

## หน่วยที่ 12

### Arduino กับลำโพงบีซเซอร์

#### สาระการเรียนรู้

1. ลำโพงบีซเซอร์ (Buzzer)
2. การสร้างความถี่เสียงของ Arduino
3. การสร้างความถี่เสียงโน้ตดนตรีของ Arduino

#### จุดประสงค์การเรียนรู้

1. เพื่อให้มีความรู้ความเข้าใจเกี่ยวกับการต่อใช้งาน Arduino กับลำโพงบีซเซอร์
2. เพื่อให้มีทักษะในการเขียนโปรแกรม Arduino กับลำโพงบีซเซอร์
3. เพื่อให้มีทัศนคติในการใช้วัสดุอุปกรณ์บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

#### จุดประสงค์เชิงพฤติกรรม

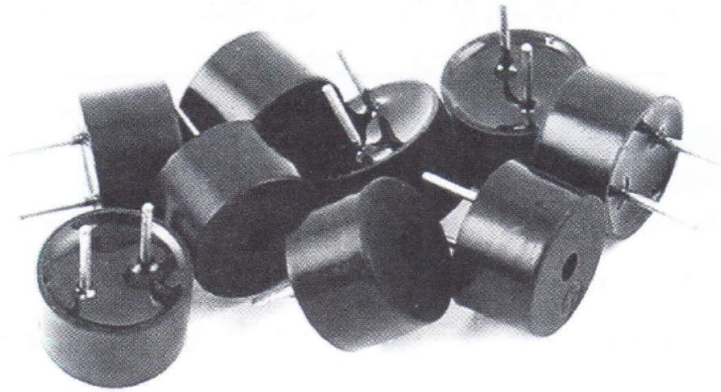
1. สามารถอธิบายการต่อใช้งาน Arduino กับลำโพงบีซเซอร์
2. สามารถเขียนโปรแกรม Arduino ควบคุมลำโพงบีซเซอร์
3. เตรียมความพร้อมด้านวัสดุ อุปกรณ์สอดคล้องกับงานได้อย่างถูกต้อง

#### ลำโพงบีซเซอร์ (Buzzer)

ลำโพงบีซเซอร์ เป็นอุปกรณ์ที่ให้กำเนิดเสียงทำหน้าที่เปลี่ยนสัญญาณไฟฟ้าให้อยู่ในรูปสัญญาณ เสียง ลำโพงบีซเซอร์มีอยู่ 2 ประเภท ได้แก่

1. แบบแอคทีฟ (Active Buzzer) ลำโพงชนิดนี้มีวงจรกำเนิดความถี่อยู่ภายในสามารถสร้างสัญญาณเสียงเตือนได้ทันทีเพียงแค่จ่ายแรงดันไฟฟ้าเข้าไป
2. แบบพาสซีฟ (Passive Buzzer) ลำโพงชนิดนี้ทำงานเหมือนลำโพงขนาดเล็ก คือ ถ้าป้อน แรงดันไฟฟ้า กระแสตรงเข้าไปไม่มีเสียง ถ้าต้องการให้มีสัญญาณเสียงต้องทำการป้อนสัญญาณความถี่ เข้าไปลำโพงชนิดนี้ สามารถกำเนิดเสียงที่มีความแตกต่างกันตามความถี่ที่ป้อนเข้ามา

ในหน่วยนี้เป็นการใช้งาน Arduino กับลำโพงบีซเซอร์แบบพาสซีฟ ดังนั้นการใช้งานต้องทำการ เขียนโปรแกรมเพื่อส่งความถี่จาก Arduino เข้าไปยังลำโพงบีซเซอร์ ปัจจุบันเครื่องใช้ไฟฟ้าโดยทั่วไป มีลำโพงบีซเซอร์ ติดตั้งอยู่ภายในด้วย เช่น ในคอมพิวเตอร์ใช้ลำโพงบีซเซอร์เพื่อส่งสัญญาณให้ทราบ ว่าสถานะของคอมพิวเตอร์มี ปัญหาอะไร หรือในเครื่องซักผ้าอัตโนมัติ เครื่องปรับอากาศ เมื่อทำการ กดปุ่มบนเครื่องหรือรีโมตคอนโทรลจะได้ยินเสียงปิ่นดังขึ้นมาด้วยเพื่อบอกให้รู้ว่าขณะนี้ทำการกด สวิตช์แล้ว



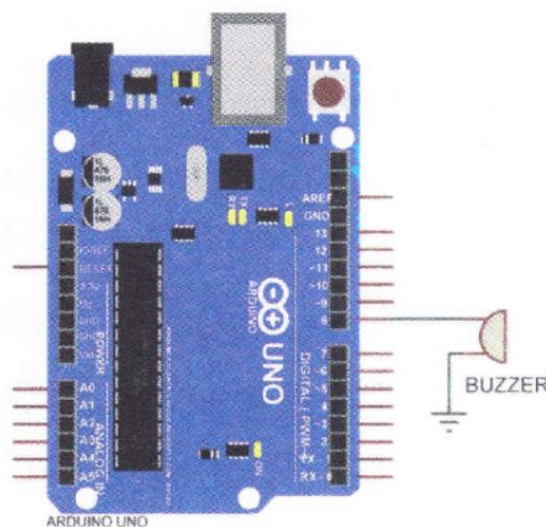
รูปที่ 12.1 ลำโพงบัสเซอร์แบบพาสซีฟ (ที่มา : <https://picclick.com/9mm-Dio-Electronic-Alarm-PCB-Panel-MountingPiez0-231744629770.htm>)

### การสร้างความถี่เสียงของ Arduino

การต่อใช้งานระหว่าง Arduino กับลำโพงบัสเซอร์สามารถต่อได้โดยตรงจากพอร์ตดิจิทัลขา อีกด้านหนึ่งของลำโพงต่อลงกราวด์ การสร้างความถี่เสียงสามารถทำได้โดยง่ายเพียงนำโปรแกรมไฟ กะพริบมาประยุกต์งานด้วยการส่งสัญญาณลอจิก “1” และลอจิก “0” ให้อยู่ในช่วงความถี่เสียงที่ มนุษย์ได้ยินซึ่งอยู่ในช่วงความถี่ 20 - 20,000 เฮิรตซ์ (Hz.)

สูตรการคำนวณหาค่าการหน่วงเวลา

$$\text{delaytime} = \frac{1\text{Second}}{2 \times \text{toneFrequency}} \quad (1 \text{ second} = 1,000,000 \text{ microsecond})$$



รูปที่ 12.2 การต่อ Arduino กับลำโพงบัสเซอร์

ตัวอย่างที่ 1 จากวงจรรูปที่ 12.2 เป็นการต่อใช้งาน Arduino กับลำโพงบัสเซอร์ เพื่อให้ผลิตความถี่ เสียงมีค่าเท่ากับ 1 กิโลเฮิร์ตซ์ (KHz.)

วิธีการคำนวณ

$$\text{delaytime} = \frac{1\text{Second}}{2 \times \text{toneFrequency}} \quad (1 \text{ second} = 1,000,000 \text{ microsecond})$$

$$\text{แทนค่า delaytime} = \frac{1,000,000}{2 \times 1,000} = 500 \text{ microsecond}$$

ดังนั้น ถ้าต้องการให้ Arduino ผลิตความถี่เสียงมีค่าเท่ากับ 1 กิโลเฮิร์ตซ์ (KHz.) ออกทาง ลำโพงบัสเซอร์ ต้องส่งค่าหน่วยเวลาในการสัญญาณลอจิก “1” และลอจิก “0” เท่ากับ 500 ไมโครวินาที (microsecond)

ตัวอย่างโปรแกรม

```
const int SpeakerPin = 8; // กำหนดตัวแปร SpeakerPin ต่อที่พอร์ตดิจิตอลขา 8
void setup() {
    pinMode(SpeakerPin, OUTPUT); // เซตพอร์ตดิจิตอลขา 8 ให้เป็นพอร์ตเอาต์พุต
}
void loop() {
    digitalWrite(SpeakerPin, HIGH); // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก “1”
    delayMicroseconds(500); // หน่วงเวลา 500 ไมโครวินาที
    digitalWrite(SpeakerPin, LOW); // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก “0”
    delayMicroseconds(500); // หน่วงเวลา 500 ไมโครวินาที
}
```

### การสร้างความถี่เสียงโน้ตดนตรีของ Arduino

Arduino มีความสามารถในการผลิตความถี่ เสียงสูง เสียงต่ำ หรือผลิตเป็นเสียงโน้ตดนตรีได้ นั่นเอง โดยความถี่เสียงโน้ตดนตรีดูได้ในตารางที่ 12.1 วิธีการคำนวณหาค่าการหน่วงเวลาใช้สมการเดียวกับ ตัวอย่างที่ 1

ตารางที่ 12.1 ความถี่เสียงโน้ตดนตรี

โน้ต	ความถี่
C	262
D	294
E	330
F	349
G	392
A	440
B	494
C	524

ตัวอย่างที่ 2 ถ้าต้องการหาความถี่เสียงโน้ต E ที่ความถี่ 330 เฮิรตซ์ต่อวินาที สามารถคำนวณ คาบเวลาได้ดังนี้  
วิธีการคำนวณ

$$\text{delaytime} = \frac{1\text{Second}}{2 \times \text{toneFrequency}} \quad (1 \text{ second} = 1,000,000 \text{ microsecond})$$

$$\text{แทนค่า delaytime} = \frac{1,000,000}{2 \times 330} = 1,515.15 \text{ microsecond}$$

ดังนั้น ถ้าต้องการให้ Arduino ผลิตความถี่เสียงโน้ต E ที่ความถี่ 330 เฮิรตซ์ต่อวินาที ออกทางลำโพงบัซเซอร์ ต้องส่งค่าหน่วยเวลาในการสัญญาณลอจิก “1” และลอจิก “0” เท่ากับ 1,515 ไมโครวินาที (microsecond) โดยนำค่านี้ไปใส่ในโปรแกรมตัวอย่างข้างบนเปลี่ยนตัวเลข 500 เป็น 1,515 แทน

```
delayMicroseconds(1515);
```

นอกจากนี้ยังคำสั่งในการผลิตความถี่เสียง เพื่อนำมาสร้างตัวโน้ตหรือเสียงเพลงได้ ซึ่งมีคำสั่ง อยู่ 2 คำสั่ง  
ดังนี้

## คำสั่ง tone();

เป็นคำสั่งในการสร้างลูกคลื่นสี่เหลี่ยมด้วยความถี่ที่กำหนดออกไปยังพอร์ตดิจิทัลที่ต้องการ ตามระยะเวลาที่กำหนด

### รูปแบบคำสั่ง

tone(ตำแหน่งพอร์ต, ความถี่, ระยะเวลา)

หรือ

tone(ตำแหน่งพอร์ต, ความถี่)

- ความถี่ คือ ความถี่ของเสียงมีค่าตั้งแต่ 21 - 65,535 เฮิรตซ์
- ระยะเวลา คือ ระยะเวลาของเสียง มีหน่วยเป็น มิลลิวินาที

ตัวอย่างเช่น ถ้าต้องการส่งความถี่ไปออกที่พอร์ตดิจิทัลขา 8 มีความถี่เสียง 494 เฮิรตซ์ มีระยะเวลา 500 ของบอร์ด Arduino ต้องใช้คำสั่ง

```
tone(8, 494, 500);
```

## คำสั่ง noTone();

เป็นคำสั่งให้หยุดผลิตความถี่เสียงในพอร์ตดิจิทัลที่กำหนด

### รูปแบบคำสั่ง

noTone(ตำแหน่งพอร์ต)

ตัวอย่างเช่น ถ้าต้องการหยุดการส่งความถี่ไปออกที่พอร์ตดิจิทัลขา 8 ของบอร์ด Arduino ต้องใช้คำสั่ง

```
noTone(8);
```

### ตัวอย่างโปรแกรม

```
#define NOTE C4 262           // ประกาศตัวแปร NOTE C4 มีค่า 262
#define NOTE D4 294           // ประกาศตัวแปร NOTE D4 มีค่า 294
#define NOTE E4 330           // ประกาศตัวแปร NOTE E4 มีค่า 330
#define NOTE F4 349           // ประกาศตัวแปร NOTE F4 มีค่า 349
#define NOTE G4 392           // ประกาศตัวแปร NOTE G4 มีค่า 392
#define NOTE A4 440           // ประกาศตัวแปร NOTE A4 มีค่า 440
#define NOTE B4 494           // ประกาศตัวแปร NOTE B4 มีค่า 494
#define NOTE C5 523           // ประกาศตัวแปร NOTE C5 มีค่า 523
```

```

const int SpeakerPin = 8;           // กำหนดตัวแปร SpeakerPin ต่อที่พอร์ตดิจิตอลขา 8

void setup() {
    pinMode(SpeakerPin, OUTPUT);    // เซตพอร์ตดิจิตอลขา 8 ให้เป็นพอร์ตเอาต์พุต
}

void loop() {
    tone(SpeakerPin, NOTE C4, 500); // ส่งความถี่เสียงออกพอร์ตที่กำหนด
    delay(500);                      // หน่วงเวลา 0.5 วินาที
    tone(SpeakerPin, NOTE D4, 500); // ส่งความถี่เสียงออกพอร์ตที่กำหนด
    delay(500);                      // หน่วงเวลา 0.5 วินาที
    tone(SpeakerPin, NOTE E4, 500); // ส่งความถี่เสียงออกพอร์ตที่กำหนด

    delay(500);                      // หน่วงเวลา 0.5 วินาที
    tone(SpeakerPin, NOTE F4, 500); // ส่งความถี่เสียงออกพอร์ตที่กำหนด
    delay(500);                      // หน่วงเวลา 0.5 วินาที
    tone(SpeakerPin, NOTE G4, 500); // ส่งความถี่เสียงออกพอร์ตที่กำหนด
    delay(500);                      // หน่วงเวลา 0.5 วินาที
    tone(SpeakerPin, NOTE A4, 500); // ส่งความถี่เสียงออกพอร์ตที่กำหนด
    delay(500);                      // หน่วงเวลา 0.5 วินาที
    tone(SpeakerPin, NOTE B4, 500); // ส่งความถี่เสียงออกพอร์ตที่กำหนด
    delay(500);                      // หน่วงเวลา 0.5 วินาที
    tone(SpeakerPin, NOTE C5, 500); // ส่งความถี่เสียงออกพอร์ตที่กำหนด
    delay(500);                      // หน่วงเวลา 0.5 วินาที
    noTone(SpeakerPin);              // หยุดส่งความถี่เสียง
    delay(2000);                     // หน่วงเวลา 2 วินาที
}

```

## สรุป

ลำโพงบัสเซอร์ เป็นอุปกรณ์ที่ให้กำเนิดเสียง ทำหน้าที่เปลี่ยนสัญญาณไฟฟ้าให้อยู่ในรูปสัญญาณ เสียง ลำโพงบัสเซอร์มีอยู่ 2 ประเภท ได้แก่ แบบแอคทีฟ (Active Buzzer) และแบบพาสซีฟ (Passive Buzzer) การใช้ งาน Arduino กับลำโพงบัสเซอร์แบบพาสซีฟ ต้องทำการเขียนโปรแกรมเพื่อส่งความถี่ จาก Arduino เข้าไปยัง ลำโพงบัสเซอร์ ปัจจุบันเครื่องใช้ไฟฟ้าโดยทั่วไปมีลำโพงบัสเซอร์ติดตั้งอยู่ภายใน ด้วย เช่น คอมพิวเตอร์, เครื่องซักผ้าอัตโนมัติ, เครื่องปรับอากาศ