


Ultra-Sonic Ranging

- ❖ Ultra-Sonic
- ❖ Ranging sensor
- ❖ C-Programming



Ultra-Sonic

- 초음파

음파의 진동수가 20KHz 이상이며, 사람의 청력으로는 들을 수 없는 주파수 영역이다. 파동을 갖는 종파의 일종이며, 지향성 및 직진성이 높고 온도변화에 민감하며, 매질에 따라 산란, 굴절, 반사, 투과하는 특성이 있다.

- 1) 진동수가 높을 수록 초음파 Beam의 방사각이 줄어든다.
- 2) Horn 직경이 클수록 동일한 진동수에서 빔의 방사각이 줄어든다.
- 3) 진동수가 낮을 수록 투과력이 높아지고 도달거리가 길어진다.
- 4) 매질의 경계면이나 불연속 면에서 주로 반사, 굴절, 회절 하며 공기층과 만나면 100% 반사 한다.

- ❖ 반사(Reflection) 특성

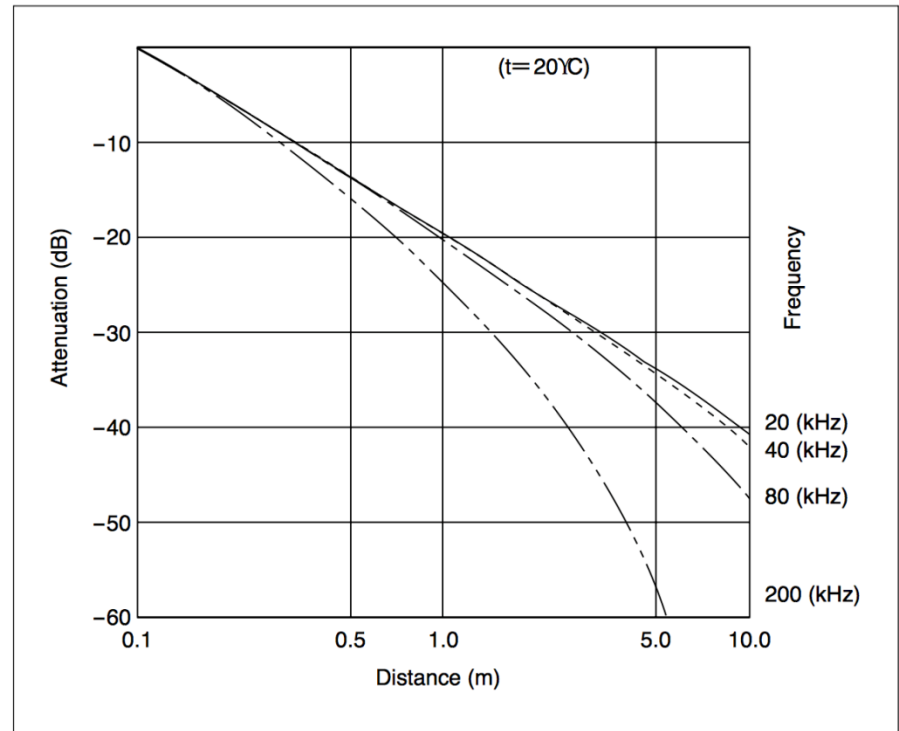
초음파의 반사특성을 이용하여 물체의 존재를 검출할 수 있으며, 재질에 따라 반사율이 다르다.


- $\approx 100\%$ 반사: 금속, 목재, 콘크리트, 유리, 고무 및 종이 등
- 흡 수: 의류, 면직물(cotton), 양모(wool) 등

Ultra-Sonic

❖ 감쇠(Attenuation) 특성

- 초음파가 공기속을 진행할 때 거리에 비례하여 세기가 감소된다. 이러한 이유는 회절현상과 흡수손실로 인한 구형 표면에서의 확산 손실에 의해 발생되며, 에너지가 매질에 의해 흡수된다.
- 초음파의 주파수가 높을수록 감쇠율이 커지며, 음파의 도달거리가 짧아진다.
- 흡수: 초음파가 지닌 에너지가 열로 변환되어 소실되는 현상.





Ultra-Sonic

❖ 굴절(Refraction) 특성

-물체가 입사되는 초음파 빔에 대하여 경사져 있을 때 빔의 방향이 굴절되는 현상이다.

❖ 산란(Scatter) 특성

-초음파가 자신의 파장보다 불규칙하거나 작은 면을 가진 반사물체에 부딪칠 때 여러 방향으로 흩어지는 현상이다.

❖ 온도 영향

음파의 전파속도는 다음 식과 같으며, 주위온도에 따라 변화한다.

$$v = 331.5 + 0.607 T_c \quad [\text{m/s}]$$

여기서, T_c : 온도[$^{\circ}\text{C}$]

Ultra-Sonic

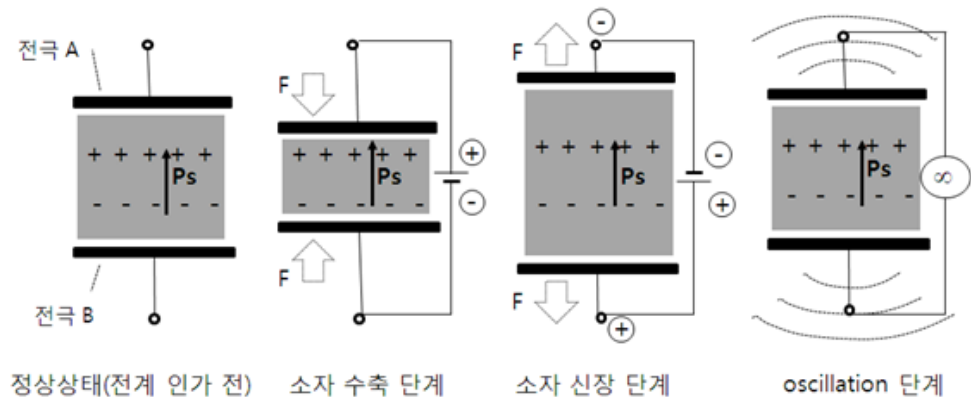
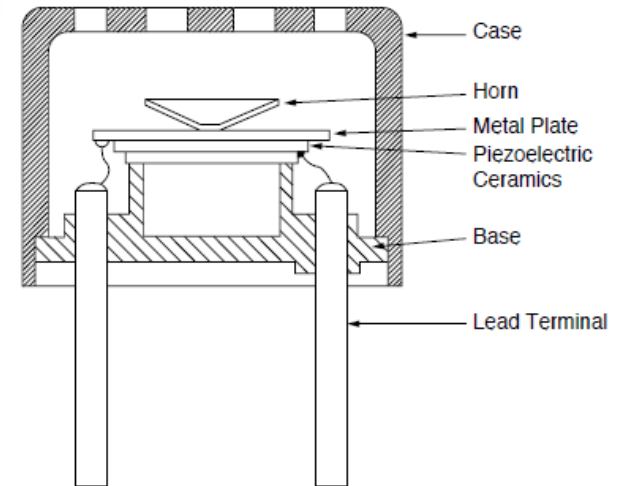
❖ 초음파센서의 구조와 동작원리


Piezoelectric Ceramics에 전압이 인가 되면, 기계적인 변형이 전압과 주파수에 따라 발생된다.

반대로 압전세라믹에 진동이 인가되면, electric charge가 생성된다.

■ 압전(Piezo-Electric)효과

일반적으로 수정, 로셀염, 티탄산바륨 등의 결정에 압력을 가하면 그 힘에 비례한 전하가 유도되고, 역으로 전기장을 인가하면 계적인 변형이 발생 되는 현상.





Ultra-Sonic

❖ 초음파의 응용

1) 어군탐지기:

1921년 P.langevin(랑지뱅,프랑스물리학자)이 초음파 SONAR를 개발하여 잠수함 탐지에 이용, 측심기로 실용화 되어 어군탐지기로 응용됨.

2) 초음파 탐상기:

1937년 S.Y. 소콜로프가 주물의 흠이나 금간 것을 발견한 것이 시초이며 철도차량의 차축검사 및 고체재료의 내부 결함, 용접결함 등 비파괴 검사로 널리 사용 되고있다.

3) 초음파 가공 및 용착:

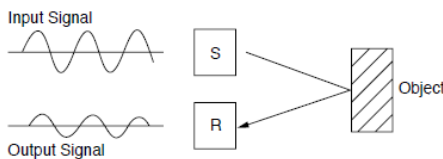
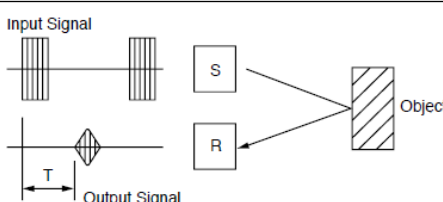
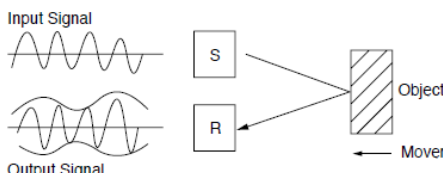
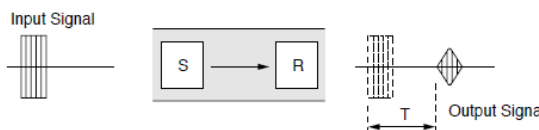
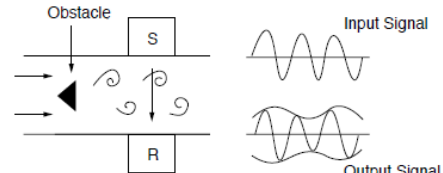
액체 위에 뜨는 작은 고체입자를 진동시켜 다른 물체 표면과 충돌 시킴으로써 유리, 보석, 초경합금 등의 가공할 수 있다. 알루미늄, 니오븀 등의 용접이 어려운 금속을 납땜 가능하다. 기타, 초음파 세척, 가습, 건조 등이 있다.


4) 의료 기기:

1942년 K.T. 뒤시크가 의료분야에 초음파를 활용하여 뇌종양을 진단하였으며 이후 산부인과 검사, 암검사 등으로 활용 되고 있다.

Ultra-Sonic

❖ 초음파의 응용

No.	Function Method	Performance Principle (S: transmitter, R: receiver)	Applications
1	Detection of Signal level of continuous wave		Counting instruments Access switches Parking meters
2	Measurement of pulse reflection time		Automatic doors Level gauges Automatic change-overs of traffic signals Back sonars of automobiles
3	Utilization of Doppler effect		Intruder alarm systems
4	Measurement of direct propagation time		Densitometers Flowmeters
5	Measurement of Karman vortex		Flowmeters



Ultra-Sonic

- Sound Velocity

- 공기(Air)속에서의 음파의 속도:

$$\begin{aligned}V_s &= 331.5 + 0.6T_C \\ &= (167.6 + 0.6T_K)\end{aligned}$$

Where

V_s is the Velocity of Sound in m/s

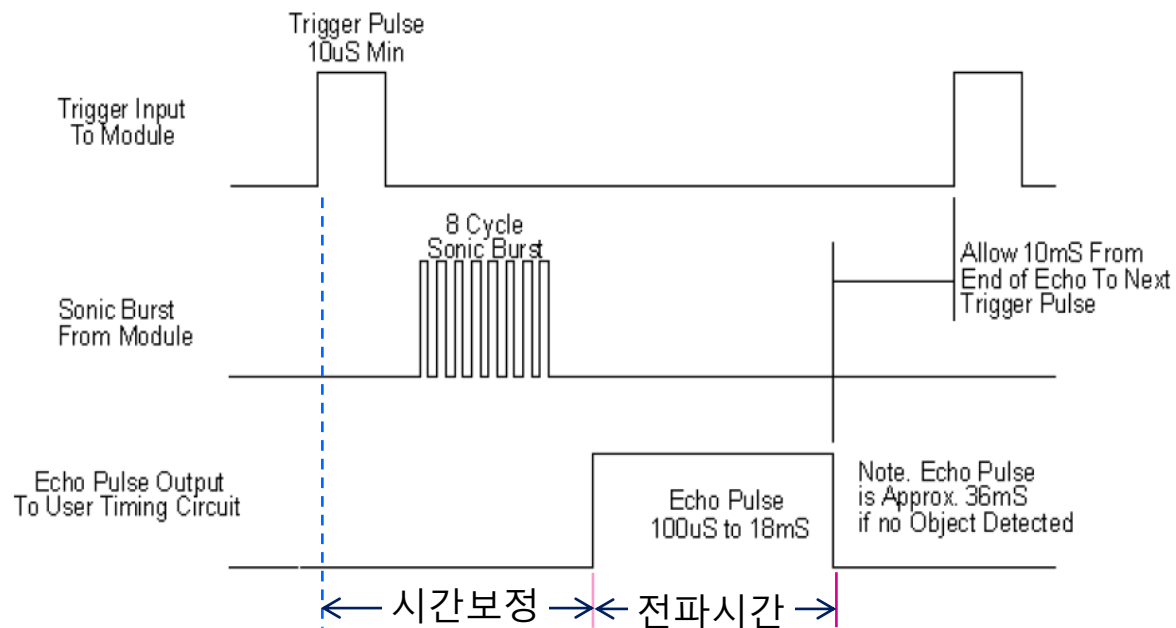
T_C is the Temperature in degrees Celsius

T_K is the Temperature in Kelvin

2. Ranging Sensor

- 초음파(Ultra-Sonic)를 이용한 거리측정

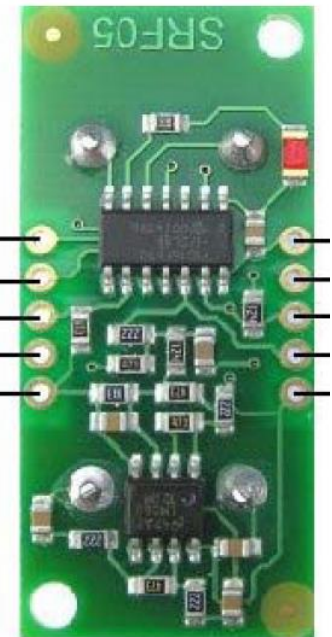
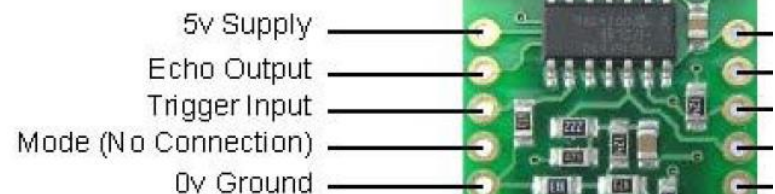
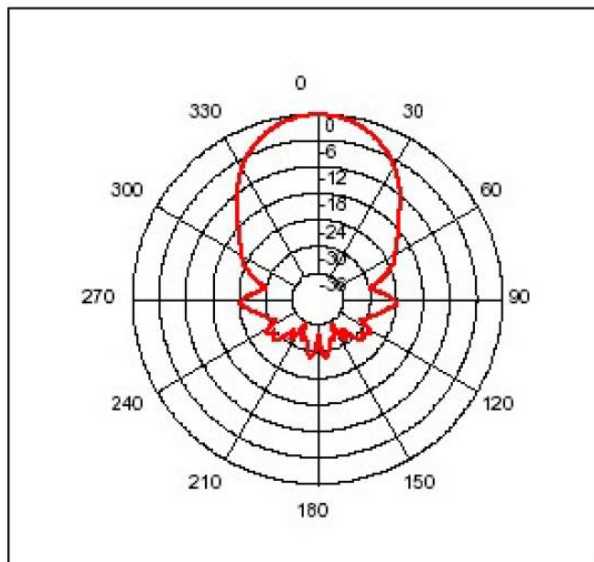
- 초음파 송출 (Trigger Pulse 발생)
- 반사음파 (Echo Signal)검출 : Echo Pulse
- 초음파 송출에서 검출까지의 시간 측정
- 거리계산하여 표시 : $S = \frac{1}{2} V_s \cdot t [m]$



Ranging Sensor: SRF-05

● 초음파 거리측정센서: SRF-05

- Voltage : 5v only required
- Low Current : 4mA Typ.
- Frequency : 40KHz
- Max Range : 4M
- Min Range : 1Cm
- Modes : mode 1 - Single pin for trig
mode 2 - Echo or 2 Pin SRF04 compatible
- Input Trigger : 10uS Min. TTL level pulse
- Echo Pulse : Positive TTL level signal, width proportional to range.
- Small Size : 43mm x 20mm x 17mm height

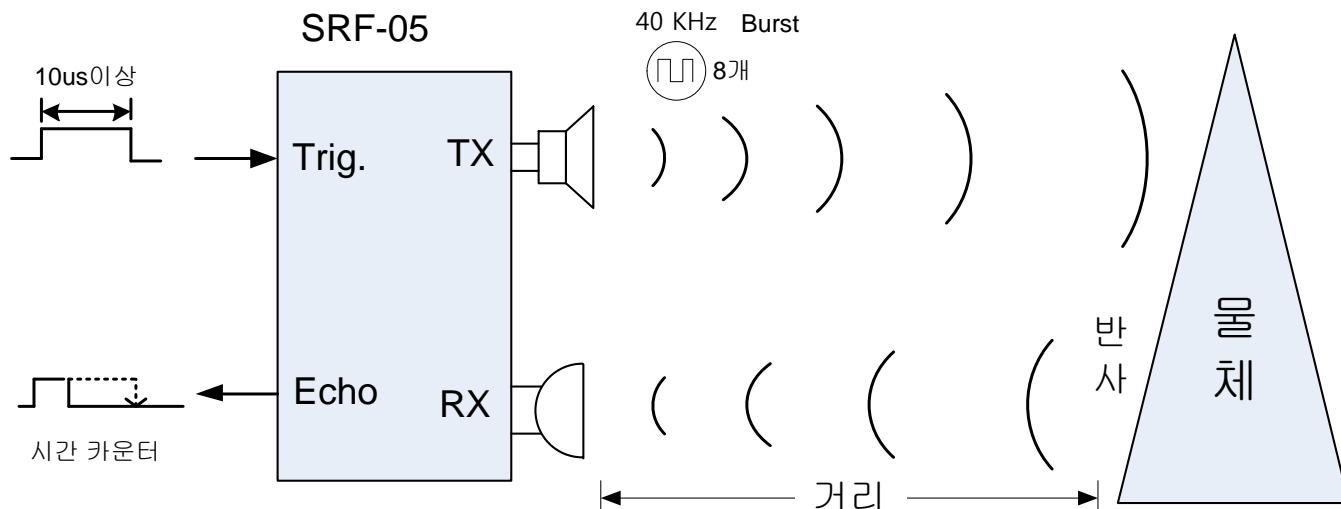


Ranging Sensor: SRF-05

❖ 초음파센서 동작 원리

초음파센서 모듈 SRF-05의 Trig. 단자에 10us이상의 pulse를 인가하면, 40khz펄스 8개로 이루어진 초음파 군집신호가 송신speaker를 통해 발사된다. 그 후 Echo출력을 High상태로 만든다.

발사된 초음파 펄스가 물체에 부딪쳐 반사되면, 수신mic를 통해 입력되고 Echo출력을 Low상태로 만든다.

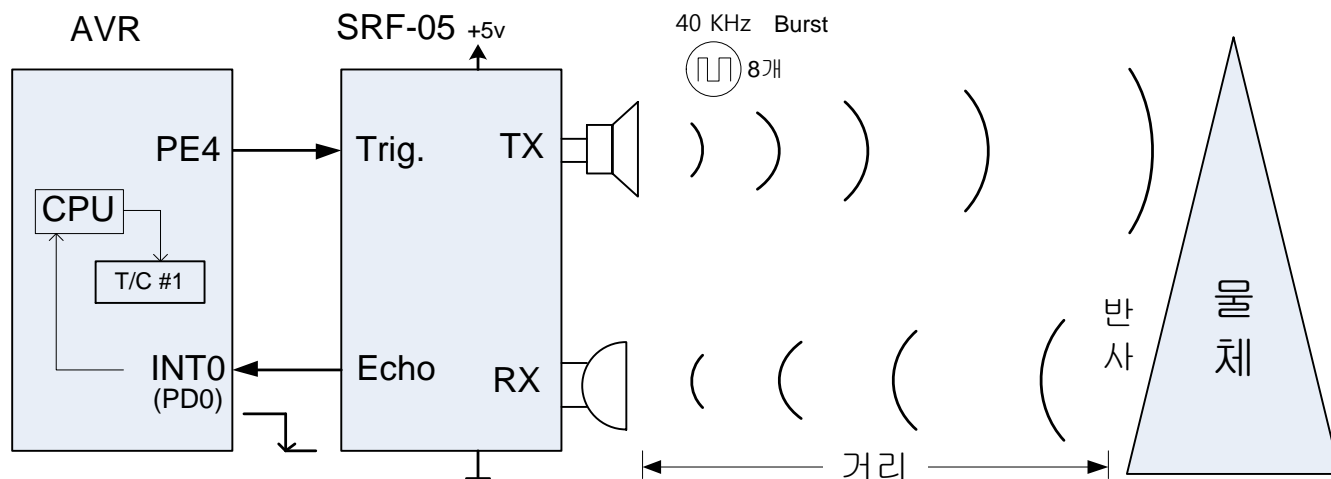


Ranging Sensor: SRF-05

- 초음파센서 이용한 거리측정

AVR이 PE4를 통해 Trig. 펄스를 출력하고, 16-bit Timer/Counter1을 동작시켜 시간을 카운터 하게한다. SRF-05는 물체로부터 반사된 초음파가 검출되면 Echo출력을 Low상태로 만든다. AVR은 Echo신호가 INT0(PD0) 인터럽트를 발생시키면 T/C#1의 카운터 동작을 멈추고 경과시간을 계산하여 물체와의 거리를 산출한다.

$$S = vt = v \cdot \frac{1}{2} T_c = 340 \times \frac{1}{2} \times 0.5us \times Count [m]$$





3. C-Programming

- C- 언어의 구조

Header

```
# include <kkk.h>
# include "bbb.h"           //사용자 header file
# define TRUE 1
unsigned char distance, time, echo;    // 전역변수 선언
(void) Ranging(void);                // 사용자함수 선언
int ultraSonic(char inp){
    ;
}                                //사용자 함수
```

Main함수

```
(void)main(void) {
    unsigned int i, j, k, srt, end; //지역변수선언
    init( );                        // 초기화 관련 사용자함수
    sei( );
    while(1) {
        ;                          // code 또는 사용자 함수
    };
}                                // main 함수 끝
```



AVR - Programming

- MCU(Micro-Control Unit) 초기화

```
void MCU_initialize(void)      /* initialize ATmega128 MCU */
{
    MCUCR = 0x80; // Enable external memory and I/O, Disable SLEEP mode
    XMCRA = 0x44; // 0x1100 - 0x7FFF (1 wait), 0x8000 - 0xFFFF (0 wait)
    XMCRB = 0x80; // Enable bus keeper. Use PC0-PC7 as high byte of address bus

    DDRB = 0xF0; // 입력/출력 방향결정, PORTB7..4 = output, PORTB3..0 = Input
    PORTB = 0x00; // I/O pin 초기값 지정

    DDRD = 0x00; // Port D = Input, "1= Output, 0= Input"
    PORTD = 0x00;

    DDRE = 0x02; // Port E1 = output, All other PORTs are Input
    PORTE = 0x00;

    DDRF = 0x00; // Port F = Input
    PORTF = 0x00;

    DDRG = 0x1F;

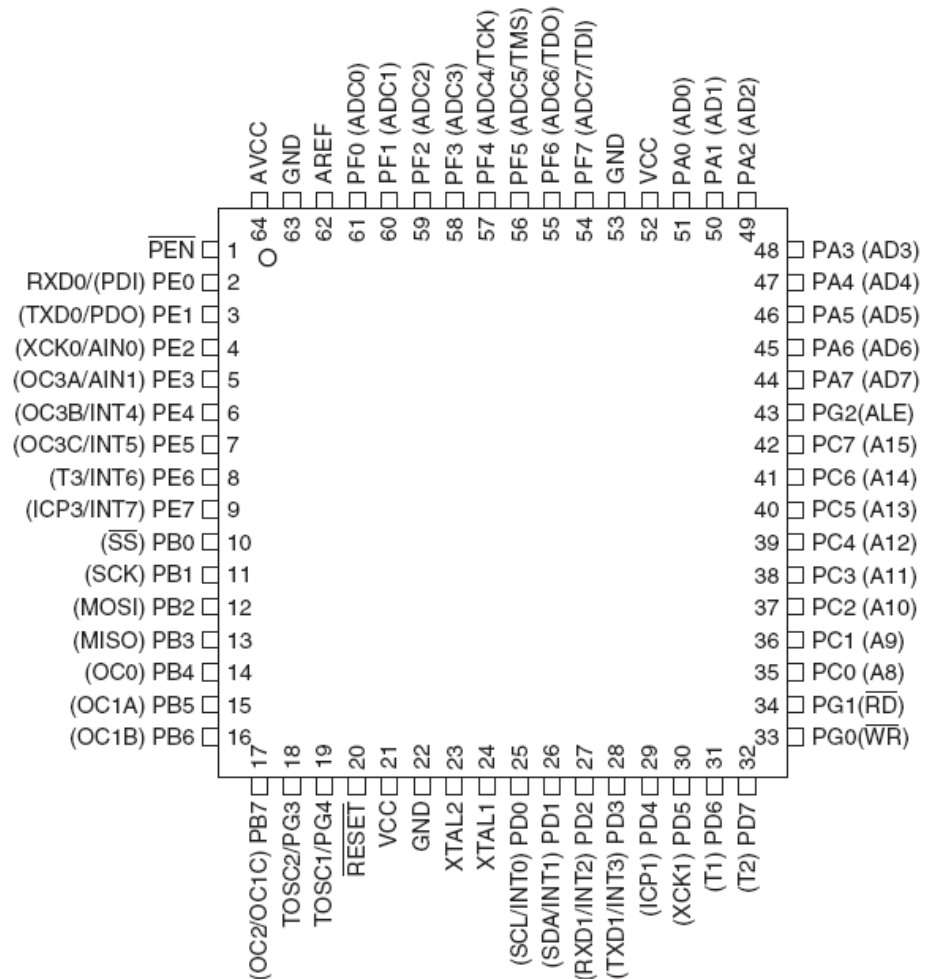
    LCD_CONTROL = 0x00; // LCD 제어
    LCD_DATABUS = 0x00; // LCD Data
}
```

AVR - Programming

● ATmega128 MCU 핀 구성도

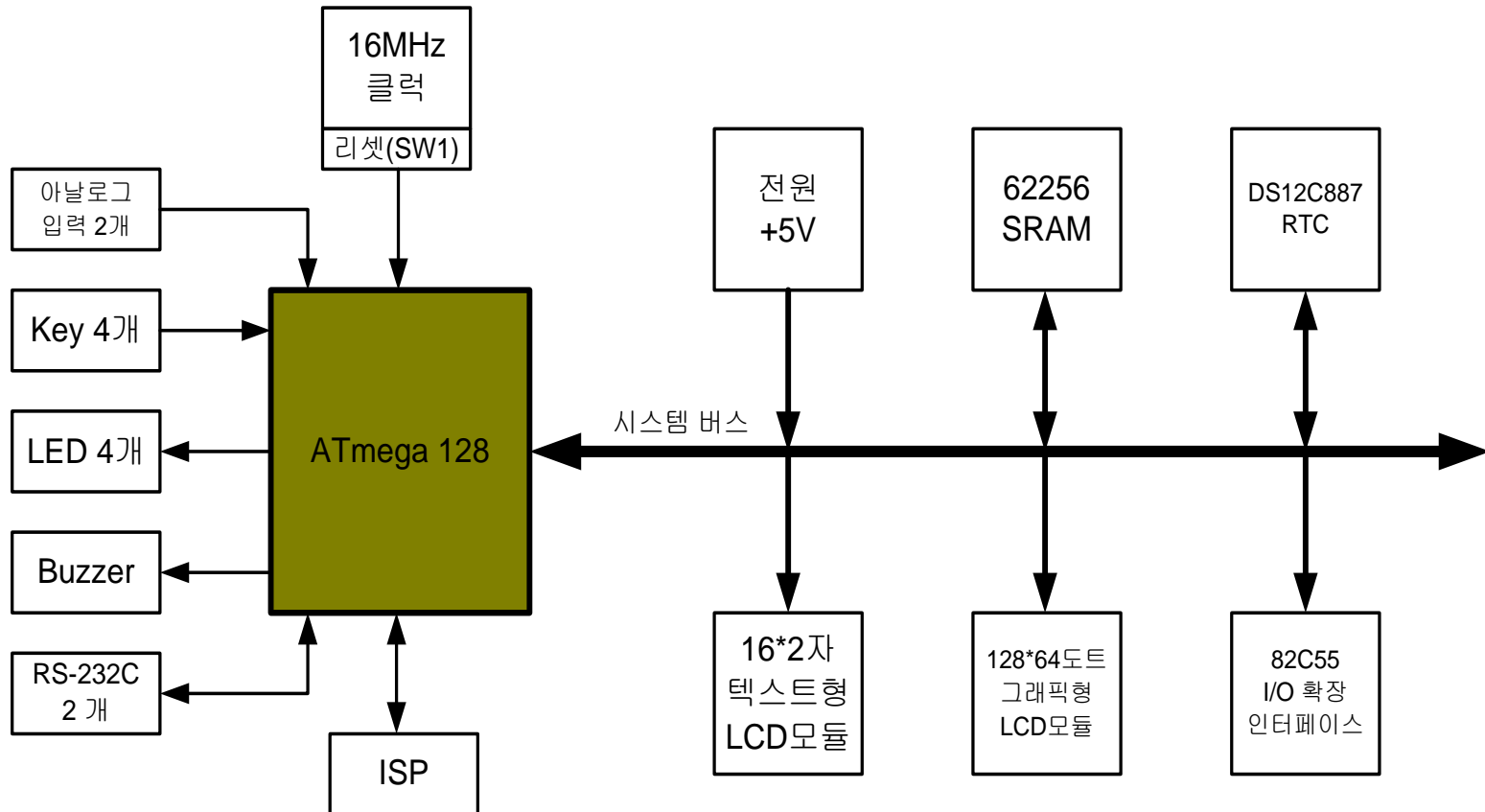
Pin Configurations

Figure 1. Pinout ATmega128



AVR - Programming

- 실험용 Board의 시스템 구성도





AVR - Programming

● Interrupt 설정

- Interrupt 는 우선순위 와 주소(종류)를 갖는 Vector 임
- 일부를 제외하고, 허용/금지가 가능함
- 형식

```
ISR(INT0_vect) {                               // 인터럽트 서비스 루틴
    Echo_count = TCNT1;
}

ISR(INT1_vect) {
    Echo_count = TCNT1;
}

ISR(INT2_vect) {
    Echo_count = TCNT1;
}

ISR(INT3_vect) {
    Echo_count = TCNT1;
}
```

AVR - Programming

● Interrupt 종류

Table 23. Reset and Interrupt Vectors

Vector No.	Program Address ⁽²⁾	Source	Interrupt Definition
1	\$0000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	\$0002	INT0	External Interrupt Request 0
3	\$0004	INT1	External Interrupt Request 1
4	\$0006	INT2	External Interrupt Request 2
5	\$0008	INT3	External Interrupt Request 3
6	\$000A	INT4	External Interrupt Request 4
7	\$000C	INT5	External Interrupt Request 5
8	\$000E	INT6	External Interrupt Request 6
9	\$0010	INT7	External Interrupt Request 7
10	\$0012	TIMER2 COMP	Timer/Counter2 Compare Match
11	\$0014	TIMER2 OVF	Timer/Counter2 Overflow
12	\$0016	TIMER1 CAPT	Timer/Counter1 Capture Event
13	\$0018	TIMER1 COMPA	Timer/Counter1 Compare Match A
14	\$001A	TIMER1 COMPB	Timer/Counter1 Compare Match B
15	\$001C	TIMER1 OVF	Timer/Counter1 Overflow

16	\$001E	TIMER0 COMP	Timer/Counter0 Compare Match
17	\$0020	TIMER0 OVF	Timer/Counter0 Overflow
18	\$0022	SPI, STC	SPI Serial Transfer Complete
19	\$0024	USART0, RX	USART0, Rx Complete
20	\$0026	USART0, UDRE	USART0 Data Register Empty
21	\$0028	USART0, TX	USART0, Tx Complete
22	\$002A	ADC	ADC Conversion Complete
23	\$002C	EE READY	EEPROM Ready
24	\$002E	ANALOG COMP	Analog Comparator
25	\$0030 ⁽³⁾	TIMER1 COMPC	Timer/Counter1 Compare Match C
26	\$0032 ⁽³⁾	TIMER3 CAPT	Timer/Counter3 Capture Event
27	\$0034 ⁽³⁾	TIMER3 COMPA	Timer/Counter3 Compare Match A
28	\$0036 ⁽³⁾	TIMER3 COMPB	Timer/Counter3 Compare Match B
29	\$0038 ⁽³⁾	TIMER3 COMPC	Timer/Counter3 Compare Match C
30	\$003A ⁽³⁾	TIMER3 OVF	Timer/Counter3 Overflow

AVR - Programming

- External Interrupt Register

Bit	7	6	5	4	3	2	1	0	
	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	EIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Bits 7..0 – INT7 – INT0: External Interrupt Request 7 - 0 Enable**

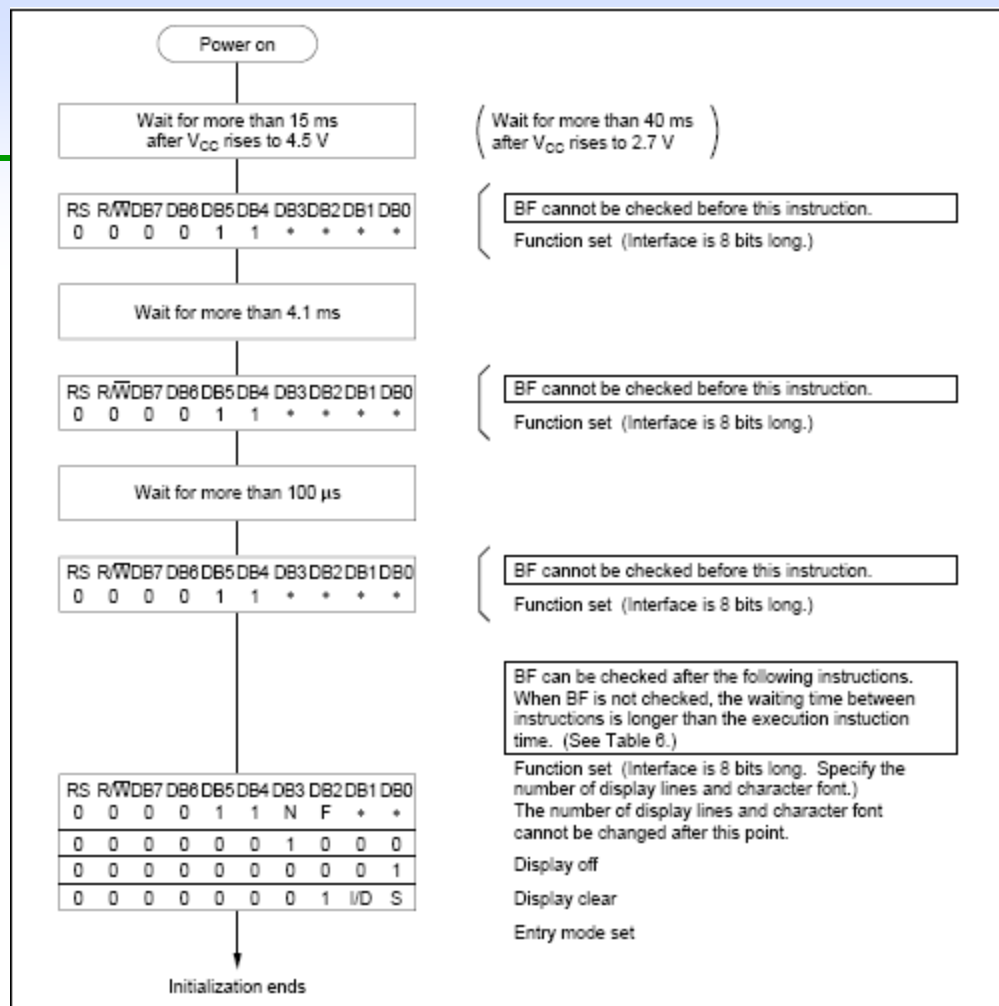
When an INT7 – INT0 bit is written to one and the I-bit in the Status Register (SREG) is set (one), the corresponding external pin interrupt is enabled. The Interrupt Sense Control bits in the External Interrupt Control Registers – EICRA and EICRB – defines whether the external interrupt is activated on rising or falling edge or level sensed. Activity on any of these pins will trigger an interrupt request even if the pin is enabled as an output. This provides a way of generating a software interrupt.

Bit	7	6	5	4	3	2	1	0	
	INTF7	INTF6	INTF5	INTF4	INTF3	INTF2	INTF1	INTF0	EIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Bits 7..0 – INTF7 - INTF0: External Interrupt Flags 7 - 0**

AVR - Programming

● LCD 초기화 및 구동



```

LCD_string(0x80,"Ultra Sonic Ranging :"); // 문자 표시
LCD_command(0xC0+11); // LCD 커서 위치 지정
LCD_4d(UltraSonic( )); // 거리 Data 표시
  
```

AVR - Programming

- 초음파센서를 이용한 거리측정 Algorithm

1 st :

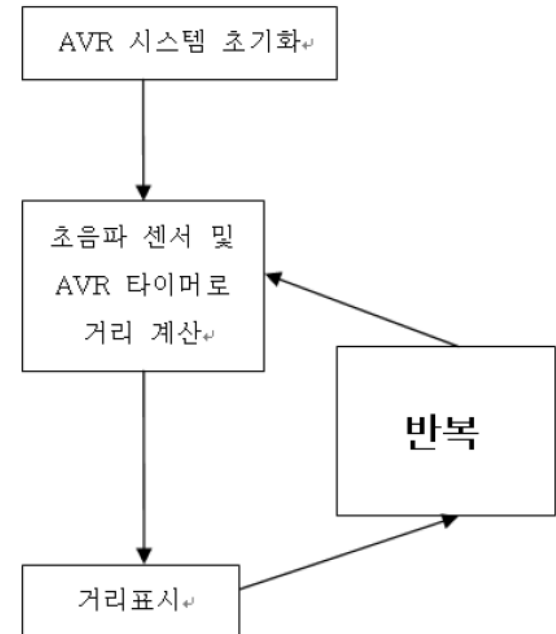
- AVR 시스템 초기화 :
MCU구성(내/외부 Memory) 초기화,
입/출력 초기화
- Timer/Counter 초기화

2nd :

- 초음파센서의 Trigger 입력에
펄스인가(초음파 송출).
- 인터럽터 INT0 입력으로부터 초음파
반사펄스 검출

3rd :

- 초음파 송출에서 부터 검출까지의
시간 측정 후, 물체와의 거리를
계산하고 LCD 및 LED에 나타낸다.





AVR - Programming

● C- Program (1/2)

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include "ATmega128_v20m.H"

unsigned int Echo_count;
unsigned int sum;

ISR(INT0_vect)
{
    Echo_count = TCNT1; //PD0
}
```

```
int UltraSonic(char ch) {
    ;
    return XXXX ;
}
```

```
int main(void) {

    MCU_initialize();

    LCD_initialize();
    LCD_string(0x80,"U-Sonic Test");
    LCD_string(0xC0," SRF - 05 ");
}
```



AVR - Programming

● C- Program (2/2)

```
TCCR1A = 0x00;           // Normal Port Operation
TCCR1B = 0x02;           // CLK/8(16MHz/8) = 2MHz(T=0.5us)
TCCR1C = 0x00;           // Do not use "force output compare"

TCNT1H = 0x00;           // Upper Byte of Timer/Counter 1
TCNT1L = 0x00;           // Lower Byte of Timer/Counter 1

EICRA = 0x0A;            // INT0, INT1: falling edge trigger
EICRB = 0x88;            // falling edge trigger

sei( );
Delay_ms(1000);

while(1) {
    LCD_string(0xC0,"US_Range:   cm");
    LCD_command(0xC0+10);
    LCD_4d(UltraSonic(0));

    if(UltraSonic(0) <= 29 && UltraSonic(0) >= 20)
        PORTB = 0x30;           // LED 4,5 is On
    else if(UltraSonic(0) <= 19 && UltraSonic(0) >= 10)
        PORTB = 0x10;           // LED4 is On
    else if(UltraSonic(0) < 9 )
        PORTB = 0x00;           // All LEDs are Off.

};

} // end of main
```



AVR - Programming

● C- Program

```
int UltraSonic(char ch) {
    int range;

    switch(ch) {
        case 0: PORTE |= 0x10; //triggering Ultrasonic Sensor
                 EIMSK |= 0x01; // PE4, output Trig Pulse
                 break; // Enable the INT0
        /* case 1: PORTE |= 0x20; // PE5
           EIMSK |= 0x02; // INT1, PD1
           break; */
        default : break;
    }
    Delay_us(12);
    PORTE &= ~0x10; // off PE4 of the PORTE
    TCNT1 = 0; // Counter1 is cleared
    EIFR = 0x00; // Clear interrupt Flag Reg.
    Delay_ms(50);
    EIMSK = 0x00; //Disable the all INTx

    Echo_count -= 1300; //set dead time, 650us(4400: SR-04)
    if(Echo_count < 180 ) Echo_count = 180; //3cm, 90us
    if(Echo_count > 36000L ) Echo_count = 36000L; //300cm, 18ms
    range = Echo_count / 116;
    return range;
}
```