

Ultrasonic Ranging

Ultrasonic Ranging

Device connection

Hardware connection

Software connection

Ultrasonic distance measurement

Control principle

Distance measurement formula

Control pin

Code analysis

Experimental results

Use the serial port to print the obstacle distance measured by the ultrasonic module.

Device connection

Hardware connection

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

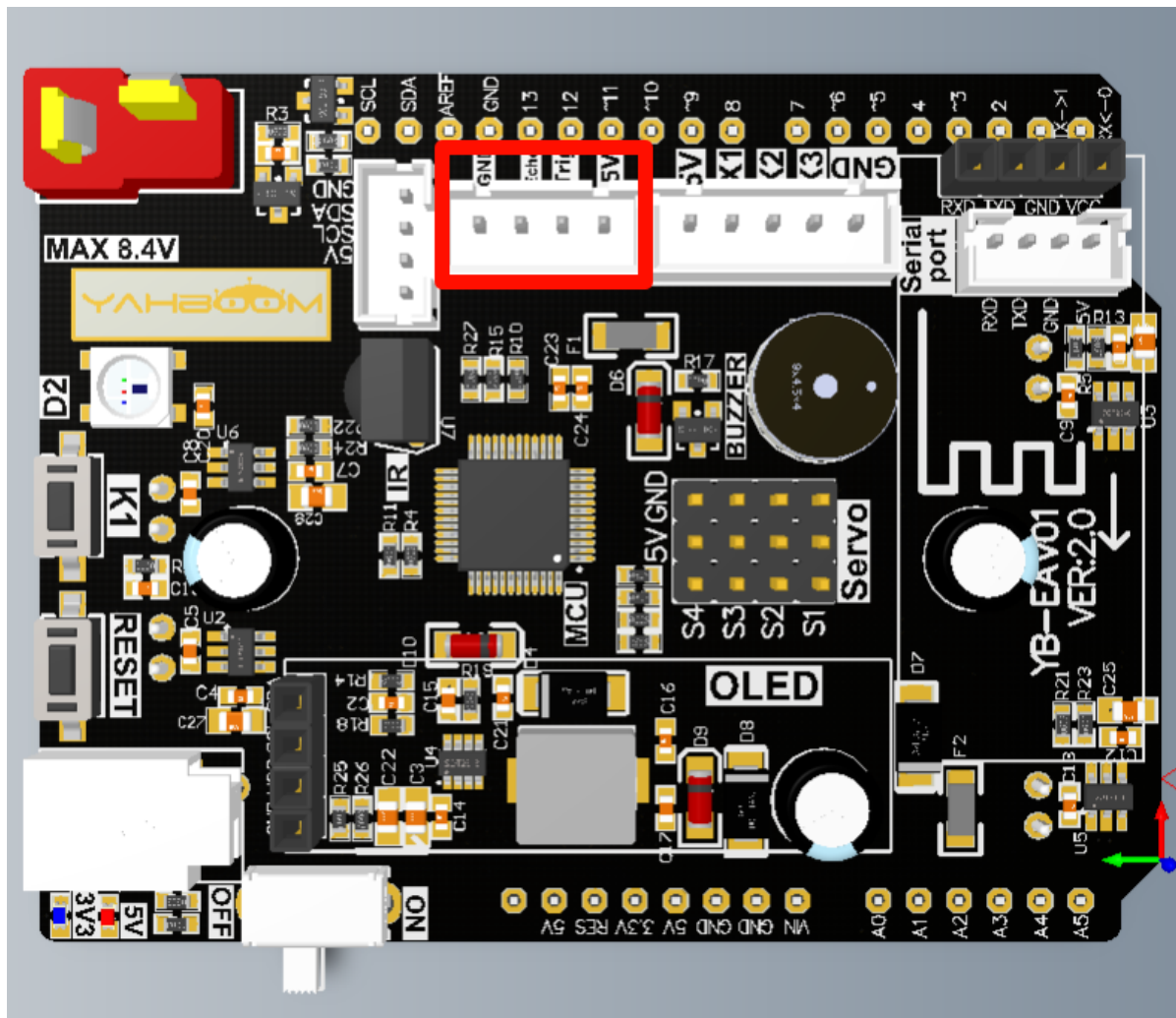
Software connection

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

Ultrasonic distance measurement

Ultrasonic waves are non-contact physical quantity measurement modules. They can accurately measure distances by transmitting and receiving ultrasonic signals.

The location selected by the red box is the location of the ultrasonic module interface:



Control principle

Transmit (trigger) signal: provide a pulse trigger signal of more than 10us;

Receive signal: the received high level time is proportional to the measured distance.

Distance measurement formula

$$Distance = \frac{highleveltime * soundspeed(340m/s)}{2}$$

Control pin

The ultrasonic module and Robduino expansion board have special interfaces for installation and connection.

Peripheral module	Arduino Uno
VCC	VCC
TRIG (trigger end)	RGB (12)
ECHO (receiver end)	ECHO (11)
GND	GND

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 ultrasonic control pins and initial distance

```
// 定义超声波模块控制引脚 Define ultrasonic control pins
#define TRIG_PIN 11
#define ECHO_PIN 12

float Distance = 0.0; // 初始化超声波距离变量 Initialize the ultrasonic distance variable
```

- Get ultrasonic distance

```
/**
 * @brief 获取超声波距离 Get ultrasonic distance
 * @param trigPin: 触发测距引脚 Trigger pin
 * @param echoPin: 接收测距引脚 Echo pin
 * @retval 转换的距离 Measured distance (cm)
 */
float getDistance(int trigPin, int echoPin) {
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);

    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    long duration = pulseIn(echoPin, HIGH);
    float distance = (duration * 0.034 / 2);
    return distance;
}
```

- Initialization Code

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

- Looping code

```

void loop() {
    delay(100);
    float newDistance = getDistance(TRIG_PIN, ECHO_PIN);
    // 获取的距离发生变化进行打印 Print only when the distance changes
    if (newDistance != Distance) {
        Serial.print("Distance: ");
        Serial.print(newDistance, 1);
        Serial.println(" cm");
        Distance = newDistance;
    }
}

```

Experimental results

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

After the program is started, the serial port will print the real-time distance in front of the ultrasonic wave (data will be printed only when the distance in front changes)!

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!

