HY17M24 Series Peripheral Driver C Library

Table of Contents

Attention：

1、HYCON Technology Corp. reserves the right to change the content of this datasheet without further notice. For most up-to-date information, please constantly visit our website：http://www.hycontek.com .

2、HYCON Technology Corp. is not responsible for problems caused by figures or application circuits narrated herein whose related industrial properties belong to third parties.

3、Specifications of any HYCON Technology Corp. products detailed or contained herein stipulate the performance, characteristics, and functions of the specified products in the independent state. We does not guarantee of the performance, characteristics, and functions of the specified products as placed in the customer's products or equipment. Constant and sufficient verification and evaluation is highly advised.

4、Please note the operating conditions of input voltage, output voltage and load current and ensure the IC internal power consumption does not exceed that of package tolerance. HYCON Technology Corp. assumes no responsibility for equipment failures that resulted from using products at values that exceed, even momentarily, rated values listed in products specifications of HYCON products specified herein.

5、Notwithstanding this product has built-in ESD protection circuit, please do not exert excessive static electricity to protection circuit.

6、Products specified or contained herein cannot be employed in applications which require extremely high levels of reliability, such as device or equipment affecting the human body, health/medical equipments, security systems, or any apparatus installed in aircrafts and other vehicles.

7、Despite the fact that HYCON Technology Corp. endeavors to enhance product quality as well as reliability in every possible way, failure or malfunction of semiconductor products may happen. Hence, users are strongly recommended to comply with safety design including redundancy and fire-precaution equipments to prevent any accidents and fires that may follow.

8、Use of the information described herein for other purposes and/or reproduction or copying without the permission of HYCON Technology Corp. is strictly prohibited.

1. Overview

1.1. C Library Introduction

This document describes the HYCON ™ HY17M24 Series driver reference manual. System-level software developers can use the HYCON ™ HY17M24 Series driver to do the fast application software development, instead of using the register level programming, which can reduce the total development time significantly.

1.2. Relative Document

User can find the following documents in our website for other relative information.
http://www.hycontek.com/

(1)HYCON HY17M24 Series Data Sheet
(2)HYCON HY17M24 Series User's Guide
(3)HYCON HY17M24 Series Hardware TOOL User Manual
(4)HYCON HY17M24 Series Software TOOL User Manual

2. SYS Driver

2.1. Introduction

The following functions are included in System Manager Section.

Item	Functions	Description
01	Sleep	Enable low power Sleep Mode
02	Idle	Enable low power Idle Mode
03	GIE_Enable	Enable GIE
04	GIE_Disable	Disable GIE
05	SYS_ReadI2CGCR	Read I2C Reset flag(GCRstIF)
06	SYS_ReadBOR2LV	ReadBOR2 flag of data(BOR2LV)
07	SYS_ReadSKERR	Read SKERR flag(SKERR)
08	SYS_ReadRST	Read reset flag of data (RST)
09	SYS_ReadIDLE	Read Idle Flag (IDL)
10	SYS_ReadWDT	Read watch dog flag (TO)
11	SYS_ReadSLEEP	Read Sleep Flag (PD)
12	SYS_ReadBOR	Read BOR flag of data (BOR)
13	SYS_ClearSKERR	Clear SKERR flag
14	SYS_ClearIDLE	Clear Idle Flag
15	SYS_ClearWDT	Clear watch dog flag
16	SYS_ClearSLEEP	Clear Sleep Flag
17	SYS_ClearBOR	Clear BOR flag
18	CSFON_Enable	Enable CSF(Chip Special Function)controller
19	CSFON_Disable	Disable CSF controller
20	SKRST_Enable	Enable SKRST function
21	SKRST_Disable	Disable SKRST function
22	ENBOR2_Enable	Enable BOR2 function
23	ENBOR2_Disable	Disable BOR2 function
24	BOR2_RSTMode	Select the reset mode of BOR2
25	BOR2_INTMode	Select the interrupt mode of BOR2
26	BOR_THSelect	Set voltage of BOR detection
27	MCLR_Enable	Enable hardware reset pin
28	MCLR_Disable	Disable hardware reset pin

2.2. Functions

2.2.1. Sleep

- **Prototype**

  Sleep();

- **Description**

  Enable low power Sleep Mode.

- **Parameters**

None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

```
/* Enable Sleep Mode. Before enable Sleep Mode, disable no use function. */
Sleep();
```

2.2.2. Idle

- **Prototype**

```
Idle();
```

- **Description**

  Enable low power Idle Mode.

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

```
/* Enable Idle Mode. Before enable Idle Mode, disable no use function. */
Idle();
```

2.2.3. GIE_Enable

- **Prototype**

```
GIE_Enable();
```

- **Description**

  Enable GIE.

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

```
/* Enable GIE */
GIE_Enable();
```

2.2.4. GIE_Disable

- **Pro]totype**

  GIE_Disable();

- **Description**

  Disable GIE.

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

  /* Disable GIE */

  GIE_Disable();

2.2.5. SYS_ReadI2CGCR

- **Prototype**

  SYS_ReadI2CGCR();

- **Description**

  Read I2C Reset flag of data.

  Read the value of register PSTAT[0].

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  0：Normal

  1：The I2C has reset

- **Example**

  /* Read I2C Reset flag of data. */

  unsigned char temp_flag;

  temp_flag= SYS_ReadI2CGCR();

2.2.6. SYS_ReadBOR2LV

- **Prototype**

  SYS_ReadBOR2LV();

- **Description**

  Read BOR2 flag of data.

  Read the value of register PSTAT[1].

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  0：VDD >BOR_TH[2:0]

  1：VDD <=BOR_TH[2:0]

- **Example**

```
/* Read BOR2 flag of data. */
unsigned char temp_flag;
temp_flag= SYS_ReadBOR2LV();
```

2.2.7. SYS_ReadSKERR

- **Prototype**

  SYS_ReadSKERR();

- **Description**

  Read Stack Error flag(SKERR) of data.

  Read the value of register PSTAT[2].

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  0：Clear by BOR, RST or command.

  1：stack has error.

- **Example**

```
/* Read SKERR flag of data. */
unsigned char temp_flag;
temp_flag= SYS_ReadSKERR();
```

2.2.8. SYS_ReadRST

- **Prototype**

  SYS_ReadRST();

- **Description**

  Read external reset pin flag(RST) of data.

  Read the value of register PSTAT[3].

- **Parameters**

  None

- **Include**

Driver/HY17M/HY17M24/RST.h

- **Return Value**

  0：Normal

  1：The Reset Pin has reset

- **Example**

```
/* Read reset flag of data*/
unsigned char flag;
flag=SYS_ReadRST();
```

2.2.9. SYS_ReadIDLE

- **Prototype**

  SYS_ReadIDLE();

- **Description**

  Read Idle flag(RST) of data.

  Read the value of register PSTAT[4].

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  0：Clear by BOR, RST or command.

  1：Chip has entered Idle Mode

- **Example**

```
/* Read Idle flag of data*/
unsigned char flag;
flag=SYS_ReadIDLE();
```

2.2.10. SYS_ReadWDT

- **Prototype**

  SYS_ReadWDT();

- **Description**

  Read watch dog flag(TO) of data.

  Read the value of register PSTAT[5].

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

0：Clear by BOR, RST or command.

1：Watch dog has triggered

- **Example**

```
/* Read watch dog flag of data. */
unsigned char flag;
flag=SYS_WdogFlagRead();
flag=SYS_ReadIDLE();
```

2.2.11. SYS_ReadSLEEP

- **Prototype**

  SYS_ReadSLEEP();

- **Description**

  Read Sleep Flag.

  Read the register PSTAT[6]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  0：Clear by BOR, RST or command.

  1：Chip has entered Sleep Mode

- **Example**

```
/* Read Sleep Flag. */
SYS_ReadSLEEP();
```

2.2.12. SYS_ReadBOR

- **Prototype**

  SYS_ReadBOR();

- **Description**

  Read BOR flag of data.

  Read the value of register PSTAT[7].

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  0：Normal, clear by command.

  1：BOR has triggered

- **Example**

```
/* Read BOR flag of data. */
unsigned char flag;
flag=SYS_ReadBOR();
```

2.2.13. SYS_ClearSKERR

- **Prototype**

  SYS_ClearSKERR();

- **Description**

  Clear Stack Error flag.

  Clear the value of register PSTAT[2].

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

```
/* Clear SKERR flag. */
SYS_ClearSKERR();
```

2.2.14. SYS_ClearIDLE

- **Prototype**

  SYS_ClearIDLE();

- **Description**

  Clear Idle flag.

  Clear the value of register PSTAT[4].

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

```
/* Clear Idle flag */
SYS_ClearIDLE();
```

2.2.15. SYS_ClearWDT

- **Prototype**

  SYS_ClearWDT();

- **Description**

  Clear watch dog flag

  Clear the register PSTAT[5]

- **Parameters**

  None

- **nclude**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

```
/* Clear watch dog flag */
SYS_ClearWDT();
```

2.2.16. SYS_ClearSLEEP

- **Prototype**

  SYS_ClearSLEEP();

- **Description**

  Clear Sleep flag

  Clear the register PSTAT[6]

- **Parameters**

  None

- **nclude**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

```
/* Clear Sleep flag */
SYS_ClearSLEEP();
```

2.2.17. SYS_ClearBOR

- **Prototype**

  SYS_ClearBOR();

- **Description**

  Clear BOR flag

  Clear the value of register PSTAT[7]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

  /* Clear BOR flag */

  SYS_ClearBOR;

2.2.18. CSFON_Enable

- **Prototype**

  CSFON_Enable();

- **Description**

  Enable CSF(Chip Special Function) controller, user must set CSFON[0] before control register CSFCN0 and CSFCN1

  Configure the register PWRCN[0]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

  /* Enable CSF */

  CSFON_Enable();

2.2.19. CSFON_Disable

- **Prototype**

  CSFON_Disable();

- **Description**

  Disable CSF(Chip Special Function) controller.

  Clear the register PWRCN[0]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

  /* Disable CSF */

  CSFON_Disable();

2.2.20. SKRST_Enable

- **Prototype**

  SKRST_Enable();

- **Description**

  Enable stack error reset function.

  Configure the register CSFCN0[7]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

```
/* Enable stack error reset function   */
CSFON_Enable();
SKRST_Enable();
```

2.2.21. SKRST_Disable

- **Prototype**

  SKRST _Disable();

- **Description**

  Disable stack error reset function.

  Clear the register CSFCN0[7].

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

```
/* Disable stack error reset function */
CSFON_Enable();
SKRST _Disable();
```

2.2.22. ENBOR2_Enable

- **Prototype**

  ENBOR2_Enable();

- **Description**

  Enable BOR2 voltage detecting function

Configure the register CSFCN1[0]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

```
/* Enable BOR2 voltage detecting function */
CSFON_Enable();
ENBOR2_Enable();
```

2.2.23. ENBOR2_Disable

- **Prototype**

  ENBOR2_Disable();

- **Description**

  Disable BOR2 voltage detecting function.

  Clear the register CSFCN1[0]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

```
/* Disable BOR2 voltage detecting function */
CSFON_Enable();
ENBOR2_Disable();
```

2.2.24. BOR2_RSTMode

- **Prototype**

  BOR2_RSTMode();

- **Description**

  Enable BOR2 reset mode.

  Configure the register CSFCN1[1]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

```
/* Enable BOR2 reset mode.  */
CSFON_Enable();
BOR2_RSTMode();
```

2.2.25. BOR2_INTMode

- **Prototype**

  BOR2_INTMode();

- **Description**

  Enable BOR2 interrupt mode.

  Clear the register CSFCN1[1]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

```
/* Enable BOR2 interrupt mode.  */
CSFON_Enable();
BOR2_INTMode();
```

2.2.26. BOR_THSelect

- **Prototype**

  BOR_THSelect(sel);

- **Description**

  Select BOR detection of voltage.

  Configure the register CSFCN1[4:2]

- **Parameters**

  sel [in] ：BOR_TH[2:0]

  BORTH_4V0 : When the battery uses 1.5V*4, voltage below 4.0V(=1.0V*4) means low voltage

  BORTH_3V65 : When the battery uses 1.5V*4, voltage below 3.6V(=0.9V*4) means low voltage

  BORTH_3V0 : When the battery uses 1.5V*3, voltage below 3V(=1.0V*3) means low voltage

  BORTH_2V75 : When the battery uses 1.5V*3, voltage below 2.7V(=0.9V*3) means low voltage

  BORTH_2V5 : for VDDA=2.4V mode, VDD>=2.45V

  BORTH_2V2 :

BORTH_2V0 : When the battery uses 1.5V*2, voltage below 2V(=1.0V*2) means low voltage

BORTH_1V7 : Chip power on preset

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

```
/* Set BOR_TH=3.65V.  */
CSFON_Enable();
BOR_THSelect(BORTH_3V65);
```

2.2.27. MCLR_Enable

- **Prototype**

  MCLR_Enable();

- **Description**

  Enable hardware reset pin, set PT1.0 as RST pin.

  Configure the register CSFCN1[7]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

  None

- **Example**

```
/* Enable MCLR.  */
CSFON_Enable();
MCLR_Enable();
```

2.2.28. MCLR_Disable

- **Prototype**

  MCLR_Disable();

- **Description**

  Disable hardware reset pin, set PT1.0 as GPIO pin.

  Clear the register CSFCN1[7]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/RST.h

- **Return Value**

None

- **Example**

```
/* Disable MCLR.   */
CSFON_Enable();
MCLR_Disable();
```

3. CLOCK Driver

3.1. Introduction

The following functions are included in Clock Manager Section.

Item	Functions	Description
01	CLK_CPUCKOpen	Open a high-speed oscillator
02	CLK_XTEnable	Enable external oscillator
03	CLK_XTDisable	Disable external oscillator
04	CLK_XTSelect	Select frequency of external oscillator
05	CLK_HAOEnable	Enable HAO oscillator
06	CLK_HAODisable	Disable HAO oscillator
07	CLK_HAOSelect	Select frequency of HAO oscillator
08	CLK_LPCCKSelect	Select the source of LPC Clock
09	CLK_OSCSelect	Select clock source of the chip
10	CLK_DHSCKSelect	Select DHS clock frequency
11	CLK_DMSCKSelect	Select DMS clock frequency
12	CLK_CPUCKSelect	Select source of CPU clock
13	CLK_ADCCKDivSelect	Select ADC clock frequency
14	CLK_TMBCKSelect	Select TMB clock frequency
15	CLK_TMBCKDivSelect	Select DTMB clock frequency

3.2. Functions

3.2.1. CLK_CPUCKOpen

- **Prototype**

  void CLK_CPUCKOpen(unsigned char haom,
  unsigned char oscs,
  unsigned char dhsck,
  unsigned char cpuck)

- **Description**

  Open a high-speed oscillator, select HAO, CPU clock, DHS_CK clock source.

  Configure the register OSCCN0[7:0]

- **Parameter**

  haom [in]：Configure frequency of HAO oscillator

HAOM_17510KHZ : 17.51MHz

HAOM_8755KHZ : 8.557MHz

HAOM_4147KHZ : 4.147MHz

HAOM_1843KHZ : 1.843MHz

oscs [in]：Select clock source of the chip

OSCS_HAO : HAO

OSCS_LPO : LPO

OSCS_XT : External oscillator

dhsck [in]：Configure DHS clock frequency

DHS_HSCKDIV8：HS_CK/8

DHS_HSCKDIV4：HS_CK/4

DHS_HSCKDIV2：HS_CK/2

DHS_HSCKDIV1：HS_CK/1

cpuck [in]：Select CPU_CK clock source

CPUS_DHSCK ：DHS_CK

CPUS_HSCK ：HS_CK

- **Include**

  Driver/HY17M/HY17M24/CLK.h

- **Return Value**

  None

- **Example**

```
/* Configure HAO frequency to 1.843MHz and use it as the chip clock source, DHS clock frequency to HS_CK/1, and CPU clock source to DHS_CK */
CLK_CPUCKOpen(HAOM_1843KHZ, OSCS_HAO, DHS_HSCKDIV1, CPUS_DHSCK );
```

3.2.2. CLK_XTEnable

- **Prototype**

  CLK_XTEnable()

- **Description**

  Enable external oscillator

  Configure the register OSCCN2[5]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/CKL.h

- **Return Value**

  None

- **Example**

```
/* Enable external oscillator */
```

```
CLK_XTEnable();
```

3.2.3. CLK_XTDisable

- **Prototype**

  CLK_XTDisable()

- **Description**

  Disable external oscillator

  Clear the register OSCCN2[5]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/CKL.h

- **Return Value**

  None

- **Example**

```
/* Disable external oscillator */
CLK_XTDisable();
```

3.2.4. CLK_XTSelect

- **Prototype**

  CLK_XTSelect(XTSel)

- **Description**

  Select external oscillator frequency

  Clear the register OSCCN2[4:3]

- **Parameter**

  XTSel [in]：Select external oscillator frequency

  XTS_16M : 2~16MHz

  XTS_2M : 2MHz(low power)

  XTS_32768 : 32768Hz

- **Include**

  Driver/HY17M/HY17M24/CKL.h

- **Return Value**

  None

- **Example**

```
/*Select 2M Hz of external oscillator */
CLK_XTSelect(XTS_2M);
```

3.2.5. CLK_HAOEnable

- **Prototype**

CLK_HAODisable()

- **Description**

  Enable HAO oscillator

  Clear the register OSCCN2[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/CLK.h

- **Return Value**

  None

- **Example**

```
/* Disable HAO oscillator*/
CLK_HAODisable();
```

3.2.6. CLK_HAODisable

- **Prototype**

  CLK_HAOEnable()

- **Description**

  Enable HAO oscillator

  Configure the register OSCCN2[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/CLK.h

- **Return Value**

  None

- **Example**

```
/* Enable HAO oscillator*/
CLK_HAOEnable();
```

3.2.7. CLK_HAOSelect

- **Prototype**

  CLK_HAOSelect(HAOSel)

- **Description**

  Select frequency of HAO oscillator

  Configure the register OSCCN2[2:1]

- **Parameter**

  HAOSel [in]：Select HAO oscillator frequency

HAOM_17510KHZ : 17.51MHz

HAOM_8755KHZ : 8.557MHz

HAOM_4147KHZ : 4.147MHz

HAOM_1843KHZ : 1.843MHz

- **Include**

  Driver/HY17M/HY17M24/CLK.h

- **Return Value**

  None

- **Example**

  /* Select 1.843M Hz as HAO oscillator */

  CLK_HAOSelect(HAOM_1843KHZ);

3.2.8. CLK_LPCCKSelect

- **Prototype**

  CLK_LPCCKSelect(LPCCKSel)

- **Description**

  Select the source of LPC Clock

  Configure register OSCCN1[6]

- **Parameter**

  LPCCKSel [in]：Select the source of LPC Clock

  LCPS_XT : External oscillator XT

  LCPS_LPO : Low-speed oscillator LPO

- **Include**

  Driver/HY17M/HY17M24/CLK.h

- **Return Value**

  None

- **Example**

  /* Select the LPC Clock from LPO */

  CLK_LPCCKSelect(LCPS_LPO);

3.2.9. CLK_OSCSelect

- **Prototype**

  CLK_OSCSelect(OSCSel)

- **Description**

  Select clock source of the chip

  Configure the register OSCCN0[7:6]

- **Parameter**

  OSCSel [in]：Select clock source of the chip

OSCS_HAO : High-speed oscillator HAO

OSCS_LPO : Low-speed oscillator LPO

OSCS_XT : External oscillator XT

- **Include**

  Driver/HY17M/HY17M24/CLK.h

- **Return Value**

  None

- **Example**

  /*Select HAO oscillator as the chip clock */

  CLK_OSCSelect(OSCS_HAO);

3.2.10. CLK_DHSCKSelect

- **Prototype**

  CLK_DHSCKSelect (DHSCKSel)

- **Description**

  Select DHS clock frequency.

  Configure the register OSCCN0[5:4]

- **Parameter**

  DHSCKSel [in]：Select DHS_CK frequency from HS_CK Pre-scale

  DHS_HSCKDIV1 : HS_CK÷1

  DHS_HSCKDIV2 : HS_CK÷2

  DHS_HSCKDIV4 : HS_CK÷4

  DHS_HSCKDIV8 : HS_CK÷8

- **Include**

  Driver/HY17M/HY17M24/CLK.h

- **Return Value**

  None

- **Example**

  /* Select HS_CK and Pre-scale 1 as DHS_CK*/

  CLK_DHSCKSelect (DHS_HSCKDIV1);

3.2.11. CLK_DMSCKSelect

- **Prototype**

  CLK_DMSCKSelect(DMSCKSel)

- **Description**

  Select DMS clock frequency

  Configure the register OSCCN0[3:1]

- **Parameter**

DMSCKSel [in]：Select DMS_CK frequency from DHS_CK and Pre-scale

DMS_DHSCKDIV2	: DHS_CK÷2
DMS_DHSCKDIV4	: DHS_CK÷4
DMS_DHSCKDIV8	: DHS_CK÷8
DMS_DHSCKDIV16	: DHS_CK÷16
DMS_DHSCKDIV32	: DHS_CK÷32
DMS_DHSCKDIV64	: DHS_CK÷64
DMS_DHSCKDIV128	: DHS_CK÷128
DMS_DHSCKDIV256	: DHS_CK÷256

- **Include**

  Driver/HY17M/HY17M24/CLK.h

- **Return Value**

  None

- **Example**

```
/* Select DHS_CK and Pre-scale 2 as DMS clock */
CLK_DMSCKSelect(DMS_DHSCKDIV2);
```

3.2.12. CLK_CPUCKSelect

- **Prototype**

  CLK_CPUCKSelect(CPUCKSel)

- **Description**

  Select source of CPU clock

  Configure the register OSCCN0[0].

- **Parameter**

  CPUCKSel [in]：Select source of CPU clock

CPUS_DHSCK	: DHS_CK
CPUS_HSCK	: HS_CK

- **Include**

  Driver/HY17M/HY17M24/CLK.h

- **Return Value**

  None

- **Example**

```
/* Select DHS_CK as the CPU clock */
CLK_CPUCKSelect(CPUS_DHSCK);
```

3.2.13. CLK_ADCCKDivSelect

- **Prototype**

  CLK_ADCCKDivSelect(DADCSel)

- **Description**

Select source of ADC clock

Configure the register OSCCN1[5:4].

- **Parameter**

DADCSel [in]：Select DHS_CK and Pre-scale as ADC clock

DADC_DHSCKDIV2	: DHS_CK÷2
DADC_DHSCKDIV4	: DHS_CK÷4
DADC_DHSCKDIV8	: DHS_CK÷8
DADC_DHSCKDIV16	: DHS_CK÷16

- **Include**

Driver/HY17M/HY17M24/CLK.h

- **Return Value**

None

- **Example**

```
/* Select DHS_CK and Pre-scale 2 as ADC clock */
CLK_ADCCKDivSelect(DADC_DHSCKDIV2);
```

3.2.14. CLK_TMBCKSelect

- **Prototype**

CLK_TMBCKSelect(TMBCKSel)

- **Description**

Select source of Timer B clock

Configure the register OSCCN1[1].

- **Parameter**

TMBCKSel [in]：Select source of TMB

TMBS_LPCCK	: LPC_CK
TMBS_HSCK	: HS_CK

- **Include**

Driver/HY17M/HY17M24/CLK.h

- **Return Value**

None

- **Example**

```
/* Select HS_CK as TMB clock */
CLK_TMBCKSelect(TMBS_HSCK);
```

3.2.15. CLK_TMBCKDivSelect

- **Prototype**

CLK_TMBCKDivSelect(TMBCKDivSel)

- **Description**

Select Timer B frequency from TMB_CK and Pre-scale

Configure the register OSCCN1[3:2].

- **Parameter**

  TMBCKSel [in]：Select DTMB_CK frequency

  DTMB_TMBCKDIV1 : TMB_CK÷1

  DTMB_TMBCKDIV2 : TMB_CK÷2

  DTMB_TMBCKDIV4 : TMB_CK÷4

- **DTMB_TMBCKDIV8 : TMB_CK÷8**

- **Include**

  Driver/HY17M/HY17M24/CLK.h

- **Return Value**

  None

- **Example**

```
/* Select TMB_CK and Pre-scale 1 as DTMB_CK frequency */
CLK_TMBCKDivSelect(TMB_CK÷1);
```

4. TIMER/WDT Driver

4.1. Introduction

The following functions are included in Timer Manager Section.

Item	Functions	Description
01	WDT_Open	Enable watchdog timer and set time-out interval
02	WDTIE_Enable	Enable WDT interrupt
03	WDTIE_Disable	Disable WDT interrupt
04	WDTIF_IsFlag	Read the WDT interrupt flag
05	WDTIF_ClearFlag	Clear the WDT interrupt flag
06	WDT_Enable	Enable WDT function
07	WDT_WDTCKDivSelect	Set the WDT time-out interval
08	WDT_Clear	Clear WDT count value
09	BZ_Enable	Enable buzzer function
10	BZ_Disable	Disable buzzer function
11	BZ_BZCKSelect	Select buzzer output frequency
12	BZ_CLKSelect	Select source of buzzer clock
13	TMA_Open	Enable timerA ,set counter value and clock source of TMA
14	TA1IE_Enable	Enable Timer-A interrupt function
15	TA1IE_Disable	Disable Timer-A interrupt function
16	TA1IF_IsFlag	Read the TMA interrupt flag
17	TA1IF_ClearFlag	Clear the TMA interrupt flag
18	TA1CIE_Enable	Enable TMA compare interrupt function
19	TA1CIE_Disable	Disable TMA compare interrupt function
20	TA1CIF_IsFlag	Read the TMA compare interrupt flag
21	TA1CIF_ClearFlag	Clear the TMA compare interrupt flag
22	TMA1Enable	Enable Timer-A function
23	TMA1Disable	Disable Timer-A function
24	TMA1_CLKSelect	Select clock source of TMA
25	TMA1_CLKDiv	Select TMA count frequency
26	TMA1ClearSet	Clear TMA1_CK and DTMA1_CK counter
27	TMA1_CompSet	Set TMA compare value
28	TMA1_CompSet	Clear TMA counter value
29	TMB_Open	Enable Timer-B ,set counter value, clock source and trigger source of TMB
30	TMBIE_Enable	Enable Timer-B interrupt function
31	TMBIE_Disable	Disable Timer-B interrupt function
32	TMBIF_IsFlag	Read the TMB interrupt flag
33	TMBIF_ClearFlag	Clear the TMB interrupt flag
34	TB1Enable	Enable Timer-B function
35	TB1Disable	Disable Timer-B function
36	TB1_ModeSelect	Select TMB mode
37	TB1_TRIG_MODE	Select TMB trigger source
38	TB1_ClearTMB1	Clear TMB counter value
39	TB1_TC1Select	Select source of CPI1 input
40	TB1_PWMO1	Configure the PWMO1 output function
41	TB1_PWMO0	Configure the PWMO0 output function
42	TB1_PWM1ModeSelect	Select the PWM1 output mode
43	TB1_PWM0ModeSelect	Select the PWM0 output mode

44	TB1_PWM1_PHASE	Set the PWM1output phase
45	TB1_PWM0_PHASE	Set the PWM0 output phase
46	TB1C0Set	Set TMB counter value
47	TB1C1Set	Set PWM0/PWM1 count condition parameter
48	TB1C2Set	Set PWM0/PWM1 count condition parameter

4.2. Functions

4.2.1. WDT_Open

- **Prototype**

  Void WDT_Open(unsigned char wdts)

- **Description**

  Enable watchdog timer(WDT) and set time-out interval.

  Configure the register WDTCN[2:0]

- **Parameter**

  wdts [in]：set the WDT time-out interval

  DWDT_WDTCKDIV16384 ：WDT_CK÷16384

  DWDT_WDTCKDIV8192 ：WDT_CK÷8192

  DWDT_WDTCKDIV4096 ：WDT_CK÷4096

  DWDT_WDTCKDIV2048 ：WDT_CK÷2048

  DWDT_WDTCKDIV64 ：WDT_CK÷64

  DWDT_WDTCKDIV32 ：WDT_CK÷32

  DWDT_WDTCKDIV16 ：WDT_CK÷16

  DWDT_WDTCKDIV8 ：WDT_CK÷8

- **Include**

  Driver/HY17M/HY17M24/WDT.h

- **Return Value**

None

- **Example**

  /* Open WDT function and set time-out interval WDT_CK÷32 */

  WDT_Open (DWDT_WDTCKDIV32);

4.2.2. WDTIE_Enable

- **Prototype**

  WDTIE_Enable()

- **Description**

  Enable WDT interrupt.

  Configure the register INTE0[4]=1

- **Parameter**

None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None.

- **Example**

  /* Enable WDT interrupt. */

  WDTIE_Enable();

4.2.3. WDTIE_Disable

- **Prototype**

  WDTIE_Disable()

- **Description**

  Disable WDT interrupt.

  Clear the register INTE0[4]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None.

- **Example**

  /* Disable WDT interrupt. */

  WDTIE_Disable();

4.2.4. WDTIF_IsFlag

- **Prototype**

  WDTIF_IsFlag()

- **Description**

  Read the WDT interrupt flag.

  Read the register INTF0[4]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：Normal

  1：WDT has overflowed.

- **Example**

```
/* Read the WDT interrupt flag. */
unsigned char flag;
flag = WDTIF_IsFlag();
```

4.2.5. WDTIF_ClearFlag

- **Prototype**

  WDTIF_ClearFlag()

- **Description**

  Clear the WDT interrupt flag.

  Clear the register INTF0[4]=0.

- **Parameter**

  None.

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None.

- **Example**

```
/* Clear the WDT interrupt flag.*/
WDTIF_ClearFlag();
```

4.2.6. WDT_Enable

- **Prototype**

  WDT_Enable()

- **Description**

  Enable WDT function.

  Configure the register WDTCN[3]=1.

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/WDT.h

- **Return Value**

  None.

- **Example**

```
/* Enable WDT function. */
WDT_Enable();
```

4.2.7. WDT_WDTCKDivSelect

- **Prototype**

  WDT_WDTCKDivSelect(Sel)

- **Description**

  Set the WDT time-out interval.

  Configure the register WDTCN[2:0]

- **Parameter**

  Sel [in]：select time-out interval.

  DWDT_WDTCKDIV16384 ：WDT_CK÷16384

  DWDT_WDTCKDIV8192 ：WDT_CK÷8192

  DWDT_WDTCKDIV4096 ：WDT_CK÷4096

  DWDT_WDTCKDIV2048 ：WDT_CK÷2048

  DWDT_WDTCKDIV64 ：WDT_CK÷64

  DWDT_WDTCKDIV32 ：WDT_CK÷32

  DWDT_WDTCKDIV16 ：WDT_CK÷16

  DWDT_WDTCKDIV8 ：WDT_CK÷8

- **Include**

  Driver/HY17M/HY17M24/WDT.h

- **Return Value**

  None.

- **Example**

```
/* Set WDT time-out interval WDT_CK÷32. */
WDT_WDTCKDivSelect(WDT_CK÷32);
```

4.2.8. WDT_Clear

- **Prototype**

  WDT_Clear()

- **Description**

  Clear WDT count value.

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/WDT.h

- **Return Value**

  None.

- **Example**

```
/* Clear WDT count. */
WDT_Clear();
```

4.2.9. BZ_Enable

- **Prototype**

  BZ_Enable()

- **Description**

  Enable buzzer function.

  Configure the register WDTCN[7]=1.

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/WDT.h

- **Return Value**

  None

- **Example**

  /* Enable buzzer function */

  BZ_Enable();

4.2.10. BZ_Disable

- **Prototype**

  BZ_Disable()

- **Description**

  Disable buzzer function.

  Clear the register WDTCN[7].

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/WDT.h

- **Return Value**

  None

- **Example**

  /* Disable buzzer function. */

  BZ_Disable();

4.2.11. BZ_BZCKSelect

- **Prototype**

  BZ_BZCKSelect(Sel)

- **Description**

  Select buzzer output frequency.

  Configure the register WDTCN[5:4].

- **Parameter**

  Sel [in]：Select buzzer output frequency

  DBZ_DZCKDIV2 ：Output frequency is BZ_CK÷2

DBZ_DZCKDIV4 : Output frequency is BZ_CK÷4

DBZ_DZCKDIV8 : Output frequency is BZ_CK÷8

DBZ_DZCKDIV16 : Output frequency is BZ_CK÷16

- **Include**

  Driver/HY17M/HY17M24/WDT.h

- **Return Value**

  None

- **Example**

```
/* Select BZ_CK and pre-scale 4 as buzzer output frequency */
BZ_BZCKSelect(DBZ_DZCKDIV4);
```

4.2.12. BZ_CLKSelect

- **Prototype**

  BZ_CLKSelect(Sel)

- **Description**

  Select source of buzzer clock.

  Configure the register WDTCN[6]

- **Parameter**

  Sel [in]：Select source of buzzer clock.

  BZS_LPCCK : LPC_CK

  BZS_LSCK : LS_CK

- **Include**

  Driver/HY17M/HY17M24/WDT.h

- **Return Value**

  None

- **Example**

```
/* Select LPC_CK as buzzer clock source. */
BZ_CLKSelect(BZS_LPCCK);
```

4.2.13. TMA_Open

- **Prototype**

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

- **Description**

  Enable timer A, set counter value and clock source of TMA.

  Configure the register TMA1CN[7:2].

- **Parameter**

  ck [in]：TMA clock(TMA1_CK) source

  TMAS1_DMSCK : From DMS_CK

TMAS1_LPCCK : From LPC_CK

cks [in]：TMA count frequency

DTMA1_TMA1CKDIV2 : TMA1_CK ÷ 2

DTMA1_TMA1CKDIV4 : TMA1_CK ÷ 4

DTMA1_TMA1CKDIV8 : TMA1_CK ÷ 18

DTMA1_TMA1CKDIV16 : TMA1_CK ÷ 16

DTMA1_TMA1CKDIV32 : TMA1_CK ÷ 32

DTMA1_TMA1CKDIV64 : TMA1_CK ÷ 64

DTMA1_TMA1CKDIV128 : TMA1_CK ÷ 128

DTMA1_TMA1CKDIV256 : TMA1_CK ÷ 256

tma_cmp [in]：TMA compare volue(TMA1C)：0x00~0xFF

- **Include**

  Driver/HY17M/HY17M24/WDT.h

- **Return Value**

  None

- **Example**

```
/* Clear Timer A. */
DrvTMA_ClearTMA();
```

4.2.14. TA1IE_Enable

- **Prototype**

  TA1IE_Enable()

- **Description**

  Enable Timer-A interrupt function .

  Configure the register INTE1[7]=1.

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Enable Timer-A interrupt function */
TA1IE_Enable();
```

4.2.15. TA1IE_Disable

- **Prototype**

  TA1IE_Disable()

- **Description**

Disable Timer-A interrupt function.

Clear the register INTE1[7].

- **Parameter**

  Nnoe

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Disable Timer-A interrupt function. */
TA1IE_Disable();`
```

4.2.16. TA1IF_IsFlag

- **Prototype**

  TA1IF_IsFlag()

- **Description**

  Read the TMA interrupt flag.

  Read the register INTF1[7]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read the TMA interrupt flag. */
unsigned char flag ;
flag = TA1IF_IsFlag();
```

4.2.17. TA1IF_ClearFlag

- **Prototype**

  TA1IF_ClearFlag()

- **Description**

  Clear the TMA interrupt flag.

  Clear the register INTF1[7].

- **Parameter**

  None

- **Include**

Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Clear Timer-A interrupt flag */

  TA1IF_ClearFlag();

4.2.18. TA1CIE_Enable

- **Prototype**

  TA1CIE_Enable()

- **Description**

  Enable TMA compare interrupt function.

  Configure the register INTE0[6]=1.

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Enable TMA compare interrupt function. */

  TA1CIE_Enable();

4.2.19. TA1CIE_Disable

- **Prototype**

  TA1CIE_Disable()

- **Description**

  Disable TMA compare interrupt function.

  Clear the register INTE0[6].

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Disable TMA compare interrupt function. */

  TA1CIE_Disable();

4.2.20. TA1CIF_IsFlag

- **Prototype**

  TA1CIF_IsFlag()

- **Description**

  Read the TMA compare interrupt flag.

  Read the register INTF0[6].

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read the TMA compare interrupt flag. */
unsigned char flag ;
flag = TA1CIF_IsFlag();
```

4.2.21. TA1CIF_ClearFlag

- **Prototype**

  TA1CIF_ClearFlag()

- **Description**

  Clear the TMA compare interrupt flag.

  Clear the register INTF0[6].

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Clear Timer-A compare interrupt flag */
TA1CIF_ClearFlag();
```

4.2.22. TMA1Enable

- **Prototype**

  TAM1Enable()

- **Description**

  Enable Timer-A function.

Configure the register TMA1CN[7]=1.

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/TMR.h

- **Return Value**

  None

- **Example**

```
/* Enable Timer-A function. */
TMA1Enable();
```

4.2.23. TMA1Disable

- **Prototype**

  TAM1Disable()

- **Description**

  Disable Timer-A function.

  Clear the register TMA1CN[7].

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/TMR.h

- **Return Value**

  None

- **Example**

```
/* Disable Timer-A function. */
TMA1Disable();
```

4.2.24. TMA1_CLKSelect

- **Prototype**

  TMA1_CLKSelect(Sel)

- **Description**

  Select clock source of TMA.

  Configure the register TMA1CN[5].

- **Parameter**

  ck [in]：Select clock source of TMA(TMA1_CK).

  TMAS1_DMSCK : DMS_CK

  TMAS1_LPCCK : LPC_CK

- **Include**

  Driver/HY17M/HY17M24/TMR.h

- **Return Value**

  None

- **Example**

  /* Select DMS_CK as TMA1_CK clock source */

  TMA1_CLKSelect(TMAS1_DMS);

4.2.25. TMA1_CLKDiv

- **Prototype**

  TMA1_CLKDiv(Sel)

- **Description**

  Select TMA count frequency.

  Configure the register TMA1CN[4:2].

- **Parameter**

  cks [in]：Select TMA count frequency

DTMA1_TMA1CKDIV2	: TMA1_CK / 2
DTMA1_TMA1CKDIV4	: TMA1_CK / 4
DTMA1_TMA1CKDIV8	: TMA1_CK / 18
DTMA1_TMA1CKDIV16	: TMA1_CK / 16
DTMA1_TMA1CKDIV32	: TMA1_CK / 32
DTMA1_TMA1CKDIV64	: TMA1_CK / 64
DTMA1_TMA1CKDIV128	: TMA1_CK / 128
DTMA1_TMA1CKDIV256	: TMA1_CK / 256

- **Include**

  Driver/HY17M/HY17M24/TMR.h

- **Return Value**

  None

- **Example**

  /* Set TMA count frequency is TMA1_CK / 16 */

  TMA1_CLKDiv(TMA1_CK / 16);

4.2.26. TMA1ClearSet

- **Prototype**

  TMA1ClearSet()

- **Description**

  Clear TMA1_CK and DTMA1_CK counter.

  Configure the register TMA1CN[6].

- **Parameter**

  None

- **Include**

Driver/HY17M/HY17M24/TMR.h

- **Return Value**

  None

- **Example**

```
/*  Clear TMA1_CK and DTMA1_CK counter */
TMA1ClearSet();
```

4.2.27. TMA1_CompSet

- **Prototype**

  TMA1_CompSet(Set)

- **Description**

  Set TMA compare value(TMA1C). When the TMA1R value is added to the same as TMA1C, the TA1CIF flag is set to 1.

  Configure the register TMA1C[7:0].

- **Parameter**

  Set [in]：0x00~0xFF

- **Include**

  Driver/HY17M/HY17M24/TMR.h

- **Return Value**

  None

- **Example**

```
/*  Set TAM1C is 0x0f.*/
TMA1_CompSet(0x0f);
```

4.2.28. TMA1_CompSet

- **Prototype**

  TMA1_CompSet(Set)

- **Description**

  Clear TMA counter value(TMA1R).

  Clear the register TMA1R[7:0].

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/TMR.h

- **Return Value**

  None

- **Example**

```
/*  Clear TMA1R value. */
TMA1_ClearTMA1();
```

4.2.29. TMB_Open

- **Prototype**

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

- **Description**

  Enable Timer-B ,set counter value, clock source and trigger source of TMB

  Configure the register TB1CN0[7:0].

- **Parameter**

  ck [in] : Selcet TMB clock source

  TMBS_LPCCK : LPC_CK

  TMBS_HSCK : HS_CK

  cks [in] : Select TMB clock frequency

  DTMB_TMBCKDIV1 : TMB_CK÷1

  DTMB_TMBCKDIV2 : TMB_CK÷2

  DTMB_TMBCKDIV4 : TMB_CK÷4

  DTMB_TMBCKDIV8 : TMB_CK÷8

  mode [in] : Select TMB mode

  TB1M_16bit : 16-bit Timer

  TB1M_17bit : 17-bit Timer

  TB1M_2_8bit : two 8-bit Timer

  TB1M_8_8bit : 8+8-bit Timer

  trig_mode [in] : Select trigger source

  TB1RT_LogicH : Logic High, always trigger

  TB1RT_CP11 : CPI1, trigger condition base on input compare result.

- **Include**

  Driver/HY17M/HY17M24/TMR.h

- **Example**

```
/* Enable TMB · Set HS_CK as TMB clock source, TMB_CK and Pre-scale 1 as DTMB_CK frequency,
TMB mode is 16-bit Timer and Logic High as trigger source */
TMB_Open(TMBS_HSCK ,DTMB1_TMB1CKDIV2 ,TB1M_16bit ,TB1RT_LogicH );
```

4.2.30. TMBIE_Enable

- **Prototype**

  TMBIE_Enable()

- **Description**

  Enable Timer-B interrupt function.

Configure the register INTE0[3]=1.

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Enable Timer-B interrupt function. */
TMBIE_Enable();
```

4.2.31. TMBIE_Disable

- **Prototype**

  TMBIE_Disable()

- **Description**

  Disable Timer-B interrupt function.

  Clear the register INTE0[3].

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Disable Timer-B interrupt function. */
TMBIE_Disable();
```

4.2.32. TMBIF_IsFlag

- **Prototype**

  TMBIF_IsFlag()

- **Description**

  Read the TMB interrupt flag.

  Read the register INTF0[3].

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

0：No interrupt

1：Interrupt occurred

- **Example**

/* Read the TMB interrupt flag. */

TMBIF_IsFlag();

4.2.33. TMBIF_ClearFlag

- **Prototype**

TMBIF_ClearFlag()

- **Description**

Clear the TMB interrupt flag.

Clear the register INTF0[3].

- **Parameter**

None

- **Include**

Driver/HY17M/HY17M24/INT.h

- **Return Value**

None

- **Example**

/* Clear the TMB interrupt flag.*/

TMBIF_ClearFlag();

4.2.34. TB1Enable

- **Prototype**

TB1Enable()

- **Description**

Enable Timer-B function.

Configure the register TB1CN0[7]=1.

- **Parameter**

None

- **Include**

Driver/HY17M/HY17M24/TMR.h

- **Return Value**

None

- **Example**

/* Enable Timer-B function. */

TMB_Enable();

4.2.35. TB1Disable

- **Prototype**

  TB1Disable()

- **Description**

  Disable Timer-B function.

  Clear the register TB1CN0[7].

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/TMR.h

- **Return Value**

  None

- **Example**

```
/* Disable Timer-B function. */
TMB_Disable();
```

4.2.36. TB1_ModeSelect

- **Prototype**

  TB1_ModeSelect(Sel)

- **Description**

  Select TMB mode.

  Configure the register TB1CN0[6:5].

- **Parameter**

  Sel [in]：Select TMB mode

  TB1M_16bit : 16-bit Timer

  TB1M_17bit : 17-bit Timer

  TB1M_2_8bit : two 8-bit Timer

  TB1M_8_8bit : 8+8-bit Timer

- **Include**

  Driver/HY17M/HY17M24/TMR.h

- **Return Value**

  None

- **Example**

```
/* Set TMB is 16-bit timer mode */
TB1_ModeSelect ( TB1M_16bit );
```

4.2.37. TB1_TRIG_MODE

- **Prototype**

  TB1_TRIG_MODE(Sel)

- **Description**

Select TMB trigger source

Configure the register TB1CN0[4:3].

- **Parameter**

trig_mode [in] : Select trigger source

TB1RT_LogicH : Logic High, always trigger

TB1RT_CP11 : CPI1, trigger condition base on input compare result.

- **Include**

Driver/HY17M/HY17M24/TMR.h

- **Return Value**

None

- **Example**

```
/* Set Logic High as TMB trigger source */
TB1_TRIG_MODE(TB1RT_LogicH);
```

4.2.38. TB1_ClearTMB1

- **Prototype**

TB1_ClearTMB1()

- **Description**

Clear TMB counter value.

Configure the register TB1CN0[2]=1

- **Parameter**

None

- **Include**

Driver/HY17M/HY17M24/TMR.h

- **Return Value**

None

- **Example**

```
/* Clear TMB counter value. */
TB1_ClearTMB1();
```

4.2.39. TB1_TC1Select

- **Prototype**

TB1_TC1Select(Set)

- **Description**

Select source of CPI1 input

Configure the register TC1CN0[6:5].

- **Parameter**

Set [in]：Select CPI1 input source

TC1S_TBI0 : TBI0

TC1S_TBI1 : TBI1

TC1S_LSCK : LS_CK

TC1S_OPC : OPC

- **Include**

  Driver/HY17M/HY17M24/TMR.h

- **Return Value**

  None

- **Example**

```
/*  Select LPC_CK as CPI1 source */
TB1_TC1Select (TC1S_LSCK);
```

4.2.40. TB1_PWMO1

- **Prototype**

  TB1_PWMO1(Sel)

- **Description**

  Configure the PWMO1 output function

  Configure the register TB1CN0[1].

- **Parameter**

  Sel [in]：Configure PWM1 output controller

  PWMO1_DISABLE ：Disable PWM1 output

  PWMO1_OUTPUT ：Enable PWM1 output

- **Include**

  Driver/HY17M/HY17M24/TMR.h

- **Return Value**

  None

- **Example**

```
/* Enable PWMO1 output */
TB1_PWMO1(PWMO1_OUTPUT) ;
```

4.2.41. TB1_PWMO0

- **Prototype**

  TB1_PWMO0(Sel)

- **Description**

  Configure the PWMO0 output function

  Configure the register TB1CN0[0].

- **Parameter**

  Sel [in]：Configure PWM0 output controller

PWMO0_DISABLE ：Disable PWM0 output

PWMO0_OUTPUT ：Enable PWM0 output

- **Include**

  Driver/HY17M/HY17M24/TMR.h

- **Return Value**

  None

- **Example**

```
/* Enable PWMO0 output */
TB1_PWMO1(PWMO0_OUTPUT) ;
```

4.2.42. TB1_PWM1ModeSelect

- **Prototype**

  TB1_PWM1ModeSelect(Sel)

- **Description**

  Select the PWM1 output mode.

  Configure the register TB1CN1[6:4].

- **Parameter**

  Sel [in]：Set PWM1 waveform

  PWMA1_PWM1O ：PWM1O

  PWMA1_PWM2O ：PWM2O

  PWMA1_PWM3O ：PWM3O

  PWMA1_PWM4O ：PWM4O

  PWMA1_PWM5O ：PWM5O

  PWMA1_PWM6O ：PWM6O

  PWMA1_PWM7O ：PWM7O

- **Include**

  Driver/HY17M/HY17M24/TMR.h

- **Return Value**

  None

- **Example**

```
/* Select PWM1 output waveform is PWM4O */
TB1_PWM1ModeSelect(PWMA1_PWM4O);
```

4.2.43. TB1_PWM0ModeSelect

- **Prototype**

  TB1_PWM0ModeSelect(Sel)

- **Description**

  Select the PWM0 output mode.

  Configure the register TB1CN1[2:0].

- **Parameter**

  Sel [in]：Set PWM0 waveform

  PWMA0_PWM1O ：PWM1O

  PWMA0_PWM2O ：PWM2O

  PWMA0_PWM3O ：PWM3O

  PWMA0_PWM4O ：PWM4O

  PWMA0_PWM5O ：PWM5O

  PWMA0_PWM6O ：PWM6O

  PWMA0_PWM7O ：PWM7O

- **Include**

  Driver/HY17M/HY17M24/TMR.h

- **Return Value**

  None

- **Example**

```
/* Select PWM0 output waveform is PWM3O */
TB1_PWM0ModeSelect(PWMA0_PWM3O);
```

4.2.44. TB1_PWM1_PHASE

- **Prototype**

  TB1_PWM1_PHASE(Sel)

- **Description**

  Set the PWM1 output phase.

  Configure the register TB1CN1[7].

- **Parameter**

  Sel [in]：Set the PWM1 output phase.

  PA1IV_INVER ：PWM1 output inver

  PA1IV_NORMAL ：PWM1 output normal

- **Include**

  Driver/HY17M/HY17M24/TMR.h

- **Return Value**

  None

- **Example**

```
/* Set PWM1 output inver */
TB1_PWM1_PHASE(PA1IV_INVER);
```

4.2.45. TB1_PWM0_PHASE

- **Prototype**

  TB1_PWM0_PHASE(Sel)

- **Description**

Set the PWM0 output phase.

Configure the register TB1CN1[3].

- **Parameter**

Sel [in]：Set the PWM0 output phase.

PA0IV_INVER ：PWM0 output inver

PA0IV_NORMAL ：PWM0 output normal

- **Include**

Driver/HY17M/HY17M24/TMR.h

- **Return Value**

None

- **Example**

```
/* Set PWM0 output inver */
TB1_PWM0_PHASE(PA0IV_INVER);
```

4.2.46. TB1C0Set

- **Prototype**

void TB1C0Set(unsigned int TMB)

- **Description**

Set TMB counter value.

Configure the register TB1C0[15:0]

- **Parameter**

TMB [in]：Set TMB counter overflow value.

0x0000~0xFFFF：The setting range is different according to the PWM mode.

- **Include**

Driver/HY17M/HY17M24/TMR.h

- **Return Value**

None

- **Example**

```
/* Set TB1C0 is 0x00ff */
TB1C0Set(0x00ff);
```

4.2.47. TB1C1Set

- **Prototype**

void TB1C1Set(unsigned int TMB)

- **Description**

Set PWM0/PWM1 count condition parameter.

Configure the register TB1C1[15:0]

- **Parameter**

TMB [in]：Set PWM count condition parameter.

0x0000~0xFFFF：The setting range is different according to the PWM mode.

- **Include**

  Driver/HY17M/HY17M24/TMR.h

- **Return Value**

  None

- **Example**

```
/* Set TB1C1 is 0x00ff */
TB1C1Set(0x00ff);
```

4.2.48. TB1C2Set

- **Prototype**

  void TB1C2Set(unsigned int TMB)

- **Description**

  Set PWM0/PWM1 count condition parameter.

  Configure the register TB1C2[15:0]

- **Parameter**

  TMB [in]：Set PWM count condition parameter.

  0x0000~0xFFFF：The setting range is different according to the PWM mode.

- **Include**

  Driver/HY17M/HY17M24/TMR.h

- **Return Value**

  None

- **Example**

```
/* Set TB1C2 is 0x00ff */
TB1C2Set(0x00ff);
```

5. GPIO Driver

5.1. Introduction

The following functions are included in GPIO Manager Section.

Item	Functions	Description
01	GPIO_OpenPT1Input	Enable the input mode and enable pull-up resistor of the specified pin
02	GPIO_OpenPT2Input	Enable the input mode and enable pull-up resistor of the specified pin
03	GPIO_OpenPT3Input	Enable the input mode and enable pull-up resistor of the specified pin
04	GPIO_OpenPT1Output	Enable the output mode of the specified pin and output high
05	GPIO_OpenPT2Output	Enable the output mode of the specified pin and output high
06	GPIO_OpenPT3Output	Enable the output mode of the specified pin and output high
07	GPIO_PT1OutputMode	Enable the output mode of the specified pin
08	GPIO_PT1OutputHigh	Set the specified pin output high
09	GPIO_PT1OutputLow	Set the specified pin output low
10	GPIO_PT1InputMode	Enable the input mode of the specified pin
11	GPIO_PT1SETPU	Enable pull-up resistor of the specified pin
12	GPIO_PT1CLRPU	Disable pull-up resistor of the specified pin
13	GPIO_PT1SETPUAll	Enable pull-up resistor of PT1
14	GPIO_PT1InputEnable	Enable digital function of the specified pin
15	GPIO_PT1InputDisable	Disable digital function of the specified pin
16	GPIO_PT1SETDA	Enable analogy function of the specified pin
17	GPIO_PT1CLRDA	Disable analogy function of the specified pin
18	GPIO_PT1GET	Read the status of the specified pin
19	E0IE_Enable	Enable PT1.0 external interrupt function
20	E0IE_Disable	Disable PT1.0 external interrupt function
21	E0IF_IsFlag	Read PT1.0 external interrupt flag
22	E0IF_ClearFlag	Clear PT1.0 external interrupt flag
23	GPIO_INTEG0Sel	Select the trigger method of PT1.0 interrupt
24	E1IE_Enable	Enable PT1.1 external interrupt function
25	E1IE_Disable	Disable PT1.1 external interrupt function
26	E1IF_IsFlag	Read PT1.1 external interrupt flag
27	E1IF_ClearFlag	Clear PT1.1 external interrupt flag
28	GPIO_INTEG1Sel	Select the trigger method of PT1.1 interrupt
29	E2IE_Enable	Enable PT1.2 external interrupt function
30	E2IE_Disable	Disable PT1.2 external interrupt function
31	E2IF_IsFlag	Read PT1.2 external interrupt flag
32	E2IF_ClearFlag	Clear PT1.2 external interrupt flag
33	INTG12_Edgerise	Set edge rise to trigger PT1.2 interrupt
34	INTG12_Edgefall	Set edge fall to trigger PT1.2 interrupt
35	E3IE_Enable	Enable PT1.3 external interrupt function
36	E3IE_Disable	Disable PT1.3 external interrupt function
37	E3IF_IsFlag	Read PT1.3 external interrupt flag
38	E3IF_ClearFlag	Clear PT1.3 external interrupt flag
39	INTG13_Edgerise	Set edge rise to trigger PT1.3 interrupt

40	INTG13_Edgefall	Set edge fall to trigger PT1.3 interrupt
41	INTE14_Enable	Enable PT1.4 interrupt function
42	INTE14_Disable	Disable PT1.4 interrupt function
43	INTF14_IsFlag	Read PT1.4 interrupt flag
44	INTF14_ClearFlag	Clear PT1.4 interrupt flag
45	INTG14_Edgerise	Set edge rise to trigger PT1.4 interrupt
46	INTG14_Edgefall	Set edge fall to trigger PT1.4 interrupt
47	INTE15_Enable	Enable PT1.5 interrupt function
48	INTE15_Disable	Disable PT1.5 interrupt function
49	INTF15_IsFlag	Read PT1.5 interrupt flag
50	INTF15_ClearFlag	Clear PT1.5 interrupt flag
51	INTG15_Edgerise	Set edge rise to trigger PT1.5 interrupt
52	INTG15_Edgefall	Set edge fall to trigger PT1.5 interrupt
53	INTE16_Enable	Enable PT1.6 interrupt function
54	INTE16_Disable	Disable PT1.6 interrupt function
55	INTF16_IsFlag	Read PT1.6 interrupt flag
56	INTF16_ClearFlag	Clear PT1.6 interrupt flag
57	INTG16_Edgerise	Set edge rise to trigger PT1.6 interrupt
58	INTG16_Edgefall	Set edge fall to trigger PT1.6 interrupt
59	GPIO_PM10Sel	Select the main function of PT1.0
60	GPIO_PM12Sel	Select the main function of PT1.2
61	GPIO_PM13Sel	Select the main function of PT1.3
62	GPIO_PM15Sel	Select the main function of PT1.5
63	GPIO_PM16Sel	Select the main function of PT1.6
64	GPIO_PT2OutputMode	Enable the output mode of the specified pin
65	GPIO_PT2OutputHigh	Set the specified pin output high
66	GPIO_PT2OutputLow	Set the specified pin output low
67	GPIO_PT2InputMode	Enable the input mode of the specified pin
68	GPIO_PT2SETPU	Enable pull-up resistor of the specified pin
69	GPIO_PT2CLRPU	Disable pull-up resistor of the specified pin
70	GPIO_PT2SETPUAll	Enable pull-up resistor of PT2
71	GPIO_PT2InputEnable	Enable digital function of the specified pin
72	GPIO_PT2InputDisable	Disable digital function of the specified pin
73	GPIO_PT2GET	Read the status of the specified pin
74	INTE20_Enable	Enable PT2.0 interrupt function
75	INTE20_Disable	Disable PT2.0 interrupt function
76	INTF20_IsFlag	Read PT2.0 interrupt flag
77	INTF20_ClearFlag	Clear PT2.0 interrupt flag
78	INTG20_Edgerise	Set edge rise to trigger PT2.1 interrupt
79	INTG20_Edgefall	Set edge fall to trigger PT2.1 interrupt
80	INTE21_Enable	Enable PT2.1 interrupt function
81	INTE21_Disable	Disable PT2.1 interrupt function
82	INTF21_IsFlag	Read PT2.1 interrupt flag
83	INTF21_ClearFlag	Clear PT2.1 interrupt flag
84	INTG21_Edgerise	Set edge rise to trigger PT2.1 interrupt
85	INTG21_Edgefall	Set edge fall to trigger PT2.1 interrupt
86	GPIO_PM20Sel	Select the main function of PT2.0
87	GPIO_PM21Sel	Select the main function of PT2.1
88	GPIO_PT3OutputMode	Enable the output mode of the specified pin
89	GPIO_PT3OutputHigh	Set the specified pin output high
90	GPIO_PT3OutputLow	Set the specified pin output low
91	GPIO_PT3InputMode	Enable the input mode of the specified pin
92	GPIO_PT3SETPU	Enable pull-up resistor of the specified pin
93	GPIO_PT3CLRPU	Disable pull-up resistor of the specified pin
94	GPIO_PT3SETPUAll	Enable pull-up resistor of PT3
95	GPIO_PT3InputEnable	Enable digital function of the specified pin
96	GPIO_PT3InputDisable	Disable digital function of the specified pin
97	GPIO_PT3SETDA	Enable analogy function of the specified pin

98	GPIO_PT3CLRDA	Disable analogy function of the specified pin
99	GPIO_PT3GET	Read the status of the specified pin
100	INTE30_Enable	Enable PT3.0 interrupt function
101	INTE30_Disable	Disable PT3.0 interrupt function
102	INTF30_IsFlag	Read PT3.0 interrupt flag
103	INTF30_ClearFlag	Clear PT3.0 interrupt flag
104	INTG30_Edgerise	Set edge rise to trigger PT3.0 interrupt
105	INTG30_Edgefall	Set edge fall to trigger PT3.0 interrupt
106	INTE31_Enable	Enable PT3.1 interrupt function
107	INTE31_Disable	Disable PT3.1 interrupt function
108	INTF31_IsFlag	Read PT3.1 interrupt flag
109	INTF31_ClearFlag	Clear PT3.1 interrupt flag
110	INTG31_Edgerise	Set edge rise to trigger PT3.1 interrupt
111	INTG31_Edgefall	Set edge fall to trigger PT3.1 interrupt
112	INTE32_Enable	Enable PT3.2 interrupt function
113	INTE32_Disable	Disable PT3.2 interrupt function
114	INTF32_IsFlag	Read PT3.2 interrupt flag
115	INTF32_ClearFlag	Clear PT3.2 interrupt flag
116	INTG32_Edgerise	Set edge rise to trigger PT3.2 interrupt
117	INTG32_Edgefall	Set edge fall to trigger PT3.2 interrupt
118	INTE33_Enable	Enable PT3.3 interrupt function
119	INTE33_Disable	Disable PT3.3 interrupt function
120	INTF33_IsFlag	Read PT3.3 interrupt flag
121	INTF30_ClearFlag	Clear PT3.3 interrupt flag
122	INTG33_Edgerise	Set edge rise to trigger PT3.3 interrupt
123	INTG33_Edgefall	Set edge fall to trigger PT3.3 interrupt
124	INTE34_Enable	Enable PT3.4 interrupt function
125	INTE34_Disable	Disable PT3.4 interrupt function
126	INTF34_IsFlag	Read PT3.4 interrupt flag
127	INTF34_ClearFlag	Clear PT3.4 interrupt flag
128	INTG34_Edgerise	Set edge rise to trigger PT3.4 interrupt
129	INTG34_Edgefall	Set edge fall to trigger PT3.4 interrupt
130	INTE35_Enable	Enable PT3.5 interrupt function
131	INTE35_Disable	Disable PT3.5 interrupt function
132	INTF35_IsFlag	Read PT3.5 interrupt flag
133	INTF35_ClearFlag	Clear PT3.5 interrupt flag
134	INTG35_Edgerise	Set edge rise to trigger PT3.5 interrupt
135	INTG35_Edgefall	Set edge fall to trigger PT3.5 interrupt
136	INTE36_Enable	Enable PT3.6 interrupt function
137	INTE36_Disable	Disable PT3.6 interrupt function
138	INTF36_IsFlag	Read PT3.6 interrupt flag
139	INTF36_ClearFlag	Clear PT3.6 interrupt flag
140	INTG36_Edgerise	Set edge rise to trigger PT3.6 interrupt
141	INTG36_Edgefall	Set edge fall to trigger PT3.6 interrupt
142	INTE37_Enable	Enable PT3.7 interrupt function
143	INTE37_Disable	Disable PT3.7 interrupt function
144	INTF37_IsFlag	Read PT3.7 interrupt flag
145	INTF37_ClearFlag	Clear PT3.7 interrupt flag
146	INTG37_Edgerise	Set edge rise to trigger PT3.7 interrupt
147	INTG37_Edgefall	Set edge fall to trigger PT3.7 interrupt
148	GPIO_PM30Sel	Select the main function of PT3.0
149	GPIO_PM32Sel	Select the main function of PT3.2
150	GPIO_PM33Sel	Select the main function of PT3.3
151	GPIO_PM34Sel	Select the main function of PT3.4
152	GPIO_PM36Sel	Select the main function of PT3.6

5.2. Functions

5.2.1. GPIO_OpenPT1Input

- **Prototype**

  void GPIO_OpenPT1Input(unsigned char InputBits, unsigned char PullHighBits)

- **Description**

  Enable the input mode and enable pull-up resistor of the specified pin.

  Configure the register TRISC1/PT1PU.

- **Parameter**

  InputBitst [in]：Enable the input mode of the specified pin

  TC10_H：PT1.0

  TC11_H：PT1.1

  TC12_H：PT1.2

  TC13_H：PT1.3

  TC14_H：PT1.4

  TC15_H：PT1.5

  TC16_H：PT1.6

  PullHighBits [in]：Enable pull-up resistor of the specified pin

  PU10_H：Enable pull-up resistor of PT1.0

  PU11_H：Enable pull-up resistor of PT1.1

  PU12_H：Enable pull-up resistor of PT1.2

  PU13_H：Enable pull-up resistor of PT1.3

  PU14_H：Enable pull-up resistor of PT1.4

  PU15_H：Enable pull-up resistor of PT1.5

  PU16_H：Enable pull-up resistor of PT1.6

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

  /* Enable the input mode and enable pull-up resistor of the pin PT1.4/PT1.5 */

  GPIO_OpenPT1Input(TC14_H, PU14_H);

  GPIO_OpenPT1Input(TC15_H, PU15_H);

5.2.2. GPIO_OpenPT2Input

- **Prototype**

  void GPIO_OpenPT2Input(unsigned char InputBits,unsigned char PullHighBits)

- **Description**

Enable the input mode and enable pull-up resistor of the specified pin.

Configure the register TRISC2/PT2PU.

- **Parameter**

InputBitst [in]：Enable the input mode of the specified pin

TC20_H：PT2.0

TC21_H：PT2.1

PullHighBits [in]：Enable pull-up resistor of the specified pin

PU20_H：Enable pull-up resistor of PT2.0

PU21_H：Enable pull-up resistor of PT2.1.

- **Include**

Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

None

- **Example**

```
/* Enable the input mode and enable pull-up resistor of the pin PT2.0/PT2.1 */
GPIO_OpenPT2Input(TC20_H, PU20_H);
GPIO_OpenPT2Input(TC21_H, PU21_H);
```

5.2.3. GPIO_OpenPT3Input

- **Prototype**

void GPIO_OpenPT3Input(unsigned char InputBits, unsigned char PullHighBits)

- **Description**

Enable the input mode and enable pull-up resistor of the specified pin

Configure the register TRISC3/PT3PU.

- **Parameter**

InputBitst [in]：Enable the input mode of the specified pin

TC30_H：PT3.0

TC31_H：PT3.1

TC32_H：PT3.2

TC33_H：PT3.3

TC34_H：PT3.4

TC35_H：PT3.5

TC36_H：PT3.6

TC37_H：PT3.6

PullHighBits [in]：Enable pull-up resistor of the specified pin

PU30_H：Enable pull-up resistor of PT3.0

PU31_H：Enable pull-up resistor of PT3.1

PU32_H：Enable pull-up resistor of PT3.2

PU33_H：Enable pull-up resistor of PT3.3

PU34_H：Enable pull-up resistor of PT3.4

PU35_H：Enable pull-up resistor of PT3.5

PU36_H：Enable pull-up resistor of PT3.6

PU37_H：Enable pull-up resistor of PT3.7

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Enable the input mode and enable pull-up resistor of the pin PT3.4/PT3.5 */
GPIO_OpenPT3Input(TC34_H, PU34_H);
GPIO_OpenPT3Input(TC35_H, PU35_H);
```

5.2.4. GPIO_OpenPT1Output

- **Prototype**

  void GPIO_OpenPT1Output(unsigned char OutputBits,unsigned char OutHighBits)

- **Description**

  Enable the output mode of the specified pin and output high.

  Configure the register TRISC1/PT1

- **Parameter**

  InputBitst [in]：Enable the output mode of the specified pin

  TC11_H：PT1.1

  TC12_H：PT1.2

  TC13_H：PT1.3

  TC14_H：PT1.4

  TC15_H：PT1.5

  TC16_H：PT1.6

  OutHighBits [in]：Set the specified pin output high

  PT11_H：PT1.1 output high

  PT12_H：PT1.2 output high

  PT13_H：PT1.3 output high

  PT14_H：PT1.4 output high

  PT15_H：PT1.5 output high

  PT16_H：PT1.6 output high

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

  /* Enable the output mode of PT1.4/PT1.5 and output high. */

  GPIO_OpenPT1Output (TC14_H ,PT14_H);

  GPIO_OpenPT1Output (TC15_H ,PT15_H);

5.2.5. GPIO_OpenPT2Output

- **Prototype**

  void GPIO_OpenPT2Output(unsigned char OutputBits,unsigned char OutHighBits)

- **Description**

  Enable the output mode of the specified pin and output high.

  Configure the register TRISC2/PT2

- **Parameter**

  InputBitst [in]：Enable the output mode of the specified pin

  TC20_H：PT2.0

  TC21_H：PT2.1

  OutHighBits [in]：Set the specified pin output high

  PT20_H：PT2.0 output high

  PT21_H：PT2.1 output high

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

  /* Enable the output mode of PT2.1/PT2.0 and output high. */

  GPIO_OpenPT2Output (TC21_H ,PT20_H);

  GPIO_OpenPT2Output (TC21_H ,PT20_H);

5.2.6. GPIO_OpenPT3Output

- **Prototype**

  void GPIO_OpenPT3Output(unsigned char OutputBits,unsigned char OutHighBits)

- **Description**

  Enable the output mode of the specified pin and output high.

  Configure the register TRISC3/PT3

- **Parameter**

  InputBitst [in]：Enable the output mode of the specified pin

  TC30_H：PT3.0

  TC31_H：PT3.1

  TC32_H：PT3.2

TC33_H：PT3.3

TC34_H：PT3.4

TC35_H：PT3.5

TC36_H：PT3.6

TC37_H：PT3.7

OutHighBits [in]：Set the specified pin output high

PT30_H：PT3.0 output high

PT31_H：PT3.1 output high

PT32_H：PT3.2 output high

PT33_H：PT3.3 output high

PT34_H：PT3.4 output high

PT35_H：PT3.5 output high

PT36_H：PT3.6 output high

PT37_H：PT3.7 output high

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Enable the output mode of PT3.4/PT3.5 and output high. */
GPIO_OpenPT3Output ( TC34_H ,PT34_H );
GPIO_OpenPT3Output ( TC35_H ,PT35_H );
```

5.2.7. GPIO_PT1OutputMode

- **Prototype**

  GPIO_PT1OutputMode(BitSet)

- **Description**

  Enable the output mode of the specified pin .

  Configure the register TRISC1

- **Parameter**

  BitSet [in]：Enable the output mode of the specified pin .

  TC11_H：PT1.1

  TC12_H：PT1.2

  TC13_H：PT1.3

  TC14_H：PT1.4

  TC15_H：PT1.5

  TC16_H：PT1.6

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

  /* Enable the output mode of the PT1.6 . */

  GPIO_PT1OutputMode(TC16_H);

5.2.8. GPIO_PT1OutputHigh

- **Prototype**

  GPIO_PT1OutputHigh(BitSet)

- **Description**

  Set the specified pin output high.

  Configure the register PT1[6:0]

- **Parameter**

  BitSet [in]：Set the specified pin output high.

  PT11_H：PT1.1

  PT12_H：PT1.2

  PT13_H：PT1.3

  PT14_H：PT1.4

  PT15_H：PT1.5

  PT16_H：PT1.6

- **include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

  /* Set the PT1.6 output high. */

  GPIO_PT1OutputMode(TC16_H);

  GPIO_PT1OutputHigh(PT16_H);

5.2.9. GPIO_PT1OutputLow

- **Prototype**

  GPIO_PT1OutputLow(BitSet)

- **Description**

  Set the specified pin output low.

  Clear the register PT1[6:0]

- **Parameter**

  BitSet [in]：Set the specified pin output low.

  PT11_H：PT1.1

PT12_H：PT1.2

PT13_H：PT1.3

PT14_H：PT1.4

PT15_H：PT1.5

PT16_H：PT1.6

- **include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Set the PT1.5 output low. */
GPIO_PT1OutputMode(TC15_H);
GPIO_PT1OutputLow(PT15_H);
```

5.2.10. GPIO_PT1InputMode

- **Prototype**

  GPIO_PT1InputMode(BitSet)

- **Description**

  Enable the input mode of the specified pin .

  Configure the register TRISC1

- **Parameter**

  BitSet [in]：Enable the input mode of the specified pin .

  TC10_H：PT1.0

  TC11_H：PT1.1

  TC12_H：PT1.2

  TC13_H：PT1.3

  TC14_H：PT1.4

  TC15_H：PT1.5

  TC16_H：PT1.6

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Enable the input mode of the PT1.6 . */
GPIO_PT1InputMode(TC16_H);
```

5.2.11. GPIO_PT1SETPU

- **Prototype**

GPIO_PT1SETPU(BitSet)

- **Description**

  Enable pull-up resistor of the specified pin.

  Configure the register PT1PU.

- **Parameter**

  BitSet [in]：Enable pull-up resistor of the specified pin.

  PU10_H：Enable PT1.0 pull-up resistor

  PU11_H：Enable PT1.1 pull-up resistor

  PU12_H：Enable PT1.2 pull-up resistor

  PU13_H：Enable PT1.3 pull-up resistor

  PU14_H：Enable PT1.4 pull-up resistor

  PU15_H：Enable PT1.5 pull-up resistor

  PU16_H：Enable PT1.6 pull-up resistor

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Enable pull-up resistor of the PT1.4. */
GPIO_PT1SETPU ( PU14_H );
```

5.2.12. GPIO_PT1CLRPU

- **Prototype**

  GPIO_PT1CLRPU(BitSet)

- **Description**

  Disable pull-up resistor of the specified pin.

  Clear the register PT1PU.

- **Parameter**

  BitSet [in]：Disable pull-up resistor of the specified pin.

  PU10_H：Disable PT1.0 pull-up resistor

  PU11_H：Disable PT1.1 pull-up resistor

  PU12_H：Disable PT1.2 pull-up resistor

  PU13_H：Disable PT1.3 pull-up resistor

  PU14_H：Disable PT1.4 pull-up resistor

  PU15_H：Disable PT1.5 pull-up resistor

  PU16_H：Disable PT1.6 pull-up resistor

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

None

- **Example**

/* Disable pull-up resistor of the PT1.4. */

GPIO_PT1CLRPU (PU14_H);

5.2.13. GPIO_PT1SETPUAll

- **Prototype**

GPIO_PT1SETPUAll()

- **Description**

Enable pull-up resistor of PT1.

Configure the register PT1PU

- **Parameter**

None

- **Include**

Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

None

- **Example**

/* Enable pull-up resistor of PT1 */

GPIO_PT1SETPUAll();

5.2.14. GPIO_PT1InputEnable

- **Prototype**

GPIO_PT1InputEnable(BitSet)

- **Description**

Enable digital function of the specified pin.

Configure the register PT1IN

- **Parameter**

BitSet [in]：Enable digital function of the specified pin.

IN10_H：Enable digital function of the PT1.0

IN11_H：Enable digital function of the PT1.1

IN12_H：Enable digital function of the PT1.2

IN13_H：Enable digital function of the PT1.3

IN14_H：Enable digital function of the PT1.4

IN15_H：Enable digital function of the PT1.5

IN16_H：Enable digital function of the PT1.6

- **Include**

Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

None

- **Example**

/* Enable digital function of the PT1.5 */

GPIO_PT1InputEnable(IN15_H);

5.2.15. GPIO_PT1InputDisable

- **Prototype**

GPIO_PT1InputDisable(BitSet)

- **Description**

Disable digital function of the specified pin.
Clear the register PT1IN

- **Parameter**

BitSet [in]：Disable digital function of the specified pin.

IN10_H：Disable digital function of the PT1.0

IN11_H：Disable digital function of the PT1.1

IN12_H：Disable digital function of the PT1.2

IN13_H：Disable digital function of the PT1.3

IN14_H：Disable digital function of the PT1.4

IN15_H：Disable digital function of the PT1.5

IN16_H：Disable digital function of the PT1.6

- **Include**

Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

None

- **Example**

/* Disable digital function of the PT1.5 */

GPIO_PT1InputDisable(IN15_H);

5.2.16. GPIO_PT1SETDA

- **Prototype**

GPIO_PT1SETDA(BitSet)

- **Description**

Enable analogy function of the specified pin.

Configure the register PT1DA

- **Parameter**

BitSet [in]：Enable analogy function of the specified pin

DA10_H：Enable analogy function of the PT1.0

DA11_H：Enable analogy function of the PT1.1

DA12_H：Enable analogy function of the PT1.2

DA13_H：Enable analogy function of the PT1.3

DA14_H：Enable analogy function of the PT1.4

DA15_H：Enable analogy function of the PT1.5

DA16_H：Enable analogy function of the PT1.6

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Enable analogy function of the PT1.5. */
GPIO_PT1SETDA(DA15_H);
```

5.2.17. GPIO_PT1CLRDA

- **Prototype**

  GPIO_PT1CLRDA(BitSet)

- **Description**

  Disable analogy function of the specified pin.

  Clear the register PT1DA

- **Parameter**

  BitSet [in]：Disable analogy function of the specified pin

  DA10_H：Disable analogy function of the PT1.0

  DA11_H：Disable analogy function of the PT1.1

  DA12_H：Disable analogy function of the PT1.2

  DA13_H：Disable analogy function of the PT1.3

  DA14_H：Disable analogy function of the PT1.4

  DA15_H：Disable analogy function of the PT1.5

  DA16_H：Disable analogy function of the PT1.6

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Disable analogy function of the PT1.5. */
GPIO_PT1CLRDA(DA15_H);
```

5.2.18. GPIO_PT1GET

- **Prototype**

  GPIO_PT1GET(BitSet)

- **Description**

Read the status of the specified pin

Read the register PT1

- **Parameter**

BitSet [in]：Read the status of the specified pin

PT10_H：Read the status of the PT1.0

PT11_H：Read the status of the PT1.1

PT12_H：Read the status of the PT1.2

PT13_H：Read the status of the PT1.3

PT14_H：Read the status of the PT1.4

PT15_H：Read the status of the PT1.5

PT16_H：Read the status of the PT1.6

- **Include**

Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

None

- **Example**

```
/* Read the status of the specified pin PT1.5 and PT1.6 */
unsigned char PT1_DATA;
PT1_DATA = GPIO_PT1GET(PT15_H | PT16_H);
```

5.2.19. E0IE_Enable

- **Prototype**

E0IE_Enable()

- **Description**

Enable PT1.0 external interrupt function.

Configure the register INTE0[0]=1

- **Parameter**

None

- **Include**

Driver/HY17M/HY17M24/INT.h

- **Return Value**

None

- **Example `**

```
/* Enable PT1.0 external interrupt function. */
E0IE_Enable();
```

5.2.20. E0IE_Disable

- **Prototype**

E0IE_Disable()

- **Description**

  Disable PT1.0 external interrupt function.

  Clear the register INTE0[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Disable PT1.0 external interrupt function. */
E0IE_Disable();
```

5.2.21. E0IF_IsFlag

- **Prototype**

  E0IF_IsFlag()

- **Description**

  Read PT1.0 external interrupt flag.

  Read the register INTF0[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read PT1.0 external interrupt flag*/
unsigned char flag;
flag = E0IF_IsFlag();
```

5.2.22. E0IF_ClearFlag

- **Prototype**

  E0IF_ClearFlag()

- **Description**

  Clear PT1.0 external interrupt flag.

  Clear the register INTF0[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Clear PT1.0 external interrupt flag*/
unsigned char flag;
E0IF_ClearFlag();
```

5.2.23. GPIO_INTEG0Sel

- **Prototype**

  GPIO_INTEG0Sel(EG0Sel)

- **Description**

  Select the trigger method of PT1.0 interrupt .

  Configure the register PT1M1[1:0]

- **Parameter**

  EG0Sel [in]：Select the trigger method of PT1.0 interrupt

  INTEG0_LEV ：0->1 or 1->0

  INTEG0_EDGERISE：0->1

  INTEG0_EDGEFALL：1->0

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/*  Select the trigger method of PT1.0 interrupt is edge fall . */
GPIO_INTEG0Sel(INTEG0_EDGEFALL);
```

5.2.24. E1IE_Enable

- **Prototype**

  E1IE_Enable()

- **Description**

  Enable PT1.1 external interrupt function.

  Configure the register INTE0[1]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

None

- **Example `**

/* Enable PT1.1 external interrupt function. */

E1IE_Enable();

5.2.25. E1IE_Disable

- **Prototype**

E1IE_Disable()

- **Description**

Disable PT1.1 external interrupt function.

Clear the register INTE0[1]

- **Parameter**

None

- **Include**

Driver/HY17M/HY17M24/INT.h

- **Return Value**

None

- **Example**

/* Disable PT1.1 external interrupt function. */

E1IE_Disable();

5.2.26. E1IF_IsFlag

- **Prototype**

E1IF_IsFlag()

- **Description**

Read PT1.1 external interrupt flag.

Read the register INTF0[1]

- **Parameter**

None

- **Include**

Driver/HY17M/HY17M24/INT.h

- **Return Value**

0：No interrupt

1：Interrupt occurred

- **Example**

/* Read PT1.1 external interrupt flag*/

unsigned char flag;

flag = E1IF_IsFlag();

5.2.27. E1IF_ClearFlag

- **Prototype**

  E1IF_ClearFlag()

- **Description**

  Clear PT1.1 external interrupt flag.

  Clear the register INTF0[1]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Clear PT1.1 external interrupt flag*/
unsigned char flag;
E1IF_ClearFlag();
```

5.2.28. GPIO_INTEG1Sel

- **Prototype**

  GPIO_INTEG1Sel(EG1Sel)

- **Description**

  Select the trigger method of PT1.1 interrupt.

  Configure the register PT1M1[3:2]

- **Parameter**

  EG1Sel [in]：Select the trigger method of PT1.1 interrupt

  INTEG1_LEV ：0->1 or 1->0

  INTEG1_EDGERISE：0->1

  INTEG1_EDGEFALL：1->0

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/*  Select the trigger method of PT1.1 interrupt is edge fall . */
GPIO_INTEG1Sel(INTE1_EDGEFALL);
```

5.2.29. E2IE_Enable

- **Prototype**

  E2IE_Enable()

- **Description**

  Enable PT1.2 external interrupt function.

  Configure the register INTE1[0]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example `**

```
/* Enable PT1.2 external interrupt function. */
E2IE_Enable();
```

5.2.30. E2IE_Disable

- **Prototype**

  E2IE_Disable()

- **Description**

  Disable PT1.2 external interrupt function.

  Clear the register INTE1[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Disable PT1.2 external interrupt function. */
E2IE_Disable();
```

5.2.31. E2IF_IsFlag

- **Prototype**

  E2IF_IsFlag()

- **Description**

  Read PT1.2 external interrupt flag.

  Read the register INTF1[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read PT1.2 external interrupt flag*/
unsigned char flag;
flag = E2IF_IsFlag();
```

5.2.32. E2IF_ClearFlag

- **Prototype**

  E2IF_ClearFlag()

- **Description**

  Clear PT1.2 external interrupt flag.

  Clear the register INTF1[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Clear PT1.2 external interrupt flag*/
unsigned char flag;
E2IF_ClearFlag();
```

5.2.33. INTG12_Edgerise

- **Prototype**

  INTG12_Edgerise()

- **Description**

  Set edge rise to trigger PT1.2 interrupt .

  Configure the register PT1INT[2]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Set edge rise to trigger PT1.2 interrupt .*/
```

INTG12_Edgerise();

5.2.34. INTG12_Edgefall

- **Prototype**

  INTG12_Edgefall()

- **Description**

  Set edge fall to trigger PT1.2 interrupt .

  Clear the register PT1INT[2]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Set edge fall to trigger PT1.2 interrupt .*/

  INTG12_Edgefall();

5.2.35. E3IE_Enable

- **Prototype**

  E3IE_Enable()

- **Description**

  Enable PT1.3 external interrupt function.

  Configure the register INTE1[1]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example `**

  /* Enable PT1.3 external interrupt function. */

  E3IE_Enable();

5.2.36. E3IE_Disable

- **Prototype**

  E3IE_Disable()

- **Description**

  Disable PT1.3 external interrupt function.

Clear the register INTE1[1]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Disable PT1.3 external interrupt function. */
E3IE_Disable();
```

5.2.37. E3IF_IsFlag

- **Prototype**

  E3IF_IsFlag()

- **Description**

  Read PT1.3 external interrupt flag.

  Read the register INTF1[1]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read PT1.3 external interrupt flag*/
unsigned char flag;
flag = E3IF_IsFlag();
```

5.2.38. E3IF_ClearFlag

- **Prototype**

  E3IF_ClearFlag()

- **Description**

  Clear PT1.3 external interrupt flag.

  Clear the register INTF1[1]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Clear PT1. 3 external interrupt flag*/

  unsigned char flag;

  E3IF_ClearFlag();

5.2.39. INTG13_Edgerise

- **Prototype**

  INTG13_Edgerise()

- **Description**

  Set edge rise to trigger PT1.3 interrupt .

  Configure the register PT1INT[3]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Set edge rise to trigger PT1.3 interrupt .*/

  INTG13_Edgerise();

5.2.40. INTG13_Edgefall

- **Prototype**

  INTG13_Edgefall()

- **Description**

  Set edge fall to trigger PT1.3 interrupt .

  Clear the register PT1INT[3]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Set edge fall to trigger PT1.3 interrupt .*/

  INTG13_Edgefall();

5.2.41. INTE14_Enable

- **Prototype**

  INTE14_Enable()

- **Description**

  Enable PT1.4 interrupt function .

  Configure the register PT1INTE[4]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Enable PT1.4 interrupt function */

  INTE14_Enable();

5.2.42. INTE14_Disable

- **Prototype**

  INTE14_Disable()

- **Description**

  Disable PT1.4 interrupt function .

  Clear the register PT1INTE[4]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Disable PT1.4 interrupt function */

  INTE14_Disable();

5.2.43. INTF14_IsFlag

- **Prototype**

  INTF14_IsFlag()

- **Description**

  Read PT1.4 interrupt flag.

  Read the register PT1INTF[4]

- **Parameter**

None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read PT1.4 interrupt flag */
unsigned char flag;
flag = INTF14_IsFlag();
```

5.2.44. INTF14_ClearFlag

- **Prototype**

  INTF14_ClearFlag()

- **Description**

  Clear PT1.4 interrupt flag.

  Clear the register PT1INTF[4]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Clear PT1.4 interrupt flag */
INTF14_ClearFlag();
```

5.2.45. INTG14_Edgerise

- **Prototype**

  INTG14_Edgerise()

- **Description**

  Set edge rise to trigger PT1.4 interrupt.

  Configure the register PT1INT[4]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Set edge rise to trigger PT1.4 interrupt. */

  INTG14_Edgerise();

5.2.46. INTG14_Edgefall

- **Prototype**

  INTG14_Edgefall()

- **Description**

  Set edge fall to trigger PT1.4 interrupt.

  Clear the register PT1INT[4]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Set edge fall to trigger PT1.4 interrupt. */

  INTG14_Edgefall();

5.2.47. INTE15_Enable

- **Prototype**

  INTE15_Enable()

- **Description**

  Enable PT1.5 interrupt function .

  Configure the register PT1INTE[5]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Enable PT1.5 interrupt function */

  INTE15_Enable();

5.2.48. INTE15_Disable

- **Prototype**

  INTE15_Disable()

- **Description**

  Disable PT1.5 interrupt function .

  Clear the register PT1INTE[5]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Disable PT1.5 interrupt function */
INTE15_Disable();
```

5.2.49. INTF15_IsFlag

- **Prototype**

  INTF15_IsFlag()

- **Description**

  Read PT1.5 interrupt flag.

  Read the register PT1INTF[5]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read PT1.5 interrupt flag */
unsigned char flag;
flag = INTF15_IsFlag();
```

5.2.50. INTF15_ClearFlag

- **Prototype**

  INTF15_ClearFlag()

- **Description**

  Clear PT1.5 interrupt flag.

  Clear the register PT1INTF[5]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Clear PT1.5 interrupt flag */

  INTF15_ClearFlag();

5.2.51. INTG15_Edgerise

- **Prototype**

  INTG15_Edgerise()

- **Description**

  Set edge rise to trigger PT1.5 interrupt.

  Configure the register PT1INT[5]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Set edge rise to trigger PT1.5 interrupt. */

  INTG15_Edgerise();

5.2.52. INTG15_Edgefall

- **Prototype**

  INTG15_Edgefall()

- **Description**

  Set edge fall to trigger PT1.5 interrupt.

  Clear the register PT1INT[5]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Set edge fall to trigger PT1.5 interrupt. */

  INTG15_Edgefall();

5.2.53. INTE16_Enable

- **Prototype**

  INTE16_Enable()

- **Description**

  Enable PT1.6 interrupt function .

  Configure the register PT1INTE[6]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Enable PT1.6 interrupt function */
INTE16_Enable();
```

5.2.54. INTE16_Disable

- **Prototype**

  INTE16_Disable()

- **Description**

  Disable PT1.6 interrupt function .

  Clear the register PT1INTE[6]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Disable PT1.6 interrupt function */
INTE16_Disable();
```

5.2.55. INTF16_IsFlag

- **Prototype**

  INTF16_IsFlag()

- **Description**

  Read PT1.6 interrupt flag.

  Read the register PT1INTF[6]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read PT1.6 interrupt flag */
unsigned char flag;
flag = INTF16_IsFlag();
```

5.2.56. INTF16_ClearFlag

- **Prototype**

  INTF16_ClearFlag()

- **Description**

  Clear PT1.6 interrupt flag.

  Clear the register PT1INTF[6]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Clear PT1.6 interrupt flag */
INTF16_ClearFlag();
```

5.2.57. INTG16_Edgerise

- **Prototype**

  INTG16_Edgerise()

- **Description**

  Set edge rise to trigger PT1.6 interrupt.

  Configure the register PT1INT[6]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

None

- **Example**

/* Set edge rise to trigger PT1.6 interrupt. */

INTG16_Edgerise();

5.2.58. INTG16_Edgefall

- **Prototype**

INTG16_Edgefall()

- **Description**

Set edge fall to trigger PT1.6 interrupt.

Clear the register PT1INT[6]

- **Parameter**

None

- **Include**

Driver/HY17M/HY17M24/INT.h

- **Return Value**

None

- **Example**

/* Set edge fall to trigger PT1.6 interrupt. */

INTG16_Edgefall();

5.2.59. GPIO_PM10Sel

- **Prototype**

GPIO_PM10Sel(PM10Sel)

- **Description**

Select the main function of PT1.0 multiple function pin.

Configure the register PT1M2[0]

- **Parameter**

PM10Sel [in]：Select the function of PT1.0

PM10_VOHL : GPIO function

PM10_TBI0_input : TBI0 input

- **Include**

Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

None

- **Example**

/* Set PT1.0 as TBI0 input pin*/

GPIO_PM10Sel(PM10_TBI0_input);

5.2.60. GPIO_PM12Sel

- **Prototype**

  GPIO_PM12Sel(PM12Sel)

- **Description**

  Select the main function of PT1.2 multiple function pin.

  Configure the register PT1M2[4]

- **Parameter**

  PM12Sel [in]： Select the function of PT1.2

  PM12_VOHL : GPIO function

  PM12_PWM1 : PWM1 output pin

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Set PT1.2 as PWM1 output pin*/
GPIO_PM12Sel(PM12_PWM1);
```

5.2.61. GPIO_PM13Sel

- **Prototype**

  GPIO_PM13Sel(PM13Sel)

- **Description**

  Select the main function of PT1.3 multiple function pin

  Configure the register PT1M2[7:6]

- **Parameter**

  PM13Sel [in]：Select the function of PT1.3

  PM13_VOHL : GPIO function

  PM13_PWM0_1 : PWM0 output pin

  PM13_SCL : I2C SCL pin

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Set PT1.3 as SCL pin */
GPIO_PM13Sel(PM13_SCL);
```

5.2.62. GPIO_PM15Sel

- **Prototype**

GPIO_PM15Sel(PM15Sel)

- **Description**

  Select the main function of PT1.5 multiple function pin.

  Configure the register PT1M3[3:2]

- **Parameter**

  PM15Sel [in]：Select the function of PT1.5

  PM15_VOHL : GPIO function

  PM15_BZ : BZ output pin

  PM15_TX : EUART TX pin

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Set PT1.5 as TX pin */
GPIO_PM15Sel(PM15_TX);
```

5.2.63. GPIO_PM16Sel

- **Prototype**

  GPIO_PM16Sel(PM16Sel)

- **Description**

  Select the main function of PT1.6 multiple function pin.

  Configure the register PT1M3[4]

- **Parameter**

  PM16Sel [in]：Select the function of PT1.6

  PM16_VOHL : GPIO function

  PM16_PWM0 : PWM0 output pin

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Set PT1.6 as PWM0 output pin*/
GPIO_PM16Sel(PM16_PWM0);
```

5.2.64. GPIO_PT2OutputMode

- **Prototype**

  GPIO_PT2OutputMode(BitSet)

- **Description**

Enable the output mode of the specified pin .

Configure the register TRISC2

- **Parameter**

BitSet [in]：Enable the output mode of the specified pin .

TC20_H：PT2.0

TC21_H：PT2.1

- **Include**

Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

None

- **Example**

/* Enable the output mode of the PT2.0 . */

GPIO_PT2OutputMode(TC20_H);

5.2.65. GPIO_PT2OutputHigh

- **Prototype**

GPIO_PT2OutputHigh(BitSet)

- **Description**

Set the specified pin output high

Configure the register PT2[1:0]

- **Parameter**

BitSet [in]：Set the specified pin output high.

PT20_H：PT2.0

PT21_H：PT2.1

- **include**

Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

None

- **Example**

/* Set the PT2.0 output high. */

GPIO_PT2OutputMode(TC2.0_H);

GPIO_PT2OutputHigh(PT2.0_H);

5.2.66. GPIO_PT2OutputLow

- **Prototype**

GPIO_PT2OutputLow(BitSet)

- **Description**

Set the specified pin output low.

Clear the register PT2[2:0]

- **Parameter**

  BitSet [in]：Set the specified pin output low.

  PT20_H：PT2.0

  PT21_H：PT2.1

- **include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Set the PT2.0 output low. */
GPIO_PT2OutputMode(TC20_H);
GPIO_PT2OutputLow(PT20_H);
```

5.2.67. GPIO_PT2InputMode

- **Prototype**

  GPIO_PT2InputMode(BitSet)

- **Description**

  Enable the input mode of the specified pin .

  Configure the register TRISC2

- **Parameter**

  BitSet [in]：Enable the input mode of the specified pin .

  TC20_H：PT2.0

  TC21_H：PT2.1

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Enable the input mode of the PT2.1 . */
GPIO_PT2InputMode(TC21_H);
```

5.2.68. GPIO_PT2SETPU

- **Prototype**

  GPIO_PT2SETPU(BitSet)

- **Description**

  Enable pull-up resistor of the specified pin.

  Configure the register PT2PU.

- **Parameter**

BitSet [in]：Enable pull-up resistor of the specified pin.

PU20_H：Enable PT2.0 pull-up resistor

PU21_H：Enable PT2.1 pull-up resistor

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Enable pull-up resistor of the PT2.1. */
GPIO_PT2SETPU ( PU21_H );
```

5.2.69. GPIO_PT2CLRPU

- **Prototype**

  GPIO_PT2CLRPU(BitSet)

- **Description**

  Disable pull-up resistor of the specified pin.

  Clear the register PT1PU.

- **Parameter**

  BitSet [in]：Disable pull-up resistor of the specified pin.

  PU20_H：Disable PT2.0 pull-up resistor

  PU21_H：Disable PT2.1 pull-up resistor

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Disable pull-up resistor of the PT2.1. */
GPIO_PT2CLRPU ( PU21_H );
```

5.2.70. GPIO_PT2SETPUAll

- **Prototype**

  GPIO_PT2SETPUAll()

- **Description**

  Enable pull-up resistor of PT2.

  Configure the register PT2PU

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

  /* Enable pull-up resistor of PT2 */

  GPIO_PT2SETPUAll();

5.2.71. GPIO_PT2InputEnable

- **Prototype**

  GPIO_PT2InputEnable(BitSet)

- **Description**

  Enable digital function of the specified pin.
  Configure the register PT2IN

- **Parameter**

  BitSet [in]：Enable digital function of the specified pin.

  IN20_H：Enable digital function of the PT2.0

  IN21_H：Enable digital function of the PT2.1

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

  /* Enable digital function of the PT2.1 */

  GPIO_PT2InputEnable(IN21_H);

5.2.72. GPIO_PT2InputDisable

- **Prototype**

  GPIO_PT2InputDisable(BitSet)

- **Description**

  Disable digital function of the specified pin.
  Clear the register PT2IN

- **Parameter**

  BitSet [in]：Disable digital function of the specified pin.

  IN20_H：Disable digital function of the PT2.0

  IN21_H：Disable digital function of the PT2.1

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Disable digital function of the PT2.1 */
GPIO_PT2InputDisable(IN21_H);
```

5.2.73. GPIO_PT2GET

- **Prototype**

  GPIO_PT2GET(BitSet)

- **Description**

  Read the status of the specified pin.

  Read the register PT2

- **Parameter**

  BitSet [in]：Read the status of the specified pin

  PT20_H：Read the status of the PT2.0

  PT20_H：Read the status of the PT2.1

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Read the status of the specified pin PT2.0 */
unsigned char PT2_DATA;
PT2_DATA = GPIO_PT2GET(PT20_H);
```

5.2.74. INTE20_Enable

- **Prototype**

  INTE20_Enable()

- **Description**

  Enable PT2.0 interrupt function .

  Configure the register PT2INTE[0]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Enable PT2.0 interrupt function */
INTE20_Enable();
```

5.2.75. INTE20_Disable

- **Prototype**

  INTE20_Disable()

- **Description**

  Disable PT20 interrupt function .

  Clear the register PT2INTE[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Disable PT2.0 interrupt function */
INTE20_Disable();
```

5.2.76. INTF20_IsFlag

- **Prototype**

  INTF20_IsFlag()

- **Description**

  Read PT2.0 interrupt flag.

  Read the register PT2INTF[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read PT2.0 interrupt flag */
unsigned char flag;
flag = INTF20_IsFlag();
```

5.2.77. INTF20_ClearFlag

- **Prototype**

  INTF20_ClearFlag()

- **Description**

  Clear PT2.0 interrupt flag.

  Clear the register PT2INTF[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Clear PT2.0 interrupt flag */
INTF20_ClearFlag();
```

5.2.78. INTG20_Edgerise

- **Prototype**

  INTG20_Edgerise()

- **Description**

  Set edge rise to trigger PT2.0 interrupt.

  Configure the register PT2INT[0]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Set edge rise to trigger PT2.0 interrupt. */
INTG20_Edgerise();
```

5.2.79. INTG20_Edgefall

- **Prototype**

  INTG20_Edgefall()

- **Description**

  Set edge fall to trigger PT2.0 interrupt.

  Clear the register PT2INT[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Set edge fall to trigger PT2.0 interrupt. */
INTG20_Edgefall();
```

5.2.80. INTE21_Enable

- **Prototype**

  INTE21_Enable()

- **Description**

  Enable PT2.1 interrupt function .

  Configure the register PT2INTE[1]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Enable PT2.1 interrupt function */
INTE21_Enable();
```

5.2.81. INTE21_Disable

- **Prototype**

  INTE21_Disable()

- **Description**

  Disable PT21 interrupt function .

  Clear the register PT2INTE[1]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Disable PT2.1 interrupt function */
INTE21_Disable();
```

5.2.82. INTF21_IsFlag

- **Prototype**

  INTF21_IsFlag()

- **Description**

Read PT2.1 interrupt flag.

Read the register PT2INTF[1]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read PT2.1 interrupt flag */
unsigned char flag;
flag = INTF21_IsFlag();
```

5.2.83. INTF21_ClearFlag

- **Prototype**

  INTF21_ClearFlag()

- **Description**

  Clear PT2.1 interrupt flag.

  Clear the register PT2INTF[1]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Clear PT2.1 interrupt flag */
INTF21_ClearFlag();
```

5.2.84. INTG21_Edgerise

- **Prototype**

  INTG21_Edgerise()

- **Description**

  Set edge rise to trigger PT2.1 interrupt.

  Configure the register PT2INT[1]=1

- **Parameter**

  None

- **Include**

Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Set edge rise to trigger PT2.1 interrupt. */

  INTG21_Edgerise();

5.2.85. INTG21_Edgefall

- **Prototype**

  INTG21_Edgefall()

- **Description**

  Set edge fall to trigger PT2.1 interrupt.

  Clear the register PT2INT[1]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Set edge fall to trigger PT2.1 interrupt. */

  INTG21_Edgefall();

5.2.86. GPIO_PM20Sel

- **Prototype**

  GPIO_PM20Sel(PM20Sel)

- **Description**

  Select the main function of PT2.0 multiple function pin.

  Configure the register PT2M1[1:0]

- **Parameter**

  PM20Sel [in]：Select the function of PT2.0

  PM20_VOHL : GPIO function

  PM20_PWM1_1 : PWM1 output pin

  PM20_TX_1 : UART TX pin

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Set PT2.0 as PWM1 output pin*/
GPIO_PM20Sel(PM20_PWM1_1);
```

5.2.87. GPIO_PM21Sel

- **Prototype**

  GPIO_PM21Sel(PM21Sel)

- **Description**

  Select the main function of PT2.1 multiple function pin.

  Configure the register PT2M1[3:2]

- **Parameter**

  PM21Sel [in]：Select the function of PT2.1

PM20_VOHL	: GPIO function
PM21_SCL_1	: I2C SCL pin
PM21_BZ_1	: BZ output pin
PM21_TBI1_input	: TBI1 input pin

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Set PT2.1 as BZ output pin*/
GPIO_PM21Sel(PM21_BZ_1);
```

5.2.88. GPIO_PT3OutputMode

- **Prototype**

  GPIO_PT3OutputMode(BitSet)

- **Description**

  Enable the output mode of the specified pin .

  Configure the register TRISC3

- **Parameter**

  BitSet [in]：Enable the output mode of the specified pin .

  TC30_H：PT3.0

  TC31_H：PT3.1

  TC32_H：PT3.2

  TC33_H：PT3.3

  TC34_H：PT3.4

  TC35_H：PT3.5

  TC36_H：PT3.6

TC37_H：PT3.7

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

  /* Enable the output mode of the PT3.6 . */

  GPIO_PT3OutputMode(TC36_H);

5.2.89. GPIO_PT3OutputHigh

- **Prototype**

  GPIO_PT3OutputHigh(BitSet)

- **Description**

  Set the specified pin output high.

  Configure the register PT3

- **Parameter**

  BitSet [in]：Set the specified pin output high.

  PT30_H：PT3.0

  PT31_H：PT3.1

  PT32_H：PT3.2

  PT33_H：PT3.3

  PT34_H：PT3.4

  PT35_H：PT3.5

  PT36_H：PT3.6

  PT37_H：PT3.7

- **include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

  /* Set the PT3.6 output high. */

  GPIO_PT3OutputMode(TC36_H);

  GPIO_PT3OutputHigh(PT36_H);

5.2.90. GPIO_PT3OutputLow

- **Prototype**

  GPIO_PT3OutputLow(BitSet)

- **Description**

  Set the specified pin output low.

Clear the register PT3

- **Parameter**

  BitSet [in]：Set the specified pin output low.

  PT30_H：PT3.0

  PT31_H：PT3.1

  PT32_H：PT3.2

  PT33_H：PT3.3

  PT34_H：PT3.4

  PT35_H：PT3.5

  PT36_H：PT3.6

  PT37_H：PT3.7

- **include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

  /* Set the PT3.5 output low. */

  GPIO_PT3OutputMode(TC35_H);

  GPIO_PT3OutputLow(PT35_H);

5.2.91. GPIO_PT3InputMode

- **Prototype**

  GPIO_PT3InputMode(BitSet)

- **Description**

  Enable the input mode of the specified pin .

  Configure the register TRISC3

- **Parameter**

  BitSet [in]：Enable the input mode of the specified pin .

  TC30_H：PT3.0

  TC31_H：PT3.1

  TC32_H：PT3.2

  TC33_H：PT3.3

  TC34_H：PT3.4

  TC35_H：PT3.5

  TC36_H：PT3.6

  TC37_H：PT3.7

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

None

- **Example**

/* Enable the input mode of the PT3.6 . */

GPIO_PT3InputMode(TC36_H);

5.2.92. GPIO_PT3SETPU

- **Prototype**

GPIO_PT3SETPU(BitSet)

- **Description**

Enable pull-up resistor of the specified pin.

Configure the register PT3PU.

- **Parameter**

BitSet [in]：Enable pull-up resistor of the specified pin.

PU30_H：Enable PT3.0 pull-up resistor

PU31_H：Enable PT3.1 pull-up resistor

PU32_H：Enable PT3.2 pull-up resistor

PU33_H：Enable PT3.3 pull-up resistor

PU34_H：Enable PT3.4 pull-up resistor

PU35_H：Enable PT3.5 pull-up resistor

PU36_H：Enable PT3.6 pull-up resistor

PU37_H：Enable PT3.7 pull-up resistor

- **Include**

Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

None

- **Example**

/* Enable pull-up resistor of the PT3.4. */

GPIO_PT3SETPU (PU34_H);

5.2.93. GPIO_PT3CLRPU

- **Prototype**

GPIO_PT3CLRPU(BitSet)

- **Description**

Disable pull-up resistor of the specified pin.

Clear the register PT3PU.

- **Parameter**

BitSet [in]：Disable pull-up resistor of the specified pin.

PU30_H：Disable PT3.0 pull-up resistor

PU31_H：Disable PT3.1 pull-up resistor

PU32_H：Disable PT3.2 pull-up resistor

PU33_H：Disable PT3.3 pull-up resistor

PU34_H：Disable PT3.4 pull-up resistor

PU35_H：Disable PT3.5 pull-up resistor

PU36_H：Disable PT3.6 pull-up resistor

PU37_H：Disable PT3.7 pull-up resistor

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Disable pull-up resistor of the PT3.4. */
GPIO_PT3CLRPU ( PU34_H );
```

5.2.94. GPIO_PT3SETPUAll

- **Prototype**

  GPIO_PT3SETPUAll()

- **Description**

  Enable pull-up resistor of PT3

  Configure the register PT3PU

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Enable pull-up resistor of PT3 */
GPIO_PT3SETPUAll();
```

5.2.95. GPIO_PT3InputEnable

- **Prototype**

  GPIO_PT3InputEnable(BitSet)

- **Description**

  Enable digital function of the specified pin.

  Configure the register PT3IN

- **Parameter**

  BitSet [in]：Enable digital function of the specified pin.

IN30_H：Enable digital function of the PT3.0

IN31_H：Enable digital function of the PT3.1

IN32_H：Enable digital function of the PT3.2

IN33_H：Enable digital function of the PT3.3

IN34_H：Enable digital function of the PT3.4

IN35_H：Enable digital function of the PT3.5

IN36_H：Enable digital function of the PT3.6

IN37_H：Enable digital function of the PT3.7

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Enable digital function of the PT3.5 */
GPIO_PT3InputEnable(IN35_H);
```

5.2.96. GPIO_PT3InputDisable

- **Prototype**

  GPIO_PT3InputDisable(BitSet)

- **Description**

  Disable digital function of the specified pin.

  Clear the register PT3IN

- **Parameter**

  BitSet [in]：Disable digital function of the specified pin.

  IN30_H：Disable digital function of the PT3.0

  IN31_H：Disable digital function of the PT3.1

  IN32_H：Disable digital function of the PT3.2

  IN33_H：Disable digital function of the PT3.3

  IN34_H：Disable digital function of the PT3.4

  IN35_H：Disable digital function of the PT3.5

  IN36_H：Disable digital function of the PT3.6

  IN37_H：Disable digital function of the PT3.7

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Disable digital function of the PT3.5 */
GPIO_PT3InputDisable(IN35_H);
```

5.2.97. GPIO_PT3SETDA

- **Prototype**

  GPIO_PT3SETDA(BitSet)

- **Description**

  Enable analogy function of the specified pin.

  Configure the register PT3DA

- **Parameter**

  BitSet [in]：Enable analogy function of the specified pin

  DA30_H：Enable analogy function of the PT3.0

  DA31_H：Enable analogy function of the PT3.1

  DA32_H：Enable analogy function of the PT3.2

  DA33_H：Enable analogy function of the PT3.3

  DA34_H：Enable analogy function of the PT3.4

  DA35_H：Enable analogy function of the PT3.5

  DA36_H：Enable analogy function of the PT3.6

  DA37_H：Enable analogy function of the PT3.7

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

  /* Enable analogy function of the PT3.5. */

  GPIO_PT3SETDA(DA35_H);

5.2.98. GPIO_PT3CLRDA

- **Prototype**

  GPIO_PT3CLRDA(BitSet)

- **Description**

  Disable analogy function of the specified pin.

  Clear the register PT3DA

- **Parameter**

  BitSet [in]：Disable analogy function of the specified pin

  DA30_H：Disable analogy function of the PT3.0

  DA31_H：Disable analogy function of the PT3.1

  DA32_H：Disable analogy function of the PT3.2

  DA33_H：Disable analogy function of the PT3.3

  DA34_H：Disable analogy function of the PT3.4

DA35_H：Disable analogy function of the PT3.5

DA36_H：Disable analogy function of the PT3.6

DA37_H：Disable analogy function of the PT3.7

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Disable analogy function of the PT3.5. */
GPIO_PT3CLRDA(DA35_H);
```

5.2.99. GPIO_PT3GET

- **Prototype**

  GPIO_PT3GET(BitSet)

- **Description**

  Read the status of the specified pin

  Read the register PT3

- **Parameter**

  BitSet [in]：Read the status of the specified pin

  PT30_H：Read the status of the PT3.0

  PT31_H：Read the status of the PT3.1

  PT32_H：Read the status of the PT3.2

  PT33_H：Read the status of the PT3.3

  PT34_H：Read the status of the PT3.4

  PT35_H：Read the status of the PT3.5

  PT36_H：Read the status of the PT3.6

  PT36_H：Read the status of the PT3.6

  PT37_H：Read the status of the PT3.7

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Read the status of the specified pin PT3.5 and PT3.6 */
unsigned char PT3_DATA;
PT3_DATA = GPIO_PT3GET(PT35_H | PT36_H);
```

5.2.100. INTE30_Enable

- **Prototype**

INTE30_Enable()

- **Description**

  Enable PT3.0 interrupt function .

  Configure the register PT3INTE[0]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Enable PT3.0 interrupt function */

  INTE30_Enable();

5.2.101. INTE30_Disable

- **Prototype**

  INTE30_Disable()

- **Description**

  Disable PT30 interrupt function .

  Clear the register PT3INTE[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Disable PT3.0 interrupt function */

  INTE30_Disable();

5.2.102. INTF30_IsFlag

- **Prototype**

  INTF30_IsFlag()

- **Description**

  Read PT3.0 interrupt flag.

  Read the register PT3INTF[0]

- **Parameter**

  None

- **Include**

Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read PT3.0 interrupt flag */
unsigned char flag;
flag = INTF30_IsFlag();
```

5.2.103. INTF30_ClearFlag

- **Prototype**

  INTF30_ClearFlag()

- **Description**

  Clear PT3.0 interrupt flag.

  Clear the register PT3INTF[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Clear PT3.0 interrupt flag */
INTF30_ClearFlag();
```

5.2.104. INTG30_Edgerise

- **Prototype**

  INTG30_Edgerise()

- **Description**

  Set edge rise to trigger PT3.0 interrupt.

  Configure the register PT3INT[0]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Set edge rise to trigger PT3.0 interrupt. */
```

INTG30_Edgerise();

5.2.105. INTG30_Edgefall

- **Prototype**

  INTG30_Edgefall()

- **Description**

  Set edge fall to trigger PT3.0 interrupt.

  Clear the register PT3INT[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Set edge fall to trigger PT3.0 interrupt. */

  INTG30_Edgefall();

5.2.106. INTE31_Enable

- **Prototype**

  INTE31_Enable()

- **Description**

  Enable PT3.1 interrupt function .

  Configure the register PT3INTE[1]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Enable PT3.1 interrupt function */

  INTE31_Enable();

5.2.107. INTE31_Disable

- **Prototype**

  INTE31_Disable()

- **Description**

  Disable PT30 interrupt function .

Clear the register PT3INTE[1]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Disable PT3.1 interrupt function */
INTE31_Disable();
```

5.2.108. INTF31_IsFlag

- **Prototype**

  INTF31_IsFlag()

- **Description**

  Read PT3.1 interrupt flag.

  Read the register PT3INTF[1]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read PT3.1 interrupt flag */
unsigned char flag;
flag = INTF31_IsFlag();
```

5.2.109. INTF31_ClearFlag

- **Prototype**

  INTF31_ClearFlag()

- **Description**

  Clear PT3.1 interrupt flag.

  Clear the register PT3INTF[1]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Clear PT3.1 interrupt flag */

  INTF31_ClearFlag();

5.2.110. INTG31_Edgerise

- **Prototype**

  INTG31_Edgerise()

- **Description**

  Set edge rise to trigger PT3.1 interrupt.

  Configure the register PT3INT[1]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Set edge rise to trigger PT3.1 interrupt. */

  INTG31_Edgerise();

5.2.111. INTG31_Edgefall

- **Prototype**

  INTG31_Edgefall()

- **Description**

  Set edge fall to trigger PT3.1 interrupt.

  Clear the register PT3INT[1]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Set edge fall to trigger PT3.1 interrupt. */

  INTG31_Edgefall();

5.2.112. INTE32_Enable

- **Prototype**

  INTE32_Enable()

- **Description**

  Enable PT3.2 interrupt function .

  Configure the register PT3INTE[2]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Enable PT3.2 interrupt function */

  INTE32_Enable();

5.2.113. INTE32_Disable

- **Prototype**

  INTE32_Disable()

- **Description**

  Disable PT32 interrupt function .

  Clear the register PT3INTE[2]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Disable PT3.2 interrupt function */

  INTE32_Disable();

5.2.114. INTF32_IsFlag

- **Prototype**

  INTF32_IsFlag()

- **Description**

  Read PT3.2 interrupt flag.

  Read the register PT3INTF[2]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read PT3.2 interrupt flag */
unsigned char flag;
flag = INTF32_IsFlag();
```

5.2.115. INTF32_ClearFlag

- **Prototype**

  INTF32_ClearFlag()

- **Description**

  Clear PT3.2 interrupt flag.

  Clear the register PT3INTF[2]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Clear PT3.2 interrupt flag */
INTF32_ClearFlag();
```

5.2.116. INTG32_Edgerise

- **Prototype**

  INTG32_Edgerise()

- **Description**

  Set edge rise to trigger PT3.2 interrupt.

  Configure the register PT3INT[2]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Set edge rise to trigger PT3.2 interrupt. */
INTG32Edgerise();
```

5.2.117. INTG32_Edgefall

- **Prototype**

  INTG32_Edgefall()

- **Description**

  Set edge fall to trigger PT3.2 interrupt.

  Clear the register PT3INT[2]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Set edge fall to trigger PT3.2 interrupt. */
INTG32_Edgefall();
```

5.2.118. INTE33_Enable

- **Prototype**

  INTE33_Enable()

- **Description**

  Enable PT3.3 interrupt function .

  Configure the register PT3INTE[3]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Enable PT3.3 interrupt function */
INTE33_Enable();
```

5.2.119. INTE33_Disable

- **Prototype**

  INTE33_Disable()

- **Description**

Disable PT33 interrupt function .

Clear the register PT3INTE[3]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Disable PT3.3 interrupt function */
INTE33_Disable();
```

5.2.120. INTF33_IsFlag

- **Prototype**

  INTF33_IsFlag()

- **Description**

  Read PT3.3 interrupt flag.

  Read the register PT3INTF[3]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read PT3.3 interrupt flag */
unsigned char flag;
flag = INTF33_IsFlag();
```

5.2.121. INTF30_ClearFlag

- **Prototype**

  INTF30_ClearFlag()

- **Description**

  Clear PT3.3 interrupt flag.

  Clear the register PT3INTF[3]

- **Parameter**

  None

- **Include**

Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Clear PT3.3 interrupt flag */

  INTF33_ClearFlag();

5.2.122. INTG33_Edgerise

- **Prototype**

  INTG33_Edgerise()

- **Description**

  Set edge rise to trigger PT3.3 interrupt.

  Configure the register PT3INT[3]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Set edge rise to trigger PT3.3 interrupt. */

  INTG33_Edgerise();

5.2.123. INTG33_Edgefall

- **Prototype**

  INTG33_Edgefall()

- **Description**

  Set edge fall to trigger PT3.3 interrupt.

  Clear the register PT3INT[3]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Set edge fall to trigger PT3.3 interrupt. */

  INTG33_Edgefall();

5.2.124. INTE34_Enable

- **Prototype**

  INTE34_Enable()

- **Description**

  Enable PT3.4 interrupt function .

  Configure the register PT3INTE[4]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Enable PT3.4 interrupt function */
INTE34_Enable();
```

5.2.125. INTE34_Disable

- **Prototype**

  INTE34_Disable()

- **Description**

  Disable PT34 interrupt function .

  Clear the register PT3INTE[4]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Disable PT3.4 interrupt function */
INTE34_Disable();
```

5.2.126. INTF34_IsFlag

- **Prototype**

  INTF34_IsFlag()

- **Description**

  Read PT3.4 interrupt flag.

  Read the register PT3INTF[4]

- **Parameter**

None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read PT3.4 interrupt flag */
unsigned char flag;
flag = INTF34_IsFlag();
```

5.2.127. INTF34_ClearFlag

- **Prototype**

  INTF34_ClearFlag()

- **Description**

  Clear PT3.4 interrupt flag.

  Clear the register PT3INTF[4]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Clear PT3.4 interrupt flag */
INTF34_ClearFlag();
```

5.2.128. INTG34_Edgerise

- **Prototype**

  INTG34_Edgerise()

- **Description**

  Set edge rise to trigger PT3.4 interrupt.

  Configure the register PT3INT[4]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Set edge rise to trigger PT3.4 interrupt. */

  INTG34_Edgerise();

5.2.129. INTG34_Edgefall

- **Prototype**

  INTG34_Edgefall()

- **Description**

  Set edge fall to trigger PT3.4 interrupt.

  Clear the register PT3INT[4]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Set edge fall to trigger PT3.4 interrupt. */

  INTG34_Edgefall();

5.2.130. INTE35_Enable

- **Prototype**

  INTE35_Enable()

- **Description**

  Enable PT3.5 interrupt function .

  Configure the register PT3INTE[5]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Enable PT3.5 interrupt function */

  INTE35_Enable();

5.2.131. INTE35_Disable

- **Prototype**

  INTE35_Disable()

- **Description**

  Disable PT35 interrupt function .

  Clear the register PT3INTE[5]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Disable PT3.5 interrupt function */
INTE35_Disable();
```

5.2.132. INTF35_IsFlag

- **Prototype**

  INTF35_IsFlag()

- **Description**

  Read PT3.5 interrupt flag.

  Read the register PT3INTF[5]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read PT3.5 interrupt flag */
unsigned char flag;
flag = INTF35_IsFlag();
```

5.2.133. INTF35_ClearFlag

- **Prototype**

  INTF35_ClearFlag()

- **Description**

  Clear PT3.5 interrupt flag.

  Clear the register PT3INTF[5]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Clear PT3.5 interrupt flag */

  INTF35_ClearFlag();

5.2.134. INTG35_Edgerise

- **Prototype**

  INTG35_Edgerise()

- **Description**

  Set edge rise to trigger PT3.5 interrupt.

  Configure the register PT3INT[5]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Set edge rise to trigger PT3.5 interrupt. */

  INTG35_Edgerise();

5.2.135. INTG35_Edgefall

- **Prototype**

  INTG35_Edgefall()

- **Description**

  Set edge fall to trigger PT3.5 interrupt.

  Clear the register PT3INT[5]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

  /* Set edge fall to trigger PT3.5 interrupt. */

  INTG35_Edgefall();

5.2.136. INTE36_Enable

- **Prototype**

  INTE36_Enable()

- **Description**

  Enable PT3.6 interrupt function .

  Configure the register PT3INTE[6]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Enable PT3.6 interrupt function */
INTE36_Enable();
```

5.2.137. INTE36_Disable

- **Prototype**

  INTE36_Disable()

- **Description**

  Disable PT36 interrupt function .

  Clear the register PT3INTE[6]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Disable PT3.6 interrupt function */
INTE36_Disable();
```

5.2.138. INTF36_IsFlag

- **Prototype**

  INTF36_IsFlag()

- **Description**

  Read PT3.6 interrupt flag.

  Read the register PT3INTF[6]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read PT3.6 interrupt flag */
unsigned char flag;
flag = INTF36_IsFlag();
```

5.2.139. INTF36_ClearFlag

- **Prototype**

  INTF36_ClearFlag()

- **Description**

  Clear PT3.6 interrupt flag.

  Clear the register PT3INTF[6]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Clear PT3.6 interrupt flag */
INTF36_ClearFlag();
```

5.2.140. INTG36_Edgerise

- **Prototype**

  INTG36_Edgerise()

- **Description**

  Set edge rise to trigger PT3.6 interrupt.

  Configure the register PT3INT[6]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

None

- **Example**

/* Set edge rise to trigger PT3.6 interrupt. */

INTG36_Edgerise();

5.2.141. INTG36_Edgefall

- **Prototype**

INTG36_Edgefall()

- **Description**

Set edge fall to trigger PT3.6 interrupt.

Clear the register PT3INT[6]

- **Parameter**

None

- **Include**

Driver/HY17M/HY17M24/INT.h

- **Return Value**

None

- **Example**

/* Set edge fall to trigger PT3.6 interrupt. */

INTG36_Edgefall();

5.2.142. INTE37_Enable

- **Prototype**

INTE37_Enable()

- **Description**

Enable PT3.7 interrupt function .

Configure the register PT3INTE[7]=1

- **Parameter**

None

- **Include**

Driver/HY17M/HY17M24/INT.h

- **Return Value**

None

- **Example**

/* Enable PT3.7 interrupt function */

INTE37_Enable();

5.2.143. INTE37_Disable

- **Prototype**

INTE37_Disable()

- **Description**

  Disable PT37 interrupt function .

  Clear the register PT3INTE[7]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Disable PT3.7 interrupt function */
INTE37_Disable();
```

5.2.144. INTF37_IsFlag

- **Prototype**

  INTF37_IsFlag()

- **Description**

  Read PT3.7 interrupt flag.

  Read the register PT3INTF[7]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read PT3.7 interrupt flag */
unsigned char flag;
flag = INTF37_IsFlag();
```

5.2.145. INTF37_ClearFlag

- **Prototype**

  INTF37_ClearFlag()

- **Description**

  Clear PT3.7 interrupt flag.

  Clear the register PT3INTF[7]

- **Parameter**

None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Clear PT3.7 interrupt flag */
INTF37_ClearFlag();
```

5.2.146. INTG37_Edgerise

- **Prototype**

  INTG37_Edgerise()

- **Description**

  Set edge rise to trigger PT3.7 interrupt.

  Configure the register PT3INT[7]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Set edge rise to trigger PT3.7 interrupt. */
INTG37_Edgerise();
```

5.2.147. INTG37_Edgefall

- **Prototype**

  INTG37_Edgefall()

- **Description**

  Set edge fall to trigger PT3.7 interrupt.

  Clear the register PT3INT[7]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/INT.h

- **Return Value**

  None

- **Example**

```
/* Set edge fall to trigger PT3.7 interrupt. */
```

INTG37 _Edgefall();

5.2.148. GPIO_PM30Sel

- **Prototype**

  GPIO_PM30Sel(PM30Sel)

- **Description**

  Select the main function of PT3.0 multiple function pin.

  Configure the register PT3M1[1:0]

- **Parameter**

  PM30Sel [in]：Select the function of PT3.0

  PM30_VOHL : GPIO function

  PM30_SCL_2 : I2C SCL pin

  PM30_TBI0_1 : TBI0 input pin

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

  /* Set PT3.0 as TBI0 input pin*/

  GPIO_PM30Sel(PM30_TBI0_1);

5.2.149. GPIO_PM32Sel

- **Prototype**

  GPIO_PM32Sel(PM32Sel)

- **Description**

  Select the main function of PT3.2 multiple function pin.

  Configure the register PT3M1[5:4]

- **Parameter**

  PM32Sel [in]：Select the function of PT3.2

  PM32_VOHL : GPIO function

  PM32_PWM0_2 : PWM0 output pin

  PM32_SCL_3 : I2C SCL pin

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

  /* Set PT3.2 as SCL pin */

  GPIO_PM32Sel(PM32_SCL_3);

5.2.150. GPIO_PM33Sel

- **Prototype**

  GPIO_PM33Sel(PM33Sel)

- **Description**

  Select the main function of PT3.3 multiple function pin.

  Configure the register PT3M1[6]

- **Parameter**

  PM33Sel [in]：Select the function of PT3.3

  PM33_VOHL : GPIO function

  PM33_PWM1_2 : PWM1 output pin

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Set PT3.3 as PWM1 output pin*/
GPIO_PM33Sel(PM33_PWM1_2);
```

5.2.151. GPIO_PM34Sel

- **Prototype**

  GPIO_PM34Sel(PM34Sel)

- **Description**

  Select the main function of PT3.4 multiple function pin.

  Configure the register PT3M2[1:0]

- **Parameter**

  PM34Sel [in]：Select the function of PT3.4

  PM34_VOHL : GPIO function

  PM34_BZ_2 : BZ output pin

  PM34_TX_3 : UART TX pin

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/* Set PT3.4 as TX pin */
GPIO_PM34Sel(PM34_TX_3);
```

5.2.152. GPIO_PM36Sel

- **Prototype**

  GPIO_PM36Sel(PM36Sel)

- **Description**

  Select the main function of PT3.6 multiple function pin.

  Configure the register PT3M2[5:4]

- **Parameter**

  PM36Sel [in]：Select the function of PT3.6

  PM36_VOHL : GPIO function

  PM36_TBI1_1 : TBI1 input pin

  PM36_TX_2 : UART TX pin

- **Include**

  Driver/HY17M/HY17M24/GPIO.h

- **Return Value**

  None

- **Example**

```
/*Set PT3.6 as TX pin */
GPIO_PM36Sel(PM36_TX_2);
```

6. ADC Driver

6.1. Introduction

The following functions are included in ADC Manager Section.

Item	Functions	Description
01	ADC_Open	Enable ADC function, set the ADC clock, OSR, the ADC signal input source, AD Gain, DC offset, and reference voltage source
02	ADC_GetData	Get the A/D conversion data
03	ADC_Enable	Enable ADC control
04	ADC_Disable	Disable ADC control
05	ADC_INT_Enable	Enable ADC interrupt
06	ADC_INT_Disable	Disable ADC interrupt
07	ADC_INT_IsFlag	Read ADC interrupt flag
08	ADC_INT_ClearFlag	Clear ADC interrupt flag
09	ADC_OSRSelect	Set the ADC OSR
10	ADC_CMFREnable	Enable Comb filter
11	ADC_CMFRDisable	Disable Comb filter
12	ADC_VRXSelect	Set the ADC full reference range select
13	ADC_GainSelect	Input signal gain for ADC
14	ADC_DCSETSelect	DC offset input voltage selection
15	ADC_ENACMEnable	Enable ACM function
16	ADC_ENACMDisable	Disable ACM function
17	ADC_ACMBufSelect	Select ACM input source
18	ADC_VINSelect	Configure the ADC signal input (positive and negative) source
19	ADC_VRINSelect	Configure the ADC reference voltage input (positive and negative) source
20	ADC_INXSelect	Configure the ADC signal input (positive and negative) exchange controller
21	ADC_ENINXCH_EnableADC_ENINXCH_Disable	Enable the ADC signal input auto switching
22	ADC_ENINXCH_Disable	Disable the ADC signal input auto switching
23	ADC_VCMSSelect	Select VCM source
24	ADC_REFOISel	Select the REFOI input sourc
25	ADC_ENTPSEnable	Enable TPS function
26	ADC_ENTPSDisable	Disable TPS function
27	ADC_TPSSelect	Select TPS output phase
28	ADC_DAFMSelect	Select the formats of Comb Filter output
29	ADC_ENCH_Enable	Enable ADC chopper
30	ADC_ENCH_Disable	Disable ADC chopper

6.2. Functions

6.2.1. ADC_Open

- **Prototype**

  void ADC_Open(unsigned char cks, unsigned char inp, unsigned char inn, unsigned char vrh, unsigned char vrl, unsigned char adgn, unsigned char vrgn, unsigned char dcset, unsigned char osr, unsigned char vcms);

- **Description**

  Enable ADC function, set the ADC clock, OSR, the ADC signal input source, AD Gain, DC offset, and reference voltage source.

  Configure the register AD1CN0/AD1CN1/AD1CN2/AD1CN3/AD1CN4/AD1CN5.

- **Parameters**

  cks [in]：Select ADC clock

  DADC_DHSCKDIV2 : DHS_CK/2
  DADC_DHSCKDIV4 : DHS_CK/4
  DADC_DHSCKDIV8 : DHS_CK/8
  DADC_DHSCKDIV16 : DHS_CK/16

  inp [in]：Select ADC positive input source

  INP_AI0 : AI0
  INP_AI1 : AI1
  INP_AI2 : AI2
  INP_AI4 : AI4
  INP_AI6 : AI6
  INP_AI8 : AI8
  INP_AI10 : AI10
  INP_AI12 : AI12
  INP_AI14 : AI14
  INP_OPOI : OPOI
  INP_VDDA : VDDA
  INP_REFOI : REFOI
  INP_TS0 : TS0
  INP_TS1 : TS1
  INP_VSS : VSS
  INP_DAOI : DAOI

  inn [in]：Select ADC negative input source

  INN_AI0 : AI0
  INN_AI1 : AI1
  INN_AI2 : AI2
  INN_AI5 : AI5
  INN_AI7 : AI7

INN_AI9 : AI9
INN_AI11 : AI11
INN_AI13 : AI13
INN_VDDDIV10 : VDD/10
INN_OPOI : OPOI
INN_VDDA : VDDA
INN_REFOI : REFOI
INN_TS0 : TS0
INN_TS1 : TS1
INN_VSS : VSS
INN_DAOI : DAOI

vrh [in]：Select the reference voltage positive input source
VRH_VDDA : VDDA
VRH_AI0 : AI0
VRH_AI10 : AI10
VRH_REFOI : REFOI

vrl [in]：Select the reference voltage negative input source
VRL_VSS : VSS
VRL_AI2 : AI2
RL_AI4 : AI4
VRL_REFOI : REFOI

adgn [in]：Select AD gain
ADGN_1DIV4 : x 1/4
ADGN_1DIV2 : x 1/2
ADGN_1 : x 1
ADGN_2 : x 2
ADGN_4 : x 4
ADGN_8 : x 8
ADGN_16 : x 16
ADGN_RSVD : x 16

vrgn [in]：Set the ADC full reference range select
VREGN_DIV2：VREF * 1/2
VREGN_DIV1：VREF* 1

dcset [in]：Select ADC dc offset
DCSET_P0 : 0
DCSET_P1DIV8 : + VREF* 1/8
DCSET_P2DIV8 : + VREF* 2/8
DCSET_P3DIV8 : + VREF* 3/8

DCSET_P4DIV8 : + VREF* 4/8
DCSET_P5DIV8 : + VREF* 5/8
DCSET_P6DIV8 : + VREF* 6/8
DCSET_P7DIV8 : + VREF* 7/8
DCSET_N0 : 0
DCSET_N1DIV8 : -VREF* 1/8
DCSET_N2DIV8 : -VREF* 2/8
DCSET_N3DIV8 : -VREF* 3/8
DCSET_N4DIV8 : -VREF* 4/8
DCSET_N5DIV8 : -VREF* 5/8
DCSET_N6DIV8 : -VREF* 6/8
DCSET_N7DIV8 : -VREF* 7/8

osr [in]：select ADC OSR
OSR_64 : ADC_CK/64
OSR_128 : ADC_CK/128
OSR_256 : ADC_CK/256
OSR_512 : ADC_CK/512
OSR_1024 : ADC_CK/1024
OSR_2048 : ADC_CK/2048
OSR_4096 : ADC_CK/4096
OSR_8192 : ADC_CK/8192
OSR_16384 : ADC_CK/16384
OSR_32768 : ADC_CK/32768
OSR_65536 : ADC_CK/65536

vcms [in]：Select VCM source
VCMS_ACMint : From ACMint
VCMS_REFOint : From REFOint

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

```
/* Set DHS_CK/2 as ADC clock, input source AI0-AI1, the reference voltage input source VDDA-VSS · AD gain is 1, DC Offset 0V · OSR is ADC_CK/65536 · REFOint as VCM source */
ADC_Open(DADC_DHSCKDIV2,INP_AI0,INN_AI1,VRH_VDDA,VRL_VSS,
        ADGN_1, VREGN_DIV1,DCSET_P0,OSR_65536,VCMS_REFOint);
```

6.2.2. ADC_GetData

- **Prototype**

long ADC_GetData(void);

- **Description**

  Get the A/D conversion data with signed.

  Configure the register AD1H:AD1M:AD1L

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  24-bit conversion data

- **Example**

```
/* Get the ADC conversion data. */
long ADC_DATA;
ADC_DATA = ADC_GetData();
```

6.2.3. ADC_Enable

- **Prototype**

  ADC_Enable()

- **Description**

  Enable A/D conversion function

  Configure the register AD1CN0[7]=1.

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

```
/* Enable ADC function */
ADC_Enable();
```

6.2.4. ADC_Disable

- **Prototype**

  ADC_Disable()

- **Description**

  Disable A/D conversion function

  Clear the register AD1CN0[7].

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

  /* Disable ADC function */

  ADC_Disable();

6.2.5. ADC_INT_Enable

- **Prototype**

  ADC_INT_Enable()

- **Description**

  Enable ADC interrupt function.

  Configure the register INTE0[5]=1.

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/HY17M/ADC.h

- **Return Value**

  None

- **Example**

  /* Enable ADC interrupt function. */

  ADC_INT_Enable();

6.2.6. ADC_INT_Disable

- **Prototype**

  ADC_INT_Disable()

- **Description**

  Disable ADC interrupt function.

  Clear the register INTE0[5].

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

  /* Disable ADC interrupt function. */

  ADC_INT_Disable();

6.2.7. ADC_INT_IsFlag

- **Prototype**

  ADC_INT_IsFlag()

- **Description**

  Read the ADC interrupt flag.

  Read the register INTF0[5].

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read the ADC interrupt flag. */
unsigned char flag;
flag = ADC_INT_IsFlag();
```

6.2.8. ADC_INT_ClearFlag

- **Prototype**

  ADC_INT_ClearFlag()

- **Description**

  Clear the ADC interrupt flag.

  Clear the register INTF0[5].

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

```
/* Clear the ADC interrupt flag. */
ADC_INT_ClearFlag();
```

6.2.9. ADC_OSRSelect

- **Prototype**

  ADC_OSRSelect(OSRSel)

- **Description**

Set the ADC OSR.

Configure the register 0x41100[5:2]

- **Parameters**

OSRSel [in]：Specify the ADC OSR. (The following output rate is calculated when clock is 2MHZ)

OSR_64	: Data Output Rate is 31250sps
OSR_128	: Data Output Rate is 15625sps
OSR_256	: Data Output Rate is 7813sps
OSR_512	: Data Output Rate is 3906sps
OSR_1024	: Data Output Rate is 1953sps
OSR_2048	: Data Output Rate is 977sps
OSR_4096	: Data Output Rate is 488sps
OSR_8192	: Data Output Rate is 244sps
OSR_16384	: Data Output Rate is 122sps
OSR_32768	: Data Output Rate is 61sps
OSR_65536	: Data Output Rate is 31sps

- **Include**

Driver/HY17M/HY17M24/ADC.h

- **Return Value**

None

- **Example**

```
/* Set the OSR of 32768 data rate 61sps. */
ADC_OSRSelect(OSR_32768);
```

6.2.10. ADC_CMFREnable

- **Prototype**

ADC_CMFREnable()

- **Description**

Enable Comb filter reset control.

Configure the register AD1CN0[0]=1

- **Parameters**

None

- **Include**

Driver/HY17M/HY17M24/ADC.h

- **Return Value**

None

- **Example**

```
/* Enable Comb filter reset control. */
ADC_CMFREnable();
```

6.2.11. ADC_CMFRDisable

- **Prototype**

  ADC_CMFRDisable()

- **Description**

  Disable Comb filter reset control.

  Clear the register AD1CN0[0].

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

```
/* Disable Comb filter reset control. */
ADC_CMFRDisable();
```

6.2.12. ADC_VRXSelect

- **Prototype**

  ADC_VRXSelect(VRXSel)

- **Description**

  Set the ADC full reference range select.

  Configure the register AD1CN1[5]

- **Parameters**

  VRXSel [in]：Specify the VREF gain. VREF= VRPS-VRNS

  VREGN_DIV2：VREF * 1/2

  VREGN_DIV1：VREF* 1

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

```
/* Set the ADC full reference range input. */
ADC_VRGainSelect(VREGN_DIV2 );
```

6.2.13. ADC_GainSelect

- **Prototype**

  ADC_GainSelect(adgn,vregn)

- **Description**

  Input signal gain for ADC.

Configure the register AD1CN1[2:0]/AD1CN1[5]

- **Parameters**

adgn [in]：Specify the ADC gain.

ADGN_1DIV4 ：x 1/4

ADGN_1DIV2 ：x 1/2

ADGN_1 ：x 1

ADGN_2 ：x 2

ADGN_4 ：x 4

ADGN_8 ：x 8

ADGN_16 ：x 16

ADGN_RSVD ：x 16

VRXSel [in]：Specify the VREF gain.

VREGN_DIV2：VREF * 1/2

VREGN_DIV1：VREF* 1

- **Include**

Driver/HY17M/HY17M24/ADC.h

- **Return Value**

None

- **Example**

```
/* Set the ADC gain is 8*2
ADC_GainConfig(ADGN_8,VREGN_DIV2);
```

6.2.14. ADC_DCSETSelect

- **Prototype**

ADC_DCSETSelect(dcset)

- **Description**

DC offset input voltage selection (VREF=REFP-REFN)

Configure the register AD1CN2[3:0]

- **Parameters**

dcset [in]：Specify the ADC DCSET.

DCSET_P0 ：0 VREF

DCSET_P1DIV8 ：+ VREF* 1/8

DCSET_P2DIV8 ：+ VREF* 2/8

DCSET_P3DIV8 ：+ VREF* 3/8

DCSET_P4DIV8 ：+ VREF* 4/8

DCSET_P5DIV8 ：+ VREF* 5/8

DCSET_P6DIV8 ：+ VREF* 6/8

DCSET_P7DIV8 ：+ VREF* 7/8

DCSET_N0 : 0VREF

DCSET_N1DIV8 : -VREF* 1/8

DCSET_N2DIV8 : -VREF* 2/8

DCSET_N3DIV8 : -VREF* 3/8

DCSET_N4DIV8 : -VREF* 4/8

DCSET_N5DIV8 : -VREF* 5/8

DCSET_N6DIV8 : -VREF* 6/8

DCSET_N7DIV8 : -VREF* 7/8

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

  /* Set the DC offset of +1/8 voltage. */

  ADC_DCSETSelec(DCSET_P1DIV8);

6.2.15. ADC_ENACMEnable

- **Prototype**

  ADC_ENACMEnable()

- **Description**

  Enable ADC common mode voltage.

  Configure the register AD1CN5[7]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

  /* Enable ADC common mode voltage. */

  ADC_ENACMEnable();

6.2.16. ADC_ENACMDisable

- **Prototype**

  ADC_ENACMDisable()

- **Description**

  Disable ADC common mode voltage.

  Clear the register AD1CN5[7]

- **Parameters**

None

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

  /* Disable ADC common mode voltage. */

  ADC_ENACMDisable();

6.2.17. ADC_ACMBufSelect

- **Prototype**

  ADC_ACMBufSelect(SELVINSel)

- **Description**

  Select ADC common mode voltage source.

  Configure the register AD1CN2[4]

- **Parameters**

  SELVINSel [in]：ACM voltage source

  SELVIN_1V2 ：1.2V

  SELVIN_VDDADIV2 ：VDDA/2

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

  /* Select 1.2V as ADC common mode voltage source. */

  ADC_ACMBufSelect(SELVIN_1V2);

6.2.18. ADC_VINSelect

- **Prototype**

  ADC_VINSelect(inp,inn)

- **Description**

  Configure the ADC signal input (positive and negative) source.

  Configure the register AD1CN3[7:0]

- **Parameters**

  inp [in]：Select positive input source

  INP_AI0 ：AI0

  INP_AI1 ：AI1

  INP_AI2 ：AI2

INP_AI4 : AI4
INP_AI6 : AI6
INP_AI8 : AI8
INP_AI10 : AI10
INP_AI12 : AI12
INP_AI14 : AI14
INP_OPOI : OPOI
INP_VDDA : VDDA
INP_REFOI : REFOI
INP_TS0 : TS0
INP_TS1 : TS1
INP_VSS : VSS
INP_DAOI : DAOI

inn [in] : Select negative input source

INN_AI0 : AI0
INN_AI1 : AI1
INN_AI2 : AI2
INN_AI5 : AI5
INN_AI7 : AI7
INN_AI9 : AI9
INN_AI11 : AI11
INN_AI13 : AI13
INN_VDDDIV10 : VDD/10
INN_OPOI : OPOI
INN_VDDA : VDDA
INN_REFOI : REFOI
INN_TS0 : TS0
INN_TS1 : TS1
INN_VSS : VSS
INN_DAOI : DAOI

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

```
/* Set the ADC signal input source AI0---AI1 */
ADC_VINSelect( INP_AI0, INN_AI1);
```

6.2.19. ADC_VRINSelect

- **Prototype**

  ADC_VRINSelect(vrh,vrl)

- **Description**

  Set the ADC reference voltage.

  Configure the register AD1CN4[7:6]/AD1CN4[5:4].

- **Parameters**

  vrh [in]：Specify the ADC VRH

  VRH_VDDA ：Reference voltage VDDA

  VRH_AI0 ：Reference voltage AI0

  VRH_AI10：Reference voltage AI10

  VRH_REFOI ：Reference voltage REFOI

  vrl [in]：Specify the ADC VRL

  VRL_VSS ：Reference voltage VSS

  VRL_AI2 ：Reference voltage AI2

  VRL_AI4 ：Reference voltage AI4

  VRL_REFOI ：Reference voltage REFOI

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

```
/* Set the ADC reference voltage VRH=AIO10, VRL=AI4 */
ADC_VRINSelect(VRH_AI10,VRL_AI4);
```

6.2.20. ADC_INXSelect

- **Prototype**

  ADC_INXSelect(INXSel)

- **Description**

  Configure the ADC signal input (positive and negative) exchange controller

  Configure the register AD1CN4[3:2]

- **Parameters**

  INXSel [in]：Select input exchange mode

  INX_NORMAL：INP→ADH, NN→ADL

  INX_INN：INN→ADH&ADL, INP floating

  INX_INP：INP→ADH&ADL, INN floating

  INX_CROSS：INP→ADL, INN→ADH

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

None

- **Example**

/* Set input exchange mode INP→ADH, INN→ADL */

ADC_INXConfig(INX_NORMAL);

6.2.21. ADC_ENINXCH_Enable

- **Prototype**

ADC_ENINXCH_Enable()

- **Description**

Enable the ADC signal input (positive and negative) auto exchange. Configure the register INX[1:0]=00b and INX[1:0]=11b change over.

Configure the register CSFCN1[5]=1

- **Parameters**

None

- **Include**

Driver/HY17M/HY17M24/ADC.h

- **Return Value**

None

- **Example**

/* Enable the ADC signal input (positive and negative) auto exchange */

CSFON_Enable();

ADC_ENINXCH_Enable();

6.2.22. ADC_ENINXCH_Disable

- **Prototype**

ADC_ENINXCH_Disable()

- **Description**

Disable the ADC signal input (positive and negative) auto exchange.

Clear the register CSFCN1[5]

- **Parameters**

None

- **Include**

Driver/HY17M/HY17M24/ADC.h

- **Return Value**

None

- **Example**

/* Disable the ADC signal input (positive and negative) auto exchange */

CSFON_Enable();

ADC_ENINXCH_Disable();

6.2.23. ADC_VCMSSelect

- **Prototype**

  ADC_VCMSSelect(vcms)

- **Description**

  Select the ADC common mode voltage source.

  Configure the register AD1CN5[5]

- **Parameters**

  cvms [in]：Select the ACM voltage source

  VCMS_REFOint ：ADC common mode voltage from REFOint

  VCMS_ACMint ：ADC common mode voltage from ACMint

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

```
/*  ADC common mode voltage from ACMint  */
ADC_VCMSSelect(VCMS_ACMint);
```

6.2.24. ADC_REFOISel

- **Prototype**

  ADC_REFOISel(Sel)

- **Description**

  Select the REFOI input sourc

  Configure the register MCCN1[6]

- **Parameters**

  Sel[in]：Select the REFOI input source

  VRSEL _REFOint ：REFOI from REFOint

  VRSEL _ACMint ：REFOI from ACMint

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

```
/* Select the REFOI input sourc from REFOint . */
ADC_REFOISel(VRSEL _REFOint);
```

6.2.25. ADC_ENTPSEnable

- **Prototype**

  ADC_ENTPSEnable()

- **Description**

  Enable TPS function.

  Configure the register AD1CN5[1]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

  /*Enable TPS function */

  ADC_ENTPSEnable();

6.2.26. ADC_ENTPSDisable

- **Prototype**

  ADC_ENTPSDisable()

- **Description**

  Disable TPS function.

  Clear the register AD1CN5[1]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

  /* Disable TPS function */

  ADC_ENTPSDisable();

6.2.27. ADC_TPSSelect

- **Prototype**

  ADC_TPSSelect(TPSSel)

- **Description**

  Configure the TPS output voltage phase control.

  Configure the register AD1CN5[0]

- **Parameters**

TPSSel [in]：TPS output voltage phase control

TPSCH_NORMAL　：normal

TPSCH_REVERSE　：inversion

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

```
/* TPS output voltage phase normal */
ADC_TPSSelect(TPSCH_NORMAL);
```

6.2.28. ADC_DAFMSelect

- **Prototype**

  ADC_DAFMSelect(DAFMSel)

- **Description**

  Configure comb filter output data format.

  Configure the register LVDCN[7]

- **Parameters**

  DAFMSel [in]：

  DAFM _NORMAL　：Normal format

  DAFM_CHOPPER　：Chopper Result format, first data(ADC1 + (ADC2))/2，second data(ADC2+ADC3)/2, third data(ADC2 +ADC3)/2, etc.

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

```
/* comb filter output data normal format */
ADC_DAFMSelect(DAFM _NORMAL);
```

6.2.29. ADC_ENCH_Enable

- **Prototype**

  ADC_ENCH_Enable()

- **Description**

  Enable ADC chopper. Need to configure the register ENINXCH=1, next configure the register ENCH=1.

  Configure the register LVDCN[6]=1

- **Parameters**

  None

- **Include**

Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

  /*Enable ADC Chopper */

  ADC_ENINXCH_Enable();

  ADC_ENCH_Enable();

6.2.30. ADC_ENCH_Disable

- **Prototype**

  ADC_ENCH_Disable()

- **Description**

  Disable ADC chopper.

  Clear the register LVDCN[6]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/ADC.h

- **Return Value**

  None

- **Example**

  /*Disable ADC Chopper */

  ADC_ENINXCH_Enable();

  ADC_ENCH_Disable();

7. UART Driver

7.1. Introduction

The Universal Asynchronous Receiver/Transmitter (UART) performs a serial-to-parallel conversion on data characters received from the peripheral such as MODEM, and a parallel-to-serial conversion on data characters received from the CPU. Details please refer to the section in the target chip specification titled UART.

Item	Functions	Description
01	UART_Open	Set UART module
02	UART_BGRSet	Configure the baud of UART
03	UART_Enable	Enable the UART function
04	UART_Disable	Disable the UART function
05	UART_TXEnable	Enable the transmit function of UART
06	UART_TXDisable	Disable the transmit function of UART
07	UART_TX9Enable	Enable the function of TX port send 9 bits
08	UART_TX9Disable	Disable the function of TX port send 9 bits
09	UART_TX9Data	Configure the bit-9 of send data
10	UART_Parity	Configure the parity of UART
11	UART_WUEEnable	Enable receive wake-up mode of UART
12	UART_WUEDisable	Disable receive wake-up mode of UART
13	UART_CREnable	Enable the receive function of UART
14	UART_CRDisable	Disable the receive function of UART
15	UART_CR9Enable	Enable the function of RC port receive 9 bits
16	UART_CR9Disable	Disable the function of RC port receive 9 bits
17	UART_ADDEnable	Enable the addrss detect function
18	UART_ADDDisable	Disable the addrss detect function
19	UART_ABDEnable	Enable the auto baud rate function
20	UART_ABDDisable	Disable the auto baud rate function
21	UART_INT_TXEnable	Enable TX interrupt function
22	UART_INT_TXDisable	Disable TX interrupt function
23	UART_INT_TXIsFlag	Read the TX interrupt flag
24	UART_INT_TXClearFlag	Clear the TX interrupt flag
25	UART_INT_RCEnable	Enable RC interrupt function
26	UART_INT_RCDisable	Disable RC interrupt function
27	UART_INT_RCIsFlag	Read the RC interrupt flag
28	UART_INT_RCClearFlag	Clear the RC interrupt flag
29	UART_GETRC9	Read the bit 9 value of receive data
30	UART_GETPERR	Read the Parity Error flag of UART
31	UART_GETFERR	Read the RC Fram error flag
32	UART_GETOERR	Read the RC Buffer over run error flag
33	UART_GETRCIDL	Read the RC port status
34	UART_GETTRMT	Read the Transmit shift register(TSR) status
35	UART_GETABDOVF	Read the auto baud rate overflow flag

7.2. Functions

7.2.1. UART_Open

- **Prototype**

  void UART_Open(unsigned int uBG0R,
  unsigned char uDataBits,
  unsigned char uParity,
  unsigned char Uartxsel);

- **Description**

  Enable the TX/RC function of UART. Configure baud, parity, TX/RC port, and data length.Enable the TX/RC interrupt function.

  Configure the register UR0CN/BA0CN/BG0RH/BG0RL/TRISC1/PT1M1/INTE1.

- **Parameter**

  uBG0R [in]：0x00~0xFFFF, Configure baud.

  uDataBits [in]：Set UART data length

  8：8 bits

  9：9 bits

  uParity [in]：Set UART parity mode

  PARITY_ODD ：Odd parity

  PARITY_Even ：Even parity

  PARITY_None ：None

  Uartxsel [in]：Select TX/RC port

  0 ： PT1.5/PT1.6

  1 ： PT2.0/PT2.1

  2 ： PT3.6/PT3.7

  3 ： PT3.4/PT3.5

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

  /* HAO=2MHZ, configure UART baud is 9600bp, 8-bit data, None parity, PT1.5/PT1.6 as TX/RC pin */

  UART_Open(0x33,8,PARITY_None,0);

7.2.2. UART_BGRSet

- **Prototype**

  UART_BGRSet(uBG0R)

- **Description**

  Configure UART baud.

  Configure the register BG0RL/BG0RH[4:0]

- **Parameter**

  uBG0R [in]：0x00~0xFFFF, Configure baud.

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

  /* HAO=2MHZ, configure UART baud is 9600bp */

  UART_BGRSet(0x33);

7.2.3. UART_Enable

- **Prototype**

  UART_Enable();

- **Description**

  Enable UART function.

  Configure the register UR0CN[7]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

  /* Enable UART function */

  UART_Enable();

7.2.4. UART_Disable

- **Prototype**

  UART_Disable();

- **Description**

  Disable UART function.

  Clear the register UR0CN[7]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

  /*Disable UART function */

  UART_Disable();

7.2.5. UART_TXEnable

- **Prototype**

  UART_TXEnable();

- **Description**

  Enable the transmit function of UART.

  Configure the register UR0CN[6]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

  /* Enable UART function */

  UART_TXEnable();

7.2.6. UART_TXDisable

- **Prototype**

  UART_TXDisable();

- **Description**

  Disable the transmit function of UART.

  Clear the register UR0CN[6]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

  /*Disable UART function */

  UART_TXDisable();

7.2.7. UART_TX9Enable

- **Prototype**

  UART_TX9Enable();

- **Description**

  Enable the function of TX send 9 bits.

  Configure the register URCON[5]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

  /* Enable the function of TX send 9 bits. */

  UART_TX9Enable();

7.2.8. UART_TX9Disable

- **Prototype**

  UART_TX9Disable();

- **Description**

  Disable the function of TX send 9 bits.

  Clear the register URCON[5]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

  /* Disable the function of TX send 9 bits. */

  UART_TX9Disable();

7.2.9. UART_TX9Data

- **Prototype**

  UART_TX9Data ();

- **Description**

  Configure the send data of bit 9.

  Configure the register UR0CN[4]

- **Parameters**

Sel [in]：Configure the bit-9 of send data

TX9D_1：Set bit-9 of send data is 1

TX9D_0：Set bit-9 of send data is 0

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

```
/* Set bit-9 of send data is 1 */
UART_TX9Enable();
UART_TX9Data(TX9D_1);
```

7.2.10. UART_Parity

- **Prototype**

  UART_Parity(Sel);

- **Description**

  Configure the parity of UART.

  Configure the register UR0CN[3]

- **Parameters**

  Sel [in]：Set UART parity mode

  PARITY_ODD ：Odd parity

  PARITY_Even ：Even parity

  PARITY_None ：None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

```
/* Set even parity of UART parity mode*/
UART_Parity(PARITY_Even);
```

7.2.11. UART_WUEEnable

- **Prototype**

  UART_WUEEnable();

- **Description**

  Enable receive wake-up mode of UART.

  Configure the register UR0CN[0]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

  /* Enable receive wake-up mode of UART */

  UART_WUEEnable();

7.2.12. UART_WUEDisable

- **Prototype**

  UART_WUEDisable();

- **Description**

  Disable receive wake-up mode of UART.

  Clear the register UR0CN[0]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

  /* Disable receive wake-up mode of UART */

  UART_WUEDisable();

7.2.13. UART_CREnable

- **Prototype**

  UART_CREnable();

- **Description**

  Enable the receive function of UART.

  Configure the register BA0CN[3]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

  /* Enable the receive function of UART. */

  UART_CREnable();

7.2.14. UART_CRDisable

- **Prototype**

  UART_CRDisable();

- **Description**

  Disable the receive function of UART.

  Clear the register BA0CN[3]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

```
/* Disable the receive function of UART. */
UART_CRDisable();
```

7.2.15. UART_CR9Enable

- **Prototype**

  UART_CR9Enable();

- **Description**

  Enable the function of RC port receive 9 bits.

  Configure the register BA0CN[2]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

```
/* Enable the function of RC port receive 9 bits. */
UART_CR9Enable();
```

7.2.16. UART_CR9Disable

- **Prototype**

  UART_CR9Disable();

- **Description**

  Disable the function of RC port receive 9 bits.

  Clear the register BA0CN[2]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

  /* Disable the function of RC port receive 9 bits. */

  UART_CR9Disable();

7.2.17. UART_ADDEnable

- **Prototype**

  UART_ADDEnable();

- **Description**

  Enable the addrss detect function.

  Configure the register BA0CN[1]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

  /* Enable the addrss detect function. */

  UART_ADDEnable();

7.2.18. UART_ADDDisable

- **Prototype**

  UART_ADDDisable();

- **Description**

  Disable the addrss detect function.

  Clear the register BA0CN[1]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

```
/* Disable the addrss detect function. */
UART_ADDDisable();
```

7.2.19. UART_ABDEnable

- **Prototype**

  UART_ABDEnable();

- **Description**

  Enable the auto baud rate function.

  Configure the register BA0CN[0]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

```
/* Enable the auto baud rate function.*/
UART_ABDEnable();
```

7.2.20. UART_ABDDisable

- **Prototype**

  UART_ABDDisable();

- **Description**

  Disable the auto-baud function.

  Clear the register BA0CN[0]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

```
/* Disable the auto-baud function.*/
UART_ABDDisable();
```

7.2.21. UART_INT_TXEnable

- **Prototype**

  UART_INT_TXIsEnable();

- **Description**

Enable TX interrupt function.

Configure the register INTE1[5]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

```
/* Enable TX interrupt function.   */
UART_INT_TXEnable();
```

7.2.22. UART_INT_TXDisable

- **Prototype**

  UART_INT_TXIsDisable();

- **Description**

  Disable TX interrupt function.

  Clear the register INTE1[5]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

```
/* Enable TX interrupt function.   */
UART_INT_TXEnable();
```

7.2.23. UART_INT_TXIsFlag

- **Prototype**

  UART_INT_TXIsFlag();

- **Description**

  Read the TX interrupt flag.

  Read the register INTF1[5]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

0：No interrupt

1：Interrupt occurred

- **Example**

```
/* Read the TX interrupt flag.  */
unsigned char flag；
flag = UART_INT_TXIsFlag();
```

7.2.24. UART_INT_TXClearFlag

- **Prototype**

  UART_INT_TXClearFlag();

- **Description**

  Clear the TX interrupt flag.

  Clear the register INTF1[5]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

```
/*Clear the TX interrupt flag.  */
UART_INT_TXClearFlag();
```

7.2.25. UART_INT_RCEnable

- **Prototype**

  UART_INT_RCIsEnable();

- **Description**

  Enable RC interrupt function.

  Configure the register INTE1[4]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

```
/* Enable RC interrupt function.  */
UART_INT_RCEnable();
```

7.2.26. UART_INT_RCDisable

- **Prototype**

  UART_INT_RCIsDisable();

- **Description**

  Disable RC interrupt function.

  Clear the register INTE1[4]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

```
/* Enable RC interrupt function.  */
UART_INT_RCEnable();
```

7.2.27. UART_INT_RCIsFlag

- **Prototype**

  UART_INT_RCIsFlag();

- **Description**

  Read the RC interrupt flag.

  Read the register INTF1[4]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read the RC interrupt flag.  */
unsigned char flag；
flag = UART_INT_RCIsFlag();
```

7.2.28. UART_INT_RCClearFlag

- **Prototype**

  UART_INT_RCClearFlag();

- **Description**

  Clear the RC interrupt flag.

Clear the register INTF1[4]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

```
/*Clear the RC interrupt flag.  */
UART_INT_RCClearFlag();
```

7.2.29. UART_GETRC9

- **Prototype**

  UART_GETRC9();

- **Description**

  Read the bit 9 value of receive data.

  Read the register UR0STA[6]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  0：The bit 9 data is 0

  1：The bit 9 data is 1

- **Example**

```
/* Read the bit 9 value of receive data.  */
unsigned char RC9D;
RC9D = UART_GETRC9()>>6;
```

7.2.30. UART_GETPERR

- **Prototype**

  UART_GETPERR();

- **Description**

  Get the Parity Error flag of UART1.

  Read the register UR0STA[5]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  0：No parity error

  1：Parity error

- **Example**

```
/* Get the PERR flag. */
unsigned char PERR_F;
PERR_F = UART_GETPERR()>>5;
```

7.2.31. UART_GETFERR

- **Prototype**

  UART_GETFERR()

- **Description**

  Get the FERR flag of UART1.

  Read the register UR0STA[4]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  0：No framing error

  1：Framing error

- **Example**

```
/* Get the FERR flag. */
unsigned char FERR_F ;
FERR_F = UART_GETFERR()>>4;
```

7.2.32. UART_GETOERR

- **Prototype**

  UART_GETOERR();

- **Description**

  Get the OERR flag of UART1.

  Read the register UR0STA[3]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  0：No overrun error

1：Overrun error

- **Example**

```
/* Get the OERR flag. */
unsigned char OERR_F;
OERR_F = UART_GETOERR()>>3;
```

7.2.33. UART_GETRCIDL

- **Prototype**

  UART_GETRCIDL() ;

- **Description**

  Read the RC port status.

  Read the register UR0STA[2]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  0：Idle

  1：Receiving

- **Example**

```
/* Get the RC port status. */
unsigned char flag ;
flag = UART_GETRCIDL()>>2;
```

7.2.34. UART_GETTRMT

- **Prototype**

  UART_GETTRMT();

- **Description**

  Read the Transmit shift register(TSR) status

  Read the register UR0STA[1]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  0：TSR is empty

  1：TSR is full

- **Example**

```
/* Get the TSR status */
```

```
unsigned char flag ;
flag = UART_GETTRMT()>>1;
```

7.2.35. UART_GETABDOVF

- **Prototype**

  UART_GETABDOVF();

- **Description**

  Read the auto baud rate overflow flag.

  Read the register UR0STA[0]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  0：baudrate not overflow

  1：baudrate has been overflow

- **Example**

```
/* Get the auto baudrate overflow flag */
unsigned char flag ;
flag = UART_GETABDOVF();
```

8. CMP Driver

8.1. Introduction

The following functions are included in Comparator(CMP) Manager Section.

Item	Functions	Description
01	CMP_Open	Enable CMP function, configure the input channel, reference voltage, and select output port
02	CMP_RLOSet	Set the vlotage source of ladder resistor, the resistor divider node, and configure hysteresis function
03	CMP_Enable	Enable CMP function
04	CMP_Disable	Disable CMP function
05	CMP_CMPResult	Read the comparison result of CMP
06	CMP_PInput	Select the CMP positive input channel
07	CMP_NInput	Select the CMP negative input channel
08	CMP_OutputSelect	Select the CMP output channel
09	CMP_OutReverse	Configure the CMP output phase control
10	CMP_2usLPFEnable	Enable the low-pass filter function
11	CMP_2usLPFDisnable	Disable the low-pass filter function
12	CMP_MODESel	Configure the CMP mode control
13	CMP_CPRLOpen	Set the ladder resistor open
14	CMP_CPRLShort	Set the ladder resistor short
15	CMP_RLO_VrefSel	Configure the voltage source of ladder resistor
16	CMP_R_DASel	Configure the resistor divider node
17	CMP_CPDMSet	Configure the hysteresis function
18	CMP_INT_Enable	Enable CMP interrupt function
19	CMP_INT_Disable	Disable CMP interrupt function
20	CMP_INT_IsFlag	Read the CMP interrupt flag
21	CMP_INT_ClearFlag	Clear the CMP interrupt flag

8.2. Functions

8.2.1. CMP_Open

- **Prototype**

  void CMP_Open(unsigned char cpps, unsigned char cpns, unsigned char cmphs, unsigned char cpor, unsigned char enrcc);

- **Description**

  Enable CMP function, configure the input channel, reference voltage, and select output port.

  Configure the register MCCN0/MCCN1

- **Parameter**

cpps [in]：Select positive input channel

CPPS_CH0 ：CH0

CPPS_CH1 ：CH1

CPPS_CH2 ：CH2

CPPS_1V2 ：1.2V

cpns [in]：Select negative input channel

CPNS_CH0 ：CH0

CPNS_CH1 ：CH1

CPNS_CH2 ：CH2

CPNS_RLO ：RLO

cmphs [in]：Select CMP mode

CMPHS_NORMAL ：Normal mode

CMP_LOWPOWER ：Low-power mode

cpor [in]：Select output phase

CPOR_INVERSE ：Inversion

CPOR_NORMAL ：Normal

enrcc [in]：Select output channel

ENRCC_DISABLE ：Disable

ENRCC_RC ：Output to UART RC

ENRCC_PT16 ：Output to PT1.6

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  None

- **Example**

  /* Enable CMP function, set input (positive/negative) channel CH1-CH2, output channel PT1.6, normal mode, output inversion */

  CMP_Open(CPPS_CH1, CPNS_CH2, CMPHS_NORMAL, CPOR_ INVERSE, ENRCC_PT16);

8.2.2. CMP_RLOSet

- **Prototype**

  void CMP_RLOSet(unsigned char cprh, unsigned char cprl, unsigned char cpda, unsigned char cpdm);

- **Description**

  Set the vlotage source of ladder resistor, the resistor divider node, and configure hysteresis function.

  Configure the register MCCN1/MCCN2/MCCN3

- **Parameter**

  cprh [in]：The ladder resistor voltage source

  CPRH_DISABLE ：Disable

CPRH_VDD : Voltage from VDD

CPRH_VDDA : Voltage from VDDA

cprl [in]：The ladder resistor short control

CPRL_OPEN : The ladder resistor open

CPRL_SHORT : The ladder resistor short

cpda [in]：The ladder resistor divider node control

CPDA_0 : 0

CPDA_1DIV32 : 1/32(CPRLH-CPRLL)

CPDA_2DIV32 : 2/32(CPRLH-CPRLL)

CPDA_3DIV32 : 3/32(CPRLH-CPRLL)

CPDA_4DIV32 : 4/32(CPRLH-CPRLL)

CPDA_5DIV32 : 5/32(CPRLH-CPRLL)

CPDA_6DIV32 : 6/32(CPRLH-CPRLL)

CPDA_7DIV32 : 7/32(CPRLH-CPRLL)

CPDA_8DIV32 : 8/32(CPRLH-CPRLL)

CPDA_9DIV32 : 9/32(CPRLH-CPRLL)

CPDA_10DIV32 : 10/32(CPRLH-CPRLL)

CPDA_11DIV32 : 11/32(CPRLH-CPRLL)

CPDA_12DIV32 : 12/32(CPRLH-CPRLL)

CPDA_13DIV32 : 13/32(CPRLH-CPRLL)

CPDA_14DIV32 : 14/32(CPRLH-CPRLL)

CPDA_15DIV32 : 15/32(CPRLH-CPRLL)

CPDA_16DIV32 : 16/32(CPRLH-CPRLL)

CPDA_17DIV32 : 17/32(CPRLH-CPRLL)

CPDA_18DIV32 : 18/32(CPRLH-CPRLL)

CPDA_19DIV32 : 19/32(CPRLH-CPRLL)

CPDA_20DIV32 : 20/32(CPRLH-CPRLL)

CPDA_21DIV32 : 21/32(CPRLH-CPRLL)

CPDA_22DIV32 : 22/32(CPRLH-CPRLL)

CPDA_23DIV32 : 23/32(CPRLH-CPRLL)

CPDA_24DIV32 : 24/32(CPRLH-CPRLL)

CPDA_25DIV32 : 25/32(CPRLH-CPRLL)

CPDA_26DIV32 : 26/32(CPRLH-CPRLL)

CPDA_27DIV32 : 27/32(CPRLH-CPRLL)

CPDA_28DIV32 : 28/32(CPRLH-CPRLL)

CPDA_29DIV32 : 29/32(CPRLH-CPRLL)

CPDA_30DIV32 : 30/32(CPRLH-CPRLL)

CPDA_31DIV32 : 31/32(CPRLH-CPRLL)

cpdm [in]：Hysteresis function control

CPDM0_ENABLE：Enable CPDA[0] hysteresis function, CPDA[0]=CPOB

CPDM1_ENABLE：Enable CPDA[1] hysteresis function, CPDA[1]=CPOB

CPDM2_ENABLE：Enable CPDA[2] hysteresis function, CPDA[2]=CPOB

CPDM3_ENABLE：Enable CPDA[3] hysteresis function, CPDA[3]=CPOB

CPDM4_ENABLE：Enable CPDA[4] hysteresis function, CPDA[4]=CPOB

CPDM_DISABLE：Hysteresis function disable

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  None

- **Example**

```
/*Set CMP negative input source from RLO, reference voltage source from VDDA, the ladder resistor short, divider node 5/32(CPRLH-CPRLL), enable CPDA[0] hysteresis function.  */
CMP_NInput(CPNS_RLO);
CMP_RLOSet(CPRH_VDDA,CPRL_OPEN,CPDA_5DIV32,0x01);
```

8.2.3. CMP_Enable

- **Prototype**

  CMP_Enable();

- **Description**

  Enable CMP function.

  Configure the register MCCN0[0]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  None

- **Example**

```
/* Enable CMP function. */
CMP_Enable();
```

8.2.4. CMP_Disable

- **Prototype**

  CMP_Disable();

- **Description**

  Disable CMP function.

  Clear the register MCCN0[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  None

- **Example**

```
/* Disable CMP function. */
CMP_Disnable();
```

8.2.5. CMP_CMPResult

- **Prototype**

  CMP_CMPResult();

- **Description**

  Read the comparison result of CMP.

  Read the register MCCN0[5].

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  0：Negative input signal > Positive input signal

  1：Positive input signal> Negative input signal

- **Example**

```
/* Read the comparison result */
unsigned char cmp_result;
cmp_result = CMP_CMPResult()>>5;
```

8.2.6. CMP_PInput

- **Prototype**

  CMP_PInput(Sel);

- **Description**

  Select the CMP positive input channel.

  Configure the register MCCN1[3:2].

- **Parameter**

  Sel [in]：the CMP positive input channel

  CPPS_CH0 ：CH0

  CPPS_CH1 ：CH1

CPPS_CH2 : CH2

CPPS_1V2 : 1.2V

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  None

- **Example**

```
/* Set the CH2 as positive input channel */
CMP_PInput(CPPS_CH2 );
```

8.2.7. CMP_NInput

- **Prototype**

  CMP_NInput(Sel);

- **Description**

  Select the CMP negative input channel.

  Configure the register MCCN1[1:0].

- **Parameter**

  Sel [in]：the CMP negative input channel

  CPNS_CH0 : CH0

  CPNS_CH1 : CH1

  CPNS_CH2 : CH2

  CPNS_1V2 : 1.2V

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  None

- **Example**

```
/* Set 1.2V as negative input channel */
CMP_NInput(CPNS_1V2 );
```

8.2.8. CMP_OutputSelect

- **Prototype**

  CMP_OutputSelect(Sel);

- **Description**

  Select the CMP output channel.

  Configure the register MCCN0[7:6]

- **Parameter**

  Sel [in]：Select the output channel.

ENRCC_DISABLE : Disable

ENRCC_RC : Output to UART RC port

ENRCC_PT16 : Output to PT1.6

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  None

- **Example**

```
/* Set the CMP output to PT1.6 */
CMP_OutputSelect(ENRCC_PT16 );
```

8.2.9. CMP_OutReverse

- **Prototype**

  CMP_OutReverse(Sel);

- **Description**

  Configure the CMP output phase control.

  Configure the register MCCN0[3].

- **Parameter**

  Sel [in]：Configure the output phase control.

  CPOR_INVERSE : Inverse

  CPOR_NORMAL : Normal

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  None

- **Example**

```
/* Set the CMP output to PT1.6 */
CMP_OutputSelect(ENRCC_PT16 );
```

8.2.10. CMP_2usLPFEnable

- **Prototype**

  CMP_2usLPFEnable();

- **Description**

  Enable the low-pass filter function.

  Configure the register MCCN0[2]=1.

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  None

- **Example**

  /* Enable the low-pass filter function. */

  CMP_2usLPFEnable();

8.2.11. CMP_2usLPFDisnable

- **Prototype**

  CMP_2usLPFDisable();

- **Description**

  Disable the low-pass filter function.

  Clear the register MCCN0[2].

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  None

- **Example**

  /* Disable the low-pass filter function. */

  CMP_2usLPFDisable();

8.2.12. CMP_MODESel

- **Prototype**

  CMP_MODESel (Sel);

- **Description**

  Configure the CMP mode control.

  Clear the register MCCN0[1].

- **Parameter**

  Sel [in] : Configure the CMP mode control.

  CMPHS_LOWPOWER : Low power mode

  CMPHS_NORMAL : Normal mode

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  None

- **Example**

  /* Set the CMP of low power */

  CMP_MODESel(CMPHS_ LOWPOWER) ;

8.2.13. CMP_CPRLOpen

- **Prototype**

  CMP_CPRLOpen();

- **Description**

  Set the ladder resistor open.

  Configure the register MCCN1[7].

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  None

- **Example**

  /* Set the ladder resistor open. */

  CMP_CPRLOpen();

8.2.14. CMP_CPRLShort

- **Prototype**

  CMP_ CPRLShort ();

- **Description**

  Set the ladder resistor open.

  Clear the register MCCN1[7].

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  None

- **Example**

  /* Set the ladder resistor short. */

  CMP_CPRLShort();

8.2.15. CMP_RLO_VrefSel

- **Prototype**

  CMP_ RLO_VrefSel(Sel);

- **Description**

  Configure the voltage source of ladder resistor.

  Configure the register MCCN1[5:4]

- **Parameter**

  Sel [in] : The ladder resistor voltage source

  CPRH_DISABLE : Disable

  CPRH_VDD : Voltage from VDD

  CPRH_VDDA : Voltage from VDDA

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  None

- **Example**

  /* Set the ladder resistor voltage from VDDA. */

  CMP_ RLO_VrefSel(CPRH_VDDA);

8.2.16. CMP_R_DASel

- **Prototype**

  CMP_ R_DASel(Sel);

- **Description**

  Configure the resistor divider node.

  Configure the register MCCN2[4:0]

- **Parameter**

  Sel [in] : Select the resistor divider node

  CPDA_0 : 0

  CPDA_1DIV32 : 1/32(CPRLH-CPRLL)

  CPDA_2DIV32 : 2/32(CPRLH-CPRLL)

  CPDA_3DIV32 : 3/32(CPRLH-CPRLL)

  CPDA_4DIV32 : 4/32(CPRLH-CPRLL)

  CPDA_5DIV32 : 5/32(CPRLH-CPRLL)

  CPDA_6DIV32 : 6/32(CPRLH-CPRLL)

  CPDA_7DIV32 : 7/32(CPRLH-CPRLL)

  CPDA_8DIV32 : 8/32(CPRLH-CPRLL)

  CPDA_9DIV32 : 9/32(CPRLH-CPRLL)

  CPDA_10DIV32 : 10/32(CPRLH-CPRLL)

  CPDA_11DIV32 : 11/32(CPRLH-CPRLL)

  CPDA_12DIV32 : 12/32(CPRLH-CPRLL)

  CPDA_13DIV32 : 13/32(CPRLH-CPRLL)

  CPDA_14DIV32 : 14/32(CPRLH-CPRLL)

  CPDA_15DIV32 : 15/32(CPRLH-CPRLL)

  CPDA_16DIV32 : 16/32(CPRLH-CPRLL)

CPDA_17DIV32 ：17/32(CPRLH-CPRLL)

CPDA_18DIV32 ：18/32(CPRLH-CPRLL)

CPDA_19DIV32 ：19/32(CPRLH-CPRLL)

CPDA_20DIV32 ：20/32(CPRLH-CPRLL)

CPDA_21DIV32 ：21/32(CPRLH-CPRLL)

CPDA_22DIV32 ：22/32(CPRLH-CPRLL)

CPDA_23DIV32 ：23/32(CPRLH-CPRLL)

CPDA_24DIV32 ：24/32(CPRLH-CPRLL)

CPDA_25DIV32 ：25/32(CPRLH-CPRLL)

CPDA_26DIV32 ：26/32(CPRLH-CPRLL)

CPDA_27DIV32 ：27/32(CPRLH-CPRLL)

CPDA_28DIV32 ：28/32(CPRLH-CPRLL)

CPDA_29DIV32 ：29/32(CPRLH-CPRLL)

CPDA_30DIV32 ：30/32(CPRLH-CPRLL)

CPDA_31DIV32 ：31/32(CPRLH-CPRLL)

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  None

- **Example**

```
/* Set the resistor divider node 5/32(CPRLH-CPRLL) . */
CMP_ R_DASel(CPDA_15DIV32);
```

8.2.17. CMP_CPDMSet

- **Prototype**

```
CMP_ CPDMSet(Set);
```

- **Description**

  Configure the hysteresis function.

  Configure the register MCCN3[4 :0]

- **Parameter**

  cpdm [in]：Hysteresis function control

  CPDM0_ENABLE：Enable CPDA[0] hysteresis function, CPDA[0]=CPOB

  CPDM1_ENABLE：Enable CPDA[1] hysteresis function, CPDA[1]=CPOB

  CPDM2_ENABLE：Enable CPDA[2] hysteresis function, CPDA[2]=CPOB

  CPDM3_ENABLE：Enable CPDA[3] hysteresis function, CPDA[3]=CPOB

  CPDM4_ENABLE：Enable CPDA[4] hysteresis function, CPDA[4]=CPOB

  CPDM_DISABLE：Hysteresis function disable

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  None

- **Example**

  /* Enable CPDA[0] hysteresis function. */

  CMP_ CPDMSet (0x01);

8.2.18. CMP_INT_Enable

- **Prototype**

  CMP_INT_Enable();

- **Description**

  Enable CMP interrupt function.

  Configure the register INTE2[2]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  None

- **Example**

  /* Enable CMP interrupt function. */

  CMP_INT_Enable();

8.2.19. CMP_INT_Disable

- **Prototype**

  CMP_INT_Disable();

- **Description**

  Disable CMP interrupt function.

  Clear the register INTE2[2]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/CMP.h

- **Return Value**

  None

- **Example**

  /* Disable CMP interrupt function. */

  CMP_INT_Disable();

8.2.20. CMP_INT_IsFlag

- **Prototype**

  CMP_INT_IsFlag();

- **Description**

  Read the CMP interrupt flag.

  Read the register INTF2[2]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read the CMP interrupt flag.  */
unsigned char flag；
flag = CMP_INT_IsFlag();
```

8.2.21. CMP_INT_ClearFlag

- **Prototype**

  CMP_INT_ClearFlag();

- **Description**

  Clear the CMP interrupt flag.

  Clear the register INTF2[2]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

```
/*Clear the CMP interrupt flag.  */
CMP_INT_ClearFlag();
```

9. OPAMP Driver

9.1. Introduction

The following functions are included in Rail-to-Rail OPAMP Manager Section.

Item	Functions	Description
01	OPA_Out2Dig_Open	Configure the OPA operate on digital mode, set the output phase, and enable the interrupt function
02	OP1_Enable	Enable OP1 function
03	OP1_Disable	Disable OP1 function
04	OP1_InputShort	Configure the OP1 input short control
05	OP1_INPSet	Configure the OP1 positive input signal source
06	OP1_INN0Set	Configure the OP1 negative input signal source
07	OP1_INN1Set	Configure the OP1 negative input signal source
08	OP1_AnalogyOutSel	Configure the OP1 analog output control
09	OP1_DigitalOutEnable	Enable the OP1 digital output function
10	OP1_DigitalOutDisable	Disable the OP1 digital output function
11	OP1_OutputReverse	Configure the OP1 output phase control
12	OP1_10pF_AppSel	Configure the OP1 built-in capacitor operating mode
13	OP1_FilterEnable	Enable filter function
14	OP1_FilterDisable	Disable filter function
15	OP1_GAINSelect	Configure the negative feedback amplification control
16	OP1_GAINSourceSel	Select the negative feedback source
17	OP1_INT_Trigger	Configure the OP1 interrupt trigger mode
18	OP1_INT_Enable	Enable the OP1 interrupt function
19	OP1_INT_Disable	Disable the OP1 interrupt function
20	OP1_INT_IsFlag	Read the OP1 interrupt flag
21	OP1_INT_ClearFlag	Clear the OP1 interrupt flag

9.2. Functions

9.2.1. OPA_Out2Dig_Open

- **Prototype**

void OPA_Out2Dig_Open(unsigned char ins, unsigned char opdr, unsigned char trig);

- **Description**

  Configure the OPA operate on digital mode, set the output phase, and enable the interrupt function.

  Configure the register OP1CN0

- **Parameter**

  ins[in]：The OP1 input short control

  OPINS_SHORT ：Short

  OPINS_NORMAL ：Open

  opdr[in]：The OP1 output phase control

  OPDR_REVERSE ：Reverse

  OPDR_NORMAL ：Normal

  trig [in]：The OP1 interrupt trigger mode

  OPOEG_EDGEFALL：Edge fall, OPC status 1→0

  OPOEG_EDGERISE ：Edge rise, OPC status 0→1

  OPOEG_LEV ：OPC status 1→0 or 0→1

- **Include**

  Driver/HY17M/HY17M24/OPA.h

- **Return Value**

  None

- **Example**

  /* Set the OPA operate to digital mode, disable the input short control, output normal phase, set the edge rise trigger mode */

  OP1_Out2Dig_Open(OPINS_NORMAL, OPDR_NORMAL, OPOEG_EDGERISE);

9.2.2. OP1_Enable

- **Prototype**

  OP1_Enable();

- **Description**

  Enable OP1 function.

  Configure the register OP1CN0[0] =1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/OPA.h

- **Return Value**

  None

- **Example**

  /* Enable OP1 function. */

  OP1_Enable();

9.2.3. OP1_Disable

- **Prototype**

  OP1_Disable();

- **Description**

  Disable OP1 function.

  Clear the register OP1CN0[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/OPA.h

- **Return Value**

  None

- **Example**

```
/* Disable OP1 function. */
OP1_Disable();
```

9.2.4. OP1_InputShort

- **Prototype**

  OP1_InputShort(Set);

- **Description**

  Configure the OP1 input (positive and negative) short control.

  Configure the register OP1CN0[7]

- **Parameter**

  Set [in]：Configure the input (positive and negative) short control.

  OPINS_SHORT ：Short

  OPINS_NORMAL ：Open

- **Include**

  Driver/HY17M/HY17M24/OPA.h

- **Return Value**

  None

- **Example**

```
/* Set the OP1 input (positive and negative) short. */
OP1_InputShort( OPINS_SHORT );
```

9.2.5. OP1_INPSet

- **Prototype**

  OP1_INP(Set);

- **Description**

Select OP1 positive input selection.

Configure the register OP1NP[7:0]

- **Parameter**

Set [in]：OP1 positive input selection.

OP1INP_AI0_ENABLE	: AI0
OP1INP_AI2_ENABLE	: AI2
OP1INP_AI3_ENABLE	: AI3
OP1INP_AI10_ENABLE	: AI10
OP1INP_REFOI_ENABLE	: REFOI
OP1INP_DAOI_ENABLE	: DAOI
OP1INP_AI13_ENABLE	: AI13
OP1INP_VSS_ENABLE	: VSS

- **Include**

Driver/HY17M/HY17M24/OPA.h

- **Return Value**

None

- **Example**

```
/* Selection the OP1 positive input of AI3. */
OP1_INPSet( OP1INP_AI3_ENABLE );
```

9.2.6. OP1_INN0Set

- **Prototype**

OP1_INN0Set(Set);

- **Description**

Select OP1 negative input selection.

Configure the register OP1NN0[7:0]

- **Parameter**

Set [in]：OP1 negative input selection register.

OP1INN0_AI1_ENABLE	: AI1
OP1INN0_AI4_ENABLE	: AI4
OP1INN0_AI9_ENABLE	: AI9
OP1INN0_REFOI_ENABLE	: REFOI
OP1INN0_R1_fb_ENABLE	: R1_fb
OP1INN0_OPOI_ENABLE	: OPOI
OP1INN0_OPOS_ENABLE	: OPOS
OP1INN0_OPO_ENABLE	: OPO

- **Include**

Driver/HY17M/HY17M24/OPA.h

- **Return Value**

None

- **Example**

/* Selection the OP1 negative input of AI4. */

OP1_INN0Set(OP1INN0_AI4_ENABLE);

9.2.7. OP1_INN1Set

- **Prototype**

OP1_INN1Set(Set);

- **Description**

Select OP1 negative input selection.

Configure the register OP1NN1[2:0]

- **Parameter**

Set [in]：OP1 negative input selection register.

OP1INN1_DAOI_ENABLE ：DAOI

OP1INN1_AI14_ENABLE ：AI14

OP1INN1_VSS_ENABLE ：VSS

- **Include**

Driver/HY17M/HY17M24/OPA.h

- **Return Value**

None

- **Example**

/* Selection the OP1 negative input of AI14. */

OP1_INN1Set(OP1INN0_AI14_ENABLE);

9.2.8. OP1_AnalogyOutSel

- **Prototype**

OP1_AnalogyOut(Sel);

- **Description**

Configure the OP1 analog output control

Configure the register OP1CN0[2:1]

- **Parameter**

Sel [in]：The analog output control

OP1OS_DIGITAL ：Disable

OP1OS_AI0 ：Output to AI0

OP1OS_R1_O ：Output to R1_O

- **Include**

Driver/HY17M/HY17M24/OPA.h

- **Return Value**

None

- **Example**

  /* Set OP1 output to AI0 */

  OP1_AnalogyOut (OP1OS_AI0);

9.2.9. OP1_DigitalOutEnable

- **Prototype**

  OP1_DigitalOutEnable();

- **Description**

  Enable the OP1 digital output function.

  Configure the register OP1CN0[3]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/OPA.h

- **Return Value**

  None

- **Example**

  /* Enable the OP1 digital output function. */

  OP1_DigitalOutEnable();

9.2.10. OP1_DigitalOutDisable

- **Prototype**

  OP1_DigitalOutDisable();

- **Description**

  Disable the OP1 digital output function.

  Clear the register OP1CN0[3]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/OPA.h

- **Return Value**

  None

- **Example**

  /* Disable the OP1 digital output function. */

  OP1_DigitalOutDisable();

9.2.11. OP1_OutputReverse

- **Prototype**

  OP1_OutpuReverse(Set);

- **Description**

  Configure the OP1 output phase control.

  Configure the register OP1CN0[6]

- **Parameter**

  Set [in]：The output phase control

  OPDR_NORMAL ：Normal

  OPDR_REVERSE ：Reverse

- **Include**

  Driver/HY17M/HY17M24/OPA.h

- **Return Value**

  None

- **Example**

```
/* Set the OP1 output normal phase. */
OP2_OutputSet(OPDR_NORMAL);
```

9.2.12. OP1_10pF_AppSel

- **Prototype**

  OP1_10pF_AppSel(Sel);

- **Description**

  Configure the OP1 built-in capacitor operating mode.

  Configure the register OP1CN0[5]

- **Parameter**

  Set [in]：The built-in capacitor operating mode

  OPCS_BUILT_IN ：Built-in capacitor mode, connect OPOI

  OPCS_SAMPLE ：Sampling capacitor mode, connect VSS

- **Include**

  Driver/HY17M/HY17M24/OPA.h

- **Return Value**

  None

- **Example**

```
/* Set the built-in capacitor operate to Sampling capacitor mode */
OP1_10pF_AppSel ( OPCS_SAMPLE );
```

9.2.13. OP1_FilterEnable

- **Prototype**

  OP1_FilterEnable();

- **Description**

  Enable filter function.

  Configure the register OP1CN0[4]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/OPA.h

- **Return Value**

  None

- **Example**

  /* Enable filter function. */

  OP1_FilterEnable();

9.2.14. OP1_FilterDisable

- **Prototype**

  OP1_FilterDisable();

- **Description**

  Disable filter function.

  Clear the register OP1CN0[4]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/OPA.h

- **Return Value**

  None

- **Example**

  /* Disable filter function. */

  OP1_FilterDisable();

9.2.15. OP1_GAINSelect

- **Prototype**

  OP1_GAINSelect(Sel);

- **Description**

  Configure the negative feedback amplification control.

  Configure the register OP1CN1[5:4]

- **Parameter**

  Set [in]：Negative feedback amplification control

  OPGAINS_4 : 4 times

  OPGAINS_8 : 8 times

  OPGAINS_16 : 16 times

  OPGAINS_32 : 32 times

- **Include**

Driver/HY17M/HY17M24/OPA.h

- **Return Value**

  None

- **Example**

  /* The negative feedback amplification 4 times */

  OP1_GAINSelect (OPGAINS_4);

9.2.16. OP1_GAINSourceSel

- **Prototype**

  OP1_GAINSourceSel(Sel);

- **Description**

  Select the negative feedback source.

  Configure the register OP1CN1[3:2]

- **Parameter**

  Set [in]：Negative feedback source

  OPDIEN_VSS ：From VSS

  OPDIEN_REFO：From REFO

  OPDIEN_AI1 ：From AI1

  OPDIEN_DAIO ：From DAIO

- **Include**

  Driver/HY17M/HY17M24/OPA.h

- **Return Value**

  None

- **Example**

  /* The negative feedback source is from AI1. */

  OP1_GAINSourceSel (OPDIEN_AI1);

9.2.17. OP1_INT_Trigger

- **Prototype**

  OP1_INT_Trigger(Sel);

- **Description**

  Configure the OP1 interrupt trigger mode.

  Configure the register OP1CN1[1:0]

- **Parameter**

  Sel [in]：The OP1 interrupt trigger mode

  OPOEG_EDGEFALL：Edge fall, OPC status 1→0

  OPOEG_EDGERISE ：Edge rise, OPC status 0→1

  OPOEG_LEV ：OPC status 1→0 or 0→1

- **Include**

Driver/HY17M/HY17M24/OPA.h

- **Return Value**

  None

- **Example**

  /* Set the OP1 interrupt trigger Edge fall mode. */

  OP1_INT_Trigger (OPOEG_EDGEFALL);

9.2.18. OP1_INT_Enable

- **Prototype**

  OP1_INT_Enable();

- **Description**

  Enable OP1 interrupt function.

  Configure the register INTE2[1]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/OPA.h

- **Return Value**

  None

- **Example**

  /* Enable OP1 interrupt function. */

  OPA_INT_Enable();

9.2.19. OP1_INT_Disable

- **Prototype**

  OP1_INT_Disable();

- **Description**

  Disable OP1 interrupt function.

  Clear the register INTE2[1]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/OPA.h

- **Return Value**

  None

- **Example**

  /* Disable OP1 interrupt function. */

  OPA_INT_Disable();

9.2.20. OP1_INT_IsFlag

- **Prototype**

  OP1_INT_IsFlag();

- **Description**

  Read the OP1 interrupt flag.

  Read the register INTF2[1]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read the OP1 interrupt flag.  */
unsigned char flag；
flag = OP1_INT_IsFlag();
```

9.2.21. OP1_INT_ClearFlag

- **Prototype**

  OP1_INT_ClearFlag();

- **Description**

  Clear the OP1 interrupt flag.

  Clear the register INTF2[1]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/UART.h

- **Return Value**

  None

- **Example**

```
/*Clear the OP1 interrupt flag.  */
OP1_INT_ClearFlag();
```

10. PWR Driver

10.1. Introduction

The following functions are included in Power Manager Section.

Item	Functions	Description
01	PWR_VDDAOpen	Enable VDDA voltage function , and VDDA voltage selection
02	PWR_BGREnable	Band gap enable control
03	PWR_BGRDisable	Band gap disable control
04	PWR_LDOEnable	VDDA LDO enable control
05	PWR_LDODisable	VDDA LDO disable control
06	PWR_VDDASel	VDDA voltage selection
07	PWR_LDOMODE	VDDA output selection
08	PWR_LDOPLEnable	Enable Output pull-down resistor
09	PWR_LDOPLDisable	Disable Output pull-down resistor
10	PWR_ENREFOEnable	Enable REFO voltage output control
11	PWR_ENREFODisable	Disable REFO voltage output control
12	PWR_REFOSel	Reference voltage selection
13	PWR_REFOISel	Select REFOI source

10.2. Functions

10.2.1. PWR_VDDAOpen

- **Prototype**

  void PWR_VDDAOpen(unsigned char vdda);

- **Description**

  Enable VDDA voltage function , and VDDA voltage selection

  Configure the register PWRCN

- **Parameter**

  vdda [in]：VDDA voltage selection

  LDOC_2V4：2.4V

  LDOC_2V6：2.6V

  LDOC_2V9：2.9V

  LDOC_3V3：3.3V

  LDOC_3V6：3.6V

LDOC_4V0：4.0V

LDOC_4V5：4.5V

LDOC_5V0：5.0V

- **Include**

  Driver/HY17M/HY17M24/PWR.h

- **Return Value**

  None

- **Example**

```
/* Enable VDDA voltage function, set VDDA =2.6V */
PWR_VDDAOpen( LDOC_2V6 );
```

10.2.2. PWR_BGREnable

- **Prototype**

  PWR_BGREnable();

- **Description**

  Band gap enable control.

  Configure the register PWRCN[7]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/PWR.h

- **Return Value**

  None

- **Example**

```
/* Enable Band gap voltage. */
PWR_BGREnable();
```

10.2.3. PWR_BGRDisable

- **Prototype**

  PWR_BGRDisable();

- **Description**

  Band gap Disable control.

  Clear the register PWRCN[7]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/PWR.h

- **Return Value**

  None

- **Example**

  /* Disable Band gap voltage. */

  PWR_BGRDisable();

10.2.4. PWR_LDOEnable

- **Prototype**

  PWR_LDOEnable();

- **Description**

  VDDA LDO enable control.

  Configure the register PWRCN[1]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/PWR.h

- **Return Value**

  None

- **Example**

  /* Enable VDDA LDO. */

  PWR_LDOEnable();

10.2.5. PWR_LDODisable

- **Prototype**

  PWR_LDODisable();

- **Description**

  VDDA LDO disable control.

  Clear the register PWRCN[1]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/PWR.h

- **Return Value**

  None

- **Example**

  /* Disable VDDA LDO. */

  PWR_LDODisable();

10.2.6. PWR_VDDASel

- **Prototype**

  PWR_VDDASel(Sel) ;

- **Description**

  Select VDDA voltage. Need to set ENLDO=1 before output VDDA.

  Configure the register PWRCN[6:4]

- **Parameter**

  Sel [in]：VDDA voltage selection

  LDOC_2V4：2.4V

  LDOC_2V6：2.6V

  LDOC_2V9：2.9V

  LDOC_3V3：3.3V

  LDOC_3V6：3.6V

  LDOC_4V0：4.0V

  LDOC_4V5：4.5V

  LDOC_5V0：5.0V

- **Include**

  Driver/HY17M/HY17M24/PWR.h

- **Return Value**

  None

- **Example**

```
/* Set VDDA=2.4V */
PWE_LDOEnable();
PWR_VDDASel(LDOC_2V4 );
```

10.2.7. PWR_LDOMODE

- **Prototype**

  PWR_LDOMODE(Sel) ;

- **Description**

  Select VDDA source. When ENLDO=0, this set will output to VDDA port.

  Configure the register PWRCN[3:2]

- **Parameter**

  Sel [in]：VDDA source selection

  LDOM_DISABLE : Disable

  LDOM_VDD : Output VDD voltage

  LDOM_HIGH : Pull high to VDD by 1.5mA

- **Include**

  Driver/HY17M/HY17M24/PWR.h

- **Return Value**

  None

- **Example**

```
/*Output VDD voltage to VDDA */
```

```
PWR_LDODisable();
PWR_LDOMODE(LDOM_VDD) ;
```

10.2.8. PWR_LDOPLEnable

- **Prototype**

  PWR_LDOPLEnable();

- **Description**

  Enable Output pull-down resistor.

  Configure the register AD1CN5[4]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/PWR.h

- **Return Value**

  None

- **Example**

```
/* Enable Output pull-down resistor. */
PWR_LDOPLEnable();
```

10.2.9. PWR_LDOPLDisable

- **Prototype**

  PWR_LDOPLDisable();

- **Description**

  Disable Output pull-down resistor.

  Clear the register AD1CN5[4]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/PWR.h

- **Return Value**

  None

- **Example**

```
/* Disable Output pull-down resistor. */
PWR_LDOPLDisable();
```

10.2.10. PWR_ENREFOEnable

- **Prototype**

  PWR_ENREFOEnable();

- **Description**

Enable REFO voltage output control.

Configure the register AD1CN5[3]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/PWR.h

- **Return Value**

  None

- **Example**

```
/* Enable REFO voltage output control. */
PWR_ENREFOEnable();
```

10.2.11. PWR_ENREFODisable

- **Prototype**

  PWR_ENREFODisable();

- **Description**

  Disable REFO voltage output control.

  Clear the register AD1CN5[3]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/PWR.h

- **Return Value**

  None

- **Example**

```
/* Disable REFO voltage output control. */
PWR_ENREFODisable();
```

10.2.12. PWR_REFOSel

- **Prototype**

  PWR_REFOSel(Sel);

- **Description**

  Reference voltage selection.

  Configure the register AD1CN1[4:3]

- **Parameter**

  Sel [in]：Reference voltage selection

  REFOS_1V2 : 1.2V

  REFOS_2V0 : 2.0V

REFOS_2V4 : 2.4V

REFOS_3V0 : 3.0V

- **Include**

  Driver/HY17M/HY17M24/PWR.h

- **Return Value**

  None

- **Example**

```
/* Set REFO voltage=1.2V */
PWR_REFOSel(REFOS_1V2);
```

10.2.13. PWR_REFOISel

- **Prototype**

  PWR_REFOISel(Sel);

- **Description**

  Select REFOI source.

  Configure the register MCCN1[6]

- **Parameter**

  Sel [in]：the input range is 0~1

  VRSEL_REFOint : From REFOint

  VRSEL_ACMint : From ACMint

- **Include**

  Driver/HY17M/HY17M24/PWR.h

- **Return Value**

  None

- **Example**

```
/* Set REFOI voltage source is from REFOint */
PWR_REFOISel(VRSEL_REFOint);
```

11. 12-bit Resistance Ladder Driver

11.1. Introduction

The following functions are included in 12-bit resistance ladder (the file:DrvDAC.h)Manager Section.

Item	Functions	Description
01	DAC_Open	Open the DAC
02	LADER_DASelect	DAC (positive/ negative) reference input selection.
03	LADDER_DALH_Enable	Enable 12-bit resistance ladder(DAC) internal output
04	LADDER_DALH_Disable	Disable 12-bit resistance ladder(DAC) internal output
05	LADDER_DAOE_Enable	DAC output enable
06	LADDER_DAOE_Disable	DAC output disable
07	LADDER_ENDA_Enable	Enable the DAC function
08	LADDER_ENDA_Disable	Disable the DACfunction

11.2. Functions

11.2.1. DAC_Open

- **Prototype**

  void DAC_Open(unsigned char daps, unsigned char dans, unsigned int dabit);

- **Description**

  Enable the DAC，select positive and negative reference input, set the initial scale value of DAC output voltage.

  Configure the register DACCN0/ DACCN1

- **Parameter**

  daps [in]：DAC positive reference input selection.

  DAPS_VDDA : VDDA

  DAPS_ REFOI : REFOI

  DAPS_ AI0 : AI0

  DAPS_ AI4 : AI4

  DAPS_ AI5 : AI5

  DAPS_ AI7 : AI7

  DAPS_ AI9 : AI9

DAPS_ AI13 : AI13

dans [in]：DAC negative reference input selection.

DANS_VSS : VSS

DANS_REFOI : REFOI

DANS_ AI0 : AI0

DANS_ AI8 : AI8

dabit [in]：The scale value of output voltage DAO/4095

DABIT[11:0]buffer from MSB to LSB, the input range is 0~4095

- **Include**

  Driver/HY17M/HY17M24/ladder.h

- **Return Value**

  None

- **Example**

  /* Enable DAC, Select positive reference input=VDDA, negative reference input=VSS, DABIT=2048 */

  DAC_Open(DAPS_VDDA,DANS_VSS, 2048);.

11.2.2. LADER_DASelect

- **Prototype**

  LADER_DASelect(DAPSel,DANSel);

- **Description**

  Select positive and negative reference input

  Configure the register DACCN0[5:4]/ DACCN0[2:0]

- **Parameter**

  daps [in]：DAC positive reference input selection.

  DAPS_VDDA : VDDA

  DAPS_ REFOI : REFOI

  DAPS_ AI0 : AI0

  DAPS_ AI4 : AI4

  DAPS_ AI5 : AI5

  DAPS_ AI7 : AI7

  DAPS_ AI9 : AI9

  DAPS_ AI13 : AI13

  dans [in]：DAC negative reference input selection.

  DANS_VSS : VSS

  DANS_REFOI : REFOI

  DANS_ AI0 : AI0

  DANS_ AI8 : AI8

- **Include**

  Driver/HY17M/HY17M24/ladder.h

- **Return Value**

  None

- **Example**

  /* Select positive reference input=VDDA, negative reference input=VSS. */

  LADER_DASelect (DAPS_VDDA,DANS_VSS);

11.2.3. LADDER_DALH_Enable

- **Prototype**

  LADDER_DALH_Enable();

- **Description**

  Enable 12-bit resistance ladder(DAC) internal output

  Configure the register DACCN1[3]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/ladder.h

- **Return Value**

  None

- **Example**

  /* Enable 12-bit resistance ladder(DAC) internal output. */

  LADDER_DALH_Enable();

11.2.4. LADDER_DALH_Disable

- **Prototype**

  LADDER_DALH_Disable();

- **Description**

  Disable 12-bit resistance ladder(DAC) internal output

  Clear the register DACCN1[3]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/ladder.h

- **Return Value**

  None

- **Example**

  /* Disable 12-bit resistance ladder(DAC) internal output. */

  LADDER_DALH_Disable();

11.2.5. LADDER_DAOE_Enable

- **Prototype**

  LADDER_DAOE_Enable();

- **Description**

  DAC output enable.

  Configure the register DACCN1[1]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/ladder.h

- **Return Value**

  None

- **Example**

  /* DAC output enable. */

  LADDER_DAOE_Enable();

11.2.6. LADDER_DAOE_Disable

- **Prototype**

  LADDER_DAOE_Disable();

- **Description**

  DAC output disable.

  Clear the register DACCN1[1]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/ladder.h

- **Return Value**

  None

- **Example**

  /* DAC output disable. */

  LADDER_DAOE_Disable();

11.2.7. LADDER_ENDA_Enable

- **Prototype**

  LADDER_ENDA_Enable();

- **Description**

  Enable the DAC function.

  Configure the register DACCN1[0]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/ladder.h

- **Return Value**

  None

- **Example**

  /* Enable the DAC function. */

  LADDER_ENDA_Enable();

11.2.8. LADDER_ENDA_Disable

- **Prototype**

  LADDER_ENDA_Disable();

- **Description**

  Disable the DAC function.

  Clear the register DACCN1[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/ladder.h

- **Return Value**

  None

- **Example**

  /* Disable the DAC function. */

  LADDER_ENDA_Disable();

12. I2C Driver

12.1. Introduction

The following functions are included in I2C Manager Section.

Item	Functions	Description
01	I2C_Open	Enable the I2C and configure the I2C bus clock
02	I2C_TimeOutOpen	Enable TimeOut function, set clock pre scale and time out limit
03	I2C_SlaveSet	Enable slave mode，set address, choose whether to enable GC
04	I2C_Ctrl	To set I2C control bit include Start, Stop, Acknowledge, Interrupt in control register.
05	I2CEnable	Enable the I2C function
06	I2CDisable	Disable the I2C function
07	I2C_TimeOutEnable	Ensable the I2C Time-out function
08	I2C_TimeOutDisable	Disable the I2C Time-out function
09	I2C_GCRstEnable	Enable I2C reset function
10	I2C_GCRstDisable	Disable I2C reset function
11	I2C_SCLCLKSet	Configure the I2C bus clock
12	I2C_SlaveModeEnable	Enable I2C slave mode
13	I2C_SlaveModeDisable	Disable I2C slave mode
14	I2C_I2CER_READ	Read the I2C error flag
15	I2C_I2CER_CLEAR	Clear the I2C error flag
16	I2C_MACTF_READ	Read the I2C master mode flag
17	I2C_SACTF_READ	Read the I2C slave mode flag
18	I2C_RDBF_READ	Read the I2C received stop/repeat-start flag
19	I2C_RWF_READ	Read the I2C read/write status flag
20	I2C_DFF_READ	Read the I2C data field flag
21	I2C_ACKF_READ	Read the I2C ACK flag
22	I2C_GCF_READ	Read the I2C general call flag
23	I2C_ARBF_READ	Read the I2C arbitration lost flag
24	I2C_I2CTF_READ	Read the Time-out flag
25	I2C_I2CTF_CLEAR	Clear the Time-out flag
26	I2C_TimeOutCLKSel	Set the time-out clock pre scale
27	I2C_TimeOutCYCSel	Set the time out limit
28	I2C_SendData	To set a byte of data to be sent.
29	I2C_ReceiveData	Read the data form receiver data buffer.
30	I2C_INT_Enable	Enable the I2C Interrupt
31	I2C_INT_Disable	Disable the I2C Interrupt
32	I2C_INT_IsFlag	Read the I2C Interrupt flag.
33	I2C_INT_ClearFlag	Clear the I2C Interrupt flag.
34	I2C_ERINT_Enable	Enable the I2C error Interrupt
35	I2C_ERINT_Disable	Disable the I2C error Interrupt
36	I2C_ERINT_IsFlag	Read the I2C error Interrupt flag.
37	I2C_ERINT_ClearFlag	Clear the I2C error Interrupt flag.

12.2. Functions

Note：Configure the IIC register after enable IIC

12.2.1. I2C_Open

- **Prototype**

  void I2C_Open(unsigned char dbr);

- **Description**

  Enable the I2C and configure the I2C bus clock

  Configure the register CFG0/CRG0/INTE1

- **Parameters**

  dbr [in]：Set CRG value. It could be 0x00~0xff. Data Baud Rate = (I2CLK/(4*(CRG+1)))

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

  /* Enable I2C and set CRG value 51 */

  I2C_Open(51);

12.2.2. I2C_TimeOutOpen

- **Prototype**

  void I2C_TimeOutOpen(unsigned int cks, unsigned char cyc);

- **Description**

  Enable Time-Out function, and set the clock pre scale and time out limit.

  Configure the register CFG0/TOC0

- **Parameters**

  cks [in]：Configure Time-out clock pre scale

  DI2C_CPUCKDIV1 : CPU_CK/1

  DI2C_CPUCKDIV2 : CPU_CK/2

  DI2C_CPUCKDIV4 : CPU_CK/4

  DI2C_CPUCKDIV8 : CPU_CK/8

  DI2C_CPUCKDIV16 : CPU_CK/16

  DI2C_CPUCKDIV32 : CPU_CK/32

  DI2C_CPUCKDIV64 : CPU_CK/64

  DI2C_CPUCKDIV128 : CPU_CK/128

  cyc [in]：Configure time out Limit

  I2CTLT_CLKPS1 : 1*CLKPS Cycle

  I2CTLT_CLKPS2 : 2*CLKPS Cycle

  I2CTLT_CLKPS3 : 3*CLKPS Cycle

I2CTLT_CLKPS4 : 4*CLKPS Cycle
I2CTLT_CLKPS5 : 5*CLKPS Cycle
I2CTLT_CLKPS6 : 6*CLKPS Cycle
I2CTLT_CLKPS7 : 7*CLKPS Cycle
I2CTLT_CLKPS8 : 8*CLKPS Cycle
I2CTLT_CLKPS9 : 9*CLKPS Cycle
I2CTLT_CLKPS10 : 10*CLKPS Cycle
I2CTLT_CLKPS11 : 11*CLKPS Cycle
I2CTLT_CLKPS12 : 12*CLKPS Cycle
I2CTLT_CLKPS13 : 13*CLKPS Cycle
I2CTLT_CLKPS14 : 14*CLKPS Cycle
I2CTLT_CLKPS15 : 15*CLKPS Cycle
I2CTLT_CLKPS16 : 16*CLKPS Cycle

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

```
/* Enable Time-Out function, set clock pre scale / 32 and time out limit=15 * CLKps Cycle */
I2C_TimeOutOpen (DI2C_CPUCKDIV32, I2CTLT_CLKPS15);
```

12.2.3. I2C_SlaveSet

- **Prototype**

  void I2C_SlaveSet(unsigned int sid);

- **Description**

  Enable slave mode, and set the location address.

  Configure the register CFG0/SID0

- **Parameters**

  sid [in]：Set SID value. It could be 0x00~0xFF.

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

```
/* Enable Slave mode, SID setting 0xA0*/
I2C_SlaveSet ( 0xA0 );
```

12.2.4. I2C_Ctrl

- **Prototype**

void I2C_Ctrl(unsigned char S, unsigned char P, unsigned char I, unsigned char A,)

- **Description**

  To set I2C control bit include Start, Stop, Acknowledge, Interrupt in control register.

  Configure the register ACT0

- **Parameters**

  S [in]：START singal

  0：Normal

  1：Generate START or repeat START signal.

  P [in]：STOP singal

  0：Normal

  1：Generate STOP singal.

  I [in]：Interrupt

  0：Normal

  1：Interrupt occurred

  A [in]：Acknowledge

  0：no reply ACK or reply NACK

  1：has replied ACK

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

```
/* Generate START signal    */
I2C_Ctrl(1,0,0,0);
```

12.2.5. I2CEnable

- **Prototype**

  I2CEnable();

- **Description**

  Enable the I2C function.

  Configure the register CFG0[0]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

```
/* Enable the I2C function. */
```

I2CEnable();

12.2.6. I2CDisable

- **Prototype**

  I2CDisable();

- **Description**

  Disable the I2C function.

  Clear the register CFG0[0]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

  /* Disable the I2C function. */

  I2CDisable();

12.2.7. I2C_TimeOutEnable

- **Prototype**

  I2C_TimeOutEnable();

- **Description**

  Enable Time-Out function.

  Configure the register CFG0[1]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

  /* Enable Time-Out function. */

  I2C_TimeOutEnable();

12.2.8. I2C_TimeOutDisable

- **Prototype**

  I2C_TimeOutDisable();

- **Description**

  Disable Time-Out function.

Clear the register CFG0[1]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

```
/* Disable Time-Out function. */
I2C_TimeOutDisable();
```

12.2.9. I2C_GCRstEnable

- **Prototype**

```
I2C_GCRstEnable();
```

- **Description**

  Enable I2C reset function.

  Configure the register CFG0[2]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

```
/* Enable I2C reset function. */
I2C_GCRstEnable();
```

12.2.10. I2C_GCRstDisable

- **Prototype**

```
I2C_GCRstDisable();
```

- **Description**

  Disable I2C reset function.

  Clear the register CFG0[2]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

  /* Disable I2C reset function. */

  I2C_GCRstDisable();

12.2.11. I2C_SCLCLKSet

- **Prototype**

  I2C_SCLCLKSet(Set);

- **Description**

  Configure the I2C bus clock. Data Baud Rate(Hz)=I2C_CK/[4*(Set+1)]

  Configure the register CRG0

- **Parameters**

  Set [in]：Set CRG value. It could be 0x00~0xff.

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

  /* Set CRG value 51*/

  I2C_SCLCLKSet(51);

12.2.12. I2C_SlaveModeEnable

- **Prototype**

  I2C_SlaveModeEnable();

- **Description**

  Enable I2C slave mode.

  Configure the register ACT0[7]=1

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

  /* Enable I2C slave mode. */

  I2C_SlaveModeEnable();

12.2.13. I2C_SlaveModeDisable

- **Prototype**

  I2C_SlaveModeDisable();

- **Description**

  Disable I2C slave mode.

  Clear the register ACT0[7]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

```
/* Disable I2C slave mode. */
I2C_SlaveModeDisable();
```

12.2.14. I2C_I2CER_READ

- **Prototype**

  I2C_I2CER_READ();

- **Description**

  Read the I2C error flag.

  Configure the register ACT0[4]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read the I2C error flag. */
unsigned char Flag;
Flag=I2C_I2CER_READ();
```

12.2.15. I2C_I2CER_CLEAR

- **Prototype**

  I2C_I2CER_CLEAR();

- **Description**

  Clear the I2C error flag.

  Clear the register ACT0[4]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

  /* Clear the I2C error flag. */

  I2C_I2CER_CLEAR();

12.2.16. I2C_MACTF_READ

- **Prototype**

  I2C_MACTF_READ();

- **Description**

  Read the I2C master mode flag.

  Read the register STA0[7]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  0：Disable master mode

  1：Enable master mode

- **Example**

  /* Read the I2C master mode flag. */

  I2C_MACTF_READ();

12.2.17. I2C_SACTF_READ

- **Prototype**

  I2C_SACTF_READ();

- **Description**

  Read the I2C slave mode flag.

  Read the register STA0[6]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  0：Disable slave mode

  1：Enable slave mode

- **Example**

```
/* Read the I2C slaver mode flag. */
I2C_SACTF_READ();
```

12.2.18. I2C_RDBF_READ

- **Prototype**

  I2C_RDBF_READ();

- **Description**

  Read the I2C received stop/repeat-start flag.

  Read the register STA0[5]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  0：Normal

  1：Received stop/repeat-start has been received or transmitted.

- **Example**

```
/* Read the I2C received stop/repeat-start flag. */
unsigned char Flag;
Flag=I2C_RDBF_READ();
```

12.2.19. I2C_RWF_READ

- **Prototype**

  I2C_RWF_READ();

- **Description**

  Read the I2C read/write status flag.

  Read the register STA0[4]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  0：Write command has been received or transmitted.

  1：Read command has been received or transmitted.

- **Example**

```
/* Read the I2C read/write status flag. */
unsigned char Flag;
Flag=I2C_RWF_READ();
```

12.2.20. I2C_DFF_READ

- **Prototype**

  I2C_DFF_READ();

- **Description**

  Read the I2C data field flag.

  Read the register STA0[3]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  0：Normal.

  1：I2C Data Byte has been transmitted or received.

- **Example**

```
/* Read the I2C data field flag. */
unsigned char Flag;
Flag=I2C_DFF_READ();
```

12.2.21. I2C_ACKF_READ

- **Prototype**

  I2C_ACKF_READ();

- **Description**

  Read the I2C ACK flag.

  Read the register STA0[2]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  0：Ack has been transmitted or received.

  1：Ack hasn’t been transmitted or received.

- **Example**

```
/* Read the I2C ACK flag. */
unsigned char Flag;
Flag=I2C_ACKF_READ();
```

12.2.22. I2C_GCF_READ

- **Prototype**

  I2C_GCF_READ();

- **Description**

  Read the I2C general call flag.

  Read the register STA0[1]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  0：Normal.

  1：Currently General Call Operation

- **Example**

```
/* Read the I2C general call flag. */
unsigned char Flag;
Flag=I2C_GCF_READ();
```

12.2.23. I2C_ARBF_READ

- **Prototype**

  I2C_ARBF_READ();

- **Description**

  Read the I2C arbitration lost flag.

  Read the register STA0[0]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  0：Normal.

  1：Arbitration Lost.

- **Example**

```
/* Read the I2C arbitration lost flag. */
unsigned char Flag;
Flag=I2C_ARBF_READ();
```

12.2.24. I2C_I2CTF_READ

- **Prototype**

  I2C_I2CTF_READ();

- **Description**

  Read the Time-out flag.

  Read the register TOC0[7]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  0：Normal.

  1：I2C Bus Clock Stretching Time-Out

- **Example**

```
/* Read the Time-out flag. */
unsigned char Flag;
Flag=I2C_I2CTF_READ();
```

12.2.25. I2C_I2CTF_CLEAR

- **Prototype**

  I2C_I2CTF_CLEAR();

- **Description**

  Clear the Time-out flag.

  Clear the register TOC0[7]

- **Parameter**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

```
/* Clear the Time-out flag. */
I2C_I2CTF_CLEAR();
```

12.2.26. I2C_TimeOutCLKSel

- **Prototype**

  I2C_TimeOutCLKSel(cks);

- **Description**

  Set the clock pre scale.

  Configure the register TOC0[6:4]

- **Parameters**

  cks [in]：Configure Time-out clock pre scale

  DI2C_CPUCKDIV1 ：CPU_CK/1

  DI2C_CPUCKDIV2 ：CPU_CK/2

DI2C_CPUCKDIV4 : CPU_CK/4
DI2C_CPUCKDIV8 : CPU_CK/8
DI2C_CPUCKDIV16 : CPU_CK/16
DI2C_CPUCKDIV32 : CPU_CK/32
DI2C_CPUCKDIV64 : CPU_CK/64
DI2C_CPUCKDIV128 : CPU_CK/128

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

```
/* Set Time-Out clock pre scale / 32. */
I2C_TimeOutCLKSel (DI2C_CPUCKDIV32);
```

12.2.27. I2C_TimeOutCYCSel

- **Prototype**

  I2C_TimeOutCYCSel(cyc);

- **Description**

  Set time out limit.

  Configure the register CFG0/TOC0

- **Parameters**

  cyc [in] : Configure time out Limit

  I2CTLT_CLKPS1 : 1*CLKPS Cycle
  I2CTLT_CLKPS2 : 2*CLKPS Cycle
  I2CTLT_CLKPS3 : 3*CLKPS Cycle
  I2CTLT_CLKPS4 : 4*CLKPS Cycle
  I2CTLT_CLKPS5 : 5*CLKPS Cycle
  I2CTLT_CLKPS6 : 6*CLKPS Cycle
  I2CTLT_CLKPS7 : 7*CLKPS Cycle
  I2CTLT_CLKPS8 : 8*CLKPS Cycle
  I2CTLT_CLKPS9 : 9*CLKPS Cycle
  I2CTLT_CLKPS10 : 10*CLKPS Cycle
  I2CTLT_CLKPS11 : 11*CLKPS Cycle
  I2CTLT_CLKPS12 : 12*CLKPS Cycle
  I2CTLT_CLKPS13 : 13*CLKPS Cycle
  I2CTLT_CLKPS14 : 14*CLKPS Cycle
  I2CTLT_CLKPS15 : 15*CLKPS Cycle
  I2CTLT_CLKPS16 : 16*CLKPS Cycle

- **Include**

Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

  /* Set time out limit=15 * CLKps Cycle */

  I2C_TimeOutOpen (I2CTLT_CLKPS15);

12.2.28. I2C_SendData

- **Prototype**

  I2C_SendData(udata);

- **Description**

  Set data to be sent.

  Configure the register TDB0

- **Parameters**

  udata [in]：The data to be sent, 0x00~0xff.

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

  /* Set byte data 0xa0 into transmitter data buffer register. */

  I2C_SendData(0xa0);

12.2.29. I2C_ReceiveData

- **Prototype**

  I2C_ReceiveData(uRec);

- **Description**

  Read the data form receiver data buffer.

  Read the register RDB0

- **Parameters**

  udata [in]：The data to be sent, 0x00~0xff.

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  Received data

- **Example**

  /* Read the data form receiver data buffer. */

  unsigned char Data;

  I2C_ReceiveData(Data);

12.2.30. I2C_INT_Enable

- **Prototype**

  I2C_INT_Enable()

- **Description**

  Enable I2C interrupt function.

  Configure the register INTE1[2]=1.

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

```
/* Enable I2C interrupt function. */
I2C_INT_Enable();
```

12.2.31. I2C_INT_Disable

- **Prototype**

  I2C_INT_Disable()

- **Description**

  Disable I2C interrupt function.

  Clear the register INTE1[2].

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

```
/* Disable I2C interrupt function. */
I2C_INT_Disable();
```

12.2.32. I2C_INT_IsFlag

- **Prototype**

  I2C_INT_IsFlag()

- **Description**

  Read the I2C interrupt flag.

  Read the register INTF1[2].

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  0：No interrupt

  1：Interrupt occurred

- **Example**

```
/* Read the I2C interrupt flag. */
unsigned char flag;
flag = I2C_INT_IsFlag();
```

12.2.33. I2C_INT_ClearFlag

- **Prototype**

  I2C_INT_ClearFlag()

- **Description**

  Clear the I2C interrupt flag.

  Clear the register INTF1[2].

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

```
/* Clear the I2C interrupt flag. */
I2C_INT_ClearFlag();
```

12.2.34. I2C_ERINT_Enable

- **Prototype**

  I2C_ERINT_Enable()

- **Description**

  Enable I2C error interrupt function.

  Configure the register INTE1[3]=1.

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

None

- **Example**

/* Enable I2C error interrupt function. */

I2C_ERINT_Enable();

12.2.35. I2C_ERINT_Disable

- **Prototype**

I2C_ERINT_Disable()

- **Description**

Disable I2C error interrupt function.

Clear the register INTE1[3].

- **Parameters**

None

- **Include**

Driver/HY17M/HY17M24/I2C.h

- **Return Value**

None

- **Example**

/* Disable I2C error interrupt function. */

I2C_ERINT_Disable();

12.2.36. I2C_ERINT_IsFlag

- **Prototype**

I2C_ERINT_IsFlag()

- **Description**

Read the I2C error interrupt flag.

Read the register INTF1[3].

- **Parameters**

None

- **Include**

Driver/HY17M/HY17M24/I2C.h

- **Return Value**

0：No interrupt

1：Interrupt occurred

- **Example**

/* Read the I2C error interrupt flag. */

unsigned char flag;

flag = I2C_ERINT_IsFlag();

12.2.37. I2C_ERINT_ClearFlag

- **Prototype**

  I2C_ERINT_ClearFlag()

- **Description**

  Clear the I2C error interrupt flag.

  Clear the register INTF1[3].

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/I2C.h

- **Return Value**

  None

- **Example**

  /* Clear the I2C error interrupt flag. */

  I2C_ERINT_ClearFlag();

13. BIE Driver

13.1. Introduction

The following functions are included in Built-In EPROM (BIE) Manager Section

Item	Functions	Description
01	BIE2_Enable	Enable BIE function
02	BIE2_Disable	Disable BIE function
03	BIE2_DataWrite	Write data to BIE
04	BIE2_Clear	Clear Information Block 2 data
05	BIE2_DataRead	Read the data of BIE

13.2. Functions

13.2.1. BIE2_Enable

- **Prototype**

  BIE2_Enable();

- **Description**

  Enable BIE function.

  Configure the register BIE2ARH[7]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/BIE.h

- **Return Value**

  None

- **Example**

  /* Enable BIE function. */

  BIE2_Enable();

13.2.2. BIE2_Disable

- **Prototype**

  BIE2_Disable();

- **Description**

Disable BIE function.

Clear the register BIE2ARH[7]

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/BIE.h

- **Return Value**

  None

- **Example**

```
/* Disable BIE function. */
BIE2_Disable();
```

13.2.3. BIE2_DataWrite

- **Prototype**

  BIE2_DataWrite();

- **Description**

  Write data to BIE. Set data into BIE writer buffer EERD0[7:0]~EERD31[7:0], before enable write control.

  Configure the register EECR1[0]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/BIE.h

- **Return Value**

  None

- **Example**

```
/* Write data 0x55 into BIE address 0. */
EERD0=0x55;
BIE2_DataWrite();
```

13.2.4. BIE2_Clear

- **Prototype**

  BIE2_Clear();

- **Description**

  Clear Information Block 2 data.

  Configure the register BIE2CN[5]=1

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/BIE.h

- **Return Value**

  None

- **Example**

  /* Clear Information Block 2 data. */

  BIE2_Clear();

13.2.5. BIE2_DataRead

- **Prototype**

  BIE2_DataRead();

- **Description**

  Read the data of BIE. BIE data will reload into EERD0~EERD31.

  Configure the register EECR2/EERD0~EERD31

- **Parameters**

  None

- **Include**

  Driver/HY17M/HY17M24/BIE.h

- **Return Value**

  None

- **Example**

  /* Read the data of BIE. */

  BIE_DataRead();

14. Library

14.1. Library File

HY17M24 Driver C Library source code under software table H08 CIDE\Driver\HY17M\HY17M24.

15. Revision History

Version	Page	Revision Summary	The Date Of Revision
V01	ALL	First edition	2020/04/28
V02	31、34	Change the defined name of HAOM HAOM_1843KHZ, HAOM_4147KHZ, HAOM_8755KHZ, HAOM_17510KHZ	2021/09/15

16. Library Change List

Date	Previous Version List		New version List	
	Version	Bug List	Version	Improvement
2020-03-24	V01	None	V01	None
2021-09-15	V01	None	V02	Change the defined name of HAOM