

* 본 자료는 서경대학교 아두이노 프로그래밍 수업 수강자를 위해 작성된 강의 교재입니다.

강의교재- 아두이노 프로그래밍

8장 초음파 센서

서경대학교 김진현

8

초음파 센서

내용

- 8.1 초음파란?
- 8.2 초음파 센서 모듈과 인터페이스
- 8.3 2개 단자를 사용하는 센서 모듈 프로그래밍
- 8.4 1개 단자를 사용하는 센서 모듈 프로그래밍(보류)
- 8.5 고찰

“(보류) 📌*” 표기 부분은 강좌 후반부에 검토할 예정입니다.

8.1 초음파란?

인간이 들을 수 있는 주파수는 보통 20Hz~20,000Hz인 것으로 알려져 있다. 초음파(Ultrasonic Wave)는 그림 8.1.1에 보인 바와 같이 사람이 들을 수 있는 가청 주파수 대역을 넘는 영역의 주파수를 가진 파동 신호이다.



그림 8.1.1 초음파의 범위 다이어그램

일부 동물들은 그림 8.1.2와 같이 초음파를 상대의 위치나 거리를 파악하는 수단으로 활용하여 온 것으로 파악되었다.

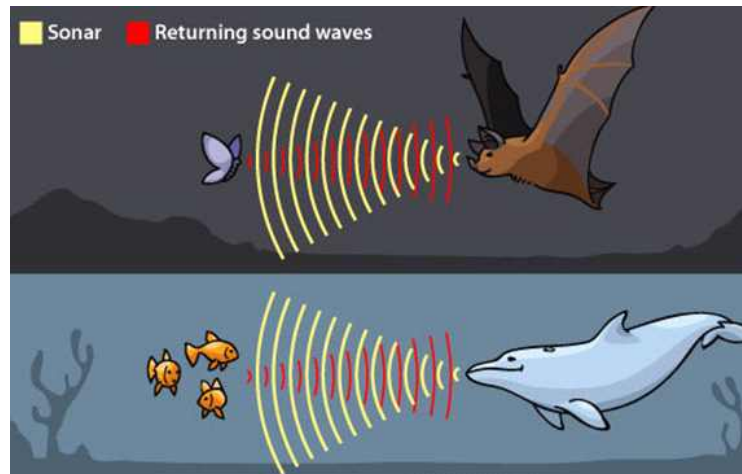


그림 8.1.2 초음파를 사용하는 동물의 사례
(박쥐:100Hz~200kHz, 돌고래:약 170KHz)

이들은 가청 주파수대의 혹은 그 이상의 소리를 발생하여 이 음파가 목표물에 부딪혀 다시 반향되어 돌아오는 시간을 측정하여 목표물과 자신간의

거리를 추정한다¹⁾. 그림 8.1.3에 이러한 원리를 이용하여 초음파센서가 거리를 측정하는 방법을 도시화하였다. 초음파센서 모듈에는 소리를 발생시키는 장치(sender, transmitter)와 물체에 부딪힌 초음파를 감지하는 센서가 내장되어서 발생시간과 감지시간과의 차이(Δt)를 이용하여 거리를 측정한다. 물체까지 도달하는데 소요된 시간은 Δt 이므로 초당 340m 정도인 소리의 속도를 이용하면 거리를 환산할 수 있을 것으로 보인다.

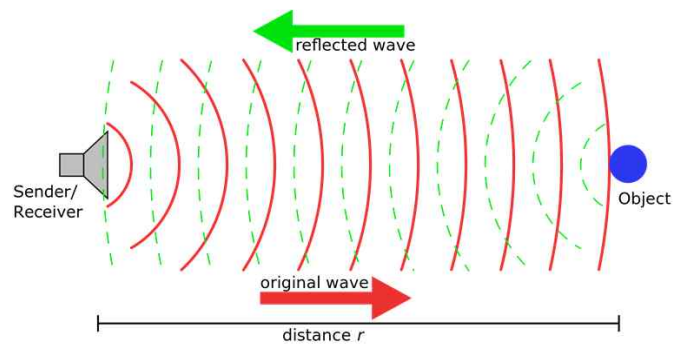


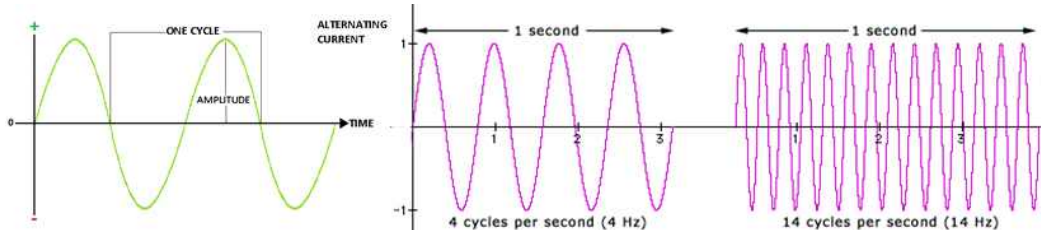
그림 8.1.3 목표의 거리를 탐색하는 원리

☞ 참고 사항 : 주기와 주파수

주기 : 신호가 1회의 진동을 완성하는 시간. 단위 : [초]

주파수 : 초당 1주기의 신호가 반복되는 회수. 단위 : [Hz]

주기(T)와 주파수(f)와의 관계 $f=1/T$



1) 박쥐의 경우는 양 쪽귀의 높낮이가 서로 달라서 이를 이용해 목표물의 3차원 위치를 추론해 내는 것으로 밝혀졌다.

초음파의 특징을 살펴보면 다음과 같다.

- 주파수가 높고, 파장이 짧으며 직진성이 높다.
- 전파 속도가 일정하다. (공기 : 340m/s, 고체 : 5,000m/s 이상)
- 투과 가능 물체가 다양하다(액체, 고체, 투명체, 불투명체)
- 액체와 고체의 경계면에서 반사, 굴절, 회절 성질을 가진다.

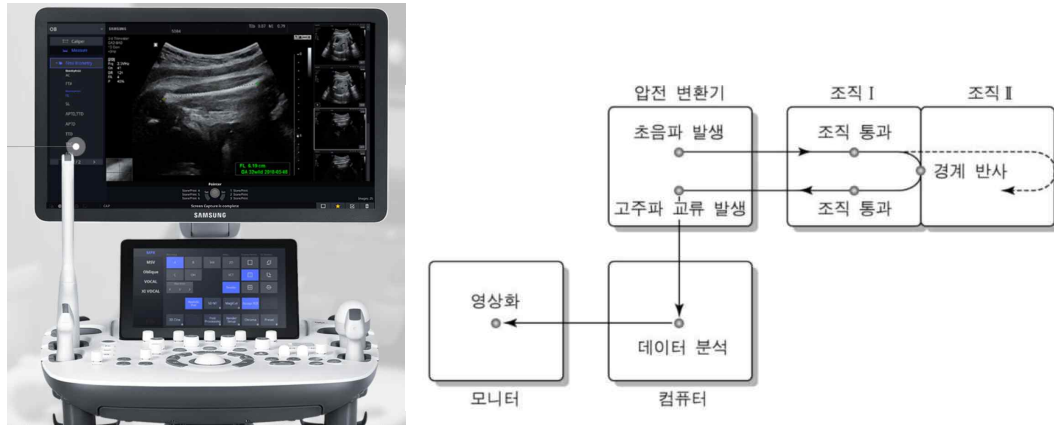
초음파는 다음과 같은 용도로 활용되고 있다.

- 속도 측정 - 초음파 유속계(流速計), 초음파 유량계, 초음파 도플러 혈류계, 초음파 도플러 유속계
- 거리 측정 - 초음파 거리계, 초음파식 근접각 센서, 초음파 레벨 센서, 초음파식 수위계, 초음파식 적설계, 초음파식 파고계, 자동차 후방 경고



BWS, Back Warning buzzer System

- 농도 측정 - 초음파 점성도계, 초음파 탁도계
- 기타 측정 - 초음파 탐상자(해저 탐사), 초음파 두께 측정, 초음파 현미경, 대 잠수함 추적, 초음파 진단장치, 초음파 CT 스캐너



초음파 영상 진단 시스템

초음파는 다음과 같은 물체의 특징에 따라 감지하기 어려울 수도 있다.

- 초음파가 반사되기 어려운 물체 - 철사 줄, 로프, 체인, 가느다란 돌출봉
- 초음파를 흡수하는 물체 - 솜, 스펀지, 섬유, 눈 등 (예: 눈에 덮여 있는 소화전)
- 난반사를 일으키는 물체 - 자갈, 모래, 거친 콘크리트

초음파 센서가 오동작을 유발할 수 있는 경우는 다음과 같다.

- 경사진 벽면에 접근할 때 - 초음파가 굴절하여 오동작
- 좁은 통로, 사방이 막혀 있는 주차장의 경우 - 여러 벽면에서 돌아오는 초음파 반사로 측정 실패

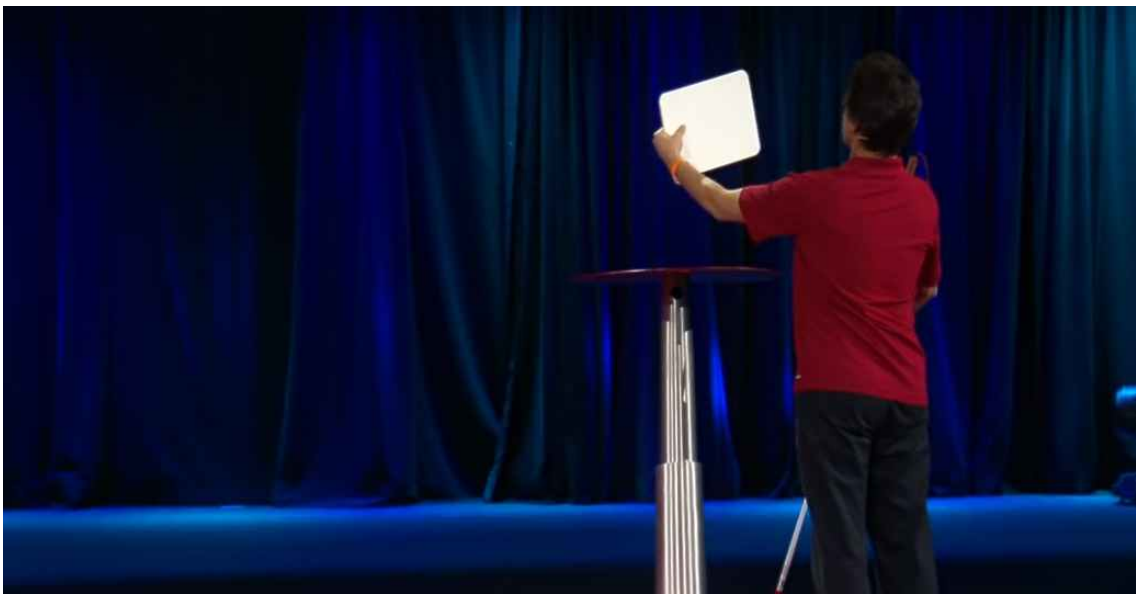
참고- 사람도 음파로 장애물을 탐지하는 사례가 언론에 노출되고 있다.

Human Echolocation

출처 : https://en.wikipedia.org/wiki/Human_echolocation

Human echolocation is the ability of humans to detect objects in their environment by sensing echoes from those objects, by actively creating sounds, for example, by tapping their canes, lightly stomping their foot, snapping their fingers, or making clicking noises with their mouths.

People trained to orient by echolocation can interpret the sound waves reflected by nearby objects, accurately identifying their location and size.



<https://www.youtube.com/watch?v=uH0aihGWB8U>

8.2 초음파 센서 모듈과 인터페이스

초음파 센서는 외부에 노출된 2개의 구성부를 가지고 있다. 하나는 초음파를 발생하여 외부로 전송하는 송신자(transmitter), 다른 하나는 물체에 부딪혀 오는 초음파를 검출하는 수신자(receiver)이다. 그림 8.2.1에 이와 같은 구성과 그 동작원리를 그림으로 표현하였다.

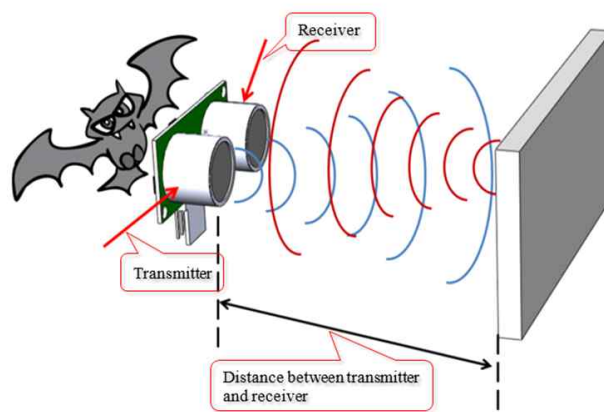


그림 8.2.1 초음파 센서의 동작 원리

아두이노 실험 키트에서는 그림 8.2.2에 보인 바와 같이 2종의 초음파 센서 모듈이 사용되고 있다.

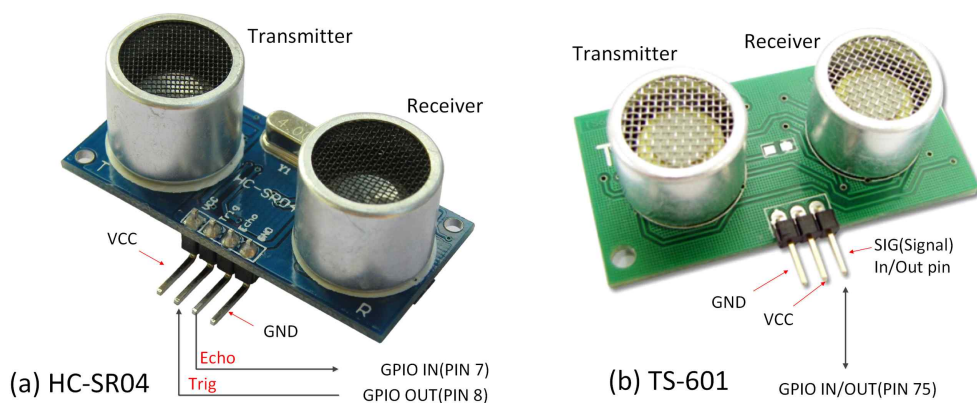


그림 8.2.2 초음파 센서 2종의 실물사진 및 신호선

그림 (a)에서는 Trigger Input 단자는 초음파를 발생하게 만드는 역할을 하며 Echo Input 단자는 반향된 초음파가 도착했음을 알리는 역할을 담당한다.

반면, 그림 (b)는 전원(Vcc, GND)외에 단 1개의 신호선을 사용한다. 본 모듈에서는 초음파 송출 제어 신호와 반향된 초음파를 감지했음을 알리는 상태 신호선의 역할을 1개의 신호선(SIG)가 담당한다²⁾. 한 단자를 동작의 절차에 따라 GPIO 출력 혹은 GPIO 입력으로 설정하여 이러한 동작을 수행한다.

2) 실험에 의하면 1개의 신호 단자를 사용하는 센서 모듈에서는 데이터 시트에 기술된 사양과는 다른 오동작이 많이 발견되었다.

8.3 2개 단자를 사용하는 센서 모듈 프로그래밍

본 사례에서 사용하는 센서를 그림 8.3.1에 보였다. 그림에서 **Trig** 단자는 센서 입장으로는 입력으로 기술되어 있지만 아두이노에서는 초음파를 방출할 것을 지시하는 출력단자이다. 또한 **Echo** 단자는 센서 입장으로는 출력이지만 아두이노에서는 입력으로 설정하여 물체의 검출을 감지해야 한다.

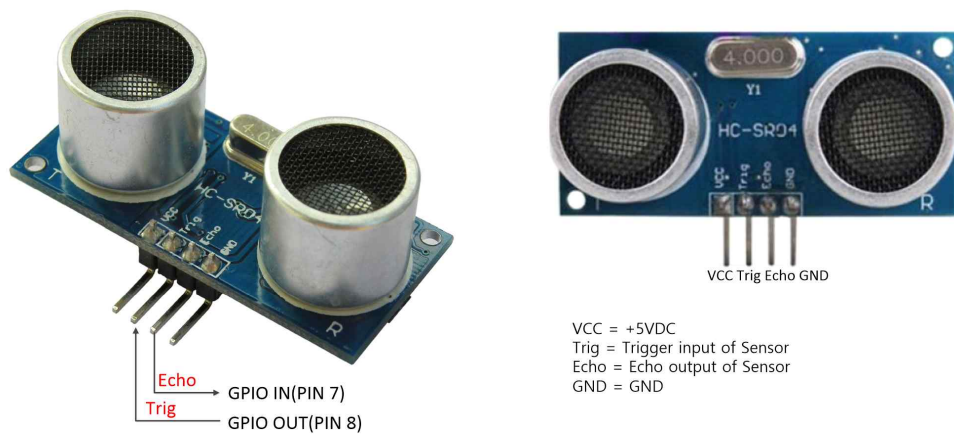


그림 8.3.1 2개의 입출력 신호 단자를 사용하는 초음파 센서 모듈

2개 제조사의 센서의 사양을 표 8.3.1에 보였다.

표 8.3.1 초음파 센서 모듈의 사양

from www.Electronics.com		from datasheet of Cytron Technologies				
Working Voltage	DC 5 V	Parameter	Min	Typ.	Max	Unit
Working Current	15mA	Operating Voltage	4.50	5.0	5.5	V
Working Frequency	40KHz	Quiescent Current	1.5	2	2.5	mA
Max Range	4m	Working Current	10	15	20	mA
Min Range	2cm	Ultrasonic Frequency	-	40	-	kHz
Measuring Angle	15 degree	Effectual Angle: <15° Ranging Distance : 2cm – 400 cm Resolution : 0.3 cm Measuring Angle: 30 degree Trigger Input Pulse width: 10uS Dimension: 45mm x 20mm x 15mm				
Trigger Input Signal	10uS TTL pulse					
Echo Output Signal	Input TTL level signal					
Dimension	45*20*15mm					

그림 8.3.2에 본 센서의 내부 동작을 보여주는 그림을 보였다.

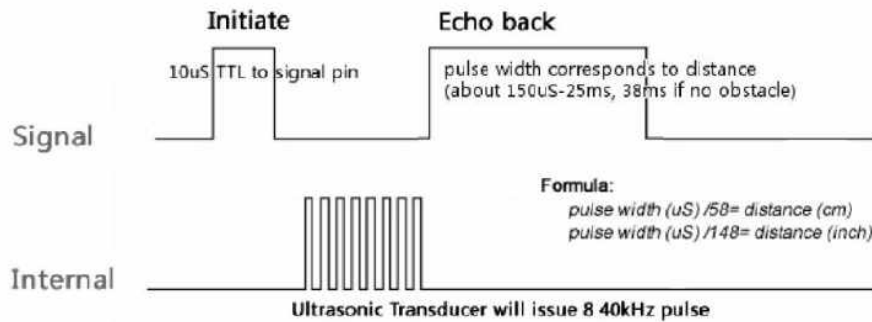


그림 8.3.2 센서의 내부 동작

이 그림과 연동하여 내부 동작을 **프로그래머가 해야 할 일**과 **센서가 내부적으로 하는 일**을 단계적으로 기술하면 다음과 같다.

① **프로그래머** : Trig 단자에 10uS의 펄스(pulse) 신호를 인가하여 센서에 초음파 발생을 지시한다. 이 동작은 Trig 단자를 H로 만든 후 10uS 동안 유지한 후 L로 만들어 이루어진다.

♦TRIG 단자를 OUTPUT 모드로 설정. ECHO 단자는 INPUT로 설정

♦10uS 폭의 pulse를 TRIG 단자를 통해 출력한다.

② **센서 모듈** : 이 입력을 받으면 센서 모듈은 스스로 8개의 40KHz 펄스로 구성된 초음파를 송신단을 통해 발사한다.

③ **센서 모듈** : 장애물에 반사된 초음파가 되돌아오면 H 신호를 센서의 Echo 단자를 통해 출력한다. 거리에 따라 150uS~25mS 폭을 가진 펄스 신호가 출력된다. H 펄스의 폭이 초음파가 발생한 이후 반향될 때까지의 시간을 나타낸 것이다. 만약 반향된 초음파가 없다면 센서는 38mS의 펄스를 출력한다.

④ **프로그래머** : Echo 단자로 유입된 펄스의 H 부분이 유지된 시간을 측정한다. 이 길이는 초음파 신호가 발사되어 검출물까지 갔다 돌아온 왕복 주행 시간에 해당한다. 따라서 실제 거리를 계산할 때는 아래와 같이 펄스의

폭을 1/2로 만들어 사용해야 한다³⁾.

$$\begin{aligned}(1) \text{ distance[m]} &= (\text{high_width_time[s]} \times \text{velocity_of_sound[m/s]}) / 2 \\(2) \text{ distance[m]} &= (\text{high_width_time[us]} \times \text{velocity_of_sound[m/s]}) / (2 * 10^6) \\(3) \text{ distance[cm]} &= 10^2 * (\text{high_width_time[us]} \times \text{velocity_of_sound[m/s]}) / (2 * 10^6) \\(4) \text{ distance[cm]} &= (\text{high_width_time[us]} \times \text{velocity_of_sound[cm/us]}) / 2\end{aligned}$$

1) 아두이노는 펄스의 시간 폭을 측정하기 위해 함수, `pulseIn()`을 제공하고 있다. 이를 이용하면 편리하게 시간을 잴 수 있다.

`pulseIn(단자번호, 상태)` 함수는 **단자**의 신호가 **상태**를 유지하는 시간[us]을 측정하여 반환한다.

♦ 활용 예: `a = pulseIn(7, HIGH)`

이 루틴은 단자 7번의 입력이 H가 될 때까지 기다리고 이것이 다시 LOW가 될 때 까지 기다렸다가 HIGH가 유지된 시간을 변수 a에 반환하는 기능을 수행한다.

2) 소리는 속도는 온도에 따라 크게 영향 받는다. 그 관계 식은 다음과 같다. 공식에서 온도는 섭씨 단위[°C]를 사용한 것이다.

$$\begin{aligned}\text{velocity_of_sound[m/s]} &= (331.5 + 0.6 * \text{Temperature}) \\ \text{velocity_of_sound[m/us]} &= (331.5 + 0.6 * \text{Temperature}) / 10^6 \\ \text{velocity_of_sound[cm/us]} &= (331.5 + 0.6 * \text{Temperature}) * 10^2 / 10^6 \\ \text{velocity_of_sound[cm/us]} &= (331.5 + 0.6 * \text{Temperature}) / 10^4\end{aligned}$$

3) 4개의 모든 식이 같은 의미를 갖는다. 단, `pulseIn()` 함수의 반환 값이 [us] 단위이고, 실제 계측이 [cm] 단위의 정밀도 수준에서 이루어지므로 식 (3) 혹은 더 간편함을 위해 (4)를 활용하기로 한다.

unsigned long pulseIn (int pinNum, Value); // Arduino 표준 함수		
unsigned long pulseIn (int pinNum, Value, timeout); // Arduino 표준 함수		
기능	Value에서 제시한 신호의 유지 시간을 측정하여 반환한다. Value가 HIGH이었다면 이 함수는 pinNum 단자가 HIGH가 될 때까지 기다렸다가 그때부터 시간을 측정하기 시작한다. 이후 해당 단자가 LOW가 될 때까지 기다렸다가 HIGH 펄스가 유지된 시간을 반환한다.	
매개변수	int pinNum	아두이노와 연결된 초음파 센서 핀 번호, 75번
	int Value	HIGH 혹은 LOW
	int timeout	Value에서 지정한 펄스가 발생할 때까지의 대기 시간. us 단위. default 1초.
리턴 값	unsigned long	us 단위의 유지시간. timeout 시간동안 Value 상태가 되지 않으면 0이 반환된다.

그림 8.3.3은 아두이노를 비롯한 MCU와 센서 모듈간의 인터페이스의 신호와 그 흐름을 일반화하여 요약해 보인 것이다.

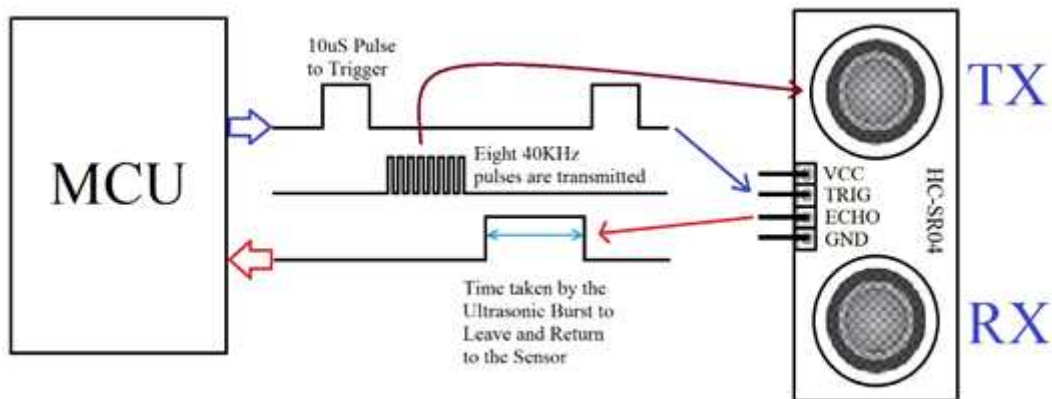


그림 8.3.3 센서와 마이크로컨트롤러 간의 동작 흐름도

□ 예제 8.3.1 : 초음파 센서를 이용하여 물체와의 거리를 출력한다.

UltraSonic1.ino : 아두이노 함수, pulseIn()

```

01 #define echoPin 7 // Echo Pin
02 #define trigPin 8 // Trigger Pin
03 #define Temperature 27 // 섭씨 온도
04 int maximumRange = 400; // Maximum range needed
05 int minimumRange = 2; // Minimum range needed
06 long duration, distance;
07 double sSpeed_m_s, sSpeed_cm_us;
08 void setup() {
09     Serial.begin (9600);
10     pinMode(trigPin, OUTPUT); pinMode(echoPin, INPUT);
11     sSpeed_m_s = 331.5 + 0.6*Temperature; // 소리의 이동 속도 [m/sec]
12     sSpeed_cm_us = (331.5 + 0.6*Temperature) * pow(10, 2) / pow(10, 6); // 이동 속도 [cm/us]
13     Serial.print("sSpeed_m_s[m/sec] ="); Serial.println(sSpeed_m_s);
14     Serial.print("sSpeed_cm_us[cm/us] ="); Serial.println(sSpeed_cm_us);
15 }
16 void loop() {
17     int range;
18     digitalWrite(trigPin, LOW); // 시작하기 전에 일단 trig 신호를 LOW로 만들어 오동작 방지.
19     delayMicroseconds(2); // 약간의 지연시간을 제공.
20     digitalWrite(trigPin, HIGH); // 10 micro sec. High Pulse 생성
21     delayMicroseconds(10);
22     digitalWrite(trigPin, LOW); // Low가 됨과 동시에 모듈은 40khz의 pulse 8개를 만들어 출력함.
23     duration = pulseIn(echoPin, HIGH); // 초음파가 발사+반사되어 돌아오는 시간(us)을 반환받는다.
24     //distance = pow(10, 2) * duration * sSpeed_m_s / (2 * pow(10, 6)); // 식 (3)
25     distance = duration/2 * sSpeed_cm_us; // 식 (4)
26     if (distance > maximumRange || distance < minimumRange) {
27         Serial.println("out of range"); delay(1000);}
28     else {
29         Serial.print("distance[cm] : ");
30         Serial.println(distance);
31         delay(1000);
32     }
32 }

```

8.4 1개 단자를 사용하는 센서 모듈 프로그래밍

=> 이 모듈은 오류를 많이 유발하여 본 강의에서는 동작 및 예제 프로그램의 분석은 제외하기로 한다.

본 모듈을 제어하기 위해서 아두이노는 센서모듈의 SIG 단자와 연결된 1개 단자(75번)를 입출력 겸용으로 사용한다.

▶ 초음파 센서 연결 방식

연결 방식	- GPIO 방식
연결 핀	- 75번 핀(Input/Output 공용)
동작 모드	- 1개의 핀을 사용해서 Digital 신호 출력 및 입력

그림 8.4.1의 초음파 센서 모듈의 타이밍도와 표 8.4.1의 타이밍 파라미터를 통해 제어 방법을 설명하기로 한다.

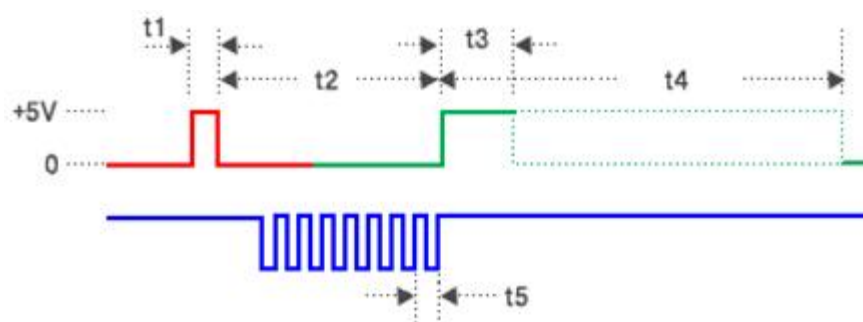







그림 8.4.1 초음파 센서 모듈의 동작 타이밍 다이어그램

표 8.4.1 초음파 센서 모듈의 타이밍 파라미터

	MCU I/O	PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNIT
	Input	Input trigger pulse	t1	-	5	-	μs
	Output	Echo postpone	t2	500	-	520	μs
		Output echo pulse MIN.	t3	110	-	140	μs
		Output echo pulse MAX.	t4	1.90	-	19.0	ms
	-	Burst pulse cycle	t5	-	25	-	μs

먼저 초음파를 발사하기 위해서는 H(로직 1) 신호를 SIG 단자에 t1 시간만큼 센서 모듈에 공급한다. 센서 모듈은 이 신호에 의해 내부에서 발진 회로가 작동하여 40kHz의 burst 펄스를 트랜스미터를 통해 발생한다. 이 동작에 t2의 시간 소요가 필요하다. t2시간이 종료되면 초음파 모듈은 SIG 단자를 통해 H를 출력하기 시작한다. 이후 물체에 초음파가 도달한 후 다시 돌아오면 SIG 단자를 L로 떨어뜨린다. 아두이노에서는 H에서 L로 되기까지 H로 유지된 시간 폭(t3+t4)을 재어서 음속을 고려하여 왕복 거리를 환산한다. **이후 재측정을 위해서는 t1 trigger pulse를 다시 보내기 전 200us를 기다려야 한다.**

프로그램 관점에서 1개 단자를 사용하는 초음파 센서의 동작의 흐름을 프로그래머와 센서의 입장에서 정리해 보면 아래와 같다.

- ① **프로그래머** - 아두이노에서 초음파 센서와 연결된 SIG 핀(75번 핀)을 OUTPUT 모드로 설정하고 Input trigger pulse(t1)를 보낸다. t1(5us) 구간의 H 펄스를 유지한 후 L로 만든다.
- ② **초음파 센서** - Input trigger pulse를 받은 초음파 센서는 자체적으로 40kHz의 초음파를 발생시킨다(t2).
- ③ **프로그래머** - SIG 핀(75번 핀)을 INPUT 모드로 설정하고 pulseIn() 함수를 이용하여 HIGH 신호가 SIG 핀(75번 핀)에 유입되는지를 기다렸다가 이때부터 시간측정을 시작한다. pulseIn() 함수는 유입되는 HIGH 펄스의 시간

폭을 측정하는 기능을 수행한다.

- ④ 초음파 센서 - SIG 핀(75번 핀)이 HIGH가 된 후 초음파가 물체에 반사되어 돌아오면 센서는 SIG 핀(75번 핀)의 입력을 스스로 LOW로 바꾸어 출력한다.
- ⑤ 프로그래머 - SIG 핀(75번 핀)의 상태 값이 HIGH -> LOW로 바뀔 때까지의 시간(t_3+t_4)을 측정하여 물체와의 거리를 계산한다.

□ 예제 8.4.1 : 초음파 센서를 이용하여 물체와의 거리를 출력한다.

UltraSonic1pin.ino : 아두이노 표준 함수 - pulseIn()

```

01 // 본 예제는 정확한 측정이 잘 되지 않는 것으로 판단된다.
02 // 실험에 의하면 물체가 고정되어도 값이 자꾸 변하는 문제가 있다.
03 #define Temperature 25
04 int SIG=75; // 초음파 센서에 연결된 아두이노 단자 번호
05 void setup( ) {
06     Serial.begin(9600); // Serial 통신 초기화
07 }
08 void loop( ) {
09     unsigned long duration=0; // Pulse 시간 값을 저장할 변수
10     int disMm=0; // 거리를 계산하여 저장할 변수
11     // trigger pulse를 출력하여 센서가 초음파를 출력하게 만든다.
12     pinMode(SIG, OUTPUT); // pulse 보내기 위해 모드 설정
13     digitalWrite(SIG, LOW); // LOW로 pulse 초기화
14     // delay 시간이 부족하면 오동작하는 경향이 있음
15     delayMicroseconds(100); // 200us. 200*10^6초
16     digitalWrite(SIG, HIGH); // HIGH로 pulse 시작
17     delayMicroseconds(5); // 정격 5 microseconds 유지
18     digitalWrite(SIG, LOW); // LOW로 pulse 마침
19     // 입력 모드로 바꾸어 H 펄스폭을 측정한다.
20     pinMode(SIG, INPUT); // 신호를 받기 위해 입력 모드 설정
21     // pulse 출력이 어느 정도 나갈 때까지 기다린다. 스펙이 제시된 바 없음.
22     delayMicroseconds(200);
23     // HIGH 값을 출력하여 SIG 신호가 LOW될 때까지의 시간을 읽어옴
24     duration = pulseIn(SIG, HIGH); // microsec 단위의 값을 반환함
25     Serial.println(duration);
26     // duration 값으로 mm 단위의 거리를 계산. 음속=340m/s
27     disMm = (331.5 + 0.6*Temperature) * ((float)duration/1000000/2) * 1000;
28     Serial.print(disMm); // 거리 값을 Serial로 출력
29     Serial.println(" mm");
30     //delay(1000); // 1초간 대기
31 }

```

8.5 고찰

□ 암기/사고/분석/조사 문제

다음 주제는 본문에서 이미 기술된 것도 있고 여러분이 새롭게 조사해야 할 내용도 있습니다. 검토해 보면서 그동안 습득한 지식을 정리해 보도록 하겠습니다.

1. 초음파 센서를 통해 거리를 측정하는 원리를 설명하시오. 또한, 성공적인 동작을 위해 주의해야 할 사항을 기술하시오.
2. 본 실험에서 사용하는 초음파 모듈의 주파수는 얼마인가?
3. 본 실험에서 사용하는 초음파센서 모듈의 원통형 소자 2개가 하는 역할은 무엇인가?

□ 설계 및 실험 미션

4. 스피커를 연결하여 센서 앞의 장애물(예를 들어 노트)을 움직여 연주(예를 들어 학교종, 비행기)하는 프로그램을 만들고 이를 시연하는 동영상을 제작하시오. 가급적 올바른 연주가 되기 위해 프로그램을 경험적인 지식을 바탕으로 튜닝하기 바라며, 스스로 취득한 필요한 경험을 보고서에 기술하기 바랍니다. 멋진 연주가 되는 것도 평가에 반영합니다. 특정 범위(예를 들어 10cm~40cm)를 넘어가면 스피커에 대해 noTone()을 수행하여 소리를 나지 않게 합니다⁴⁾.

4) 그래야 조용히 실험할 수 있을 것으로 보입니다.

5. 장애물까지의 거리를 mm 단위로 반환하는 함수를 다음과 같이 제작하고자 한다.

1) digitalWrite(), digitalRead(), pulseIn() 함수 등의 표준 함수를 사용하여 위 조건을 만족하는 함수, Read_mm_Dist(echoPin, trigPin, Temperature)를 제작한다.

2) 다른 표준함수는 사용하지만, pulseIn() 표준함수를 사용하지 않고 동일한 동작을 하는 자체 MyPulseIn() 함수를 설계하고 이를 기반으로 제시된 함수 Read_mm_Dist2()를 설계한다.

3) Read_mm_Dist()와 Read_mm_Dist2()의 거리 측정 오차가 어느 정도(%)인지 실험을 통해 비교하시오. 다음 관점에서 분석하기 바랍니다.

- ① 10회 이상의 평균을 이용하여 산출
- ② micros() 함수의 평균오차와 비교

힌트 : millis() 혹은 micros() 함수를 활용하여 자체 함수 제작
이 함수들은 호출할 때마다 자체 시간을 ms, 혹은 us 단위로 반환한다. 해당 예제는 [링크](#)를 참조바랍니다. 어떤 신호의 H 유지시간을 측정하고자 할 때 H가 된 것을 보았을 때 micros() 함수를 호출하였을 때 반환 값을 start 변수에 기록하고, 그 신호가 다시 L로 떨어지는 것을 관찰했을 때 micros() 함수를 수행했을 때의 반환 값을 end라고 한다면, H 펄스가 유지되었던 시간 $t = \text{end} - \text{start}[\text{us}]$ 가 될 것입니다.

6. 20cm 이하 거리에 근접하면 점점 고주파의 음향을 울리는 프로그램을 작성하시오.

7. 스피커를 연결하여 거리가 가까워지면 고주파를 발생하는 프로그램을 작성하라. 이를 이용하여 연주를 시연하시오.

가능한 연속적인 음을 출력하면 유리하다.
사례 : 10cm~50cm까지 거리에 따라 2000Hz~100Hz의 음정을 출력한다. 범위를 벗어나면 소리 출력을 중지한다.

8. pulseIn()과 같은 기능을 하는 함수를 인터럽트를 이용하여 구현하시오.

14장의 인터럽트 학습 후 재검토 요망 🚨!