



CANTUS-CAN

- *GPIO* -

32bits EISC Microprocessor *CANTUS*

Ver 1.1
April 24, 2013

Advanced Digital Chips Inc.

History

2013-02-19	Released	
2013-04-24	Modified	CANTUS-CAN

CANTUS-CAN Application Note : #0002B GPIO

©Advanced Digital Chips Inc.

All right reserved.

No part of this document may be reproduced in any form without written permission from Advanced Digital Chips Inc.

Advanced Digital Chips Inc. reserves the right to change in its products or product specification to improve function or design at any time, without notice.

Office

(Gwanyang-dong, Keumkang Pentierium IT Tower) 22F, A-Tower,
282, Hagui-ro, Dongan-gu, Anyang-si, Gyeonggi-do, SEOUL 431-810 Rep. of KOREA

Tel : +82-31-463-7500

Fax : +82-31-463-7588

URL : <http://www.adc.co.kr>

— Table of Contents —

1 SUMMARY.....6

2 GPIO INPUT MODE.....7

2.1 REGISTER SET7

2.2 FUNCTION SET.....9

3 GPIO EDGE DETECT11

3.1 REGISTER SET11

3.2 FUNCTION SET 1.....14

3.3 FUNCTION SET 2.....16

4 GPIO OUTPUT MODE.....17

4.1 REGISTER SET17

4.2 FUNCTION SET 1.....19

— List of Figures —

그림 2-1 GPIO Input Mode Register Set 7
그림 3-1 GPIO Edge Detect Register Set 11
그림 4-1 GPIO Output Mode Register Set..... 17

— List of Tables —

<i>Æ 2-1 Port Alternate Function 0 Register</i>	7
<i>Æ 2-2 Port Direction Input Mode Setting Register (PxIDIR)</i>	8
<i>Æ 2-3 Port Input Data Level Register (PxILEV)</i>	8
<i>Æ 3-1 Port Rising Edge Detect Register (PxRED)</i>	12
<i>Æ 3-2 Port Falling Edge Detect Register (PxFED)</i>	12
<i>Æ 3-3 Port Edge Detect Status Register (PxEDS)</i>	13
<i>Æ 4-1 Port Direction Output Mode Setting Register (PxODIR)</i>	18
<i>Æ 4-2 Port Output Data High Level Setting Register (PxOHIGH)</i>	18
<i>Æ 4-3 Port Output Data Low Level Setting Register (PxOLOW)</i>	18
<i>Æ 4-4 Port Output Data Level Register (PxOLEV)</i>	18

1 Summary

이 문서는 CANTUS의 GPIO(General Purpose I/O)에 대한 Application Note이다.

CANTUS의 GPIO는 8개의 Pin으로 구성된 PORT0, 1, 2, 3, 4, 5와 5개의 Pin으로 구성된 PORT6로 이루어져 총 45개의 I/O Port를 제공한다. 각 Port는 Input / Output Mode를 지원하며, Input Mode에서 Edge Detect(IRQ) Mode를 사용할 수 있다.

또한 Output Mode에서 Port Pin 단위의 High/Low 출력이 가능 하다.

이 문서는 CANTUS의 GPIO를 Input Mode / Input Mode에서 Edge Detect / Output Mode로 사용하기 위한 방법을 기술한다.

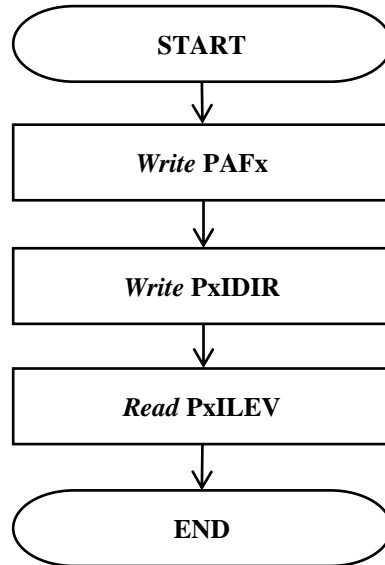
CANTUS의 GPIO는 CANTUS Datasheet ‘8 GPIO(GENERAL PURPOSE I/O)’를 참조 하라.

2 GPIO Input Mode

2.1 Register Set

CANTUS의 GPIO를 Input Mode로 사용하기 위해 다음과 같은 순서로 Register를 설정한다.

그림 2-1 GPIO Input Mode Register Set



PAFx

해당 Port의 Pin을 GPIO로 사용하기 위해 ‘11b’을 쓴다. Port의 Pin의 Alternation은 해당 bit(2)에 쓰는 값으로 설정 된다.

표 2-1 Port Alternate Function 0 Register

Group	Index	Pin	1 st	2 nd	3 rd	4 th (default)
			00	01	10	11
PAF0 0x80020020	0	76			AD[0]	P0.0
	1	77			AD[1]	P0.1
	2	78			AD[2]	P0.2
	3	79			AD[3]	P0.3
	4	82			AD[4]	P0.4
	5	83			AD[5]	P0.5
	6	84			AD[6]	P0.6
	7	85			AD[7]	P0.7

PxIDR (Port Direction Input Mode Setting Register)

해당 Port의 Pin을 Input Mode로 설정하기 위해 '1b'을 해당 bit(1)에 쓴다.

표 2-2 Port Direction Input Mode Setting Register (PxIDIR)

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 8	R	Reserved	-
7	W	Px.7IDR : Px.7 Direction Input Mode Setting bit	-
6	W	Px.6IDR : Px.6 Direction Input Mode Setting bit	-
5	W	Px.5IDR : Px.5 Direction Input Mode Setting bit	-
4	W	Px.4IDR : Px.4 Direction Input Mode Setting bit	-
3	W	Px.3IDR : Px.3 Direction Input Mode Setting bit	-
2	W	Px.2IDR : Px.2 Direction Input Mode Setting bit	-
1	W	Px.1IDR : Px.1 Direction Input Mode Setting bit	-
0	W	Px.0IDR : Px.0 Direction Input Mode Setting bit	-

PxILEV (Port Input Data Level Register)

해당 Port의 Input Level을 읽는다.

표 2-3 Port Input Data Level Register (PxILEV)

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 8	R	Reserved	-
7	R	Px.7ILEV : Px.7 Input Level bit 0 : Low Level 1 : High Level	Pull-up
6	R	Px.6ILEV : Px.6 Input Level bit 0 : Low Level 1 : High Level	Pull-up
5	R	Px.5ILEV : Px.5 Input Level bit 0 : Low Level 1 : High Level	Pull-up
4	R	Px.4ILEV : Px.4 Input Level bit 0 : Low Level 1 : High Level	Pull-up
3	R	Px.3ILEV : Px.3 Input Level bit 0 : Low Level 1 : High Level	Pull-up
2	R	Px.2ILEV : Px.2 Input Level bit 0 : Low Level 1 : High Level	Pull-up
1	R	Px.1ILEV : Px.1 Input Level bit 0 : Low Level 1 : High Level	Pull-up
0	R	Px.0ILEV : Px.0 Input Level bit 0 : Low Level 1 : High Level	Pull-up

2.2 Function Set

아래는 CANTUS의 Port 0의 7번 Pin을 PAF0로 GPIO로 설정하고, P0IDIR로 Input Mode로 설정한 후 P0ILEV를 읽어 Input Level 상태에 따라 UART로 string을 출력하는 예이다.

```
main()
{
  ...

  *R_PAF0 = F_PAF0_7_GPIO;
  *R_P0IDIR = (1<<7);

  while(1)
  {
    if( (1<<7) == (*R_P0ILEV & (1<<7)) )
    {
      debugstring("P0.7 High Level\r\n");
    }
    else
    {
      debugstring("P0.7 Low Level\r\n");
    }
  }
  ...
}
```

*R_PAF0

Port Alternate Function Register는 SDK/Include/CANTUS/paf.h에

```
#define R_PAF0 ((volatile U32*)0x80020020)
#define F_PAF0_7_GPIO      (3 <<14)
```

로 정의되어 있다.

▶ 여기서는 Port Alternate Function Register 0에 F_PAF0_7_GPIO을 대입하여 Port 0의 7번 Pin을 GPIO로 설정한다.

*R_P0IDIR

Port Direction Input Mode Setting Register는 SDK/Include/CANTUS/gpio.h에

```
#define R_P0IDIR ((volatile unsigned int*)0x80023404)
```

로 정의되어 있다.

▶ 여기서는 Port Direction Input Mode Setting Register 0에 (1<<7)을 대입하여 Port 0의 7번 Pin을 Input Mode로 설정 한다.

***R_POILEV**

Port Input Data Level Register는 SDK/Include/CANTUS/gpio.h에

```
#define R_POILEV ((volatile unsigned int*)0x80023410)
```

로 정의되어 있다.

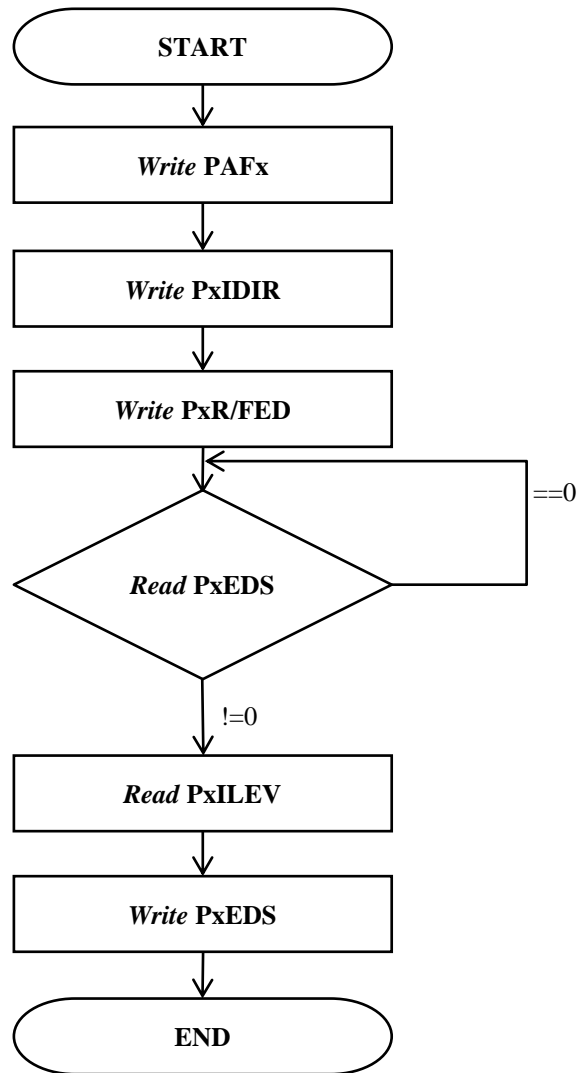
▶ 여기서는 Port Input Data Level Register 0를 ($1 \ll 7$)와 비교하여 Port 0의 7번 Pin의 입력 Level을 판단한다.

3 GPIO Edge Detect

3.1 Register Set

CANTUS의 GPIO를 Input Mode로 사용하면 Rising, Falling Edge Detection이 가능하다. 또한 Edge Detection의 결과로 Core에 GPIO Interrupt를 요청할 수 있다. 이를 위한 Register 설정은 다음과 같다.

그림 3-1 GPIO Edge Detect Register Set



PxRED (Port Rising Edge Detect Register)

해당 Port Pin의 Rising Edge를 Detection하기 위해 '1b'을 해당 bit(1)에 쓴다.

표 3-1 Port Rising Edge Detect Register (PxRED)

Bit	R/W	Description	Default Value
31 : 8	R	Reserved	-
7	R/W	Px.7RED : Px.7 Rising Edge Detect bit 0 : Disable 1 : Enable	0
6	R/W	Px.6RED : Px.6 Rising Edge Detect bit 0 : Disable 1 : Enable	0
5	R/W	Px.5RED : Px.5 Rising Edge Detect bit 0 : Disable 1 : Enable	0
4	R/W	Px.4RED : Px.4 Rising Edge Detect bit 0 : Disable 1 : Enable	0
3	R/W	Px.3RED : Px.3 Rising Edge Detect bit 0 : Disable 1 : Enable	0
2	R/W	Px.2RED : Px.2 Rising Edge Detect bit 0 : Disable 1 : Enable	0
1	R/W	Px.1RED : Px.1 Rising Edge Detect bit 0 : Disable 1 : Enable	0
0	R/W	Px.0RED : Px.0 Rising Edge Detect bit 0 : Disable 1 : Enable	0

PxFED (Port Falling Edge Detect Register)

해당 Port Pin의 Falling Edge를 Detection하기 위해 '1b'을 해당 bit(1)에 쓴다.

표 3-2 Port Falling Edge Detect Register (PxFED)

Bit	R/W	Description	Default Value
31 : 8	R	Reserved	-
7	R/W	Px.7FED : Px.7 Falling Edge Detect bit 0 : Disable 1 : Enable	0
6	R/W	Px.6FED : Px.6 Falling Edge Detect bit 0 : Disable 1 : Enable	0
5	R/W	Px.5FED : Px.5 Falling Edge Detect bit 0 : Disable 1 : Enable	0
4	R/W	Px.4FED : Px.4 Falling Edge Detect bit 0 : Disable 1 : Enable	0
3	R/W	Px.3FED : Px.3 Falling Edge Detect bit 0 : Disable 1 : Enable	0
2	R/W	Px.2FED : Px.2 Falling Edge Detect bit 0 : Disable 1 : Enable	0
1	R/W	Px.1FED : Px.1 Falling Edge Detect bit 0 : Disable 1 : Enable	0
0	R/W	Px.0FED : Px.0 Falling Edge Detect bit 0 : Disable 1 : Enable	0

- PxRED / PxFED를 모두 설정하면 Any Edge Mode로 동작한다. Any Edge Mode에서 Detection된 Type은 PxILEV를 읽어 판단 한다.

PxEDS (Port Edge Detect Status Register)

해당 Port Pin의 Edge Detection 여부를 읽는다. Edge Detection을 다시 시작(Clear) 하기 위해 '1b'를 해당 bit(1)에 쓴다.

표 3-3 Port Edge Detect Status Register (PxEDS)

<i>Bit</i>	<i>R/W</i>	<i>Description</i>	<i>Default Value</i>
31 : 8	R	Reserved	-
7	R/W	Px.7EDS : Px.7 Edge Detect Status bit	0
6	R/W	Px.6EDS : Px.6 Edge Detect Status bit	0
5	R/W	Px.5EDS : Px.5 Edge Detect Status bit	0
4	R/W	Px.4EDS : Px.4 Edge Detect Status bit	0
3	R/W	Px.3EDS : Px.3 Edge Detect Status bit	0
2	R/W	Px.2EDS : Px.2 Edge Detect Status bit	0
1	R/W	Px.1EDS : Px.1 Edge Detect Status bit	0
0	R/W	Px.0EDS : Px.0 Edge Detect Status bit	0

3.2 Function Set 1

아래는 CANTUS의 Port 0의 7번 Pin을 PAF0로 GPIO로 설정하고, P0IDIR로 Input Mode로 설정한 후 P0RED와 P0FED를 설정하여 Any Edge Mode로 설정 한다. 그 후 P0EDS를 읽어 Edge가 Detection되면 P0ILEV를 읽어 Input Level을 판단하고 string을 출력하며, Edge Detection을 다시 시작(Clear) 하기 위해 P0EDS에 (1<<7)를 쓰는 예이다.

```
main()
{
  ...

  *R_PAF0 = F_PAF0_7_GPIO;
  *R_P0IDIR = (1<<7);
  *R_P0RED = (1<<7);
  *R_P0FED = (1<<7);

  while(1)
  {
    if( (1<<7) == (*R_P0EDS & (1<<7)) )
    {
      if((1<<7) == (*R_P0ILEV & (1<<7)) )
      {
        debugstring("P0.7 Rising Edge Detected\r\n");
      }
      else
      {
        debugstring("P0.7 Falling Edge Detected\r\n");
      }
      *R_P0EDS = (1<<7);
    }
  }
  ...
}
```

***R_P0RED**

Port Rising Edge Detect Register는 SDK/Include/CANTUS/gpio.h에

```
#define R_P0RED ((volatile unsigned int*)0x80023420)
```

로 정의되어 있다.

▶ 여기서는 Port Rising Edge Detect Register 0에 (1<<7)를 대입하여 Port 0의 7번 Pin을 Rising Edge Detect를 Enable로 설정한다.

***R_P0FED**

Port Falling Edge Detect Register는 SDK/Include/CANTUS/gpio.h에

```
#define R_P0FED ((volatile unsigned int*)0x80023424)
```

로 정의되어 있다.

▶ 여기서는 Port Falling Edge Detect Register 0에 (1<<7)를 대입하여 Port 0의 7번 Pin을 Falling Edge Detect를 Enable로 설정한다.

***R_P0EDS**

Port Edge Detect Status Register는 SDK/Include/CANTUS/gpio.h에

```
#define R_P0EDS ((volatile unsigned int*)0x80023428)
```

로 정의되어 있다.

▶ 여기서는 Port Edge Detect Status Register 0를 읽어 Edge Detection된 Pin을 알 수 있다. 또한 Edge Detection을 다시 시작(Clear) 하기 위해 (1<<7)를 쓴다.

***R_P0ILEV**

Port Input Data Level Register는 SDK/Include/CANTUS/gpio.h에

```
#define R_P0ILEV ((volatile unsigned int*)0x80023410)
```

로 정의되어 있다.

▶ 여기서는 Port Input Data Level Register 0를 읽어 Edge Detection된 후의 Input Level을 알 수 있다. Rising Edge / Falling Edge가 모두 Enable되어 있는 경우 Detection Type의 판별은 Port Input Data Level Register를 읽어 판별 한다.

3.3 Function Set 2

아래는 CANTUS의 Port 0의 7번 Pin을 PAF0로 GPIO로 설정하고, P0IDIR로 Input Mode로 설정한 후 P0RED와 P0FED를 설정하여 Any Edge Mode로 설정 한다. GPIO0 Interrupt를 설정하여, Edge가 Detection되면 GPIO0_ISR()로 분기하여 P0ILEV를 읽어 Input Level을 판단하고 string을 출력하며, Edge Detection을 다시 시작(Clear) 하기 위해 P0EDS에 (1<<7)를 쓰는 예이다. Interrupt에 관련된 내용은 여기서는 다루지 않는다.

```
void GPIO0_ISR(void)
{
    if( (1<<7) == (*R_P0ILEV & (1<<7)) )
    {
        debugstring("P0.7 Rising Edge Detected\r\n");
    }
    else
    {
        debugstring("P0.7 Falling Edge Detected\r\n");
    }
    *R_P0EDS = (1<<7);
}
main()
{
    ...

    *R_PAF0 = F_PAF0_7_GPIO;
    *R_P0IDIR = (1<<7);
    *R_P0RED = (1<<7);
    *R_P0FED = (1<<7);

    setinterrupt(INTNUM_GPIO0,GPIO0_ISR);
    EnableInterrupt(INTNUM_GPIO0,TRUE);

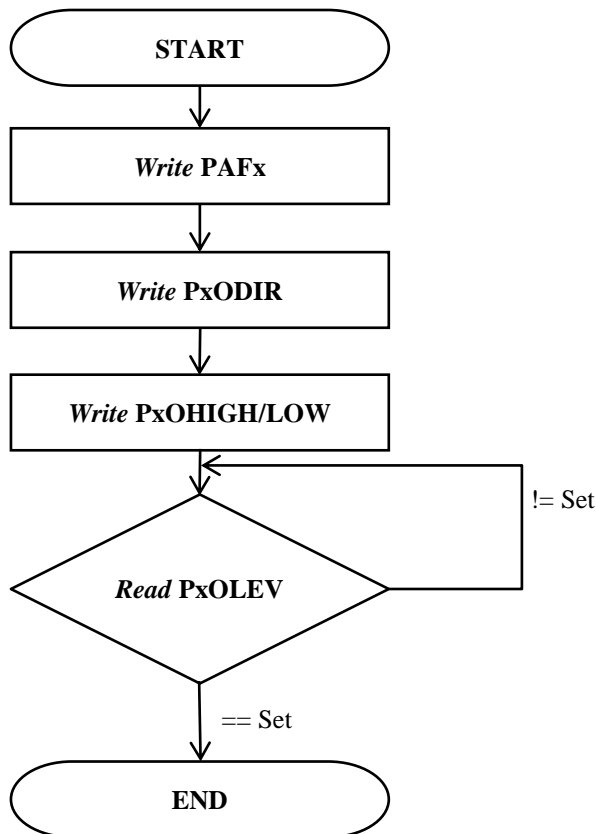
    while(1)
    {
        }
    ...
}
```


4 GPIO Output Mode

4.1 Register Set

CANTUS의 GPIO를 Output Mode로 사용하기 위해 다음과 같은 순서로 Register를 설정한다. Output Mode에서는 High/Low Level Output을 각각 설정 한다.

그림 4-1 GPIO Output Mode Register Set



Set : PxOHIGH/LOW에 쓴 값을 의미한다.

PxODIR (Port Direction Output Mode Setting Register)

해당 Port의 Pin을 Output Mode로 설정하기 위해 '1b'을 해당 bit(1)에 쓴다.

PxOHIGH (Port Output Data High Level Setting Register)

해당 Port의 Pin에 High Level을 출력하기 위해 '1b'을 해당 bit(1)에 쓴다.

4.2 Function Set 1

아래는 CANTUS의 Port 6의 4번 Pin을 PAF6로 GPIO로 설정하고, P6ODIR로 Output Mode로 설정한 후 P6OHIGH로 High Level을, P6OLOW로 Low Level을 출력하여 반복하는 예이다.

```
main()
{
  ...

  *R_PAF6 = F_PAF6_4_GPIO;
  *R_P6ODIR = (1<<4);

  while(1)
  {
    *R_P6OHIGH = (1<<4);
    while(!(*R_P6OLEV & (1<<4)));

    *R_P6OLOW = (1<<4);
    while(*R_P6OLEV & (1<<4));
  }
  ...
}
```

*R_PAF6

Port Alternate Function Register는 SDK/Include/CANTUS/paf.h에

```
#define R_PAF6 ((volatile U32*)0x80020038)
#define F_PAF6_4_GPIO      (3 << 8)
```

로 정의되어 있다.

▶ 여기서는 Port Alternate Function Register 6에 F_PAF6_4_GPIO을 대입하여 Port 6의 4번 Pin을 GPIO로 설정 한다.

*R_P6ODIR

Port Direction Output Mode Setting Register 는 SDK/Include/CANTUS/gpio.h에

```
#define R_P6ODIR ((volatile unsigned int*)0x80023600)
```

로 정의되어 있다.

▶ 여기서는 Port Direction Output Mode Setting Register 6에 (1<<4)를 대입하여 Port 6의 4번 Pin을 Output Mode로 설정 한다.

*R_P6OHIGH

Port Output Data High Level Setting Register는 SDK/Include/CANTUS/gpio.h에

```
#define R_P6OHIGH ((volatile unsigned int*)0x80023608)
```

로 정의되어 있다.

▶ 여기서는 Port Output Data High Level Setting Register 6에 (1<<4)를 대입하여 Port 6의 4번 Pin을 Output Data High Level로 설정하여 High Level을 출력 한다.

***R_P6OLOW**

Port Output Data Low Level Setting Register는 SDK/Include/CANTUS/gpio.h에

```
#define R_P6OLOW ((volatile unsigned int*)0x8002360c)
```

로 정의되어 있다.

▶ 여기서는 Port Output Data Low Level Setting Register 6에 (1<<4)를 대입하여 Port 6의 4번 Pin을 Output Data Low Level로 설정하여 Low Level을 출력 한다.

***R_P6OLEV**

Port Output Data Level Register는 SDK/Include/CANTUS/gpio.h에

```
#define R_P6OLEV ((volatile unsigned int*)0x80023608)
```

로 정의되어 있다.

▶ 여기서는 Port Output Data Level Register 6를 읽어 (1<<4)와 비교하여 Port 6의 4번 Pin의 출력 Level을 판단 한다.