마이크로프로세서응용
(초음파센서)

2023-2학기

Kookmin Univ. EMCO Lab.

Contents

0.3cm
Resolution

<15
Effectual Angle

<2mA
Quiescent Current

2~450cm
Detection Distance

Motor Control Lab.
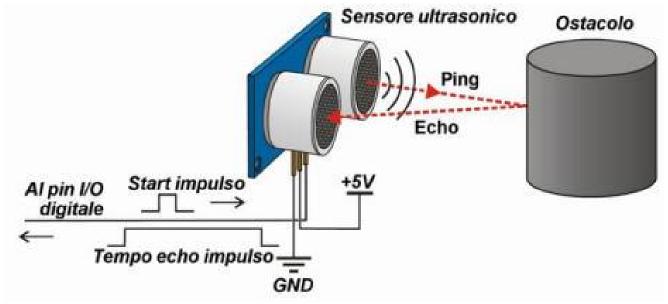
1. SR-HC04

- HW동작

$$The\ \ range = high\ \ level\ \ time / 2 * velocity(340M / S)$$

2. eTimer

- MPC5604P eTimer block diagram

Figure 27-2. eTimer channel block diagram

- Counter

Figure 27-9. Counter register (CNTR)

*Counter register: 클럭신호의 edge때마다 그 값이 1씩 더해지는 레지스터
*사용하는 클럭신호의 주파수에 따라 카운팅되는 속도가 달라짐
*강의자료 코드에서는 64MHz를 128분주해서 사용 (0.5MHz, 1카운트당 2μs)

- Capture

Figure 27-5. Capture register 1 (CAPT1)

on echo_rising edge

Figure 27-6. Capture register 2 (CAPT2)

on echo_falling edge

*Capture: 특정 이벤트가 발생하면 카운터 레지스터 값을 읽어오는 기능

- 계산식

```
If(CAPT1<CAPT2)
{
 counts_high = CAPT2-CAPT1;
}
else
{
 counts_high = 0xFFFF-CAPT1+CAPT2;
}
```
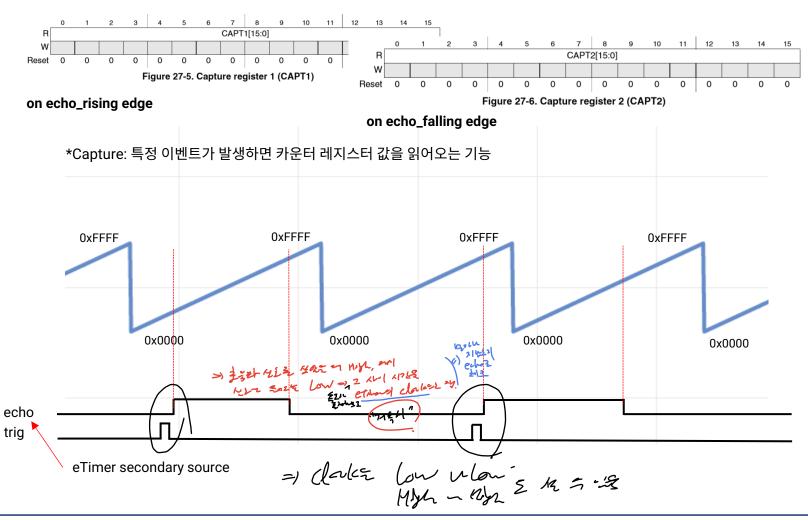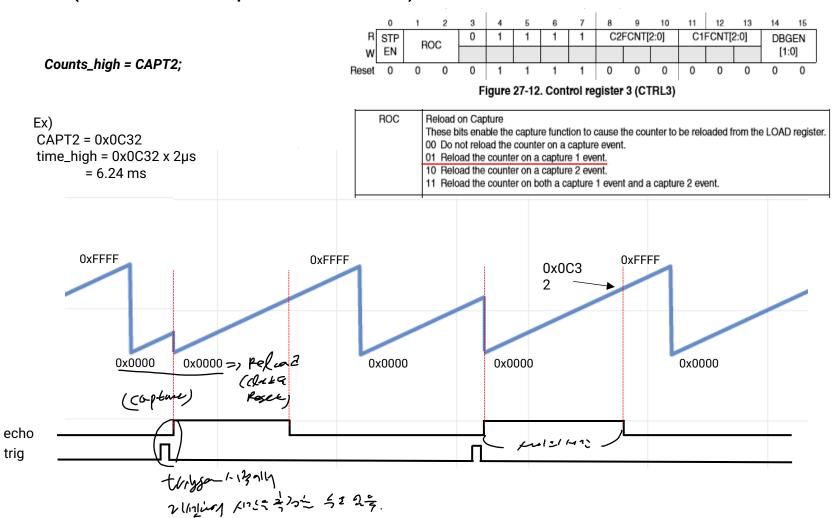
Ex)
 CAPT1 = 0x00F3, CAPT2 = 0x0A03
 counts = 0x0A03-0x00F3
        = 0x0910

Time_high = 0x0910 x 2µs   (카운트횟수 x 카운트 하나당 걸리는 시간)
          = 4.64 ms

Motor Control Lab.

- 계산식 (Reload on Capture 기능 사용)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	STP EN	ROC		0	1	1	1	1	C2FCNT[2:0]			C1FCNT[2:0]			DBGEN [1:0]	
W																
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

Figure 27-12. Control register 3 (CTRL3)

Counts_high = CAPT2;

ROC	Reload on Capture
	These bits enable the capture function to cause the counter to be reloaded from the LOAD register.
	00 Do not reload the counter on a capture event.
	01 Reload the counter on a capture 1 event.
	10 Reload the counter on a capture 2 event.
	11 Reload the counter on both a capture 1 event and a capture 2 event.

Ex)
 CAPT2 = 0x0C32
 time_high = 0x0C32 x 2µs
        = 6.24 ms

- eTimer 초기화　　=> Capture 기능으로 사용

```c
void init_eTimer(void)
{

// =============== eTimer initialization =======
    ETIMER_0.ENBL.R = 0b000000;

    ETIMER_0.CHANNEL[0].CTRL.B.CNTMODE = 0b001;      // Count rising edges of primary source
    ETIMER_0.CHANNEL[0].CTRL.B.PRISRC  = 0b11111;    // From Table 'Count source values'
    ETIMER_0.CHANNEL[0].CTRL.B.ONCE    = 0;          // Count repeatedly
    ETIMER_0.CHANNEL[0].CTRL.B.LENGTH  = 0;          // Continue counting to roll over
    ETIMER_0.CHANNEL[0].CTRL.B.DIR     = 0;          // Count up
    ETIMER_0.CHANNEL[0].CTRL.B.SECSRC  = 0b00000;    // From Table 'Count source values'

    ETIMER_0.CHANNEL[0].CCCTRL.B.CPT2MODE = 0b01;    // Capture falling edges    => Capture 2에서
    ETIMER_0.CHANNEL[0].CCCTRL.B.CPT1MODE = 0b10;    // Capture rising edges              인터럽트
    ETIMER_0.CHANNEL[0].CCCTRL.B.ONESHOT = 0;        // Free-running mode is selected
    ETIMER_0.CHANNEL[0].CCCTRL.B.ARM = 1;            // Input capture operation as specified
                                                     // by the CPT1MODE and CPT2MODE bits is enabled

    ETIMER_0.CHANNEL[0].CTRL3.B.ROC    = 0b01;       // Reload the counter on a capture 1 event

    ETIMER_0.CHANNEL[0].INTDMA.B.ICF2IE = 1;         // Input Capture 2 flag interrupt Enable

    ETIMER_0.ENBL.R = 0b000001;         ( Capture~1       // ETC[0] Enabled
                                          reload )
// =============== eTimer PIN initialization =======
    SIU.PCR[0].R = 0x0500;                           // eTimer_0 Func_ETC[0] PIN A[0]
}
```

- eTimer 루틴

```
void eTimer_0(void)
{
    ETIMER_0.CHANNEL[0].STS.B.ICF2 = 1;

    echo = ETIMER_0.CHANNEL[0].CAPT2.R;
    /*
    time_high  = 128/64M sec x echo (2 micro sec per 1 count)
    distance   = 340 x time_high / 2
               = 340 x echo
    */
    distance = 340 * echo / 10000;   // centimeter
}
```

2. eTimer

```c
#include "MPC5604P_M26V.h"
#include "freemaster.h"
#include "init_base.h"


/******************** Variable ********************/
volatile int i = 0;
unsigned int distance=0,echo=0,eTimerCnt=0,Pit0cnt=0;


/******************** Function ********************/
void init_eTimer(void);
void eTimer_0(void);
void init_PIT(void);
void PIT0ISR(void);

int main(void)
{
    initModesAndClock();
    disableWatchdog();
    enableIrq();
    FMSTR_Init();
    init_INTC();
    init_Linflex0();
    init_PIT();
    init_eTimer();
    INTC_InstallINTCInterruptHandler(eTimer_0,157,14);
    INTC_InstallINTCInterruptHandler(PIT0ISR,59,6);

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

        i++;
    }
}
```

3. 실습

1. 초당 10번 값을 가져올 수 있도록 센서의 **trigger** 신호를 만드세요
   Hint) PIT

2. **CAPT2**로 가져온 값을 이용해 거리를 **cm**단위로 표현하세요
    Hint) CLK_primary source: 0.5Mhz

실습 답안

- 1번 문제

```
void init_PIT(void)
{
    PIT.PITMCR.R = 0x00000001;          // PIT Enabled and Config stop in debug mode
    PIT.CH[0].LDVAL.R = 0x00000280;     // 10 us
    PIT.CH[0].TCTRL.R = 0x3;            // Timer interrupt enabled & start

    SIU.PCR[55].R = 0x0200;             // PD[7]
}



void PIT0ISR(void)
{
    PIT.CH[0].TFLG.B.TIF = 1;           // Clear PIT0 Flag
    Pit0cnt++;

    if(Pit0cnt==10000)
    {
        SIU.GPDO[55].B.PDO = 1;
        Pit0cnt=0;
    }
    else
    {
        SIU.GPDO[55].B.PDO = 0;
    }

}
```

=> 0.1초마다 on/off

실습 답안

- 2번 문제

```
void eTimer_0(void)
{

    ETIMER_0.CHANNEL[0].STS.B.ICF2 = 1;

    echo = ETIMER_0.CHANNEL[0].CAPT2.R;
    /*
    time_high  = 128/64M sec x echo (2 micro sec per 1 count)
    distance   = 340 x time_high / 2
               = 340 x echo
    */
    distance = 340 * echo / 10000;  // centimeter
}
```