

KEY Control Buzzer

KEY Control Buzzer

Device connection

Hardware connection

Software connection

Button control RGB

Control principle

Control pin

Code analysis

Experimental results

The buzzer sound is controlled by controlling the button (K1) on the Robduino expansion board.

Device connection

Hardware connection

Use Type-B data cable to connect Arduino Uno and computer.

Software connection

Open the "Arduino IDE" software and select the model and serial port number corresponding to the development board.

Button control RGB

Control principle

Associate the button state with the buzzer, and switch the buzzer state by pressing and releasing the button.

Control pin

Peripheral module	Arduino Uno
K1	7
BUZZER	10

Code analysis

Here is only a brief introduction to the code content. For detailed code, please refer to the corresponding code file, which is provided in the download area!

- Define control pins, button status and buzzer status

```
// 定义按键引脚和控制状态 Define pin and key(button) states
#define BUZZER_PIN 10
bool bBuzzerState = false;

// 定义按键引脚和控制状态 Define pin and key(button) states
#define KEY_PIN 7
#define Press_KEY 0
#define Release_KEY 1
```

- Button detection

Note: This method will only respond to the button being pressed when the button is released, that is, it will only return to the pressed state once it is pressed and released!

```
/**
 * @brief 获取按键状态 Get key(button) status
 * @param pin: 按键控制引脚 Control key(button) pins
 * @retval 按键状态 Key(button) Status
 */
int getKeyState(uint8_t pin) {
    if (digitalRead(pin) == LOW) {
        delay(20);
        if (digitalRead(pin) == LOW) {
            while (digitalRead(pin) == LOW)
                ;
            return Press_KEY;
        }
        return Release_KEY;
    } else {
        return Release_KEY;
    }
}
```

- Initialization Code

```
void setup() {
    pinMode(KEY_PIN, INPUT_PULLUP); // 设置按键KEY引脚上拉输入模式 Set the key(button)
    pin to pull-up input mode
    pinMode(BUZZER_PIN, OUTPUT);    // 设置蜂鸣器引脚输出模式 Set the buzzer pin
    output mode
}
```

- Looping code

```
void loop() {  
  // 根据按键切换蜂鸣器发声 Switch the buzzer sound according to the key  
  if (getKeyState(KEY_PIN) == Press_KEY) {  
    bBuzzerState = !bBuzzerState;  
  }  
  if (bBuzzerState) {  
    tone(BUZZER_PIN, 1000);  
    delay(20);  
  }  
  noTone(BUZZER_PIN);  
}
```

Experimental results

After compiling the program successfully, upload the code to the Arduino Uno development board.

After the program starts, press and release the button, the buzzer will switch the sound state (sound or no sound)!

The burning program cannot use other programs to occupy the serial port or external serial communication module (for example: WiFi camera module), otherwise the program cannot be burned or an error message will be prompted!