

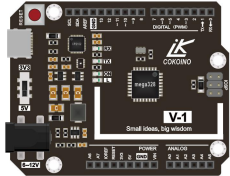
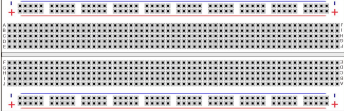

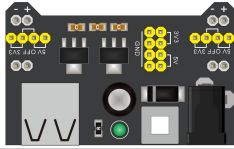








## 14.Electronic password lock

### ABOUT THIS PROJECT:

#### You will learn:

◆ How to make a electronic password lock

#### 1、 Things used in this project:

Hardware components	Picture	Quantity
V-1 board		1 PCS
Breadboard		1 PCS
Battery button (you need to buy 9V battery yourself)		1 PCS
Breadboard power module		1 PCS
Male to Male DuPont Cable		12 PCS
Type C USB Cable		1 PCS
SS8050 Transistor		1 PCS
IN4148 diode		1 PCS
Active buzzer		1 PCS
4*4 membrane button		1 PCS
F3 Red LED Light		1 PCS
220R Resistance		2 PCS

---

## 2.Overview

This experiment is an extension of Lesson 13, adding resistors and LED lights, and its circuitry is a bit more complicated. In the experiment, if you enter the correct password in the 4\*4 matrix keyboard, the LED in the circuit will light up.

\*: Re-enter password

#:confirm password

For example, the initial setting password is 123456. After entering the 123456 password on the 4\*4 matrix keyboard, press the # key again and the LED will light.

## 3.Experiment of electronic password lock

Copy the keypad library file to the libraries folder in the IDE installation directory.

### 3.1 Sketch

```
#include <Keypad.h>
#define buzzer 11
#define led 10
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
//define the symbols on the buttons of the keypads
char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {2, 3, 4, 5}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {6, 7, 8, 9}; //connect to the column pinouts of the keypad
//initialize an instance of class NewKeypad
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
char passWord[7]="123456";
char temp_password[7]="";
int count=0;
void setup(){
  Serial.begin(9600);
  pinMode(led,OUTPUT);
  setPwmFrequency(11, 8);
}
void loop(){
  char customKey = customKeypad.getKey();
  if (customKey){
    analogWrite(buzzer,30);
    if(count<=5 && customKey!='#' && customKey!='*')
      temp_password[count]=customKey;
    count++;
    if(count>=100)
      count=100;
    delay(100);
    analogWrite(buzzer,0);
    Serial.println(customKey);
  }
  if(customKey=='#'){ //confirm password
    int a=0;
```

---

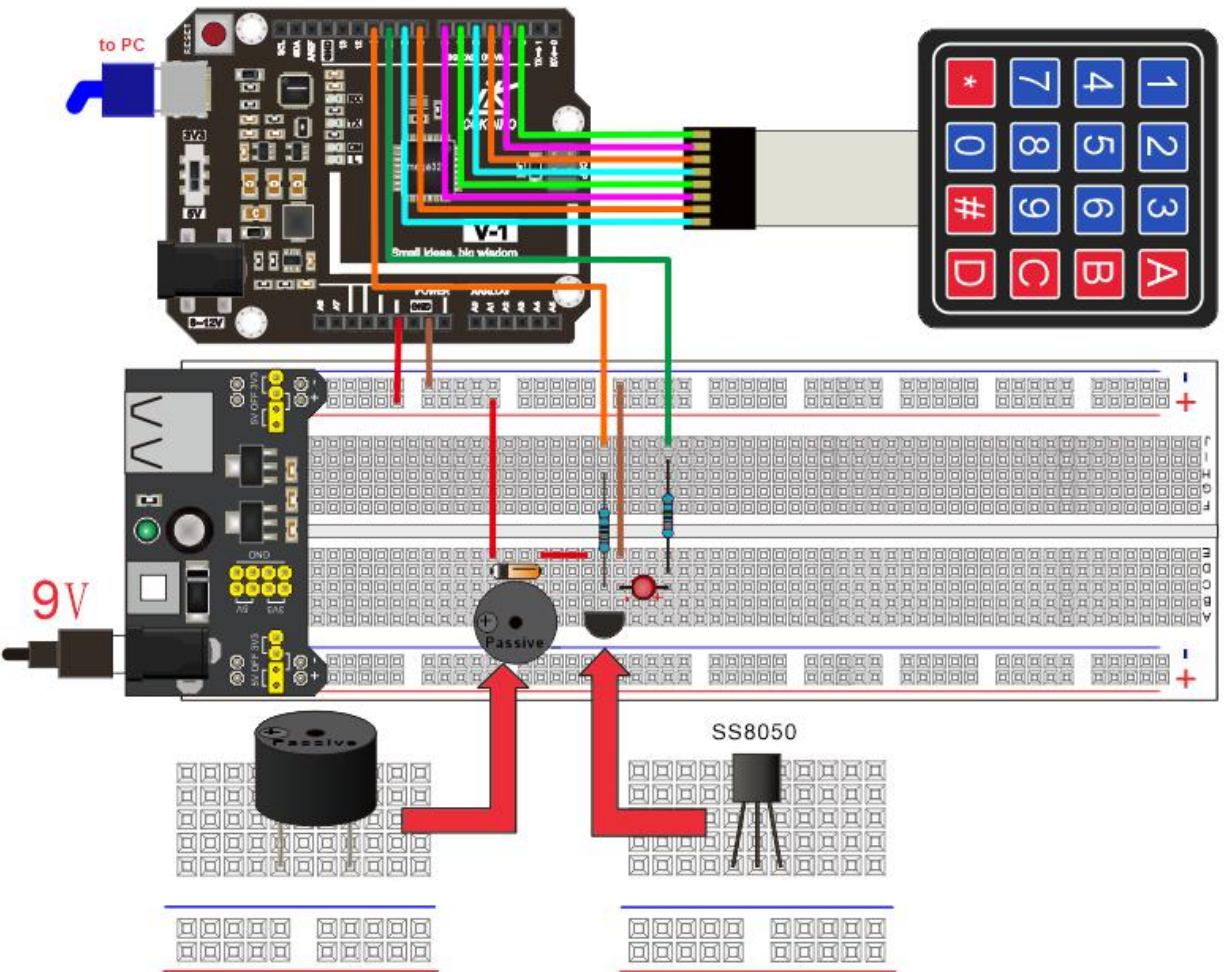
```

    for(int b=0;b<6;b++){
        if(temp_password[b]==passWord[b])
            a=a+1;
    }
    if(a==6)
        digitalWrite(led,HIGH);
    count=0;
}
if(customKey=='*'){    //reset password
    for(int c=0;c<6;c++){
        temp_password[c]=' ';
    }
    count=0;
    digitalWrite(led,LOW);
}
}

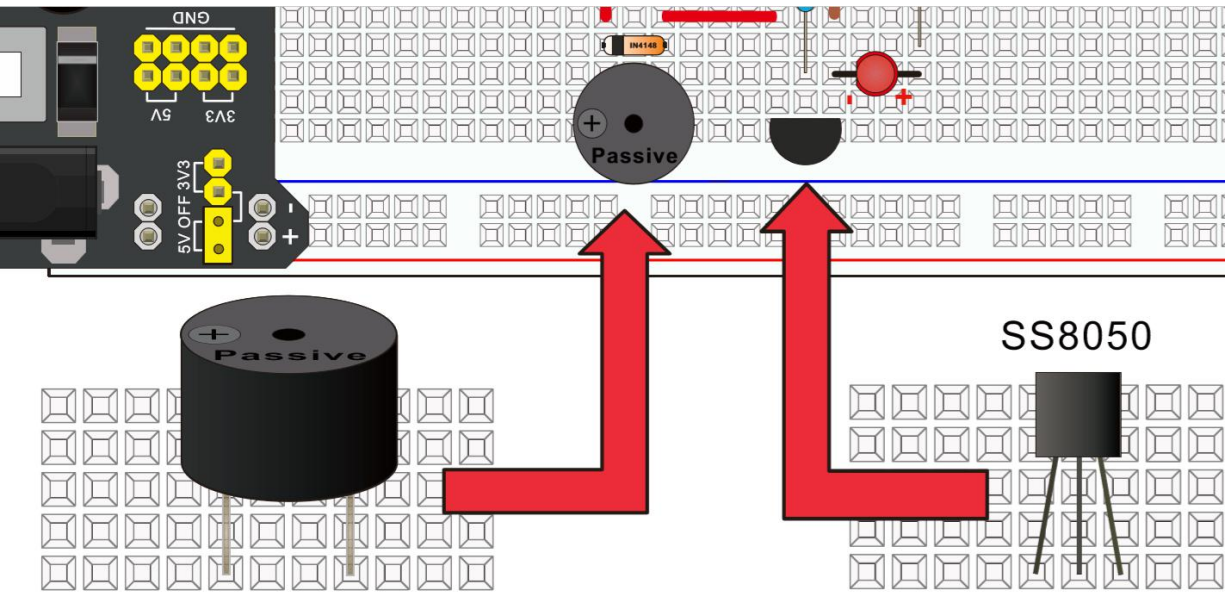
void setPwmFrequency(int pin, int divisor) {
    byte mode;
    if(pin == 5 || pin == 6 || pin == 9 || pin == 10) {
        switch(divisor) {
            case 1: mode = 0x01; break;
            case 8: mode = 0x02; break;
            case 64: mode = 0x03; break;
            case 256: mode = 0x04; break;
            case 1024: mode = 0x05; break;
            default: return;
        }
        if(pin == 5 || pin == 6) {
            TCCR0B = TCCR0B & 0b11111000 | mode;
        } else {
            TCCR1B = TCCR1B & 0b11111000 | mode;
        }
    } else if(pin == 3 || pin == 11) {
        switch(divisor) {
            case 1: mode = 0x01; break;
            case 8: mode = 0x02; break;
            case 32: mode = 0x03; break;
            case 64: mode = 0x04; break;
            case 128: mode = 0x05; break;
            case 256: mode = 0x06; break;
            case 1024: mode = 0x07; break;
            default: return;
        }
        TCCR2B = TCCR2B & 0b11111000 | mode;
    }
}
}

```

3.2 Wiring Diagram

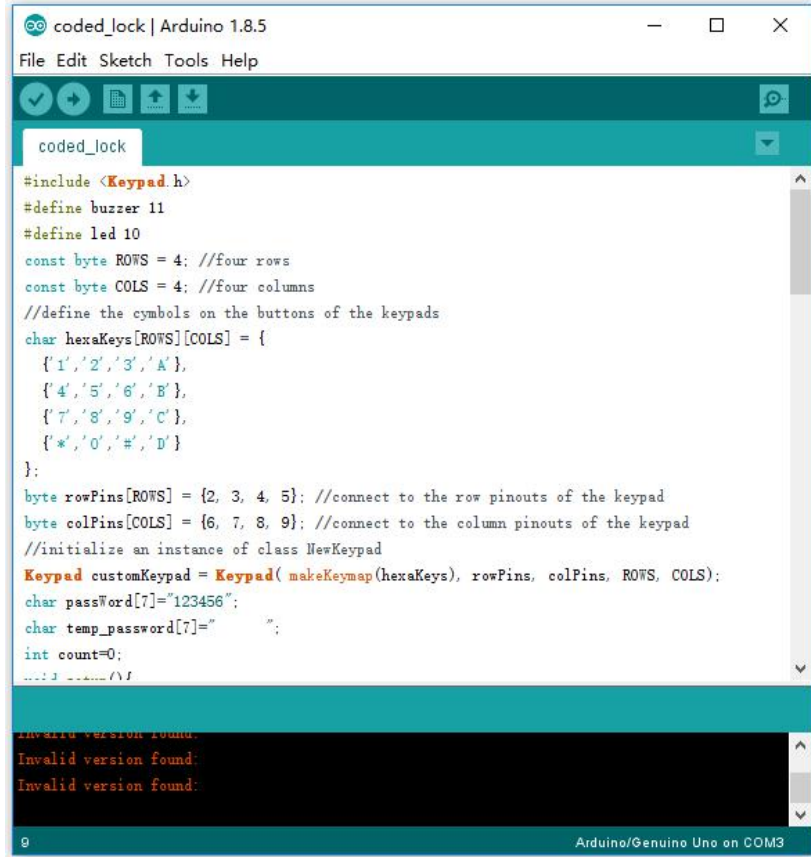


Detail enlargement



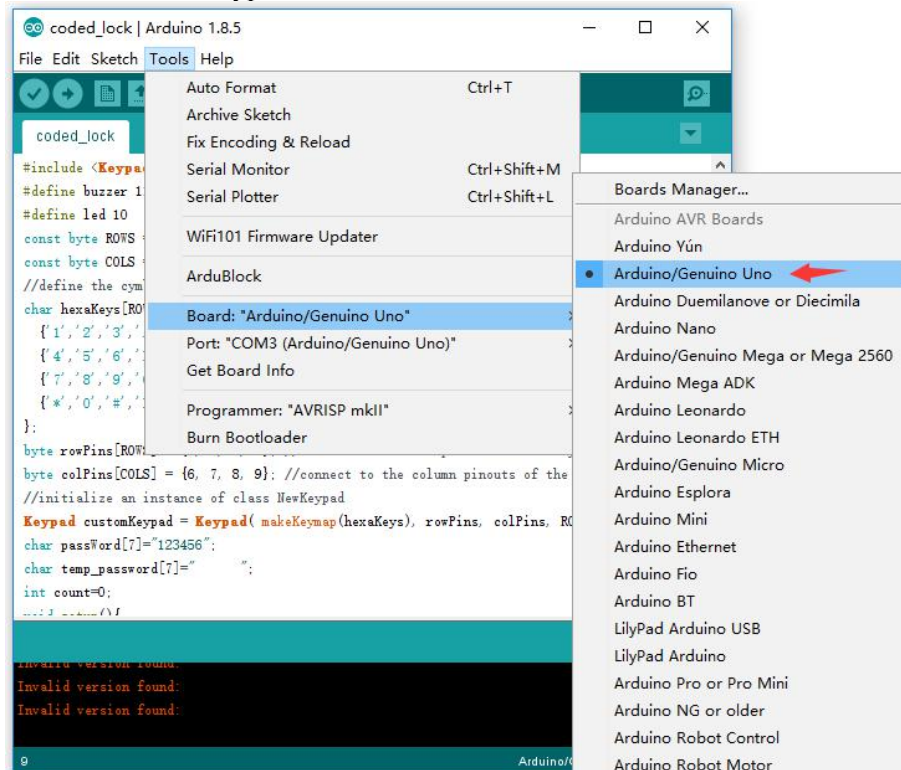
### 3.3 Step

3.3.1 Connect the computer and V-1 board with a USB cable and copy the above sample code to the Arduino IDE as shown below:



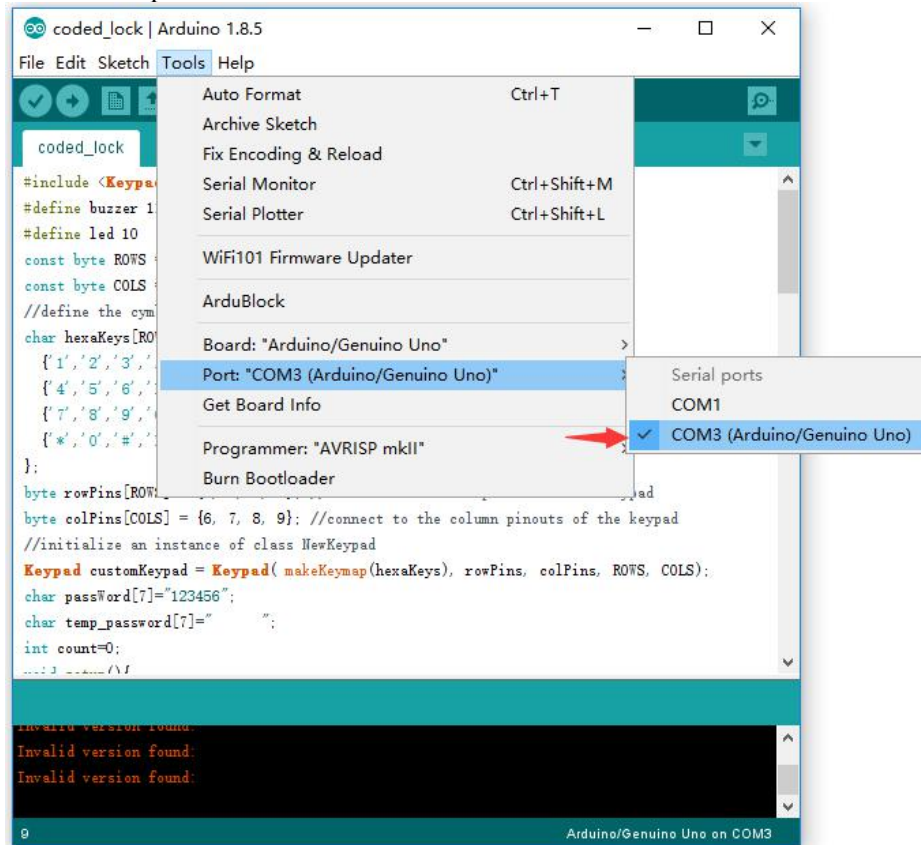
```
#include <Keypad.h>
#define buzzer 11
#define led 10
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
//define the symbols on the buttons of the keypad
char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {2, 3, 4, 5}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {6, 7, 8, 9}; //connect to the column pinouts of the keypad
//initialize an instance of class NewKeypad
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
char password[7]="123456";
char temp_password[7]="";
int count=0;
void setup() {
  pinMode(buzzer, OUTPUT);
  pinMode(led, OUTPUT);
}
```

### 3.3.2 Select board type

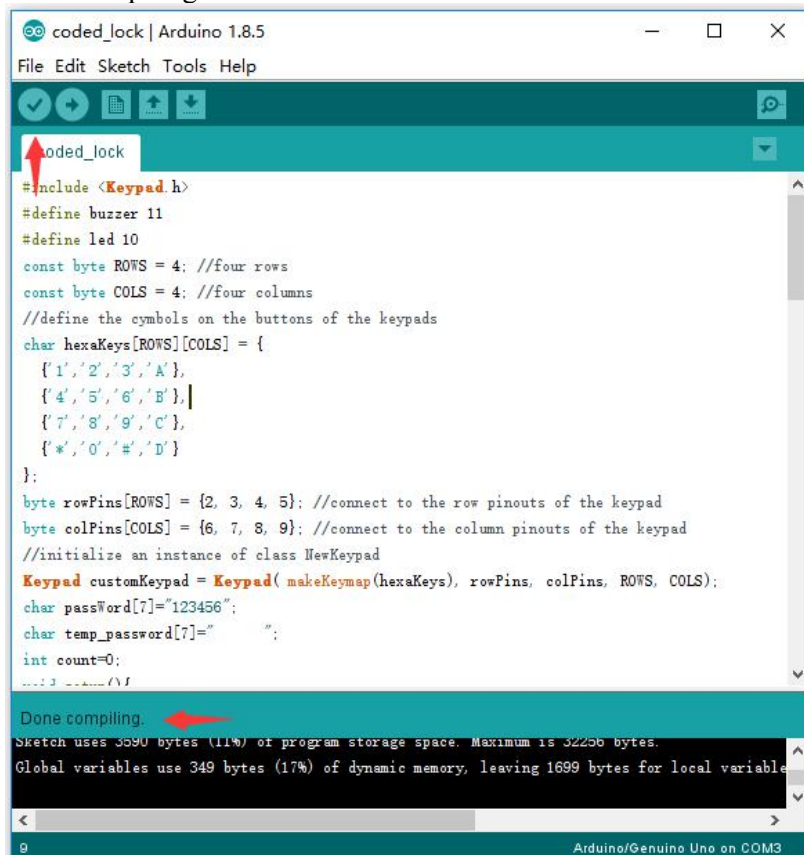




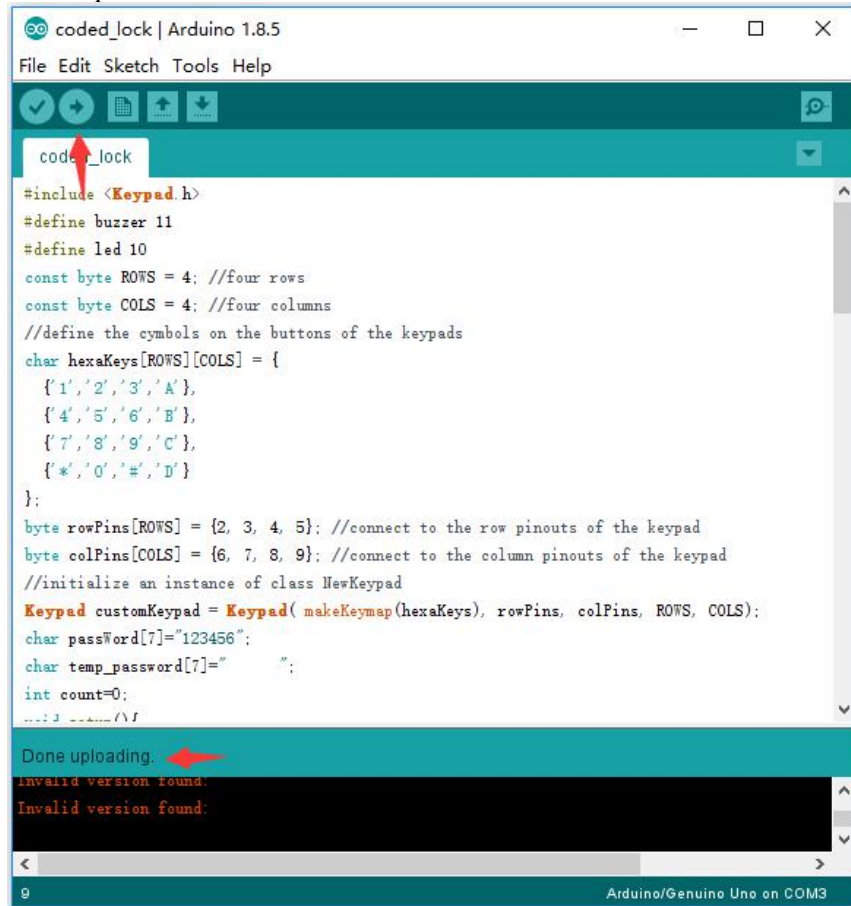
### 3.3.3 Select port



### 3.3.4 Compiling



### 3.3.5 Upload the sketch



The screenshot shows the Arduino IDE interface with the sketch 'coded\_lock' loaded. The code defines a 4x4 keypad with a password of '123456'. The status bar at the bottom indicates 'Arduino/Genuino Uno on COM3'. The upload button (a right-pointing arrow) is highlighted with a red arrow. Below the code editor, a message box says 'Done uploading.' with a red arrow pointing to it. The console window shows two 'Invalid version found.' messages.

```
#include <Keypad.h>
#define buzzer 11
#define led 10
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
//define the symbols on the buttons of the keypads
char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {2, 3, 4, 5}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {6, 7, 8, 9}; //connect to the column pinouts of the keypad
//initialize an instance of class NewKeypad
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
char passWord[7]="123456";
char temp_password[7]=" ";
int count=0;
```

### 3.3.6、 Result

Unplug the USB cable from the V-1 board, connect the power module to the external power supply, and then turn on the switch of the power module on the breadboard. entering the 123456 password on the 4\*4 matrix keyboard, press the # key again and the LED will light.

