址，Flash空間是從0x90000開始，所以該位址值只需寫16位值，輸入範圍0~0xffff，且一個sector包含32page，所以位址的間隔步長為32*128；所以首位址是以頁來計算的，且需要對齊，位址為0xu000（u為可變的值）。如要想從0x91000開始，函數參數只需填寫0x1000值就可以。

Delay time[in]：延時時間

- **包含標頭檔**

Peripheral_lib/DrvFlash.h

- **函數返回值**

0xFF 設置成功

- **函數用法**

/* 從0x91000位址開始連續清除一個sector的資料*/

SectorErase(0x1000, 0x2000);

15.3 Flash 存儲空間結構

1 page=128 Bytes

1 sector=32 pages=4096 Bytes

Sector & Page			
Sector	Page	Address Range (Byte)	
0	0	000000H	00007FH
	1	000080H	0000FFH
	…	…	…
	30	000F00H	000F7FH
	31	000F80H	000FFFH
1	32	001000H	00107FH
	33	001080H	0010FFH
	…	…	…
	63	001F80H	001FFFH
…	…	…	…
15	480	00F000H	00F07FH
	…	…	…
	511	00FF80H	00FFFFH

16 Revision History

Version	Page	Revision Summary	The Date Of Revision
V14	ALL	First edition	2017/02/20
V15	ALL	1. 新增 Flash 自我燒錄注意事項, 工作電壓 VDD3V 必須高於 2.7V.	2018/02/18
V16	章節 3 章節 13 章節 14 章節 16	新增函式 DrvCLOCK_EnableENHAO 與 DrvCLOCK_SelectIHOSC_CalHAO 新增函式 DrvRTC_EnableWUEn 與 DrvRTC_DisableWUEn 新增函式 DrvI2C_EnableSEn 與 DrvI2C_DisableSEn 與 DrvI2C_EnableI2Cen 修正 Flash Function 的函數返回值描述	2022/12/15

17 C Library Change List

Date	舊版本 Queries List		新版本改善	
	版本	Bug List	版本	改善
2017-2-20	V1.4	無		
2018-2-18	V1.4	1. 使用函式 DrvI2C_SlaveSet 沒有功能 2. 使用以下函式 uint32_t DrvSPI32_IsRxBufferFull(void) uint32_t DrvSPI32_IsTxBufferFull(void) 沒有功能	V1.5	1. 修正 DrvI2C_SlaveSet 函式使用問題. 2. 修正 uint32_t DrvSPI32_IsRxBufferFull(void) uint32_t DrvSPI32_IsTxBufferFull(void) 函式用問題
2022/12/15	V1.5	1. DrvPWM_Open 函式設置問題, 從 111 開始往 110 設置會失效. 2. DrvUART_Open/ DrvUART2_Open 除頻設置問題, 在某些數值運算會有 N+1 的誤差	V1.7	1. 修正函式 DrvPWM_Open, DrvUART_Open/ DrvUART2_Open 2. 新增函式 DrvRTC_EnableWUEn, DrvRTC_DisableWUEn, DrvCLOCK_EnableENHAO, DrvI2C_EnableSEn, DrvI2C_DisableSEn, DrvI2C_EnableI2Cen 3. 新增函式 DrvCLOCK_SelectIHOSC_CalHAO