마이크로프로세서응용
(ADC)

2023. 2학기

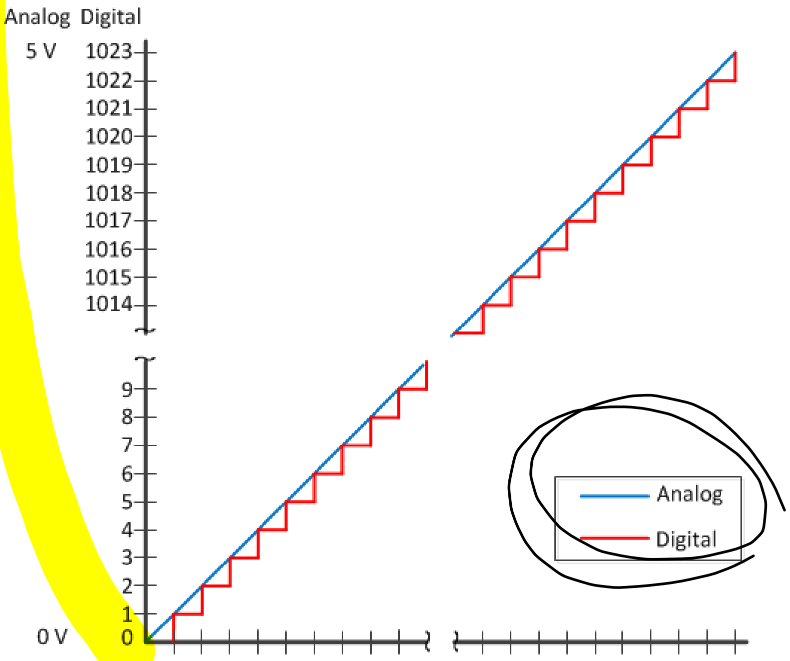
Kookmin Univ. EMCO Lab.

Contents

1. ADC

- Analog to Digital Converter
- 아날로그 (전압) 신호를 디지털 신호로 변환
- Resolution : 10bit ($2^{10} = 1024, 0 \sim 1023$)
- 2개의 module (2 x 11개 + 4채널 공유)

ADC0 , ADC1



< 전압 값에 따른 디지털 값 >

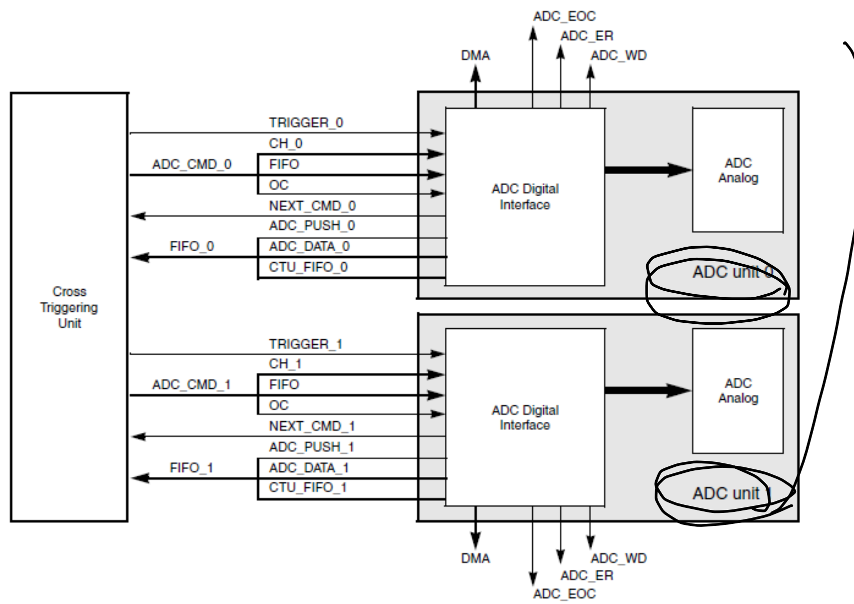


Figure 24-1. ADC Implementation

< ADC Block Diagram >

- ADC 초기화 코드

MCR

CTR

manual set

NCMR

CDR

MCR

```
void init_ADC1(void)
{
    ADC_1.MCR.B.ABORT = 1;          // Abort ADC_1

    ADC_1.MCR.B.OWREN = 0;          // disable overwritting
    ADC_1.MCR.B.WLSIDE = 0;         // conversion data is written right_aligned
    ADC_1.MCR.B.MODE = 0;           // One Shot mode
    ADC_1.MCR.B.CTUEN = 0;          // disable CTU triggered
    ADC_1.MCR.B.ADCLKSEL = 0;       // Set ADClock 32MHz
//  ADC_1.MCR.B.ADCLKSEL = 1;       // Set ADClock 64MHz
    ADC_1.MCR.B.ACKO = 0;           // disable auto clock off
    ADC_1.MCR.B.PWDN = 0;           // disable power down mode

    ADC_1.CTR[0].R = 0x00008208;
    //  Phase duration Latch(INPLATCH)       : Enabled(Always)    1 clock Cycle
    //  Input Sampling Duration(INPSAMP)     : 8 (INPSAMP >= 8)   7 clock Cycles
    //  Input Comparison Duration(INPCMP)    : 0b01               12 clock Cycles
    //  Conversion Time                      : 7 + 12 + 1(Tck)    20 clock Cycles

    ADC_1.NCMR[0].R = 0x00000020;   // Select ANS5 inputs for conversion

    ADC_1.CDR[5].R = 0x00000000;    // Channel[5] Set default

    ADC_1.MCR.B.ABORT = 0;          // Exit Abort state
}
```

ADC_1 → ch 5 set

→ ABORT을 풀어서 동작

2. ADC 초기화

- Main 함수 시작시 초기화
  - ADC 모듈1에 대한 초기화 선언

```
int main(void)
{
    initModesAndClock();
    disableWatchdog();
    enableIrq();
    initOutputClock();
    FMSTR_Init();
    init_INTC();
    init_Linflex0();
    init_ADC1();
```

2. ADC 초기화

- ADC 초기화 코드

ADC 모듈에 대한 기본적인 설정

MCR

```
void init_ADC1(void)
{
    ADC_1.MCR.B.ABORT = 1;          // Abort ADC_1

    ADC_1.MCR.B.OWREN = 0;          // disable overwritting
    ADC_1.MCR.B.WLSIDE = 0;         // conversion data is written right_aligned
    ADC_1.MCR.B.MODE = 0;           // One Shot mode
    ADC_1.MCR.B.CTUEN = 0;          // disable CTU triggered
    ADC_1.MCR.B.ADCLKSEL = 0;       // Set ADClock 32MHz
//  ADC_1.MCR.B.ADCLKSEL = 1;       // Set ADClock 64MHz
    ADC_1.MCR.B.ACK0 = 0;           // disable auto clock off
    ADC_1.MCR.B.PWDN = 0;           // disable power down mode
```

24.4.2.1 Main Configuration Register (MCR)

The Main Configuration Register (MCR) provides configuration settings for the ADC.

Address: Base + 0x0000 Access: User read/write

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	OWREN	WLSIDE	MODE	0	0	0	0	NSTART	0	JTRGEN	JEDGE	JSTART	0	0	CTUEN	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0	0	0	0	0	0	0	ADCLK SEL	ABORT CHAIN	ABORT	ACKO	0	0	0	0	PWDN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 24-8. Main Configuration Register (MCR)

- ADC 초기화 코드

ADC 모듈에 대한 기본적인 설정

24.4.2.1 Main Configuration Register (MCR)

The Main Configuration Register (MCR) provides configuration settings for the ADC.

Address: Base + 0x0000 Access: User read/write

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	OWREN	WLSIDE	MODE	0	0	0	0	NSTART	0	JTRGEN	JEDGE	JSTART	0	0	CTUEN	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0	0	0	0	0	0	0	ADCLK SEL	ABORT CHAIN	ABORT	ACKO	0	0	0	0	PWDN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 24-8. Main Configuration Register (MCR)

Table 24-12. MCR field descriptions

Field	Description
OWREN	Overwrite enable This bit enables or disables the functionality to overwrite unread converted data. 0 Prevents overwrite of unread converted data; new result is discarded 1 Enables converted data to be overwritten by a new conversion
WLSIDE	Write left/right-aligned 0 The conversion data is written right-aligned. 1 Data is left-aligned (from 15 to (15 – resolution + 1)). The WLSIDE bit affects all the CDR registers simultaneously. See Figure 24-23 and Figure 24-23.
MODE	One Shot/Scan 0 One Shot Mode—Configures the normal conversion of one chain. 1 Scan Mode—Configures continuous chain conversion mode; when the programmed chain conversion is finished it restarts immediately.

Field	Description
JTRGEN	Injection external trigger enable 0 External trigger disabled for channel injection 1 External trigger enabled for channel injection
JEDGE	Injection trigger edge selection Edge selection for external trigger, if JTRGEN = 1. 0 Selects falling edge for the external trigger 1 Selects rising edge for the external trigger
JSTART	Injection start Setting this bit will start the configured injected analog channels to be converted by software. Resetting this bit has no effect, as the injected chain conversion cannot be interrupted.
CTUEN	Cross trigger unit conversion enable 0 CTU triggered conversion disabled 1 CTU triggered conversion enabled
ADCLKSEL	Analog clock select This bit can only be written when ADC in Power-Down mode 0 ADC clock frequency is half Peripheral Set Clock frequency 1 ADC clock frequency is equal to Peripheral Set Clock frequency
ABORTCHAIN	Abort Chain When this bit is set, the ongoing Chain Conversion is aborted. This bit is reset by hardware as soon as a new conversion is requested. 0 Conversion is not affected 1 Aborts the ongoing chain conversion
ABORT	Abort Conversion When this bit is set, the ongoing conversion is aborted and a new conversion is invoked. This bit is reset by hardware as soon as a new conversion is invoked. If it is set during a scan chain, only the ongoing conversion is aborted and the next conversion is performed as planned. 0 Conversion is not affected 1 Aborts the ongoing conversion
ACKO	Auto-clock-off enable If set, this bit enables the Auto clock off feature. 0 Auto clock off disabled 1 Auto clock off enabled
PWDN	Power-down enable When this bit is set, the analog module is requested to enter Power Down mode. When ADC status is PWDN, resetting this bit starts ADC transition to IDLE mode. 0 ADC is in normal mode 1 ADC has been requested to power down

2. ADC 초기화

- ADC 초기화 코드

ADC Sampling 수에 따른 변환 시간 설정

CTR

```
ADC_1.CTR[0].R = 0x00008208;
//  Phase duration Latch(INPLATCH)        : Enabled(Always)    1 clock Cycle
//  Input Sampling Duration(INPSAMP)      : 8 (INPSAMP >= 8)   7 clock Cycles
//  Input Comparison Duration(INPCMP)     : 0b01               12 clock Cycles
//  Conversion Time                       : 7 + 12 + 1(Tck)    20 clock Cycles
```

24.4.6 Conversion timing registers CTR[0]

CTR0 = associated to internal precision channels (from 0 to 15)

Address: Base + 0x0094 (CTR0)   Access: User read/write

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	INPLATCH	0	OFFSHIFT		0	INPCMP		0	INPSAMP							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1

8 2 6 8

Figure 24-19. Conversion timing registers CTR[0]

Field	Description
INPLATCH	Configuration bit for latching phase duration
OFFSHIFT	Configuration for offset shift characteristic 00 No shift (that is the transition between codes 000h and 001h) is reached when the A_{VIN} (analog input voltage) is equal to 1 LSB. 01 Transition between code 000h and 001h is reached when the A_{VIN} is equal to1/2 LSB 10 Transition between code 00h and 001h is reached when the A_{VIN} is equal to 0 11 Not used **Note:** Available only on CTR0
INPCMP	Configuration bits for comparison phase duration
INPSAMP	Configuration bits for sampling phase duration

2. ADC 초기화

- 변환시간 계산 방법

Address: Base + 0x0094 (CTR0) Access: User read/write

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	INPLATCH	0	OFFSHIFT		0	INPCMP		0	INPSAMP							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1

Figure 24-19. Conversion timing registers CTR[0]

- ADC_0

$$T_{sample} = (INPSAMP - ndelay) \times T_{ck}$$

Always : $INPSAMP \geq 3$

- $INPSAMP \leq 6 : ndelay = 0.5$
- $INPSAMP > 6 : ndelay = 1$

$$T_{eval} = 10 \times T_{biteval} = 10 \times INPCMP \times T_{ck}$$

Always : $INPCMP \geq 1$ and $INPLATCH < INPCMP$

$$T_{conv} = T_{sample} + T_{eval} + (ndelay \times T_{ck})$$

- ADC_1

$$T_{sample} = (INPSAMP - 1) \times T_{ck}$$

Always : $INPSAMP \geq 8$: $ndelay = 1$

$$T_{eval} = 12 \times T_{biteval} = 12 \times INPCMP \times T_{ck} \quad (INPCMP \geq 1)$$
$$= 12 \times 4 \times T_{ck} \quad (INPCMP = 0)$$

$$T_{conv} = T_{sample} + T_{eval} + T_{ck}$$

- ADC 초기화 코드

ADC Channel 사용 설정

NCMR

```
ADC_1.NCMR[0].R = 0x00000020;    // Select ANS5 inputs for conversion
```

24.4.7.2 Normal Conversion Mask Registers (NCMR[0])

NCMR0 = Enable bits of normal sampling for channel 0 to 15 (precision channels)

Address: Base + 0x00A4 Access: User read/write

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R/W	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 24-20. Normal Conversion Mask Register 0 (NCMR0)

CH5 사용

2. ADC 초기화

- ADC 초기화 코드 (void init_ADC0(void)-CDR)

ADC 데이터 저장 변수 초기화

CDR

```
ADC_1.CDR[5].R = 0x00000000;     // Channel[5] Set default
```

24.4.9.2 Channel Data Register (CDR[0..15])

CDR[0..15] = precision channels

Each data register also gives information regarding the corresponding result as described below.

Address: See Table 24-10 Access: User read/write

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0	0	0	0	0	0	0	0	0	0	0	0	VALID	OVERW	RESULT	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0	0	0	0	0	0	CDATA[0:9] (MCR[WLSIDE] = 0)									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- ADC Read 변환 함수

```
int R_adc=0;

void ADCRead_1(void)
{
    ADC_1.MCR.B.NSTART = 1; // Module 1 Conversion Start
    asm("nop");

    while(ADC_1.MCR.B.NSTART) asm("nop");
    R_adc= ADC_1.CDR[5].B.CDATA;
}
```

Assembly어 ⇒ 시작 전까지 기다려라

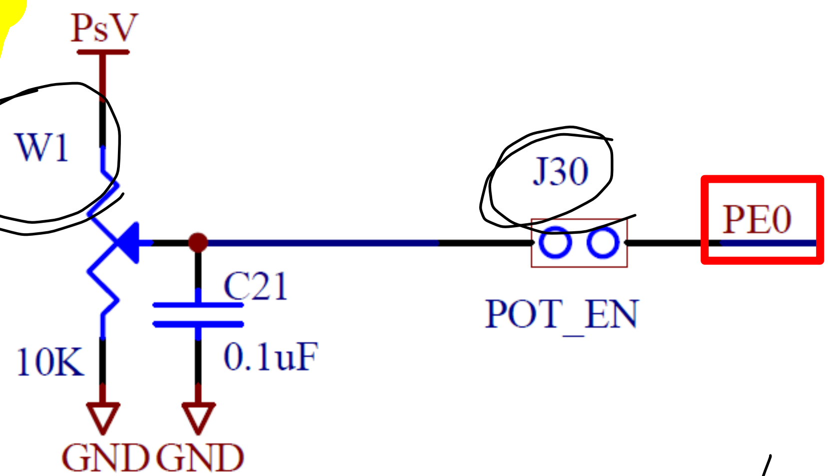
→ 변환시 필요한 Data를 input

A→D 1 : ~~완료~~ 변환중 , A→D 0 : 변환완료

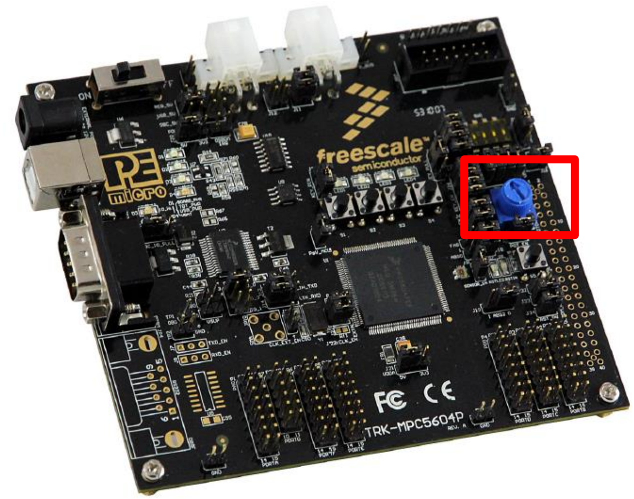
- ADC Read 변환 함수 실행 위치

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

    SW_Func();
    ADCRead_1();

    i++;
}
}
```

4. 가변저항회로

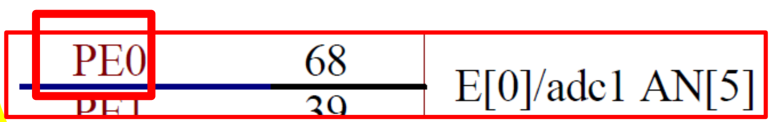


< picture3. 가변저항 >

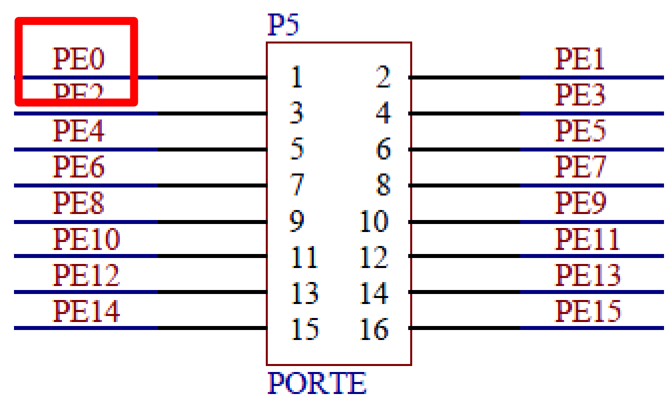


< picture4. MPC5604p >

MPU Port E



< picture5. 가변저항에 연결된 MCU Pin >



< picture6. MCU pin 과 연결된 pad >

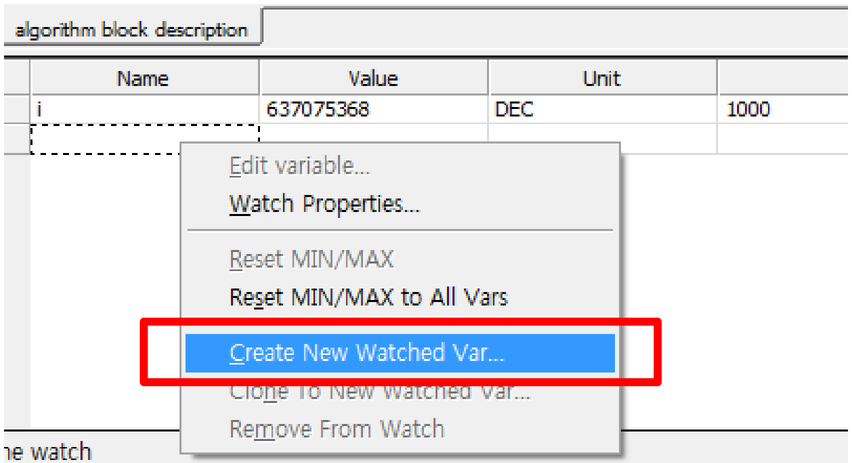
- 사용할 핀 설정

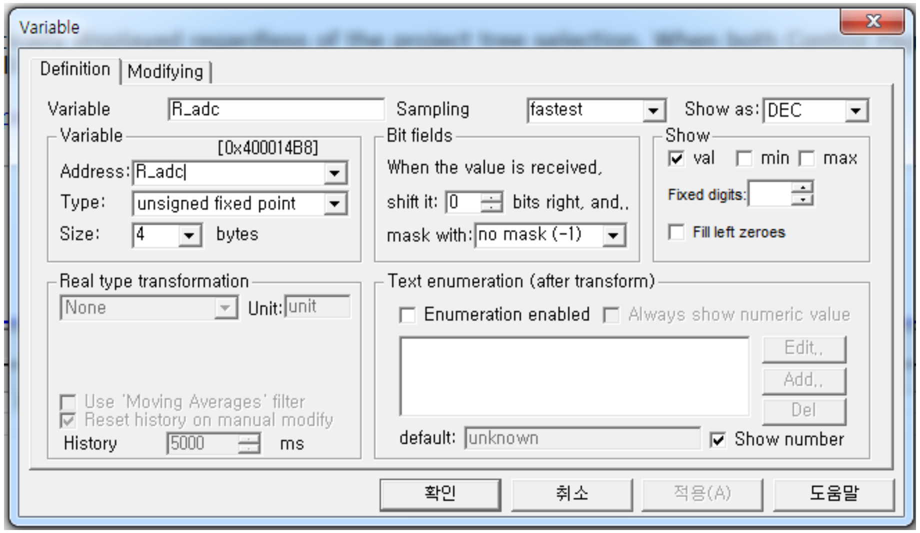
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0	SMC	APC	0	PA[1:0]		OBE	IBE	0	0	ODE	0	0	SRC	WPE	WPS
W																
Reset[1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

```
void siu_portE_init(void)
{
    /* ---------------------------------------------------------------- */
    /* Pad Configuration Register PCR[64] AN_1[5]/PE[0]_I (68) */
    /* ---------------------------------------------------------------- */
    SIU.PCR[64].R = 0x2400;
        /* Selected Function : ADC1 AN[5] (ALT n)1          */
        /* Input Buffers : Disabled                          */
        /* Safe Mode Control: Disabled                       */
        /* Analog Pad Switch : Enabled                       */
```

Port pin	Pad configuration register (PCR)	Alternate function[1,2]	Functions	Peripheral[3]	I/O direction[4]	Pad speed[5]		Pin No.	
						SRC = 0	SRC = 1	100-pin	144-pin
E[0]	PCR[64]	ALT0 ALT1 ALT2 ALT3 —	GPIO[64] — — — AN[5]	SIUL — — — ADC_1	Input only	—	—	46	68

- Freemaster에서 변수 추가





- 결과

Name	Value	Unit	Period
i	10541716	DEC	1000
LED1	1	DEC	0
LED2	1	DEC	0
LED3	1	DEC	0
Blank	1000	DEC	0
R_adc	1023	DEC	0

Name	Value	Unit	Period
i	18780393	DEC	1000
LED1	1	DEC	0
LED2	1	DEC	0
LED3	1	DEC	0
Blank	1000	DEC	0
R_adc	498	DEC	0

Name	Value	Unit	Period
i	23886281	DEC	1000
LED1	1	DEC	0
LED2	1	DEC	0
LED3	1	DEC	0
Blank	1000	DEC	0
R_adc	0	DEC	0

1023

2048 1024 512 256 128 64 32 16 8 4 2 1

1023 >> 6

768 816

2048 1024 512 256 128 64 32 16 8 4 2 1

0 1 0 0

768

132

5. 실습

LED_DZS = Read() >> 6;

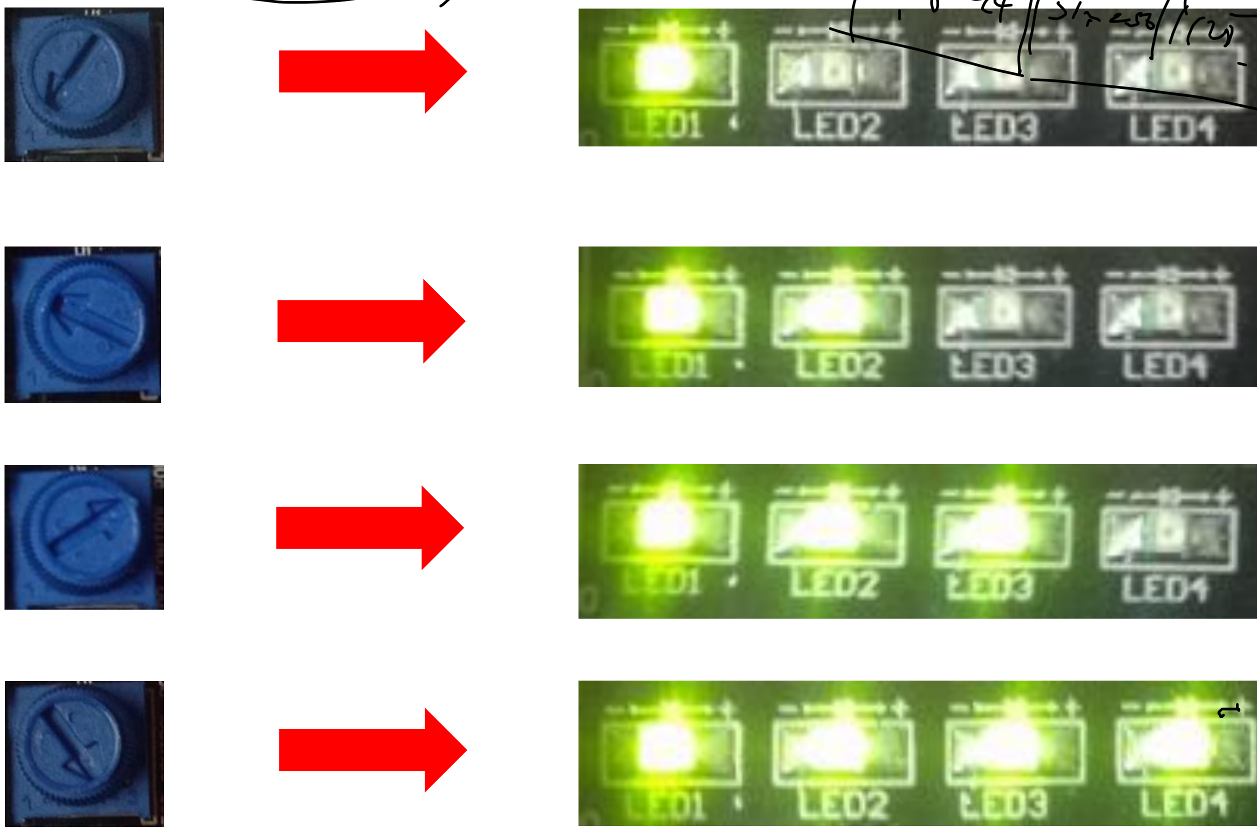
LED3 LED2 LED1 LED0

ADC >>

LED3 = LED_DZS & 0x08
LED2 = LED_DZS & 0x04
LED1 = LED_DZS & 0x02
LED0 = LED_DZS & 0x01

- 실습 1: 가변저항을 변화시킴에 따라 LED4개를 차례로 키고 끌 수 있도록 코드를 작성하시오.

1024 -> 16



1010

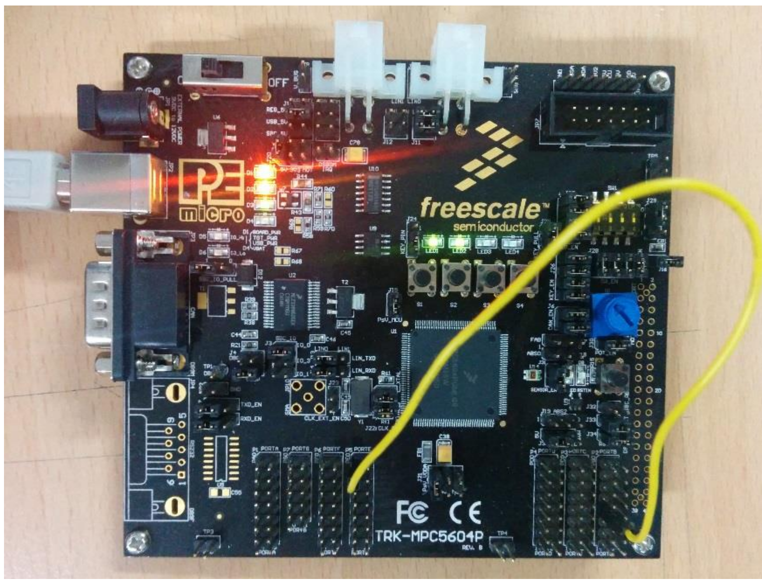
10 0x000A

< 가변 저항 변화에 따른 LED의 변화 >

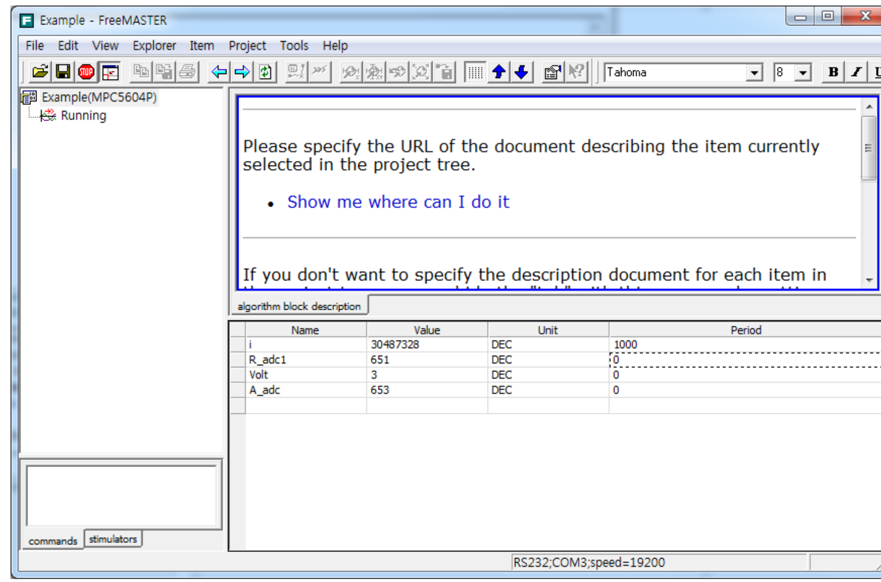
5. 실습

ADC

- 실습 2: 가변저항 pad를 이용하여 여분의 ADC pad를 연결하고 값을 0~5V로 scale 한 값을 나타내시오. (여분의 ADC는 pad B 8을 이용하고 ADC.0를 이용해 값을 받으시오.)



< 가변저항 pad 와 pab B[8] 연결>



< ADC0를 이용한 전압 측정 >