

Lesson 2 Use the Buzzer to Plays Music

2.1 Overview

This lesson focuses on using the buzzer on the Adeept Pixie Board to play various tones and a song. By following the steps and code examples provided, you can learn how to control the buzzer's frequency and duration to create different musical sounds.

2.2 Required Components

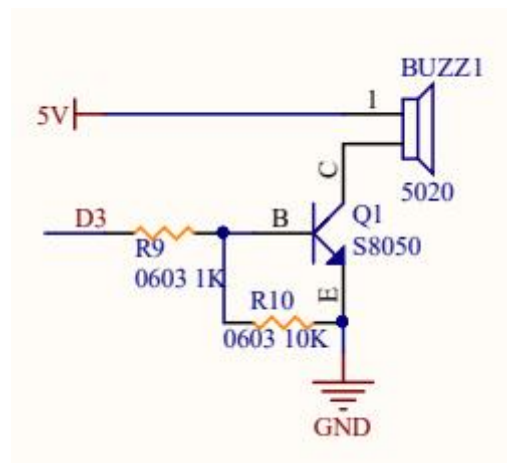
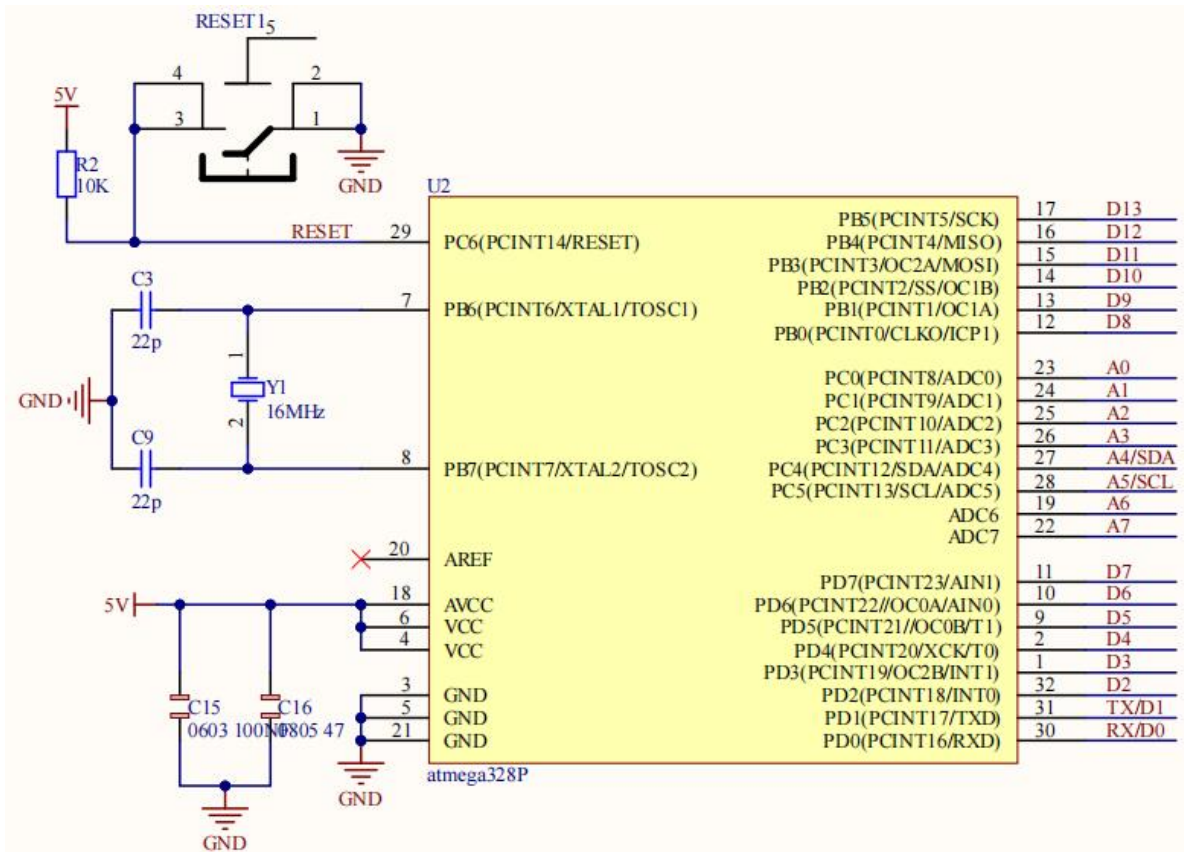
Components	Quantity	Picture
Adeept Pixie Board	1	
Type-C USB Cable	1	

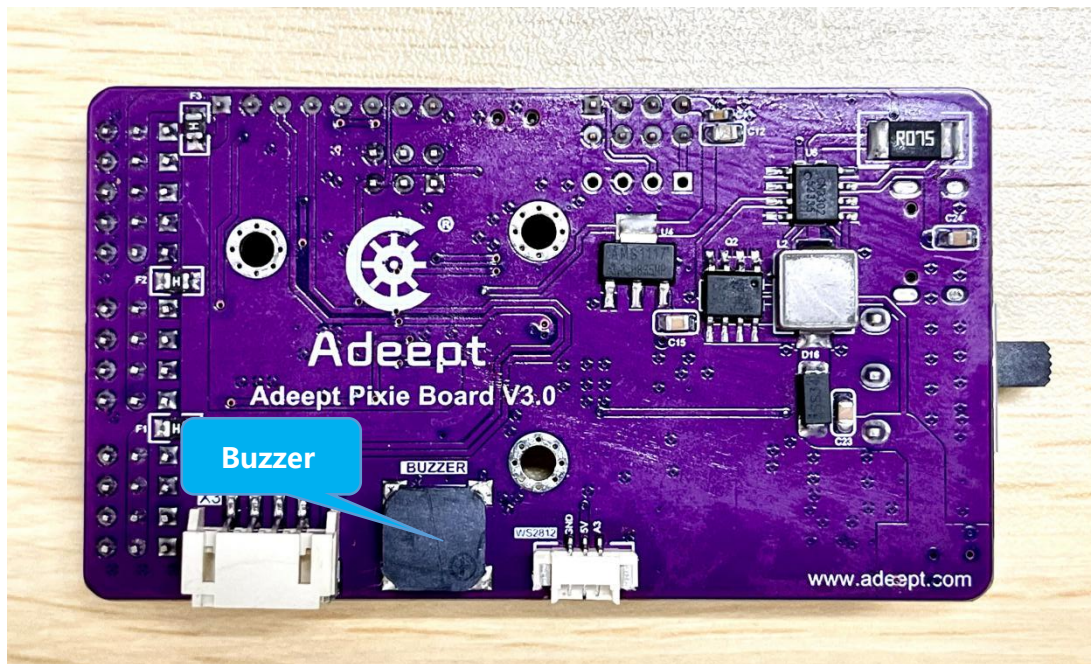
2.3 Principle Introduction

The Adeept Pixie Board has a passive buzzer integrated into it. The buzzer is connected to pin D3 of the board. When an appropriate electrical signal is sent to this pin, the buzzer can produce sounds of different frequencies. The frequency of the electrical signal determines the pitch of the sound, and the duration of the signal controls how long the sound lasts.

2.4 Wiring Diagram

A passive buzzer is built into the Adeept Pixie Board.

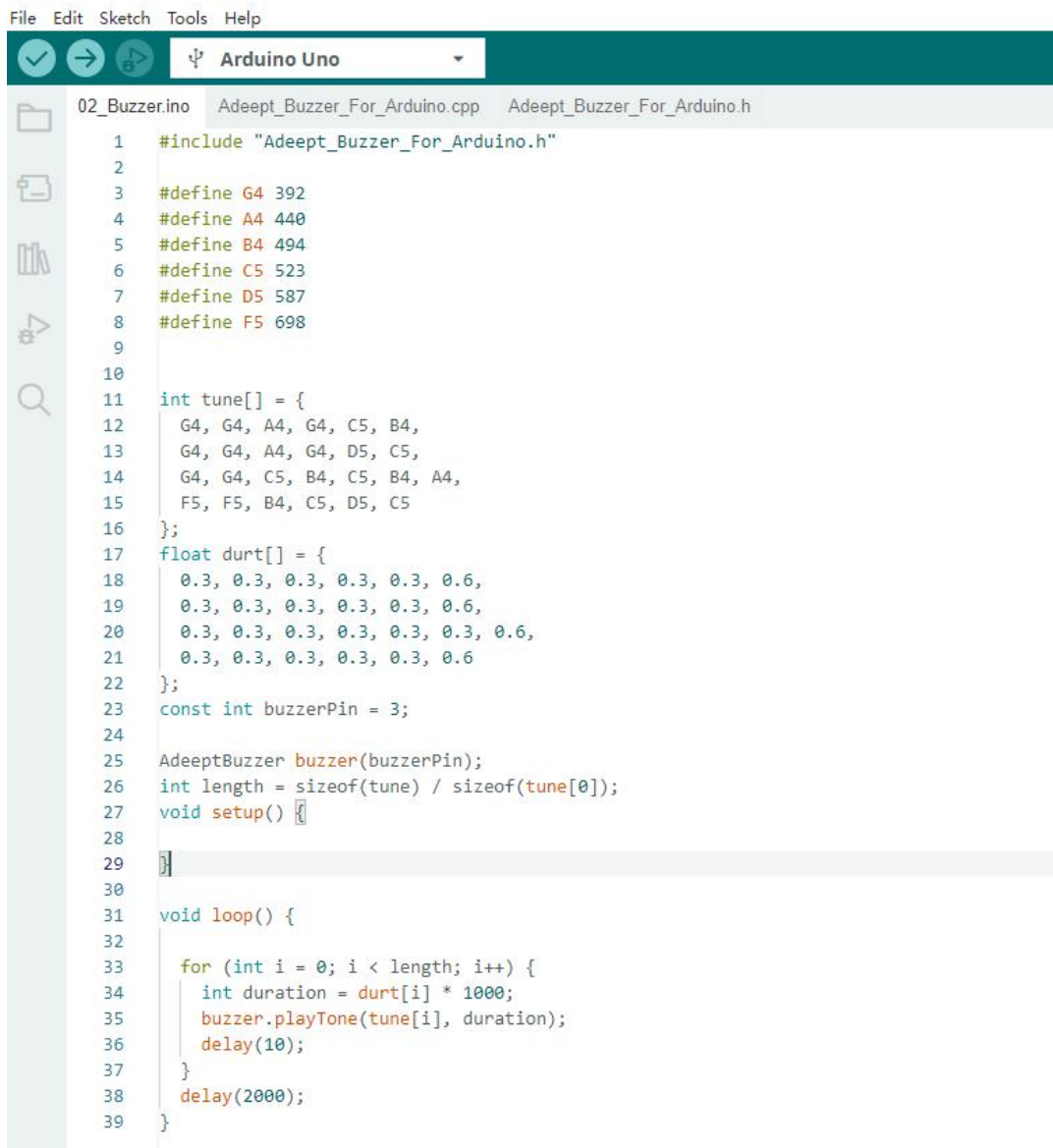




PINS of Adeept Pixie Board	Buzzer
D3	Buzzer

2.5 Demonstration

1. Connect your computer and Adeept Pixie Board with a USB cable.
2. Open **"02_Buzzer"** folder in **"/Code"** , double-click **"02_Buzzer.ino"** .



```

File Edit Sketch Tools Help
Arduino Uno
02_Buzzer.ino Adeept_Buzzer_For_Arduino.cpp Adeept_Buzzer_For_Arduino.h

1  #include "Adeeps_Buzzer_For_Arduino.h"
2
3  #define G4 392
4  #define A4 440
5  #define B4 494
6  #define C5 523
7  #define D5 587
8  #define F5 698
9
10
11 int tune[] = {
12     G4, G4, A4, G4, C5, B4,
13     G4, G4, A4, G4, D5, C5,
14     G4, G4, C5, B4, C5, B4, A4,
15     F5, F5, B4, C5, D5, C5
16 };
17 float durt[] = {
18     0.3, 0.3, 0.3, 0.3, 0.3, 0.6,
19     0.3, 0.3, 0.3, 0.3, 0.3, 0.6,
20     0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.6,
21     0.3, 0.3, 0.3, 0.3, 0.3, 0.6
22 };
23 const int buzzerPin = 3;
24
25 AdeepsBuzzer buzzer(buzzerPin);
26 int length = sizeof(tune) / sizeof(tune[0]);
27 void setup() {
28
29 }
30
31 void loop() {
32
33     for (int i = 0; i < length; i++) {
34         int duration = durt[i] * 1000;
35         buzzer.playTone(tune[i], duration);
36         delay(10);
37     }
38     delay(2000);
39 }

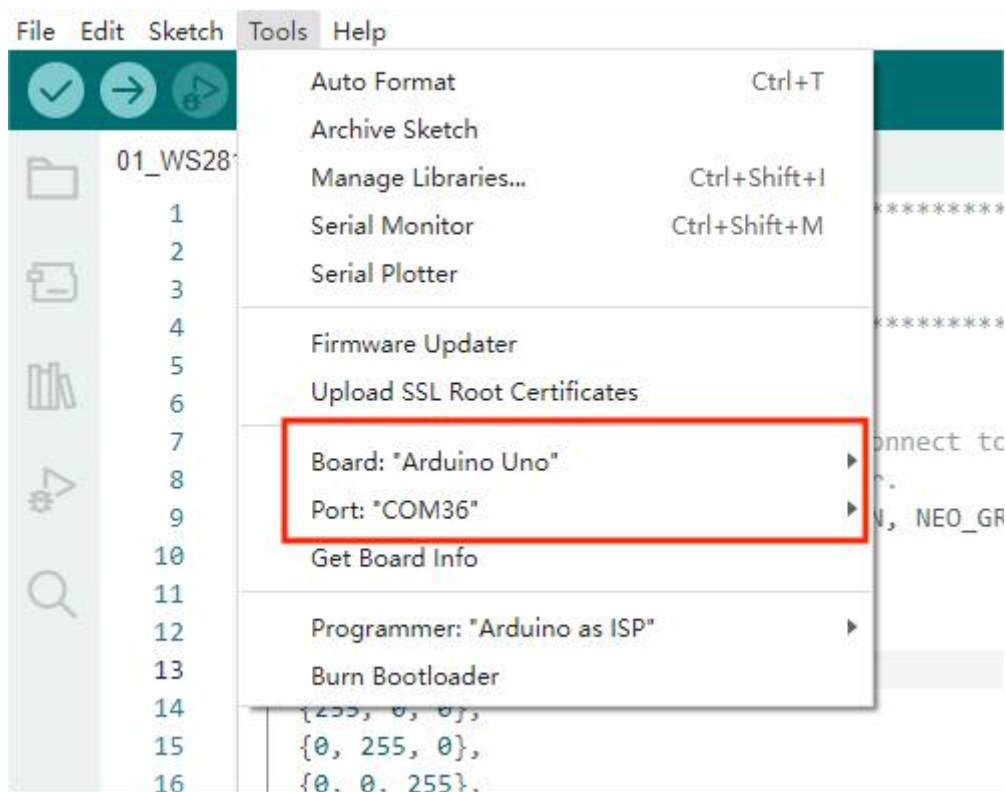
```

3. Select development board and serial port.

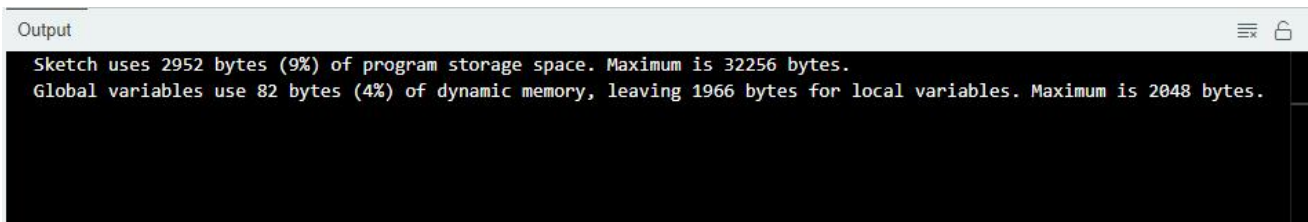
Board: **Tools---**>**Board---**>**Arduino AVR Boards---**>**Arduino Uno**

Port: **Tools --->Port---**>**COMx**

Note: The port number will be different in different computers.



4. After opening, click  to upload the code program to the Arduino. If there is no error warning in the console below, it means that the Upload is successful.



5. After the program runs successfully, the onboard buzzer will play the "Happy Birthday" song.

2.6 Code

```
01 #include "Adeept_Buzzer_For_Arduino.h"
02
03 #define G4 392
04 #define A4 440
05 #define B4 494
06 #define C5 523
07 #define D5 587
08 #define F5 698
09
```

```

10
11 int tune[] = {
12     G4, G4, A4, G4, C5, B4,
13     G4, G4, A4, G4, D5, C5,
14     G4, G4, C5, B4, C5, B4, A4,
15     F5, F5, B4, C5, D5, C5
16 };
17 float durt[] = {
18     0.3, 0.3, 0.3, 0.3, 0.3, 0.6,
19     0.3, 0.3, 0.3, 0.3, 0.3, 0.6,
20     0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.6,
21     0.3, 0.3, 0.3, 0.3, 0.3, 0.6
22 };
23 const int buzzerPin = 3;
24
25 AdeeptBuzzer buzzer(buzzerPin);
26 int length = sizeof(tune) / sizeof(tune[0]);
27 void setup() {
28
29 }
30
31 void loop() {
32
33     for (int i = 0; i < length; i++) {
34         int duration = durt[i] * 1000;
35         buzzer.playTone(tune[i], duration);
36         delay(10);
37     }
38     delay(2000);
39 }

```

Complete code refer to [Adeapt_Buzzer_For_Arduino.cpp](#)

```

01 #include "Adeapt_Buzzer_For_Arduino.h"
02 #include <Arduino.h>
03
04 AdeeptBuzzer::AdeaptBuzzer(int pin) {
05     buzzerPin = pin;
06     pinMode(buzzerPin, OUTPUT);
07 }
08
09 void AdeeptBuzzer::playTone(int frequency, int duration) {
10     tone(buzzerPin, frequency, duration);
11     delay(duration);+
12     noTone(buzzerPin);
13 }
14

```

Complete code refer to [Adeapt_Buzzer_For_Arduino.h](#)

```

01 // Adeapt_Buzzer_For_Arduino.h
02 #ifndef ADEEPT_BUZZER_FOR_ARDUINO_H
03 #define ADEEPT_BUZZER_FOR_ARDUINO_H
04
05 class AdeeptBuzzer {
06 public:
07     AdeeptBuzzer(int pin);

```

```
08     void playTone(int frequency, int duration);  
09     private:  
10         int buzzerPin;  
11     };  
12  
13 #endif
```

Code explanation

Initialization phase:

In the current code, the interior of the setup() function is temporarily empty and no specific initialization operation has been performed.

Loop control process:

Stage 1: Drive the onboard buzzer to sound sequentially based on the preset note frequency array tune and duration array durt. Note frequency includes G4 (392Hz) 、 A4 (440Hz) 、 B4 (494Hz) 、 C5 (523Hz) 、 D5 (587Hz) 、 F5 (698Hz) Wait, the duration of each note is determined by multiplying the corresponding value in the durt array by 1000 (in milliseconds). After playing each note, the program will pause for 10 milliseconds.

Stage 2: After completing a round of playing all preset notes, the buzzer stops sounding and the program enters a delay state, waiting for 2 seconds.

Forming a cyclic effect, Make the buzzer play periodically according to the preset note sequence and duration, and pause for 2 seconds after completing each round of playback.