

LESSON 6: BUZZER (ACTIVE)

INTRODUCTION

You can use a buzzer whenever you want to make some noise.

COMPONENTS

- 1x Arduino Uno board
- 1x Breadboard
- 1x USB cable
- 1x Buzzer (Active)
- Jumper wires

PRINCIPLE

As a type of electronic buzzer with integrated structure, buzzers, which are supplied by DC power, are widely used in computers, printers, photocopiers, alarms, electronic toys, automotive electronic devices, telephones, timers and other electronic products for voice devices. Buzzers can be categorized as active and passive ones (see the following picture). Turn the pins of two buzzers face up, and the one with a green circuit board is a passive buzzer, while the other enclosed with a black tape is an active one.



ACTIVE

PASSIVE

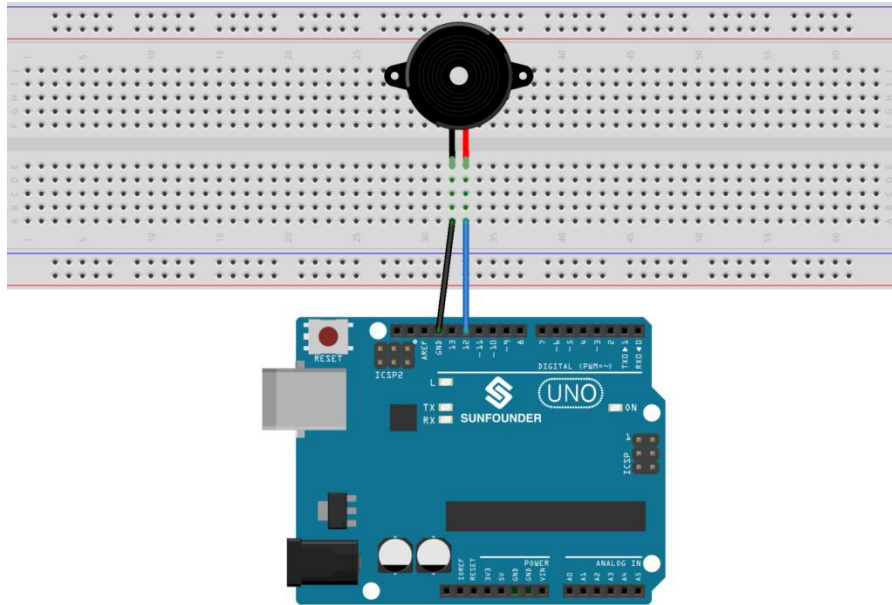
The difference between an active buzzer and a passive buzzer is:

An active buzzer has a built-in oscillating source, so it will make sounds when electrified. But a passive buzzer does not have such source, so it will not tweet if DC signals are used; instead, you need to use square waves whose frequency is between 2K and 5K to drive it. The active buzzer is often more expensive than the passive one because of multiple built-in oscillating circuits.

In this experiment, we use the active buzzer.

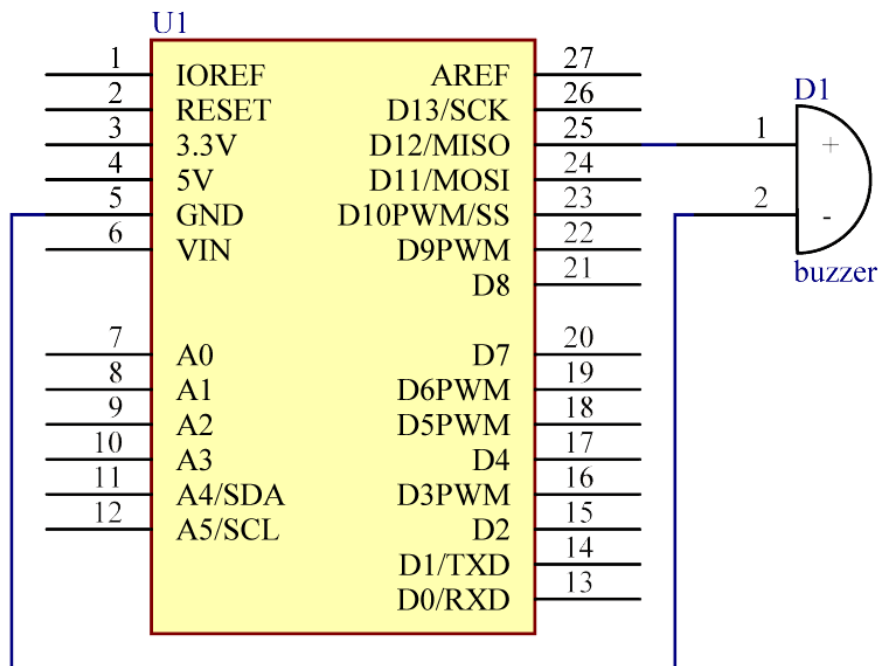
CONNECTION

STEP 1: Build the circuit



fritzing

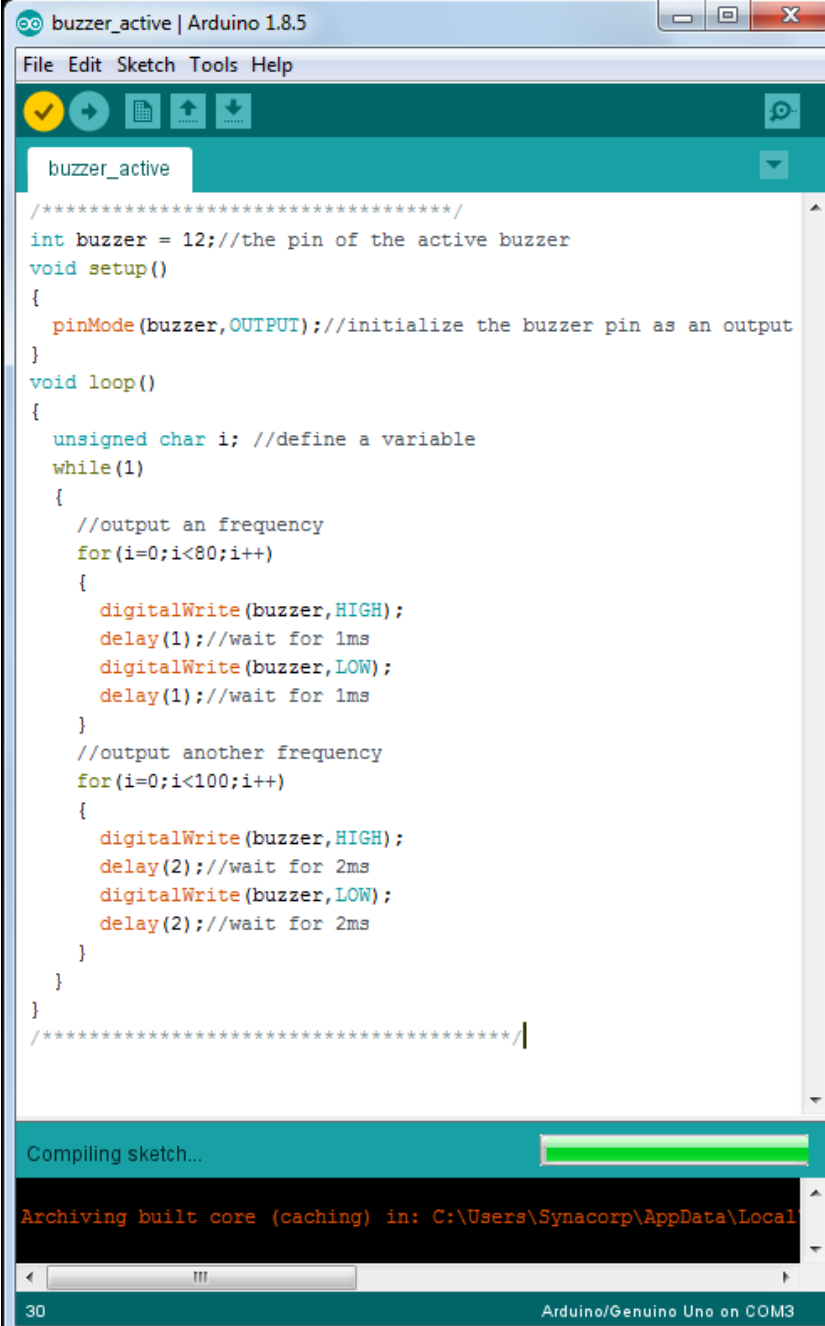
Schematic diagram



STEP 2: Program

```
/*  
int buzzer = 12;//the pin of the active buzzer  
void setup()  
{  
  pinMode(buzzer,OUTPUT);//initialize the buzzer pin as an output  
}  
void loop()  
{  
  unsigned char i; //define a variable  
  while(1)  
  {  
    //output an frequency  
    for(i=0;i<80;i++)  
    {  
      digitalWrite(buzzer,HIGH);  
      delay(1);//wait for 1ms  
      digitalWrite(buzzer,LOW);  
      delay(1);//wait for 1ms  
    }  
    //output another frequency  
    for(i=0;i<100;i++)  
    {  
      digitalWrite(buzzer,HIGH);  
      delay(2);//wait for 2ms  
      digitalWrite(buzzer,LOW);  
      delay(2);//wait for 2ms  
    }  
  }  
}  
*/
```

STEP 3: Compile the code

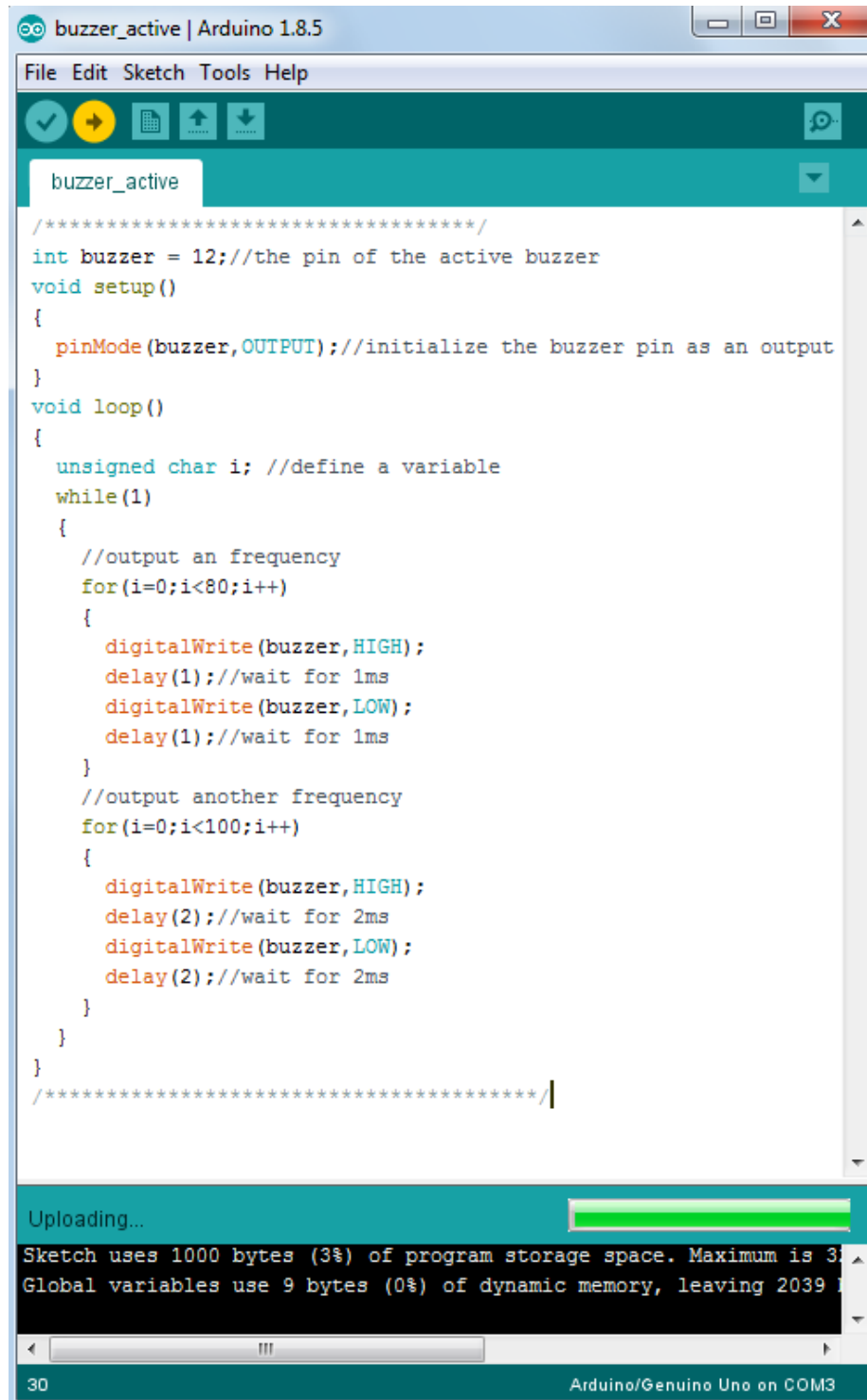


The screenshot shows the Arduino IDE interface with the sketch 'buzzer_active' open. The code is as follows:


```
/*  
int buzzer = 12;//the pin of the active buzzer  
void setup()  
{  
  pinMode(buzzer,OUTPUT);//initialize the buzzer pin as an output  
}  
void loop()  
{  
  unsigned char i; //define a variable  
  while(1)  
  {  
    //output an frequency  
    for(i=0;i<80;i++)  
    {  
      digitalWrite(buzzer,HIGH);  
      delay(1);//wait for 1ms  
      digitalWrite(buzzer,LOW);  
      delay(1);//wait for 1ms  
    }  
    //output another frequency  
    for(i=0;i<100;i++)  
    {  
      digitalWrite(buzzer,HIGH);  
      delay(2);//wait for 2ms  
      digitalWrite(buzzer,LOW);  
      delay(2);//wait for 2ms  
    }  
  }  
}  
*/
```

At the bottom of the IDE, a progress bar indicates 'Compiling sketch...'. Below the progress bar, the output window shows the message: 'Archiving built core (caching) in: C:\Users\Synacorp\AppData\Local'. The status bar at the bottom left shows the page number '30' and the board name 'Arduino/Genuino Uno on COM3'.

STEP 4: Upload the sketch to Arduino UNO board.



```
*****/
int buzzer = 12;//the pin of the active buzzer
void setup()
{
  pinMode(buzzer,OUTPUT);//initialize the buzzer pin as an output
}
void loop()
{
  unsigned char i; //define a variable
  while(1)
  {
    //output an frequency
    for(i=0;i<80;i++)
    {
      digitalWrite(buzzer,HIGH);
      delay(1);//wait for 1ms
      digitalWrite(buzzer,LOW);
      delay(1);//wait for 1ms
    }
    //output another frequency
    for(i=0;i<100;i++)
    {
      digitalWrite(buzzer,HIGH);
      delay(2);//wait for 2ms
      digitalWrite(buzzer,LOW);
      delay(2);//wait for 2ms
    }
  }
}
*****/
```

Uploading... 

Sketch uses 1000 bytes (3%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes free.

30 Arduino/Genuino Uno on COM3

RESULT: Now, you should hear the buzzer make sounds.

