KEY Read

Read the button (K1) on the Robduino expansion board and print the button status through the serial port.

Device connection

Hardware connection

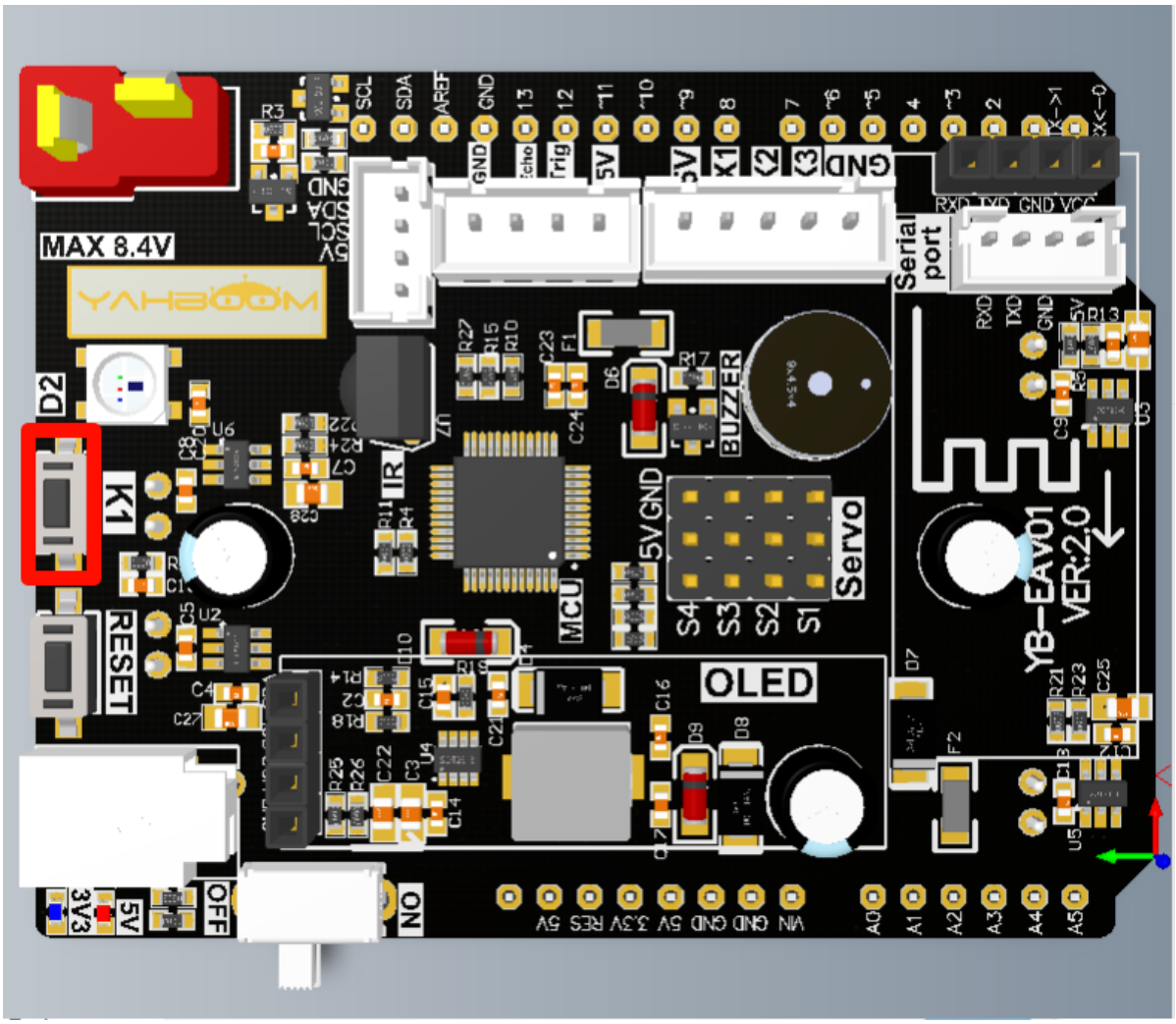
Use a Type-B data cable to connect Arduino Uno and the computer.

Software connection

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

Read button

The location selected by the red box is the location of the button K1 module:



Control principle

Determine whether the button is pressed or released by reading the high and low levels of the pins corresponding to the button.

Press the button: read the low level

Release the button: read the high level

Control pin

Peripheral module	Arduino Uno
K1	7

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 pin and key(button) states

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

- Key detection

Note: This method directly reports the key status

```
/**
 * @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) {
      return Press_KEY;
    }
    return Release_KEY;
  } else {
    return Release_KEY;
  }
}
```

- Initialization Code

```
void setup() {
  Serial.begin(115200);            // 初始化串口波特率115200 Initialize serial
communication at 115200 bps
  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
}
```

- Looping code

```
void loop() {
  // 获取按键KEY状态并打印信息 Get the key(button) status and print it
  if (getKeyState(KEY_PIN) == Press_KEY) {
    Serial.println("Press_KEY");
  } else {
    Serial.println("Release_KEY");
  }
}
```

Experimental results

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

After the program starts, press or release the button K1, and the serial port will print the corresponding status of the button!

If there is no display content, you can check whether the serial port baud rate is consistent with the code setting, and then press the RESET button on the development board.

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