마이크로프로세서응용

(GPIO)

↳ Global input , output

2023. 2학기

Kookmin Univ. EMCO Lab.

KMU 국민대학교
자동차공학 전문대학원

Motor Control Lab.
eMS

Contents

1. LED 회로

2. LED 동작 설정

3. LED 동작 1

4. LED 동작 2

5. SWITHC 동작

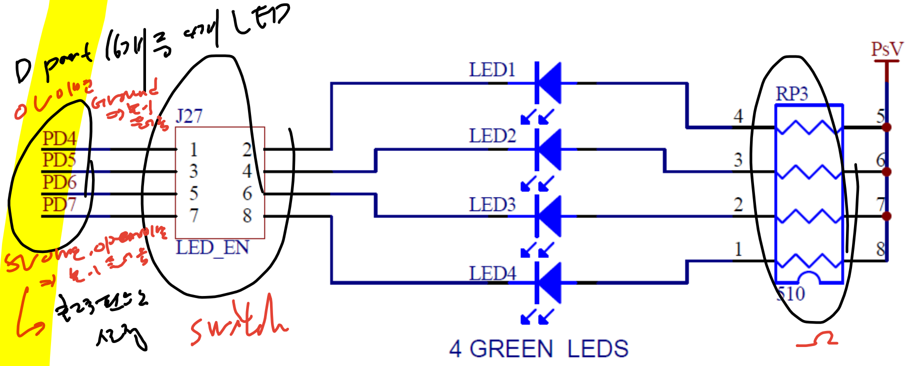
Evaluation Board

↳ 평가보드

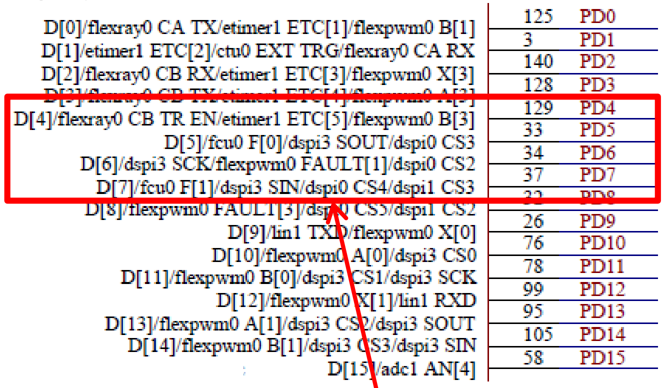
Switch Input
LED output

1. LED 회로

- LED light

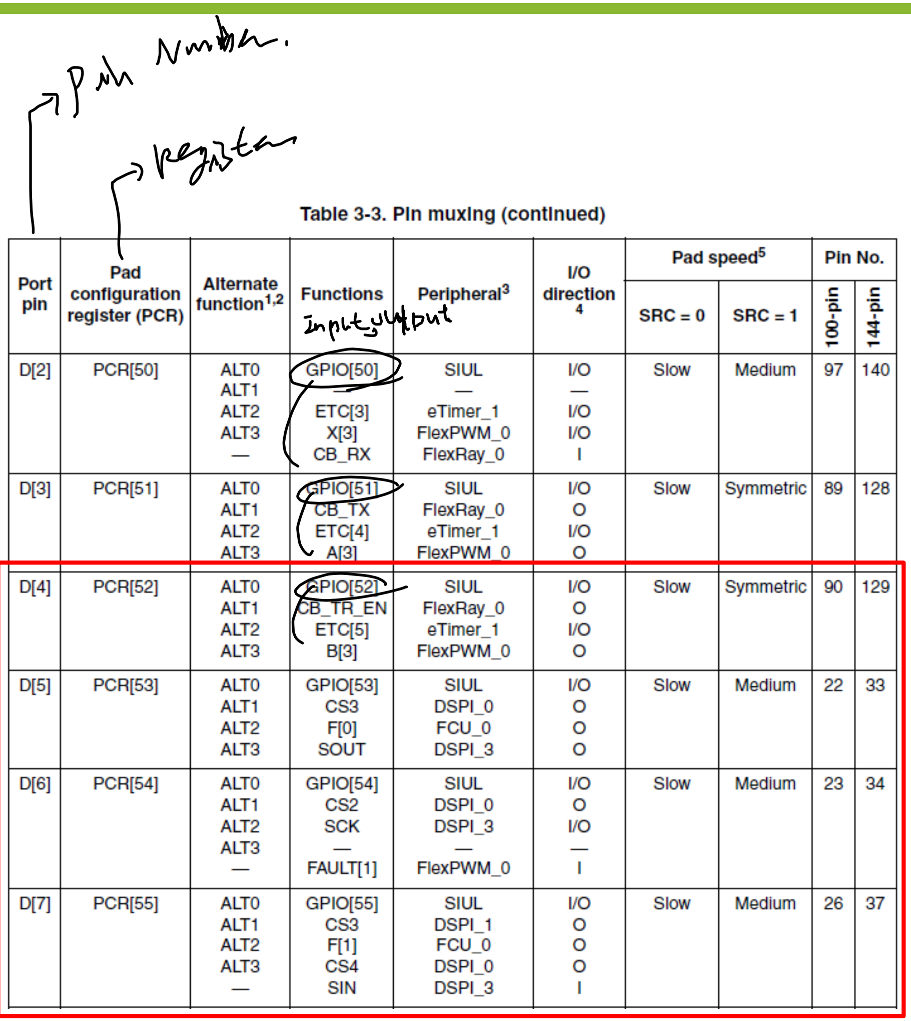


Port D



D[0]/flexray0 CA TX/etimer1 ETC[1]/flexpwm0 B[1]	125	PD0
D[1]/etimer1 ETC[2]/ctu0 EXT TRG/flexray0 CA RX	3	PD1
D[2]/flexray0 CB RX/etimer1 ETC[3]/flexpwm0 X[3]	140	PD2
D[3]/flexray0 CB TX/etimer1 ETC[4]/flexpwm0 A[3]	128	PD3
D[4]/flexray0 CB TR EN/etimer1 ETC[5]/flexpwm0 B[3]	129	PD4
D[5]/fcu0 F[0]/dspi3 SOUT/dspi0 CS3	33	PD5
D[6]/dspi3 SCK/flexpwm0 FAULT[1]/dspi0 CS2	34	PD6
D[7]/fcu0 F[1]/dspi3 SIN/dspi0 CS4/dspi1 CS3	37	PD7
D[8]/flexpwm0 FAULT[3]/dspi0 CS5/dspi1 CS2	32	PD8
D[9]/lin1 TXD/flexpwm0 X[0]	26	PD9
D[10]/flexpwm0 A[0]/dspi3 CS0	76	PD10
D[11]/flexpwm0 B[0]/dspi3 CS1/dspi3 SCK	78	PD11
D[12]/flexpwm0 X[1]/lin1 RXD	99	PD12
D[13]/flexpwm0 A[1]/dspi3 CS2/dspi3 SOUT	95	PD13
D[14]/flexpwm0 B[1]/dspi3 CS3/dspi3 SIN	105	PD14
D[15]/adc1 AN[4]	58	PD15

LED에 연결된 GPIO Pin



→ Pin Number.

→ register

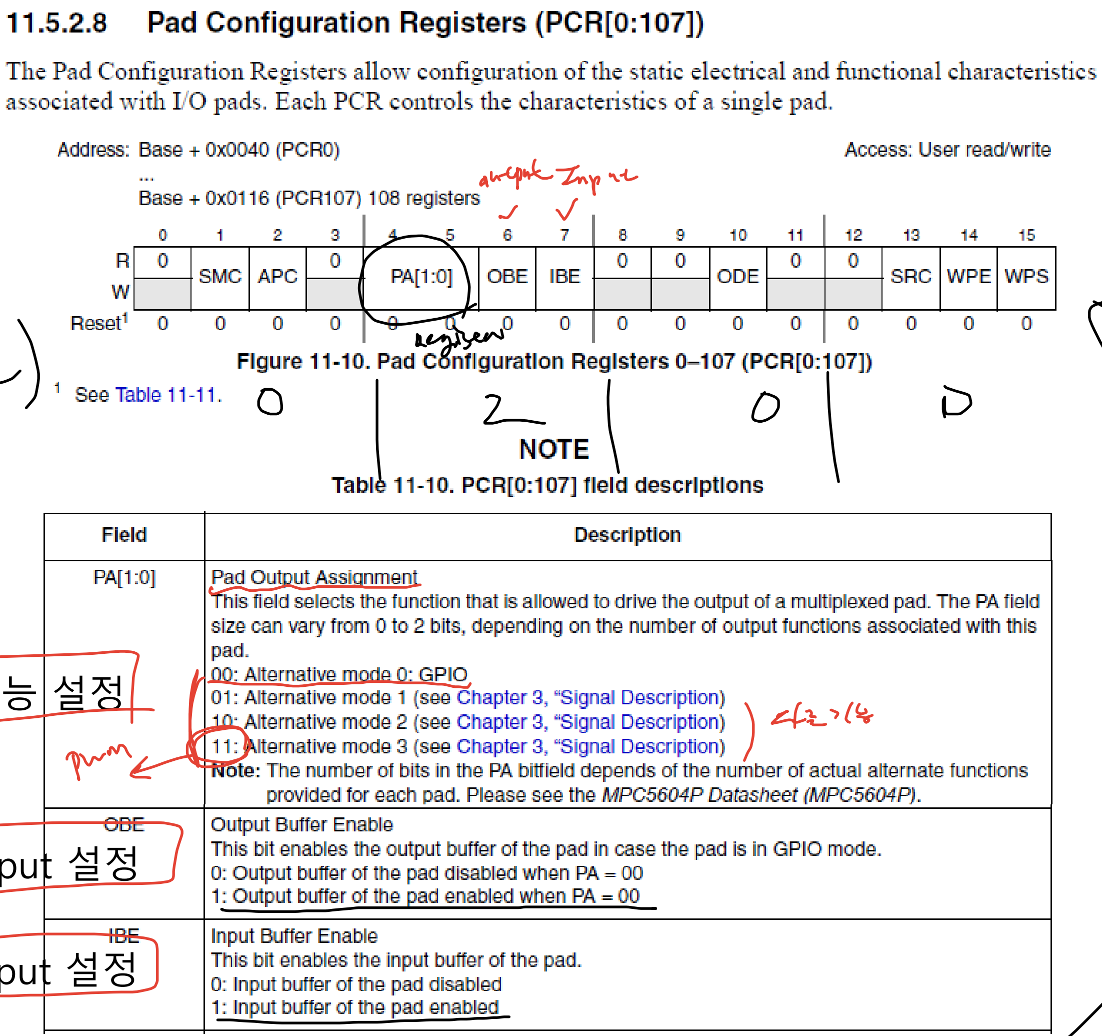
Table 3-3. Pin muxing (continued)

Port pin	Pad configuration register (PCR)	Alternate function[1,2]	Functions	Peripheral[3]	I/O direction[4]	Pad speed[5]		Pin No.	
			Input/Output			SRC = 0	SRC = 1	100-pin	144-pin
D[2]	PCR[50]	ALT0 ALT1 ALT2 ALT3 —	GPIO[50] — ETC[3] X[3] CB_RX	SIUL — eTimer_1 FlexPWM_0 FlexRay_0	I/O — I/O I/O I	Slow	Medium	97	140
D[3]	PCR[51]	ALT0 ALT1 ALT2 ALT3	GPIO[51] CB_TX ETC[4] A[3]	SIUL FlexRay_0 eTimer_1 FlexPWM_0	I/O O I/O O	Slow	Symmetric	89	128
D[4]	PCR[52]	ALT0 ALT1 ALT2 ALT3	GPIO[52] CB_TR_EN ETC[5] B[3]	SIUL FlexRay_0 eTimer_1 FlexPWM_0	I/O O I/O O	Slow	Symmetric	90	129
D[5]	PCR[53]	ALT0 ALT1 ALT2 ALT3	GPIO[53] CS3 F[0] SOUT	SIUL DSPI_0 FCU_0 DSPI_3	I/O O O O	Slow	Medium	22	33
D[6]	PCR[54]	ALT0 ALT1 ALT2 ALT3 —	GPIO[54] CS2 SCK — FAULT[1]	SIUL DSPI_0 DSPI_3 — FlexPWM_0	I/O O I/O — I	Slow	Medium	23	34
D[7]	PCR[55]	ALT0 ALT1 ALT2 ALT3 —	GPIO[55] CS3 F[1] CS4 SIN	SIUL DSPI_1 FCU_0 DSPI_0 DSPI_3	I/O O O O I	Slow	Medium	26	37

2. LED 동작 설정

- LED light

PCR을 설정하는 Register(16bit)



11.5.2.8 Pad Configuration Registers (PCR[0:107])

The Pad Configuration Registers allow configuration of the static electrical and functional characteristics associated with I/O pads. Each PCR controls the characteristics of a single pad.

Address: Base + 0x0040 (PCR0)
...
Base + 0x0116 (PCR107) 108 registers

Access: User read/write

output Input

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0	SMC	APC	0	PA[1:0]		OBE	IBE	0	0	ODE	0	0	SRC	WPE	WPS
W																
Reset[1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

register

Figure 11-10. Pad Configuration Registers 0–107 (PCR[0:107])

[1] See Table 11-11.

0 2 0 D

NOTE

Table 11-10. PCR[0:107] field descriptions

Field	Description
PA[1:0]	Pad Output Assignment This field selects the function that is allowed to drive the output of a multiplexed pad. The PA field size can vary from 0 to 2 bits, depending on the number of output functions associated with this pad. 00: Alternative mode 0: GPIO 01: Alternative mode 1 (see Chapter 3, "Signal Description) 10: Alternative mode 2 (see Chapter 3, "Signal Description) 11: Alternative mode 3 (see Chapter 3, "Signal Description) **Note:** The number of bits in the PA bitfield depends of the number of actual alternate functions provided for each pad. Please see the *MPC5604P Datasheet (MPC5604P)*.
OBE	Output Buffer Enable This bit enables the output buffer of the pad in case the pad is in GPIO mode. 0: Output buffer of the pad disabled when PA = 00 1: Output buffer of the pad enabled when PA = 00
IBE	Input Buffer Enable This bit enables the input buffer of the pad. 0: Input buffer of the pad disabled 1: Input buffer of the pad enabled

다른 기능

pwm

GPIO의 기능 설정

Output 설정

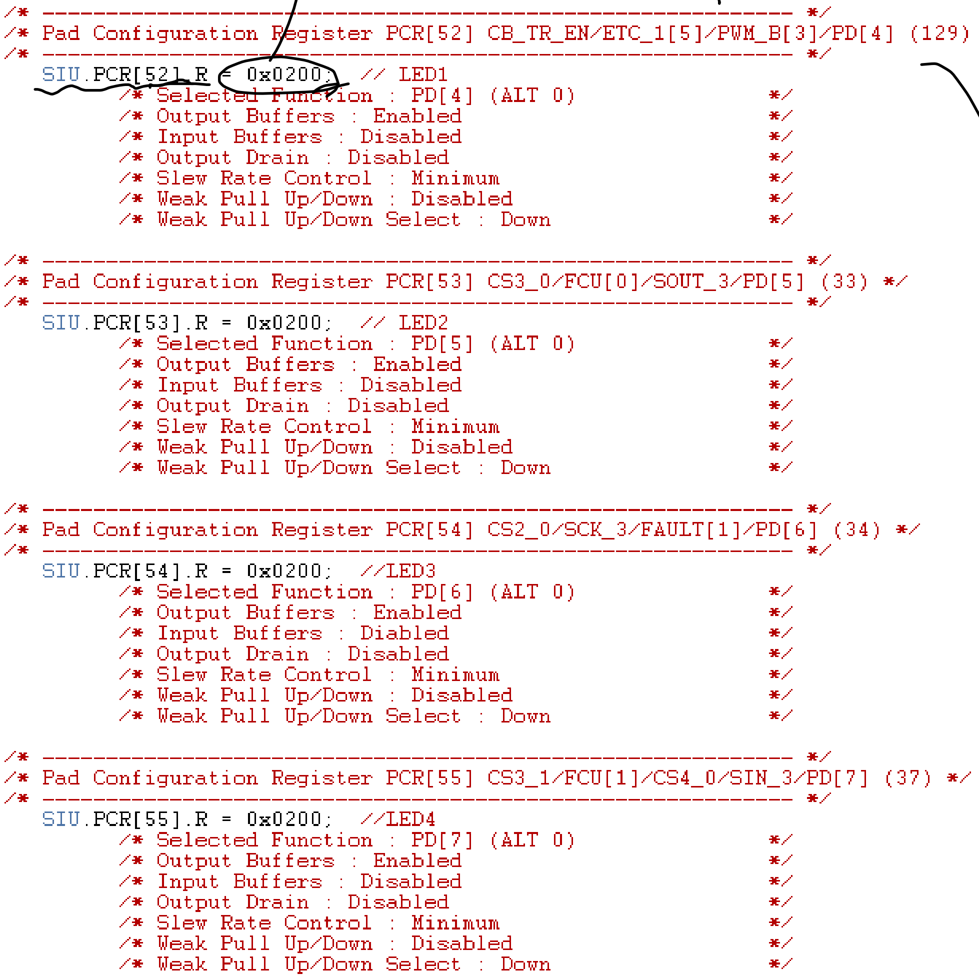
Input 설정

2. LED 동작 설정

- LED light
  - 기본 Pin Setting

GPIO => Input, output 기능

"OBE" Enable => output으로 사용



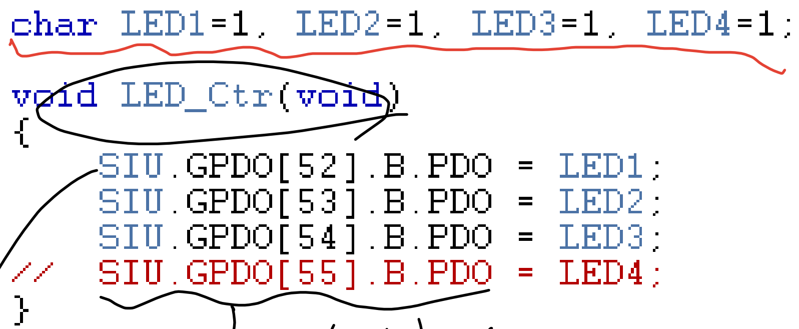
```
/* ------------------------------------------------------------ */
/* Pad Configuration Register PCR[52] CB_TR_EN/ETC_1[5]/PWM_B[3]/PD[4] (129)
/* ------------------------------------------------------------ */
   SIU.PCR[52].R = 0x0200;   // LED1
         /* Selected Function : PD[4] (ALT 0)                   */
         /* Output Buffers : Enabled                            */
         /* Input Buffers : Disabled                            */
         /* Output Drain : Disabled                             */
         /* Slew Rate Control : Minimum                         */
         /* Weak Pull Up/Down : Disabled                        */
         /* Weak Pull Up/Down Select : Down                     */

/* ------------------------------------------------------------ */
/* Pad Configuration Register PCR[53] CS3_0/FCU[0]/SOUT_3/PD[5] (33) */
/* ------------------------------------------------------------ */
   SIU.PCR[53].R = 0x0200;   // LED2
         /* Selected Function : PD[5] (ALT 0)                   */
         /* Output Buffers : Enabled                            */
         /* Input Buffers : Disabled                            */
         /* Output Drain : Disabled                             */
         /* Slew Rate Control : Minimum                         */
         /* Weak Pull Up/Down : Disabled                        */
         /* Weak Pull Up/Down Select : Down                     */

/* ------------------------------------------------------------ */
/* Pad Configuration Register PCR[54] CS2_0/SCK_3/FAULT[1]/PD[6] (34) */
/* ------------------------------------------------------------ */
   SIU.PCR[54].R = 0x0200;   //LED3
         /* Selected Function : PD[6] (ALT 0)                   */
         /* Output Buffers : Enabled                            */
         /* Input Buffers : Diabled                             */
         /* Output Drain : Disabled                             */
         /* Slew Rate Control : Minimum                         */
         /* Weak Pull Up/Down : Disabled                        */
         /* Weak Pull Up/Down Select : Down                     */

/* ------------------------------------------------------------ */
/* Pad Configuration Register PCR[55] CS3_1/FCU[1]/CS4_0/SIN_3/PD[7] (37) */
/* ------------------------------------------------------------ */
   SIU.PCR[55].R = 0x0200;   //LED4
         /* Selected Function : PD[7] (ALT 0)                   */
         /* Output Buffers : Enabled                            */
         /* Input Buffers : Disabled                            */
         /* Output Drain : Disabled                             */
         /* Slew Rate Control : Minimum                         */
         /* Weak Pull Up/Down : Disabled                        */
         /* Weak Pull Up/Down Select : Down                     */
```

4개 모두 output으로 설정

3. LED 동작1

- FreeMASTER를 이용하여 LED를 On/Off

3.1 함수



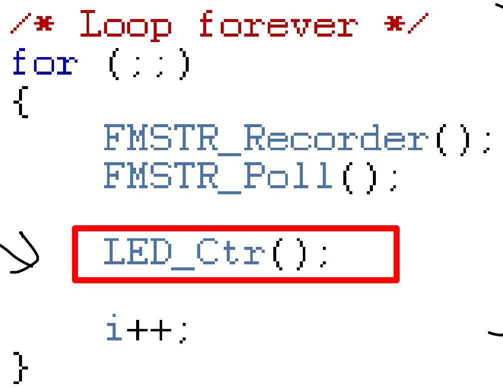
```
char LED1=1, LED2=1, LED3=1, LED4=1;
void LED_Ctr(void)
{
	SIU.GPDO[52].B.PDO = LED1;
	SIU.GPDO[53].B.PDO = LED2;
	SIU.GPDO[54].B.PDO = LED3;
//	SIU.GPDO[55].B.PDO = LED4;
}
```

- FreeMASTER에서 쓸 변수

- LED On/Off 함수

↳ output buffer

3.2 실행 위치(in main함수)



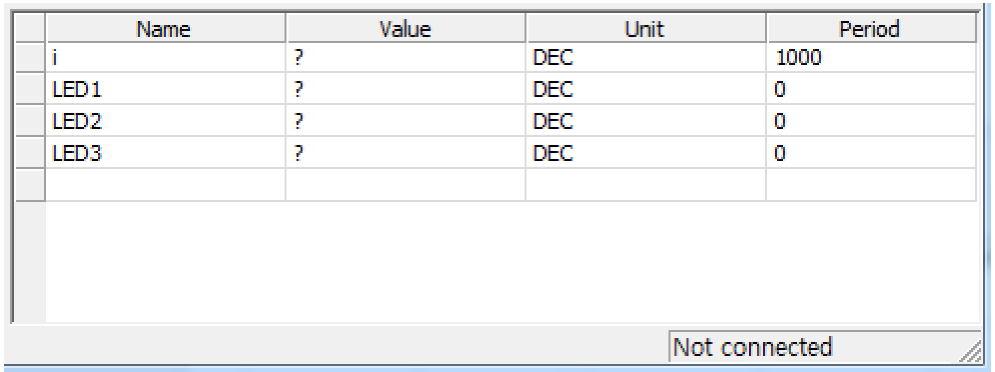
```
/* Loop forever */
for (;;)
{
	FMSTR_Recorder();
	FMSTR_Poll();

	LED_Ctr();

	i++;
}
```

⇒ 무한 loop

3. LED 동작1

- FreeMASTER를 이용하여 LED를 On/Off

3.5 FreeMASTER 변수 추가



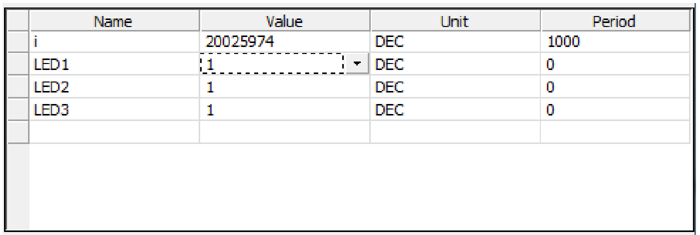
Name	Value	Unit	Period
i	?	DEC	1000
LED1	?	DEC	0
LED2	?	DEC	0
LED3	?	DEC	0

Not connected

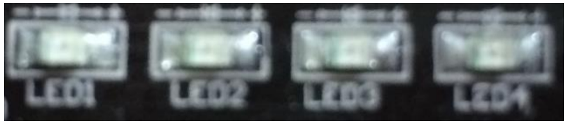
3. LED 동작1

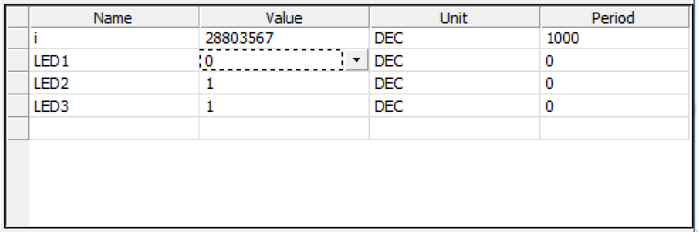
- FreeMASTER를 이용하여 LED를 On/Off

3.6 결과

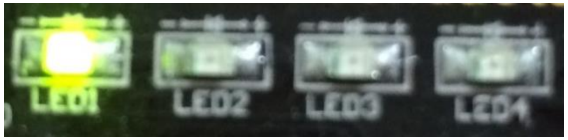


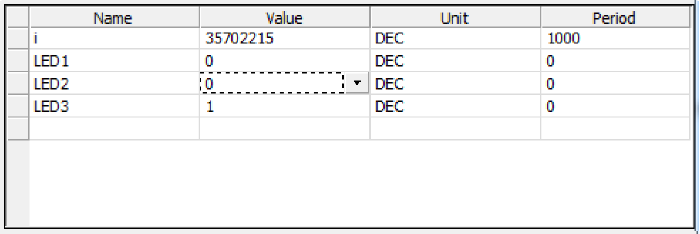
Name	Value	Unit	Period
i	20025974	DEC	1000
LED1	1	DEC	0
LED2	1	DEC	0
LED3	1	DEC	0



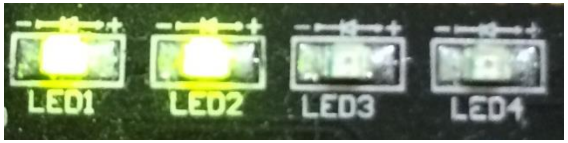


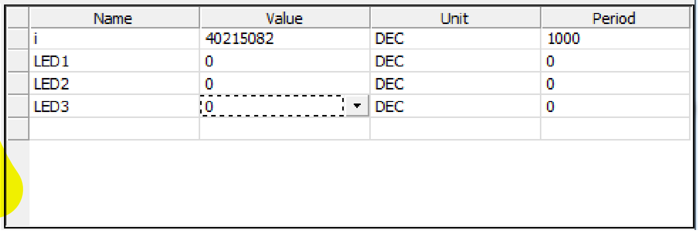
Name	Value	Unit	Period
i	28803567	DEC	1000
LED1	0	DEC	0
LED2	1	DEC	0
LED3	1	DEC	0



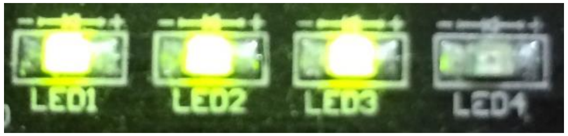


Name	Value	Unit	Period
i	35702215	DEC	1000
LED1	0	DEC	0
LED2	0	DEC	0
LED3	1	DEC	0





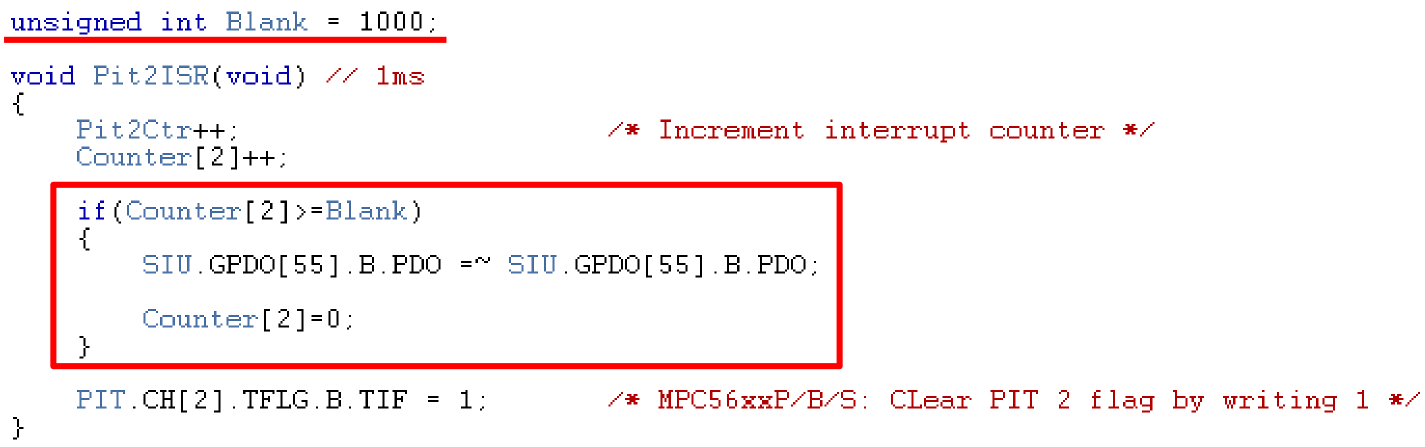
Name	Value	Unit	Period
i	40215082	DEC	1000
LED1	0	DEC	0
LED2	0	DEC	0
LED3	0	DEC	0



4. LED 동작2

- FreeMASTER를 이용하여 LED On/Off 주기 변화

4.1 함수



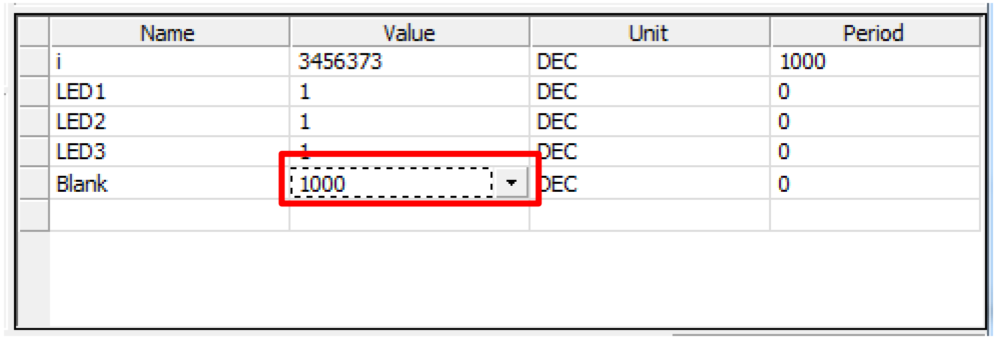
```
unsigned int Blank = 1000;

void Pit2ISR(void) // 1ms
{
    Pit2Ctr++;                        /* Increment interrupt counter */
    Counter[2]++;

    if(Counter[2]>=Blank)
    {
        SIU.GPDO[55].B.PDO =~ SIU.GPDO[55].B.PDO;

        Counter[2]=0;
    }

    PIT.CH[2].TFLG.B.TIF = 1;         /* MPC56xxP/B/S: CLear PIT 2 flag by writing 1 */
}
```

1ms PIT 함수 내부에 위와 같이 입력

4. LED 동작2

- FreeMASTER를 이용하여 LED On/Off 주기 변화

4.2 FreeMASTER 변수 추가



Name	Value	Unit	Period
i	3456373	DEC	1000
LED1	1	DEC	0
LED2	1	DEC	0
LED3	1	DEC	0
Blank	1000	DEC	0

Blank변수 값을 변경하여 LED4의 On/Off 주기를 조절
1ms 함수 안에 있으므로 1000번 카운트하면 1s가 된다.

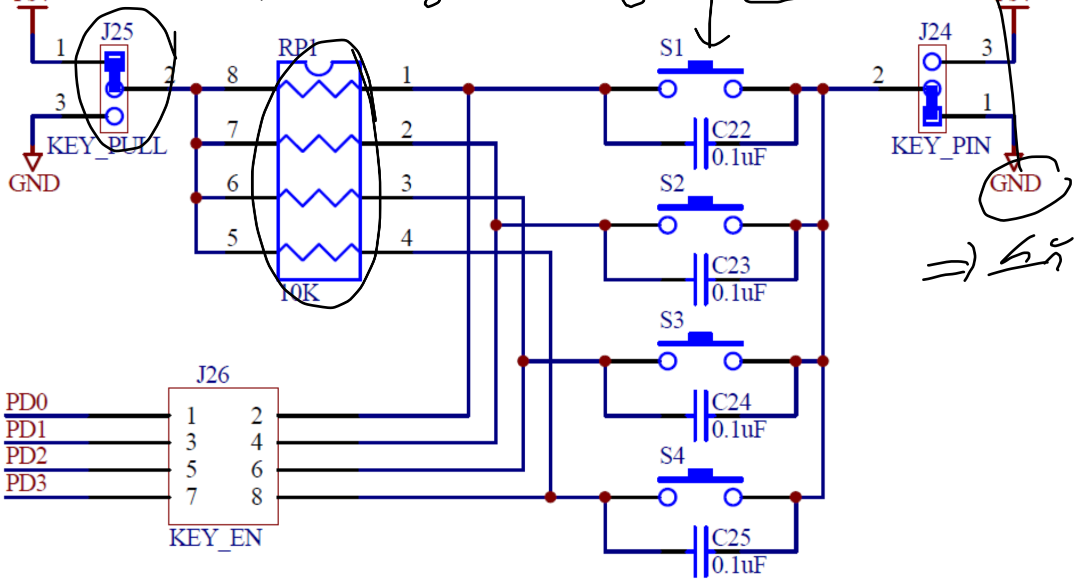
5. Switch 동작

- Switch 회로

안쪽에 A가는 10kΩ을 통해 5V걸림

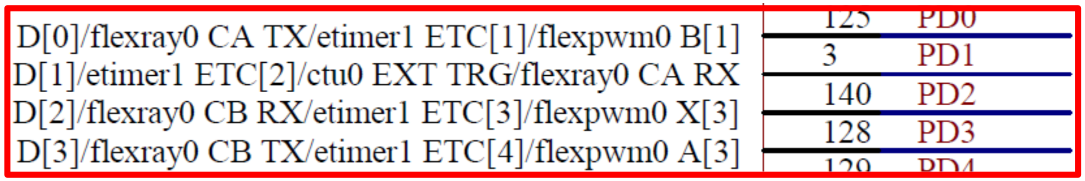
누르면 GND가 전송됨 → 0 (GND)

5V → 1



→ 스위치를 누르면 0
안누르면 1

MPU Port D



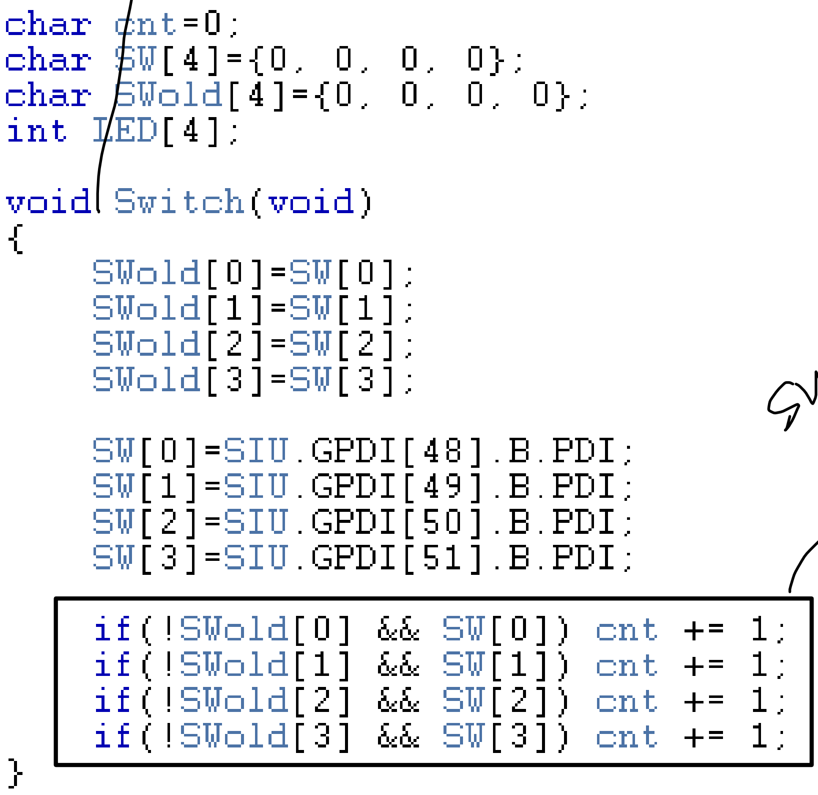
	125	PD0
D[0]/flexray0 CA TX/etimer1 ETC[1]/flexpwm0 B[1]	3	PD1
D[1]/etimer1 ETC[2]/ctu0 EXT TRG/flexray0 CA RX	140	PD2
D[2]/flexray0 CB RX/etimer1 ETC[3]/flexpwm0 X[3]	128	PD3
D[3]/flexray0 CB TX/etimer1 ETC[4]/flexpwm0 A[3]	129	PD4

Switch에 연결된 GPIO Pin

5. Switch 동작

- Switch를 이용하여 간단한 Code 작성

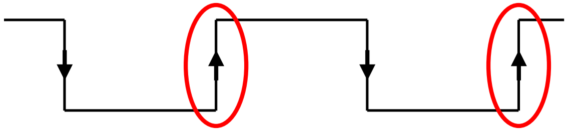
  - Switch Count Code



```
char cnt=0;
char SW[4]={0, 0, 0, 0};
char SWold[4]={0, 0, 0, 0};
int LED[4];

void Switch(void)
{
    SWold[0]=SW[0];
    SWold[1]=SW[1];
    SWold[2]=SW[2];
    SWold[3]=SW[3];

    SW[0]=SIU.GPDI[48].B.PDI;
    SW[1]=SIU.GPDI[49].B.PDI;
    SW[2]=SIU.GPDI[50].B.PDI;
    SW[3]=SIU.GPDI[51].B.PDI;

    if(!SWold[0] && SW[0]) cnt += 1;
    if(!SWold[1] && SW[1]) cnt += 1;
    if(!SWold[2] && SW[2]) cnt += 1;
    if(!SWold[3] && SW[3]) cnt += 1;
}
```

switch는 안 눌려 있다가 눌리면 cnt+1

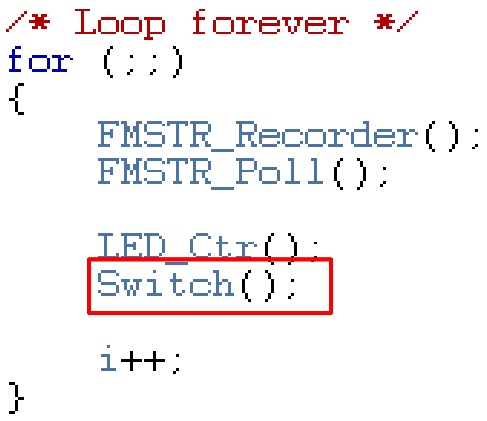
~~switch가 눌릴 때에~~



저장된 old 값과 새로운 값을
비교하여 rising 순간 Count

5. Switch 동작

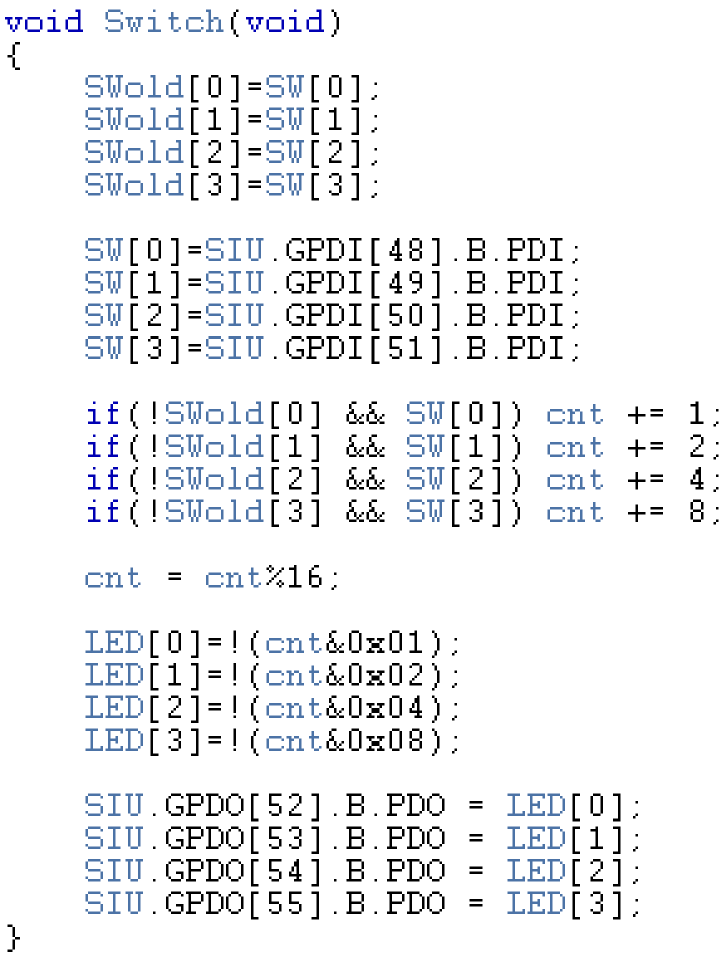
- Switch를 이용하여 간단한 Code 작성
  - Switch Count Code 실행 위치(in main함수)



```
/* Loop forever */
for (;;)
{
    FMSTR_Recorder();
    FMSTR_Poll();

    LED_Ctr();
    Switch();

    i++;
}
```

5. Switch 동작

- Switch를 이용하여 간단한 Code 작성
  - Switch를 이용하여 LED로 2진법 표현(최대 15까지)



```
void Switch(void)
{
    SWold[0]=SW[0];
    SWold[1]=SW[1];
    SWold[2]=SW[2];
    SWold[3]=SW[3];

    SW[0]=SIU.GPDI[48].B.PDI;
    SW[1]=SIU.GPDI[49].B.PDI;
    SW[2]=SIU.GPDI[50].B.PDI;
    SW[3]=SIU.GPDI[51].B.PDI;

    if(!SWold[0] && SW[0]) cnt += 1;
    if(!SWold[1] && SW[1]) cnt += 2;
    if(!SWold[2] && SW[2]) cnt += 4;
    if(!SWold[3] && SW[3]) cnt += 8;

    cnt = cnt%16;

    LED[0]=!(cnt&0x01);
    LED[1]=!(cnt&0x02);
    LED[2]=!(cnt&0x04);
    LED[3]=!(cnt&0x08);

    SIU.GPDO[52].B.PDO = LED[0];
    SIU.GPDO[53].B.PDO = LED[1];
    SIU.GPDO[54].B.PDO = LED[2];
    SIU.GPDO[55].B.PDO = LED[3];
}
```