
마이크로컨트롤러 기능

제 8장 TC를 이용한 외부 입력 받기



ICAT

Integrated Circuits for Advanced Technology Lab.

TC를 이용한 외부 입력 받기

1. 외부 입력 카운터
2. 외부 입력 캡처



ICAT

Integrated Circuits for Advanced Technology Lab.

실습 11 : 외부 입력 카운터

□ 실습 개요

- ❖ 타이머/카운터1을 통해서 외부 입력 카운터를 계수하는 실습
- ❖ 외부 클럭 입력 포트를 사용하여 스위치를 누를 횟수를 계수한다

□ 실습 목표

- ❖ 타이머1의 외부 입력 카운터 동작원리 이해
- ❖ 16비트 타이머/카운터 관련 레지스터 이해



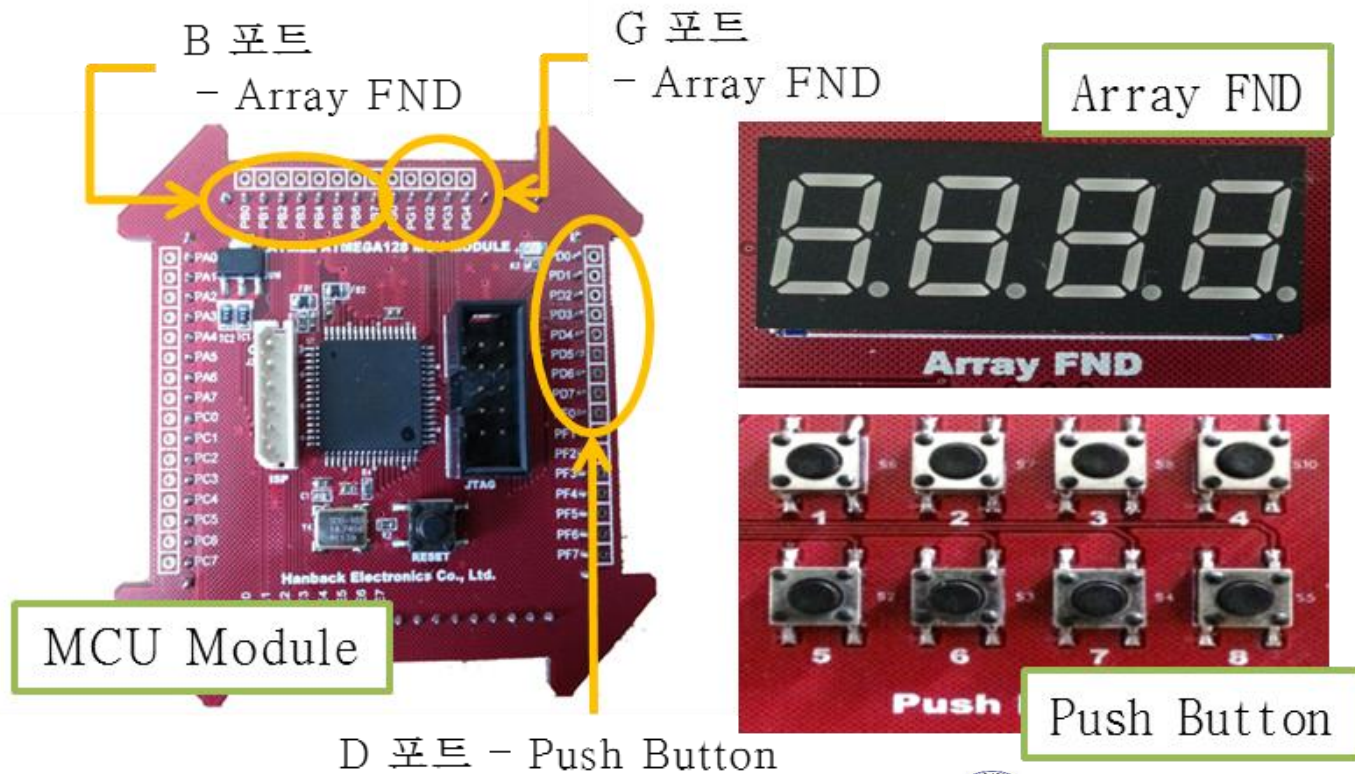
ICAT

Integrated Circuits for Advanced Technology Lab.

실습 11 : 외부 입력 카운터

□ 사용 모듈

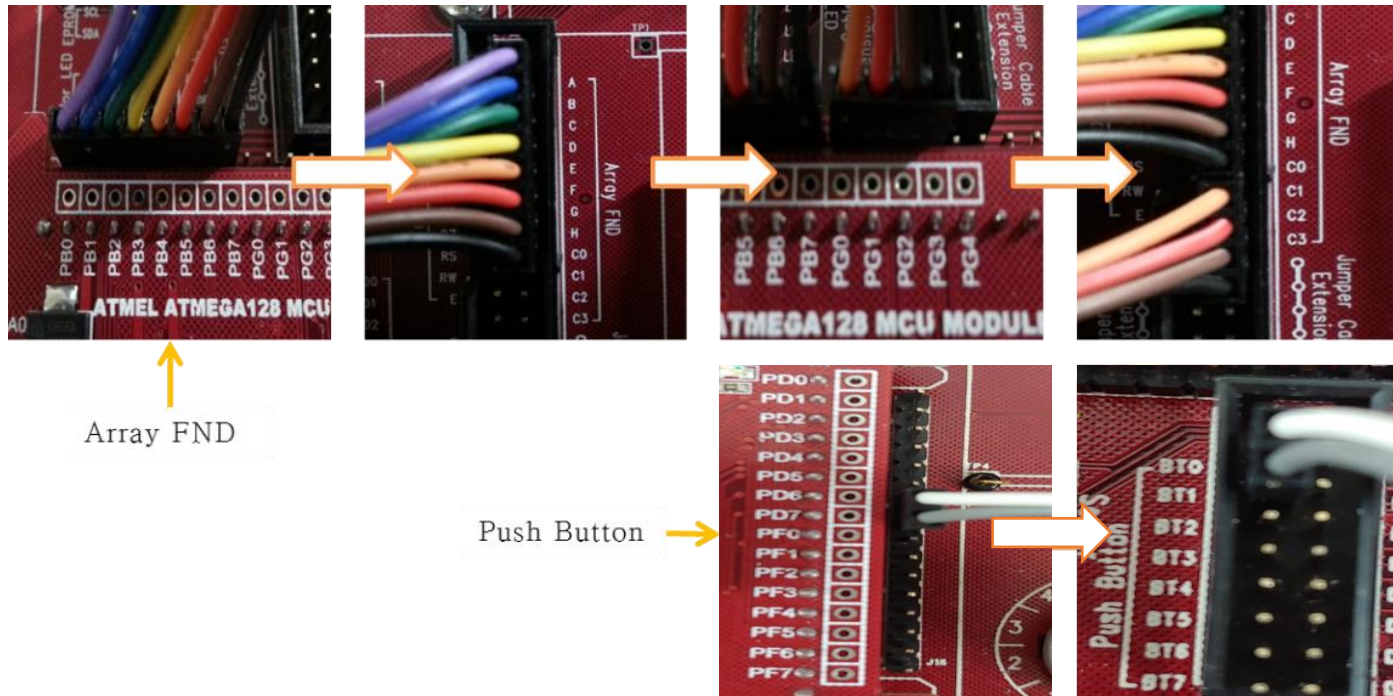
- ❖ MCU 모듈, Array FND 모듈, Push Button 모듈



실습 11 : 외부 입력 카운터

□ 모듈 결선 방법

- ❖ MCU 모듈 포트B의 PB0~PB7을 Array FND의 A~H까지 연결
- ❖ MCU 모듈 포트G의 PG0~PG3을 Array FND의 C0~C3까지 연결
- ❖ MCU 모듈 포트D의 PD6를 푸시 스위치 모듈의 BT0에 연결



Array FND

Push Button



ICAT

Integrated Circuits for Advanced Technology Lab.

실습 11 : 외부 입력 카운터

❖ 구동 프로그램 : 사전 지식

❖ 하나의 신호 및 다양한 신호로 발생하는 외부신호를 ICPn 핀을 통해 사용 가능

❖ PD6 포트 핀의 부가기능 : T1(타이머/카운터1 클럭 입력)

❖ 타이머

- 내부 클럭(clk)을 이용하여 일정시간 간격의 펄스를 만들어 내거나 일정시간 경과후에 인터럽트를 발생시키는 기능을 말하며 사용하는 내부 클럭 이고 분주가 가능하며 범위 내에서 클럭(0~16MHz) 선택도 가능

❖ 카운터

- 외부핀(TOSC1, TOSC2, T1, T2, T3)을 통해 입력되는 펄스를 계수(Edge Detect)하여 Event, 카운터로 동작하는 것을 말하며 사용하는 클럭은 외부 클럭이며 분주가 불가능함

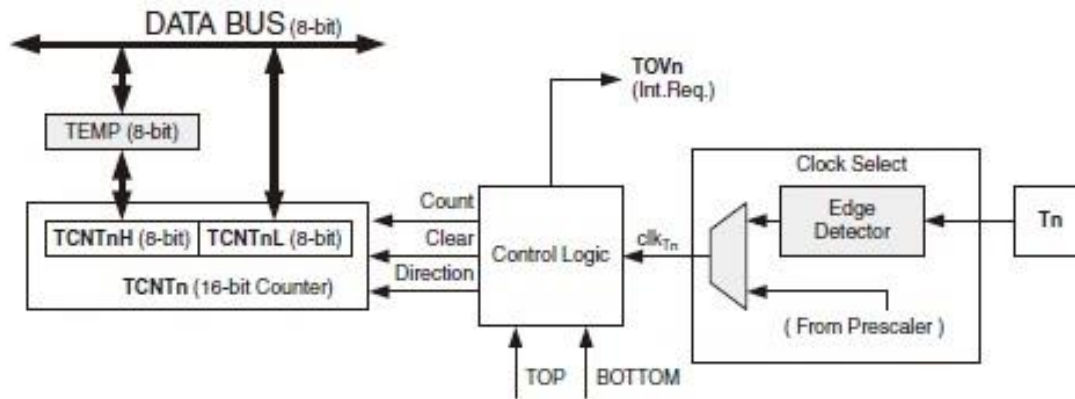


ICAT

Integrated Circuits for Advanced Technology Lab.

실습 11 : 외부 입력 카운터

- ❖ 구동 프로그램 : 사전 지식(카운터의 이해)
 - ❖ 16비트 타이머/카운터의 카운터 블록 다이어그램



- ❖ 클럭 소스인 clk_{tn} 을 내부 클럭(From Prescaler)으로 받아 2분주하는 플립플롭으로 구성
- ❖ 각각의 연속된 상태가 2분주되기 때문에 n 개의 플립플롭으로 구성된 타이머는 $2n$ 으로 입력 클럭 주파수를 나누게 된다
- ❖ 단계의 출력은 타이머가 Overflow된 것을 뜻하며 플립플롭이나 플래그 비트(I/O 제어 레지스터)에게 이를 알려줌
- ❖ 클럭 소스인 clk_{tn} 을 외부 클럭 입력(T_n)으로 받아 분주없이 외부 클럭을 그대로 사용하며, Edge Detector를 사용하여 **상향 엣지 또는 하향 엣지에서 카운팅한다**

실습 11 : 외부 입력 카운터

□ 구동 프로그램 : 소스 분석

❖ Timer1_Counter.c

1)	<pre>#include<avr/io.h> #include<util/delay.h></pre>
2)	<pre>enum{C0,C1,C2,C3}; unsigned char FND_DATA_TBL[]={0x3F,0X06,0X5B,0X4F,0X66,0X6D,0X7C,0X07, 0X7F,0X67,0X77,0X7C,0X39,0X5E,0X79,0X71,0X08,0X80}; void print_FND(unsigned char selCx, unsigned char data); //원하는 FND 포트에 data 출력</pre>
3)	<pre>int main(void) { unsigned char digit0, digit1, digit2, digit3; unsigned int temp=0; unsigned int c_value;</pre>



ICAT

Integrated Circuits for Advanced Technology Lab.

실습 11 : 외부 입력 카운터

3)	<pre>DDRB = 0xFF; //data port DDRG = 0x0F; //control port 0-3 PIN PORTB = 0x00; //초기 값은 segment가 전부 꺼져있게 만든다. PORTG = 0xFF; //초기 값은 FND 스위치를 전부 선택하지 않는다 digit0=digit1=digit2=digit3=0; //FND 표시 초기값은 "0000"</pre>
4)	<pre>TCCR1B=0x07; //타이머 1의 클럭 소스를 외부 클럭(T1) 입력으로 설정(상승에지) while(1) {</pre>
5)	<pre>c_value=TCNT1L (TCNT1H<<8); digit0= c_value/1000; //카운트 값을 1000으로 나눈 몫 //즉 1000의 자리 숫자가 된다 temp = c_value % 1000; //카운트 값을 1000으로 나눈 나머지 //즉 1000의 자리를 뺀 100단위 숫자 digit1 = temp /100; //100의 자리 수 temp = temp %100; digit2 = temp/10; //10의 자리 수 digit3= temp % 10; //1의 자리 수</pre>



실습 11 : 외부 입력 카운터

6)	<pre>print_FND(C3, digit3); //4번째 COM 에 1초단위 출력 _delay_ms(1); print_FND(C2, digit2); //3번째 COM에 10초단위 출력 _delay_ms(1); print_FND(C1, digit1); //2번째 COM에 1분 단위 출력 _delay_ms(1); print_FND(C0, digit0); //1번째 COM에 10분단위 출력 _delay_ms(1); } }</pre>
7)	<pre>void print_FND(unsigned char selCx, unsigned char data){ switch(selCx){ case C3: //4번째 FND (좌측기준) PORTG &= 0xF0; PORTG = 0x07; break; case C2: //3번째 FND PORTG &= 0xF0; PORTG = 0x0B; break; case C1: //2번째 FND PORTG &= 0xF0; PORTG = 0x0D; break; }</pre>



실습 11 : 외부 입력 카운터

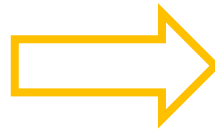
```
7) case C0: //1번째 FND
PORTG   &=      0xF0;
PORTG   |=      0x0E;
break;
default:
PORTG   &=      0xF0;
PORTG   |=      0x0F;
break;
}
PORTB = FND_DATA_TBL[data];
}
```



실습 11 : 외부 입력 카운터

□ 실행 결과

- ❖ 스위치 모듈의 푸쉬 스위치를 1번 누를 때마다 Array FND의 값이 하나씩 증가한다.



실습 12 : 외부 입력 캡처

□ 실습 개요

- ❖ 타이머/카운터1을 통해서 외부 입력을 캡처 하는 실습
- ❖ 외부 입력 캡처를 사용하여 이벤트가 발생한 시간을 저장한다

□ 실습 목표

- ❖ 타이머1의 외부 입력 카운터 동작원리 이해
- ❖ 16비트 타이머/카운터 관련 레지스터 이해



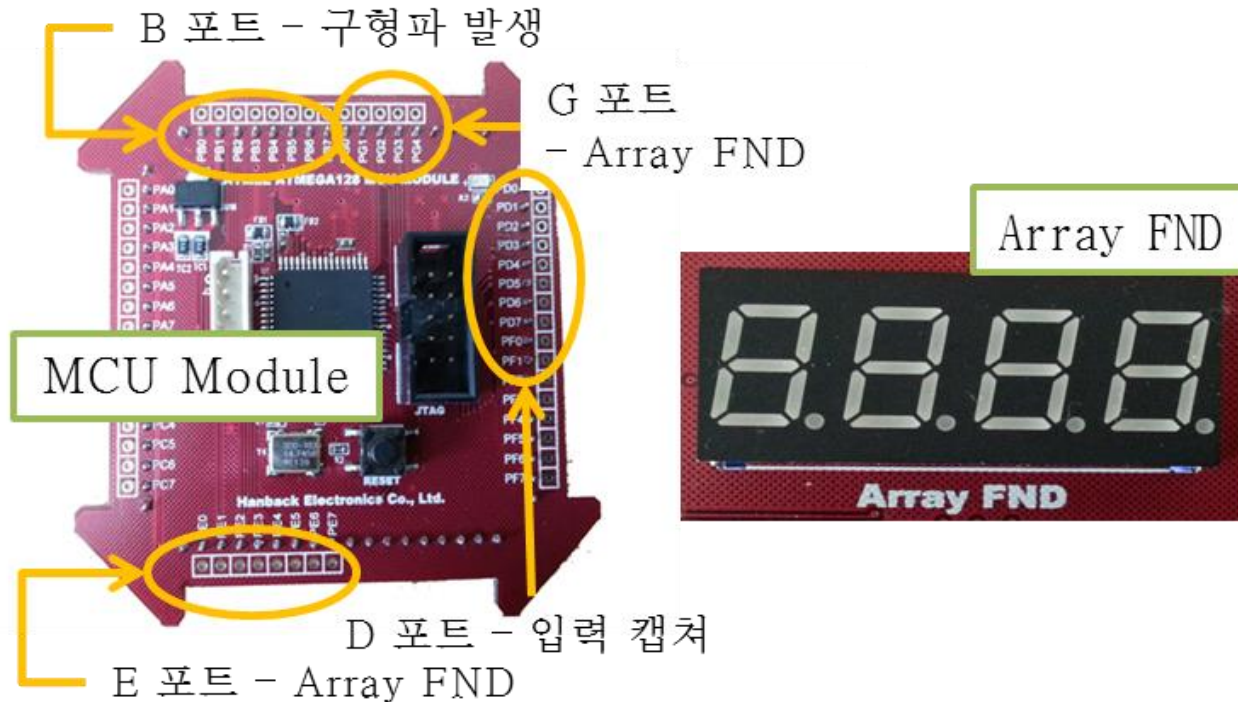
ICAT

Integrated Circuits for Advanced Technology Lab.

실습 12 : 외부 입력 캡처

□ 사용 모듈

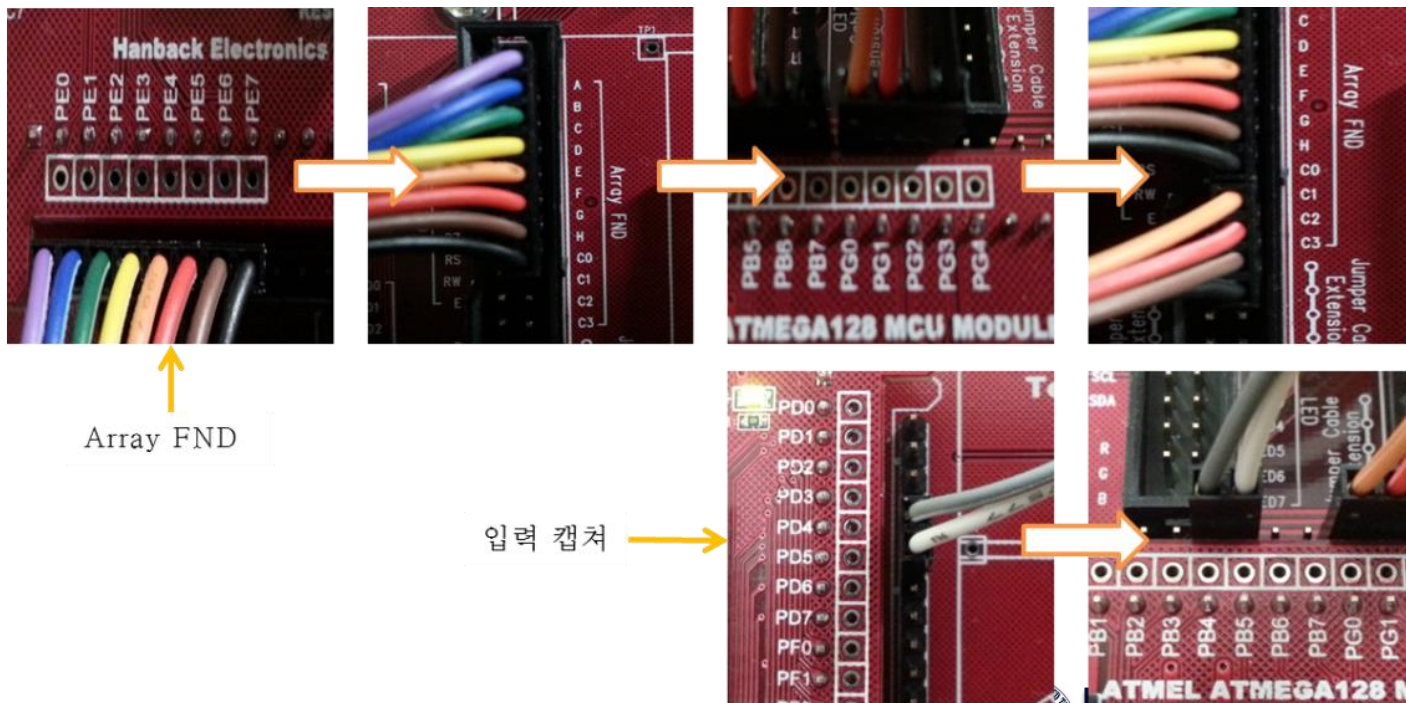
- ❖ MCU 모듈, Array FND 모듈



실습 12 : 외부 입력 캡처

□ 모듈 결선 방법

- ❖ MCU 모듈 포트 E의 PE0~PE7을 Array FND 모듈의 A~H까지 연결.
- ❖ MCU 모듈 포트 G의 PG0~PG3을 Array FND 모듈의 C0~C3까지 연결.
- ❖ MCU 모듈 포트 B의 PB4를 MCU 모듈 포트 D의 PD4핀에 연결.



실습 12 : 외부 입력 캡처

□ 구동 프로그램 : 사전 지식

- ❖ 외부 신호는 타이머 입력 캡처 핀(ICPn) 또는 아날로그 비교기의 출력(ACO)에서 발생한 신호 변화를 검출하여 캡처가 이루어 짐
- ❖ 일반적으로 입력 캡처 핀(ICPn)에 의해 사용되며 아날로그 비교기의 출력 (ACO)을 사용할 때에는 ACSR 레지스터의 ACIC 비트를 설정하여 선택하여 사용
- ❖ 부가적으로 노이즈 제거기 선택 가능
 - TCCRnB 레지스터의 입력 캡처 잡음 제거기(ICNC) 비트를 “1”로 설정
- ❖ 16비트 타이머/카운터에만 존재
- ❖ 타이머 입력 캡처 핀(ICPn)의 로직 레벨 변화(High or Low)가 있거나 ICRn 레지스터가 TOP으로 사용되는 PWM 동작모드에서 TCNTn값이 TOP이 될 때 인터럽트가 발생한다.
- ❖ 캡처가 발생되면 카운터 값(TCNTn)은 입력 캡처 레지스터(ICRn)에 저장되고, 동시에 입력 캡처 플래그(ICFn)는 Set 된다.
- ❖ PD4 포트 핀 : IC1(타이머/카운터1 입력 캡처 트리거)

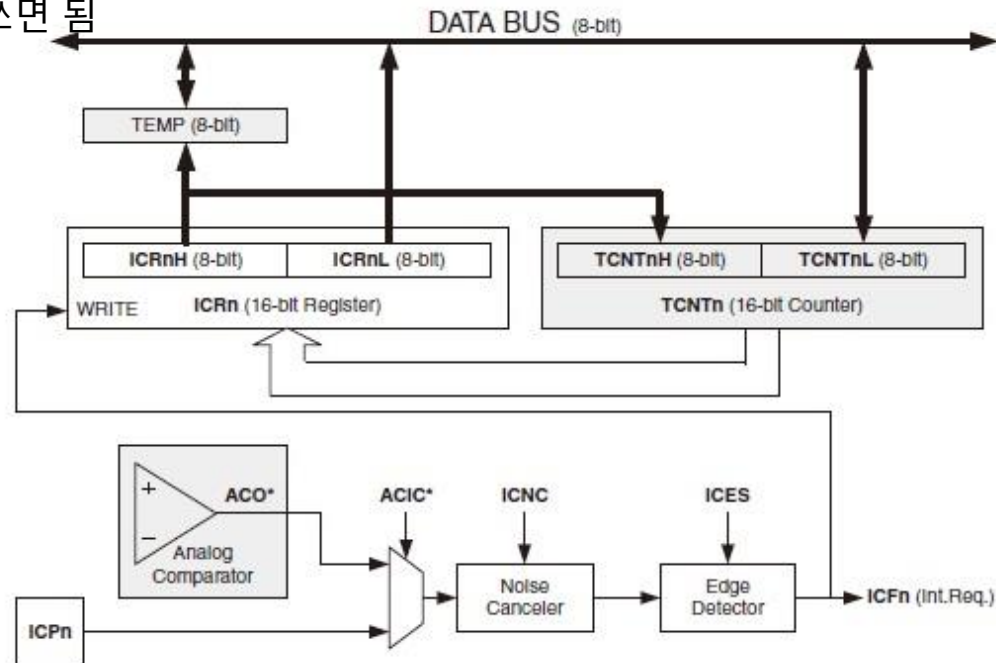


실습 12 : 외부 입력 캡처

□ 구동 프로그램 : 사전 지식

❖ 16비트 입력 캡처 장치 블록 다이어그램

- ICRn을 읽을 경우에는 하위 바이트(ICRnL)를 읽고 나서 상위 바이트(ICRnH)를 읽어야 함, 하위 바이트를 읽는 동안 상위 바이트는 임시 저장
- ICRn 레지스터에 데이터를 쓸 경우에는 읽을 때는 상위 바이트를 먼저 쓰고 하위 바이트를 나중에 쓰면 됨



실습 12 : 외부 입력 캡처

□ 구동 프로그램 : 소스 분석

❖ Timer1_inputcapture.c

1)	<pre>#include<avr/io.h> #include<avr/interrupt.h> #include<util/delay.h></pre>
2)	<pre>enum{C0,C1,C2,C3}; unsigned char FND_DATA_TBL[]={0x3F,0X06,0X5B,0X4F,0X66,0X6D,0X7C,0X07, 0X7F,0X67,0X77,0X7C,0X39,0X5E,0X79,0X71,0X08,0X80}; void print_FND(unsigned char selCx, unsigned char data); //원하는 FND 포트에 data 출력 volatile unsigned int c_value;</pre>
3)	<pre>int main(void) { unsigned char digit0, digit1, digit2, digit3; unsigned int temp=0; DDRB = 0x10; //PB4(OCR0)핀 출력 설정</pre>



ICAT

Integrated Circuits for Advanced Technology Lab.

실습 12 : 외부 입력 캡처

3)	<pre>DDRE = 0xFF; //data port DDRG = 0x0F; //control port 0-3 PIN PORTE = 0x00; //초기 값은 segment가 전부 꺼져있게 만든다. PORTG = 0xFF; //초기 값은 FND 스위치를 전부 선택하지 않는다. digit0=digit1=digit2=digit3=0; //FND 표시 초기값은 "0000"</pre>
4)	<pre>TCCR0 = 0x1E; //CTC Mode, 256분주, 50% duty (00011110) TCNT0 = 0; OCR0=150;</pre>
5)	<pre>TIMSK = 0x20; //Timer1 Input Capture 인터럽트 허용 TCCR1B = 0x44; //(01000100) ICES1=1→ 상승엿지에서 입력캡처, CSn2:0=100 → 256분주 sei(); while(1) {</pre>
6)	<pre>digit0= c_value/1000; //카운트 값을 1000으로 나눈 몫 //즉 1000의 자리 숫자가 된다 temp = c_value % 1000; //카운트 값을 1000으로 나눈 나머지 //즉 1000의 자리를 뺀 100단위 숫자</pre>



실습 12 : 외부 입력 캡처

6)	<pre>digit1 = temp /100; //100의 자리 수 temp = temp %100; digit2 = temp/10; //10의 자리 수 digit3= temp % 10; //1의 자리 수 print_FND(C3, digit3); //4번째 COM 에 1초단위 출력 _delay_ms(1); print_FND(C2, digit2); //3번째 COM에 10초단위 출력 _delay_ms(1); print_FND(C1, digit1); //2번째 COM에 1분 단위 출력 _delay_ms(1); print_FND(C0, digit0); //1번째 COM에 10분단위 출력 _delay_ms(1); } }</pre>
7)	<pre>SIGNAL(SIG_INPUT_CAPTURE1) { c_value = ICR1L (ICR1H<<8); TCNT1 = 0; }</pre>
8)	<pre>void print_FND(unsigned char selCx, unsigned char data){ switch(selCx){</pre>



실습 12 : 외부 입력 캡처

```
8) case C3: //4번째 FND (좌측기준)
PORTG   &=      0xF0;
PORTG   |=      0x07;
break;
case C2: //3번째 FND
PORTG   &=      0xF0;
PORTG   |=      0x0B;
break;
case C1: //2번째 FND
PORTG   &=      0xF0;
PORTG   |=      0x0D;
break;
case C0: //1번째 FND
PORTG   &=      0xF0;
PORTG   |=      0x0E;
break;
default:
PORTG   &=      0xF0;
PORTG   |=      0x0F;
break;
}
PORTE = FND_DATA_TBL[data];
}
```



실습 12 : 외부 입력 캡처

□ 실행 결과

- ❖ OCR0의 값이 “150”이므로 TCNT0=150일때 PB4의 구형파(OC0)가 1->0으로 변경, TCNT0=0으로 클리어, 이때 상승에지가 아니므로 IC1 인터럽트가 아직 발생하지 않고 분주비가 같으므로 TCNT1=151
- ❖ 두번째 TCNT0=150일때, PB4는 0->1로 변경, 상승에지가 발생하여 IC1 인터럽트가 발생하고 이때의 TCNT1=302 값이 ICR1에 저장됨
- ❖ Array FND에 “0302”가 다음과 같이 출력된다.

