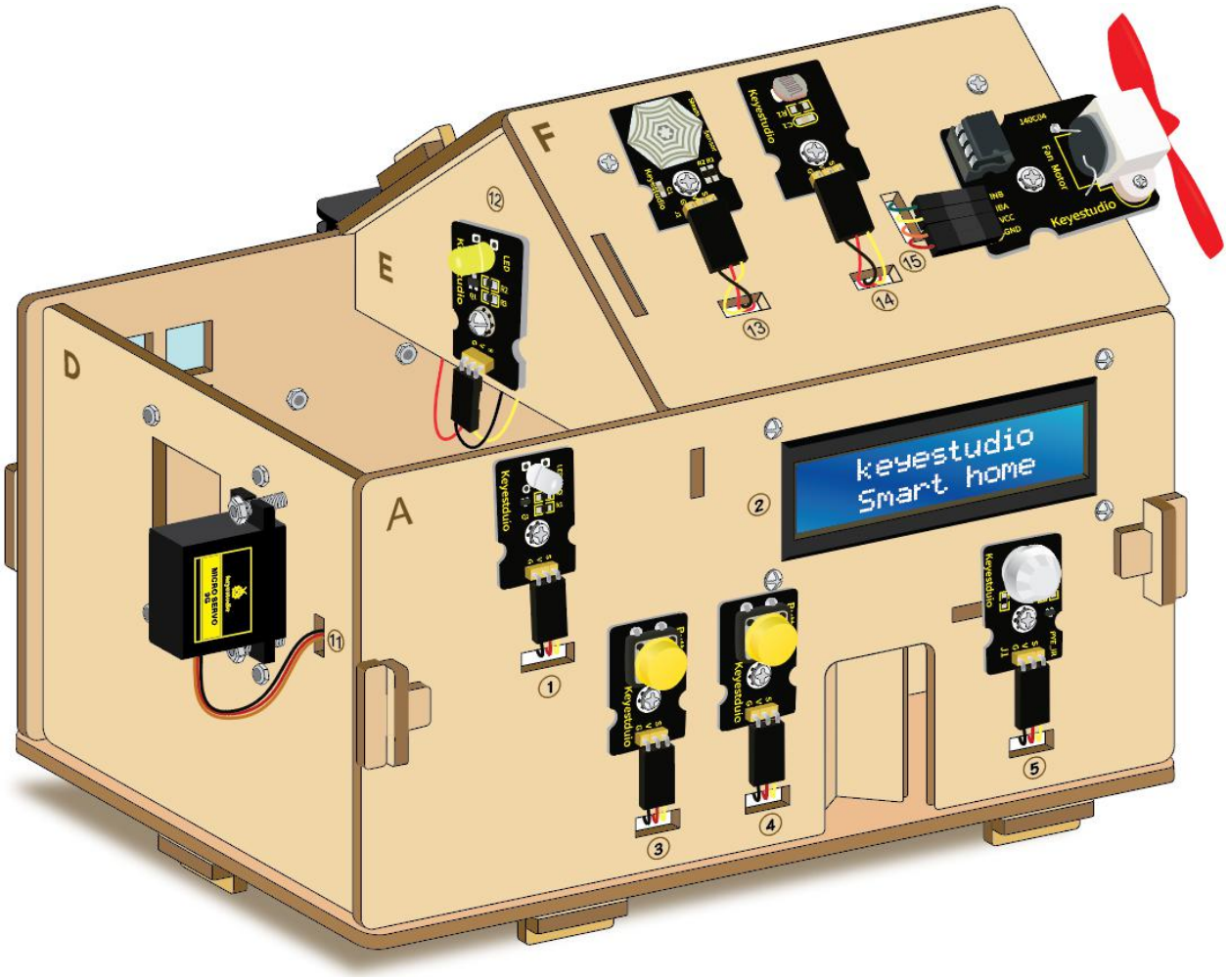
Keyestudio Smart Home Kit For Arduino

1. Overview:



The smart home kit based on Arduino is newly issued by Keyestudieo company. The concept is to make people enjoy life.

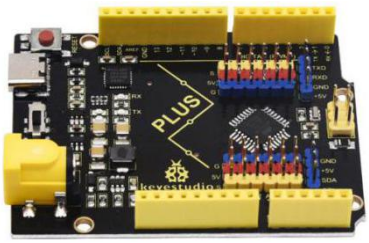
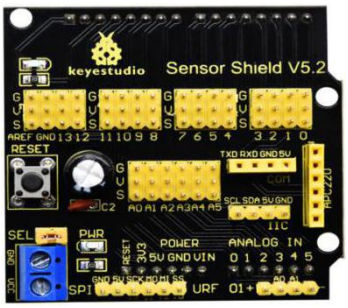
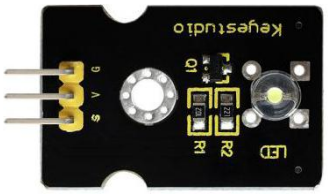
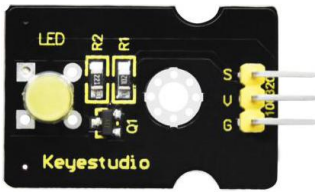
You could remotely control your assorted smart systems through your phone or computer when you're out, like turning on air conditioner and water heater on the way home; the electronic door lock and led light will automatically run when you get home.

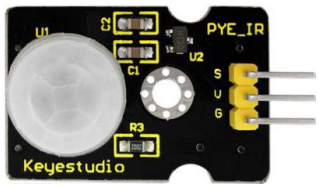
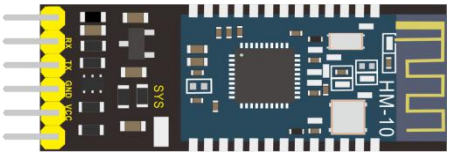
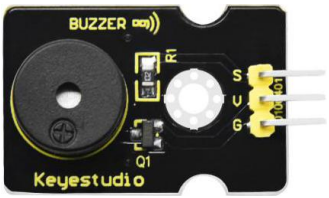
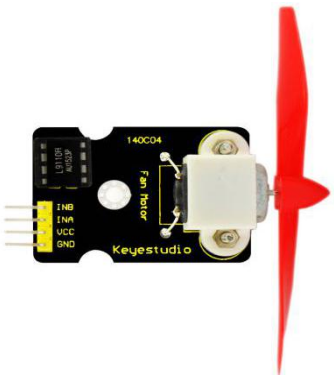
Additionally, the intelligent lighting system can be used to select preset lighting scenes to create a comfortable and quiet atmosphere, which contributes to make your brain completely relaxed. You just need a remote control to finish all the process. What surprises you is that the smart system can save electricity as well. As Bill Gates puts it, in the future, homes without smart home systems will be as unfashionable as homes without internet access today.

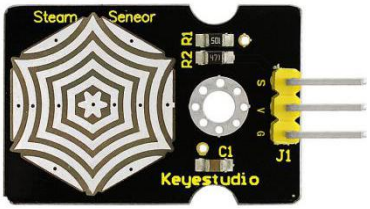
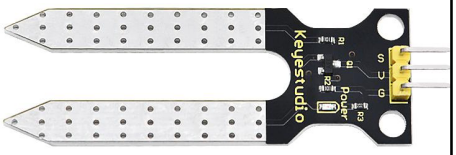
Let's know more about smart home kit!

Contents

2. Kit List

No.	Name	QTY	Picture
1	keyestudio PLUS Control Board (Compatible with Arduino UNO)	1	
2	keyestudio Shield V 5.2 Expansion Board	1	
3	Wooden Board*10 T=3MM	1	
4	White LED Module	1	
5	Yellow LED Module	1	

6	Button Sensor	2	
7	Photocell Sensor	1	
8	PIR Motion Sensor	1	
9	MQ-2 Gas Sensor	1	
10	Relay Module	1	
11	Bluetooth HM-10 Module	1	
12	Passive Buzzer Sensor	1	
13	Fan module	1	

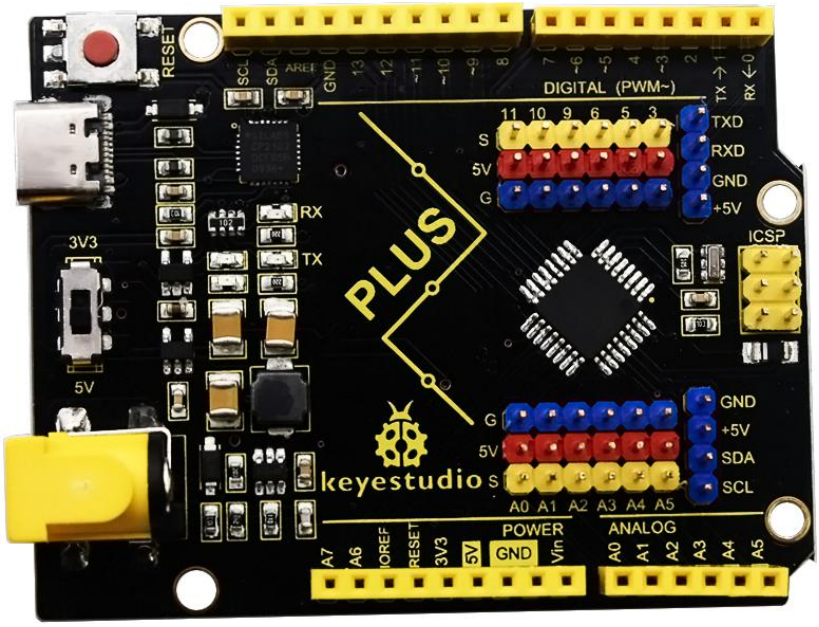
14	Steam Sensor	1	
15	Servo Motor	2	
16	LCD1602 Display Module	1	
17	Soil Humidity Sensor	1	
18	USB Cable	1	
19	Female to Female Dupont lines	1	
20	Male to female dupont line	0.15	
21	M3 Nickel Plated Nut	25	
22	M2*12MM Round Head Screw	6	
23	M2 Nickel Plated Nut	6	
24	M3*10MM Dual-pass Copper	4	

	Pillar		
25	M3*6MM Round Head Screw	8	
26	M3 304 Stainless Steel Self-locking Nut	4	
27	M3*10MM Round Head Screw	20	
28	M2.5*10MM Round Head Screw	7	
29	M2.5 Nickel plated nut	6	
30	M3*12MM Round Head Screw	6	
31	M3*10MM Flat Head Screw	2	
32	M1.2*5MM Round Head Self-tapping Screw	10	
33	6-cell AA battery holder with DC head and 15cm dew line	1	
34	Black-yellow Handle 3*40MM Cross Screwdriver	1	
35	2.54 3Pin F-F jumper wire 20cm	13	

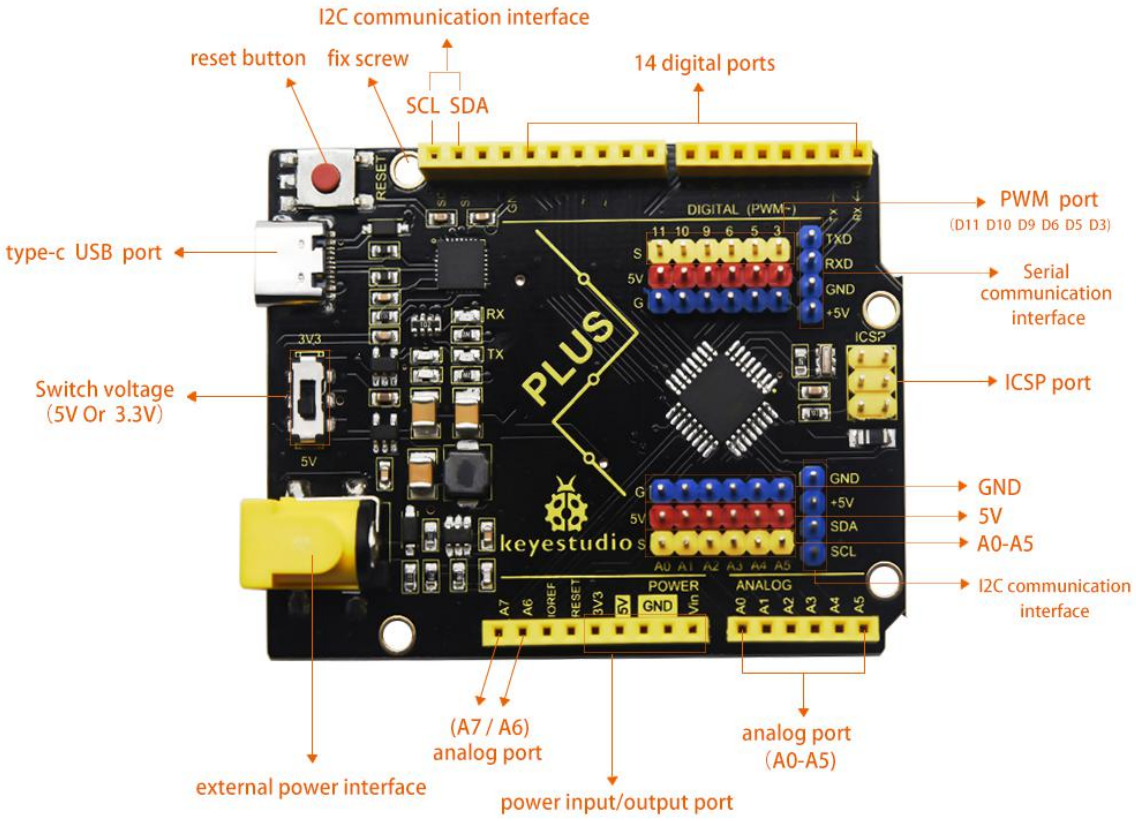
36	2.54 4Pin F-F jumper wire 20cm	2	

3. Keyestudio PLUS Control Board

Description



Keyestudio PLUS control board is fully compatible with Arduino IDE development environment. It contains all the functions of the Arduino UNO board. Moreover, some improvements we made highly strengthen its function. It is the best choice to learn how to build circuit and write code as well. Let's get started!



Serial communication interface: D0 is RX, D1 is TX

PWM interface (pulse width modulation): D3 D5 D6 D9 D10 D11

External interrupt interface: D2 (interrupt 0) and D3 (interrupt 1)

SPI communication interface: D10 is SS, D11 is MOSI, D12 is MISO, D13 is SCK

IIC communication port: A4 is SDA, A5 is SCL

Introduction:

When we get control board, we need to download Arduino IDE and driver firstly.

You could download Arduino IDE from the official website:

https://www.arduino.cc/en/Main/OldSoftwareReleases#1.5.x.

There are various versions of the IDE in the official link, here we download a Windows system, version 1.8.10.

click 

to download software directly, you don't need to install，click 

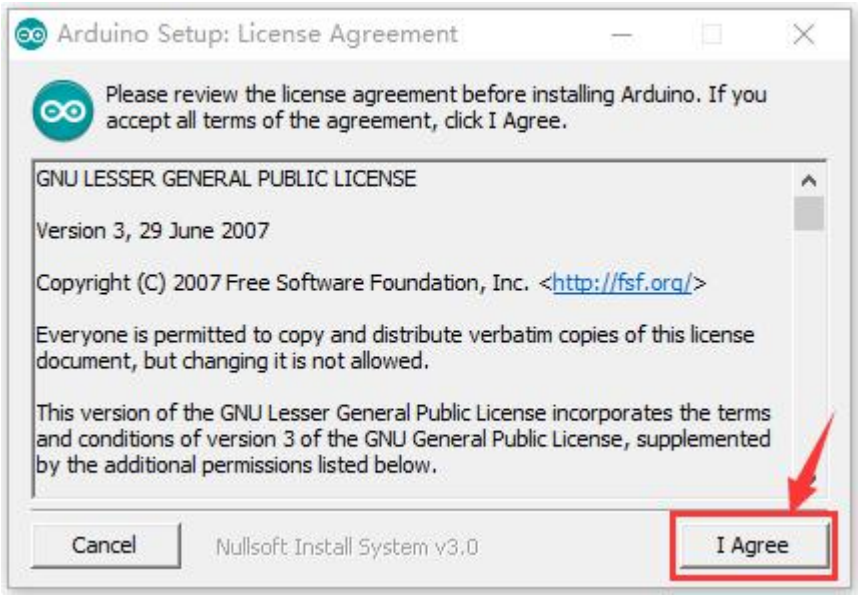
, windows installer stands for installation file, then click 

to download installation file.

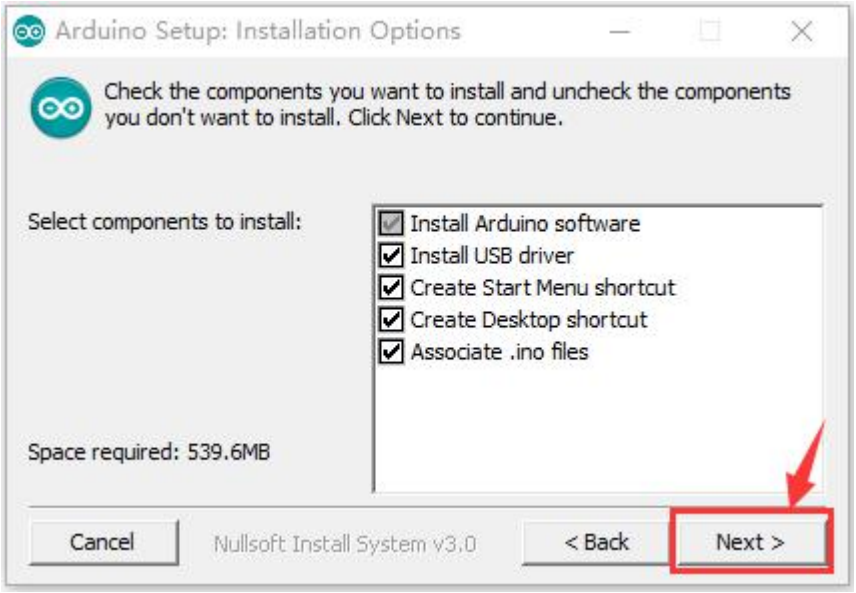
3.1. **Installing Arduino IDE**

Double click arduino-1.8.10-windows.exe to start.

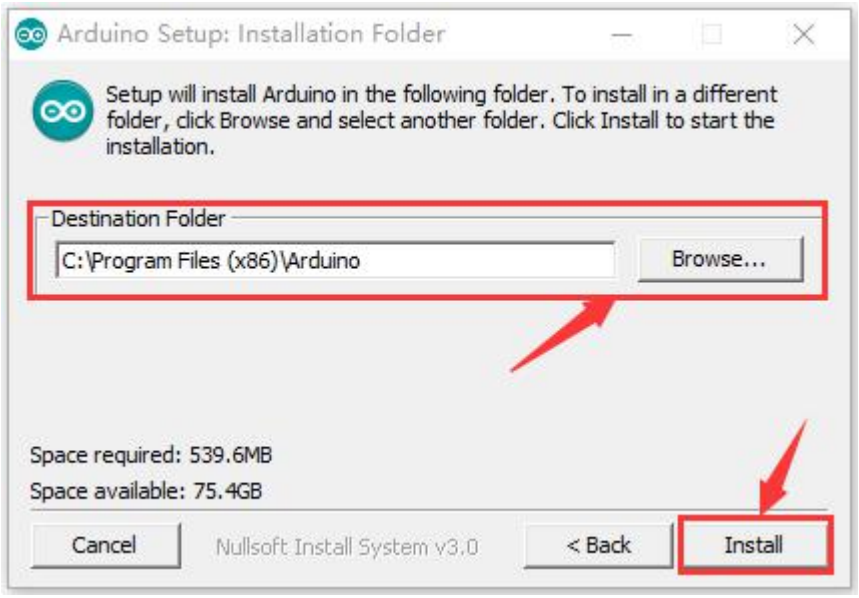
Select "I Agree"to accept license agreement.



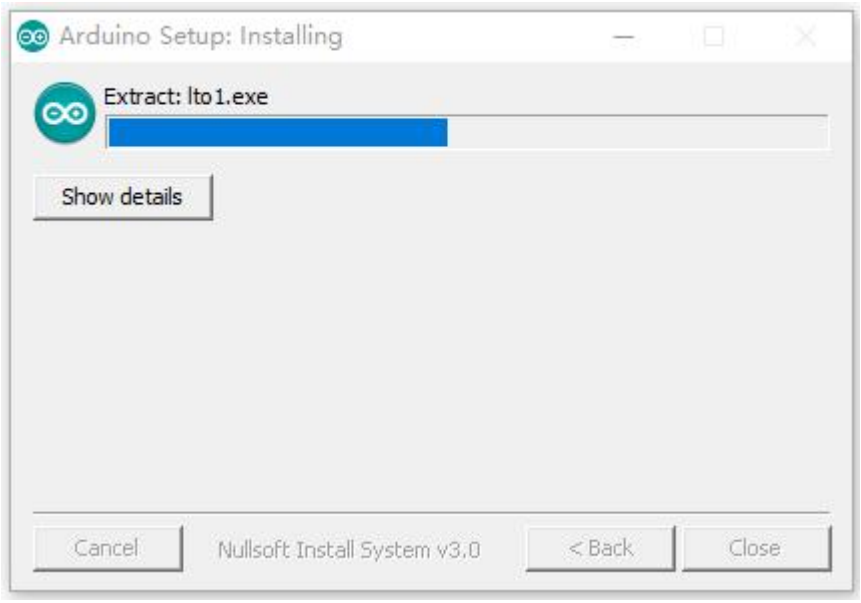
Select components to install and click "Next".

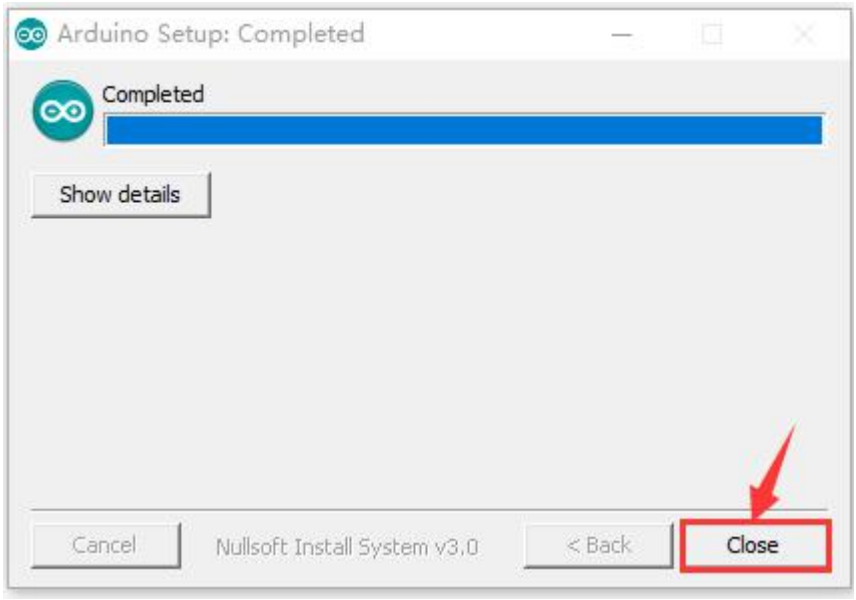


Click "Browse" and select another folder. Click "Install" to start the installation.



Finally, wait for a few minutes to finish.

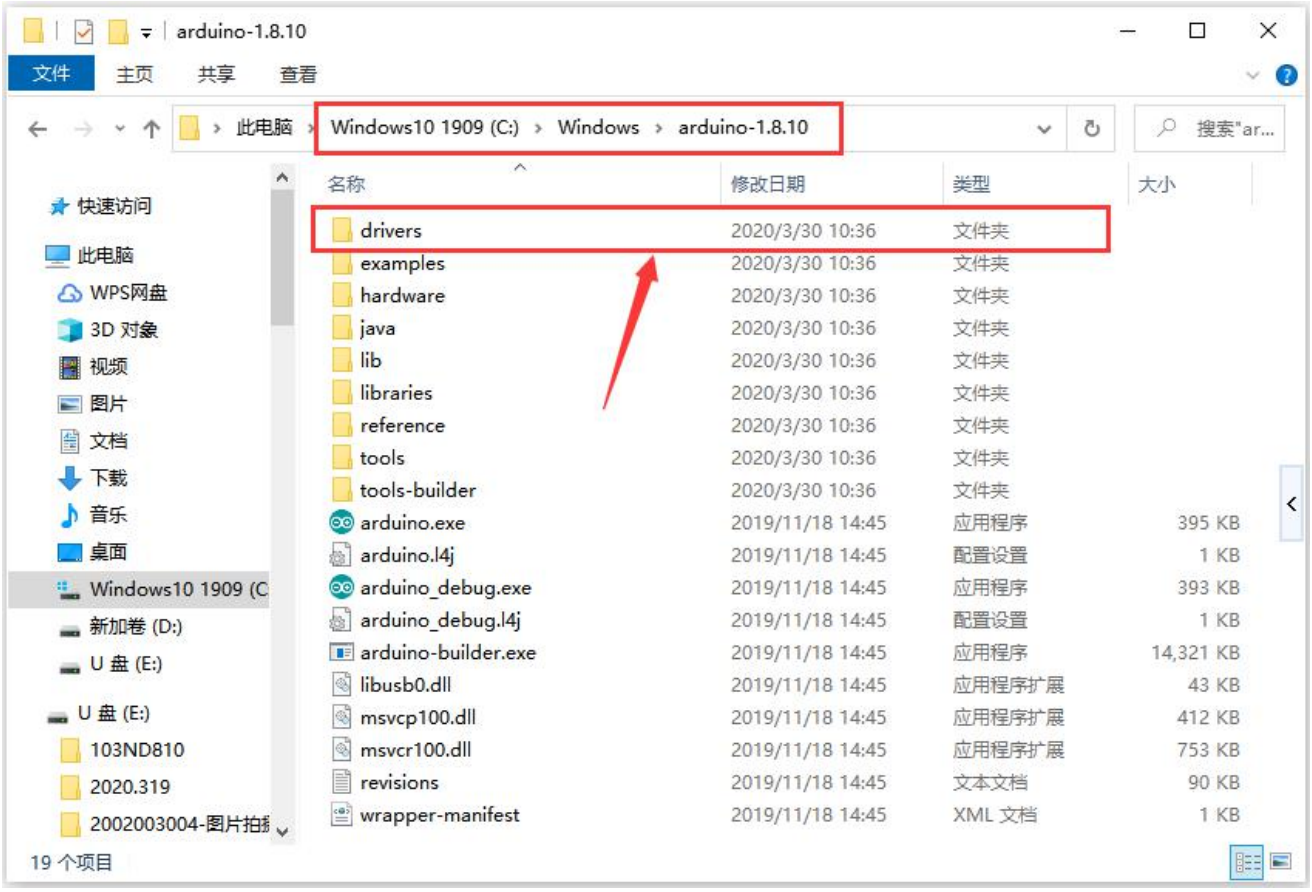


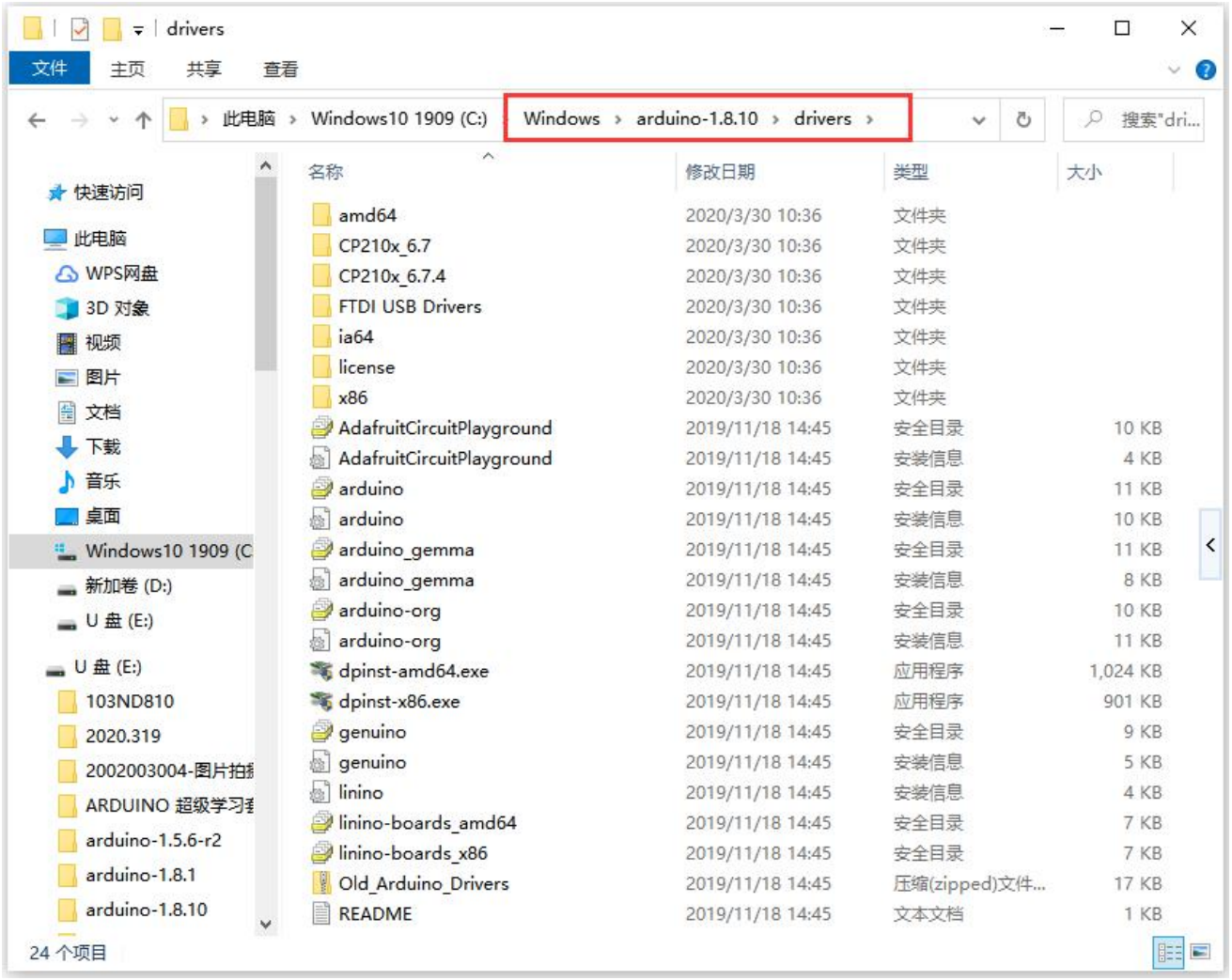


3.2 **Installing Driver**

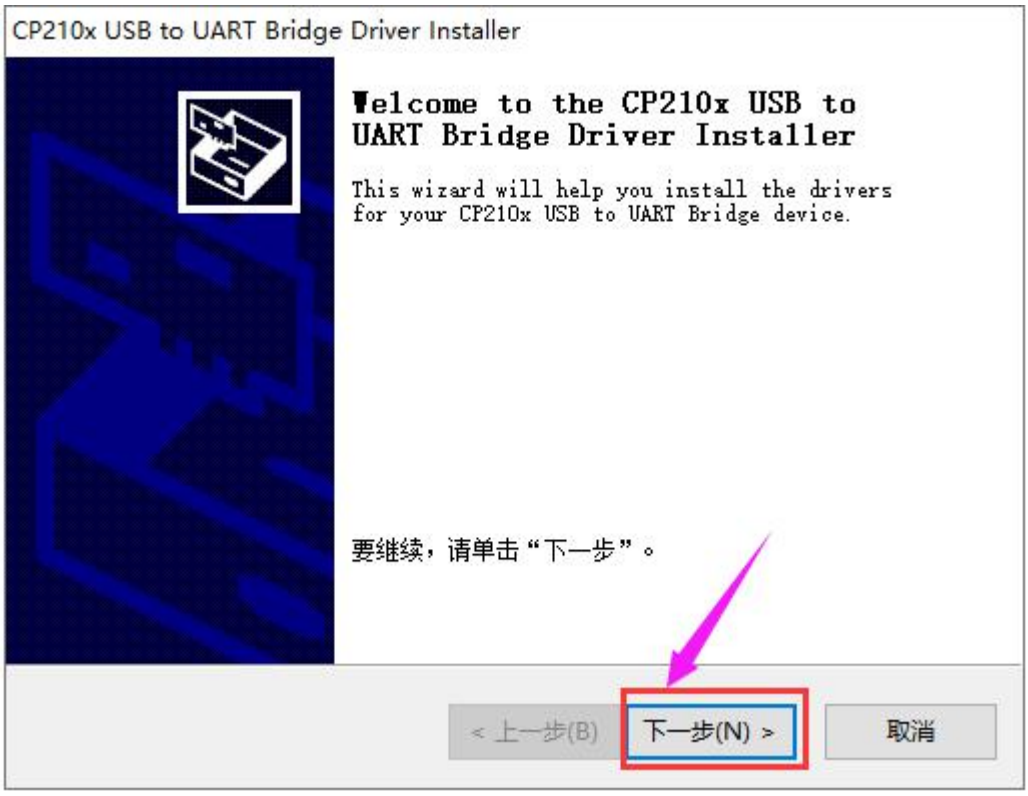
Finish the download, now, let's install the driver of keyestudio PLUS control board. Its chip is CP2102 serial chip.

(1) If download the Arduino1.8.x development software, the driver file is included in it.

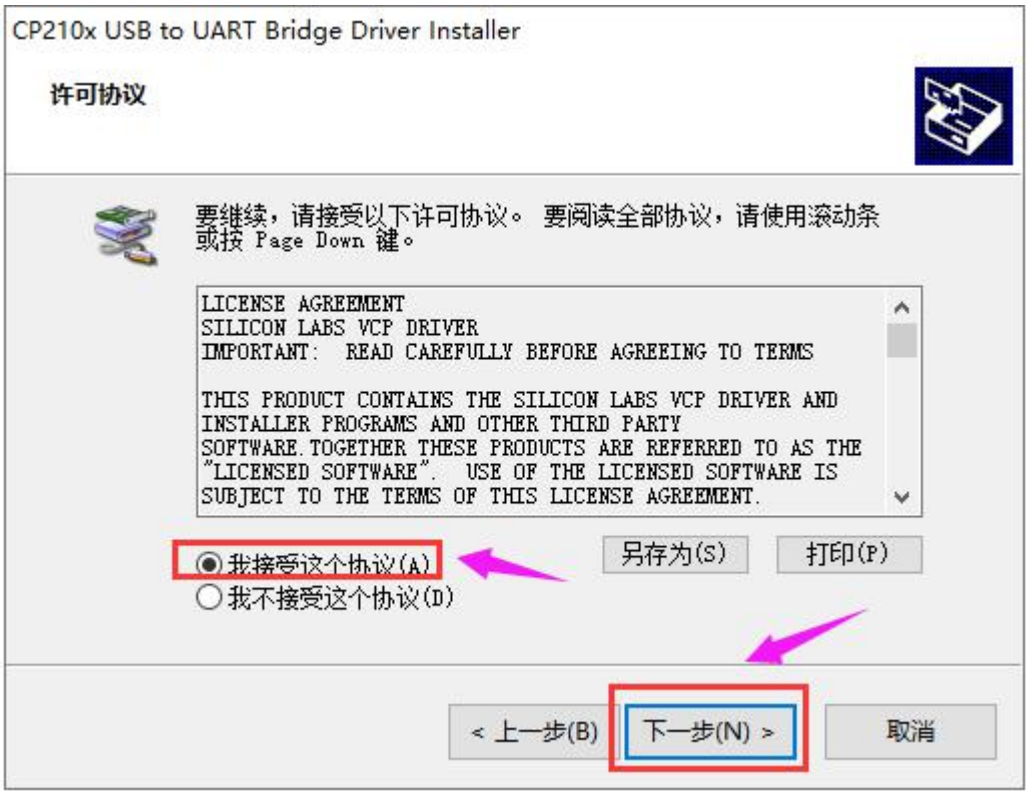

arduino-1.8.10
文件
主页
共享
查看
此电脑 › Windows10 1909 (C:) › Windows › arduino-1.8.10
搜索"ar...
快速访问
此电脑
WPS网盘
3D 对象
视频
图片
文档
下载
音乐
桌面
Windows10 1909 (C
新加卷 (D:)
U 盘 (E:)
U 盘 (E:)
103ND810
2020.319
2002003004-图片拍
19 个项目
名称
修改日期
类型
大小
drivers 2020/3/30 10:36 文件夹
examples 2020/3/30 10:36 文件夹
hardware 2020/3/30 10:36 文件夹
java 2020/3/30 10:36 文件夹
lib 2020/3/30 10:36 文件夹
libraries 2020/3/30 10:36 文件夹
reference 2020/3/30 10:36 文件夹
tools 2020/3/30 10:36 文件夹
tools-builder 2020/3/30 10:36 文件夹
arduino.exe 2019/11/18 14:45 应用程序 395 KB
arduino.l4j 2019/11/18 14:45 配置设置 1 KB
arduino_debug.exe 2019/11/18 14:45 应用程序 393 KB
arduino_debug.l4j 2019/11/18 14:45 配置设置 1 KB
arduino-builder.exe 2019/11/18 14:45 应用程序 14,321 KB
libusb0.dll 2019/11/18 14:45 应用程序扩展 43 KB
msvcp100.dll 2019/11/18 14:45 应用程序扩展 412 KB
msvcr100.dll 2019/11/18 14:45 应用程序扩展 753 KB
revisions 2019/11/18 14:45 文本文档 90 KB
wrapper-manifest 2019/11/18 14:45 XML 文档 1 KB



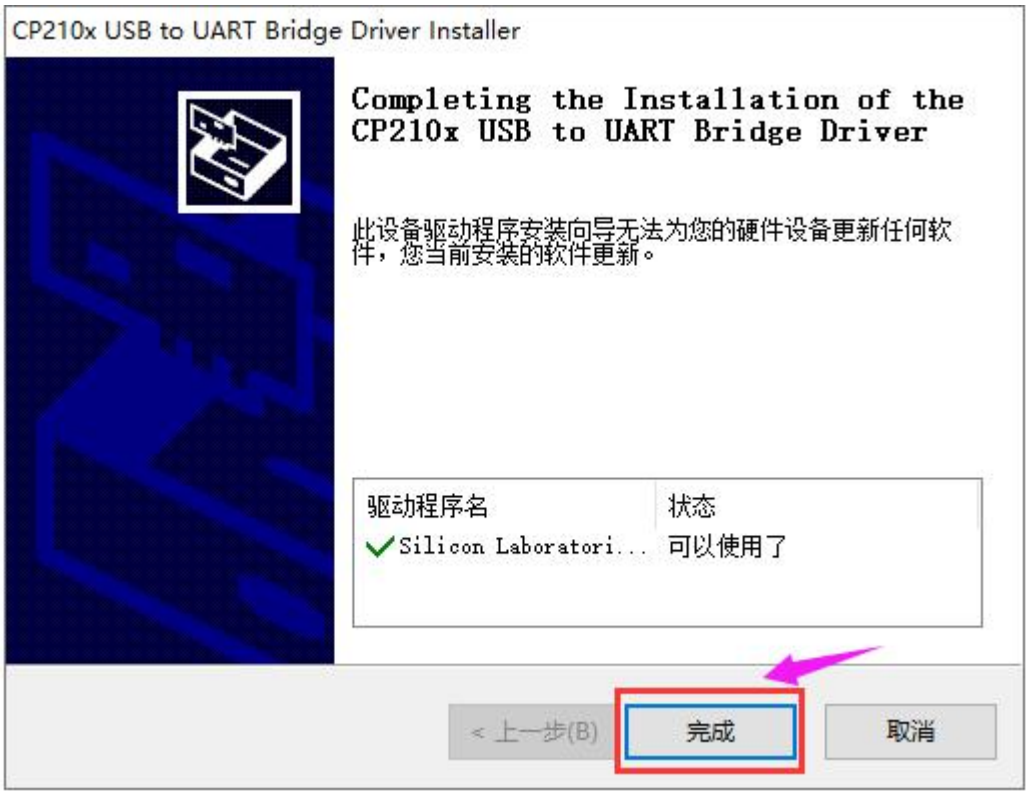
Double click dpinst-amd64.exe , click to next step, as shown below:



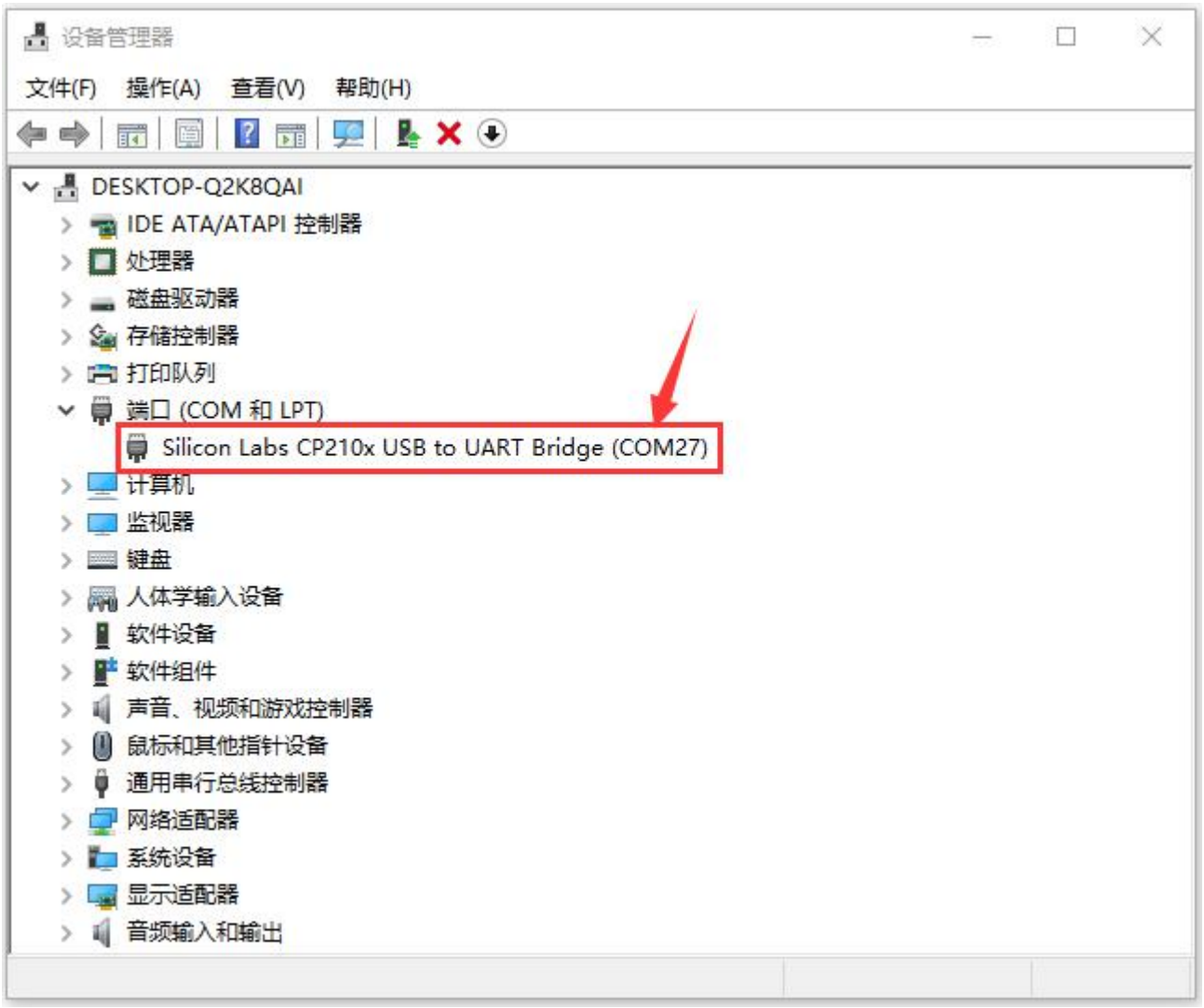
Click “Next step”, as shown below



Install successfully, click to “Finish” to end installation, as shown below.



Then you can right click "Computer" —>"Properties"—>"Device manager", you should see the corresponding COM port as the figure shown below.



(2) If download the Arduino1.6.x or Arduino1.5.x earlier edition, need to download the driver file.

There are driver files for each Windows system in the link:

https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers

(Note: choose the corresponding software according to Windows system of your computer)

Download for Windows 10 Universal (v10.1.8)

Note: The latest version of the Universal Driver can be automatically installed from Windows Update.

Platform	Software	Release Notes
Windows 10 Universal	Download VCP (2.3 MB)	Download VCP Revision History

Download for Windows 7/8/8.1 (v6.7.6)

Platform	Software	Release Notes
Windows 7/8/8.1	Download VCP (5.3 MB) (Default)	Download VCP Revision History
Windows 7/8/8.1	Download VCP with Serial Enumeration (5.3 MB) Learn More »	Download VCP Revision History

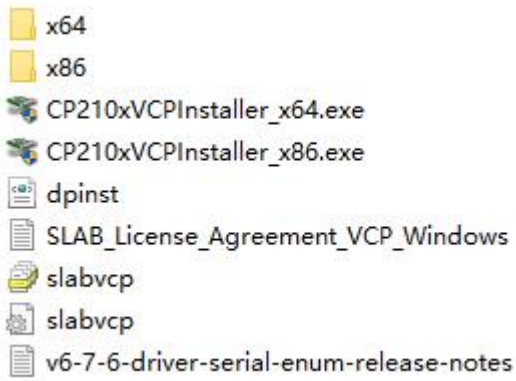
Download for Windows XP/Server 2003/Vista/7/8/8.1 (v6.7)

Platform	Software	Release Notes
Windows XP/Server 2003/Vista/7/8/8.1	Download VCP (3.66 MB)	Download VCP Revision History

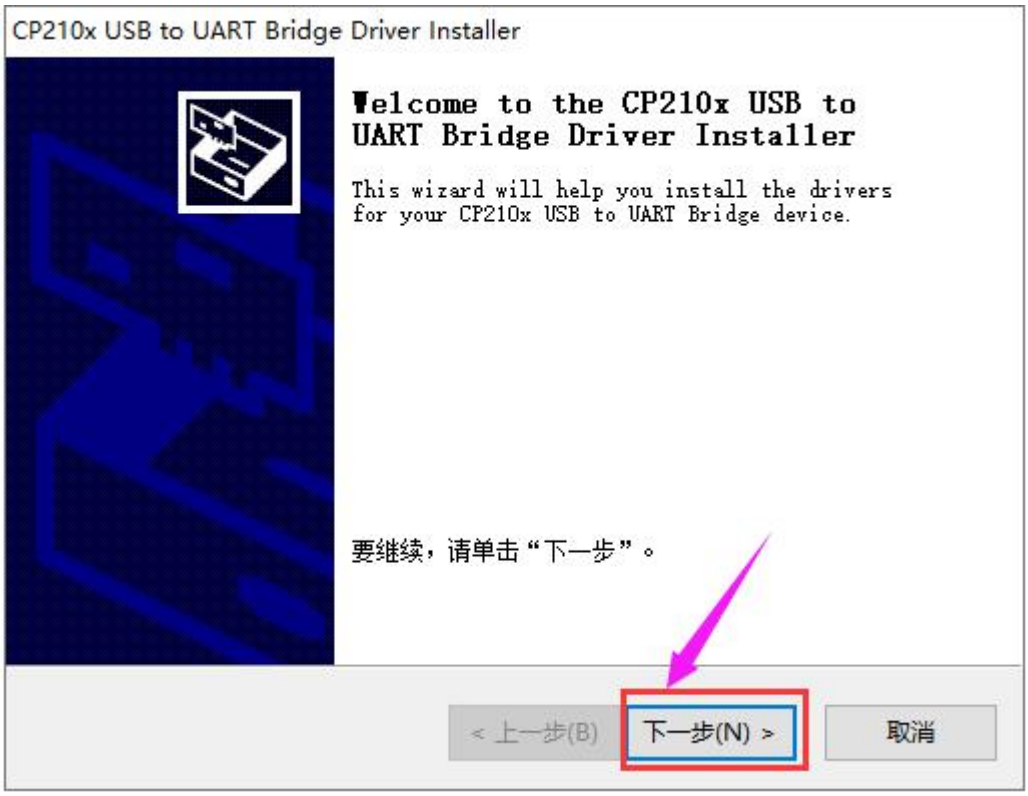
Please refer to the following link for installing driver on MAC system

https://v.qq.com/x/page/s0367yegedj.html

Next to show how to install the driver on Windows system, open driver file, as shown below.



Double click CP210xVCPInstaller_x64.exe (my own PC)，click "Next step", as shown below.

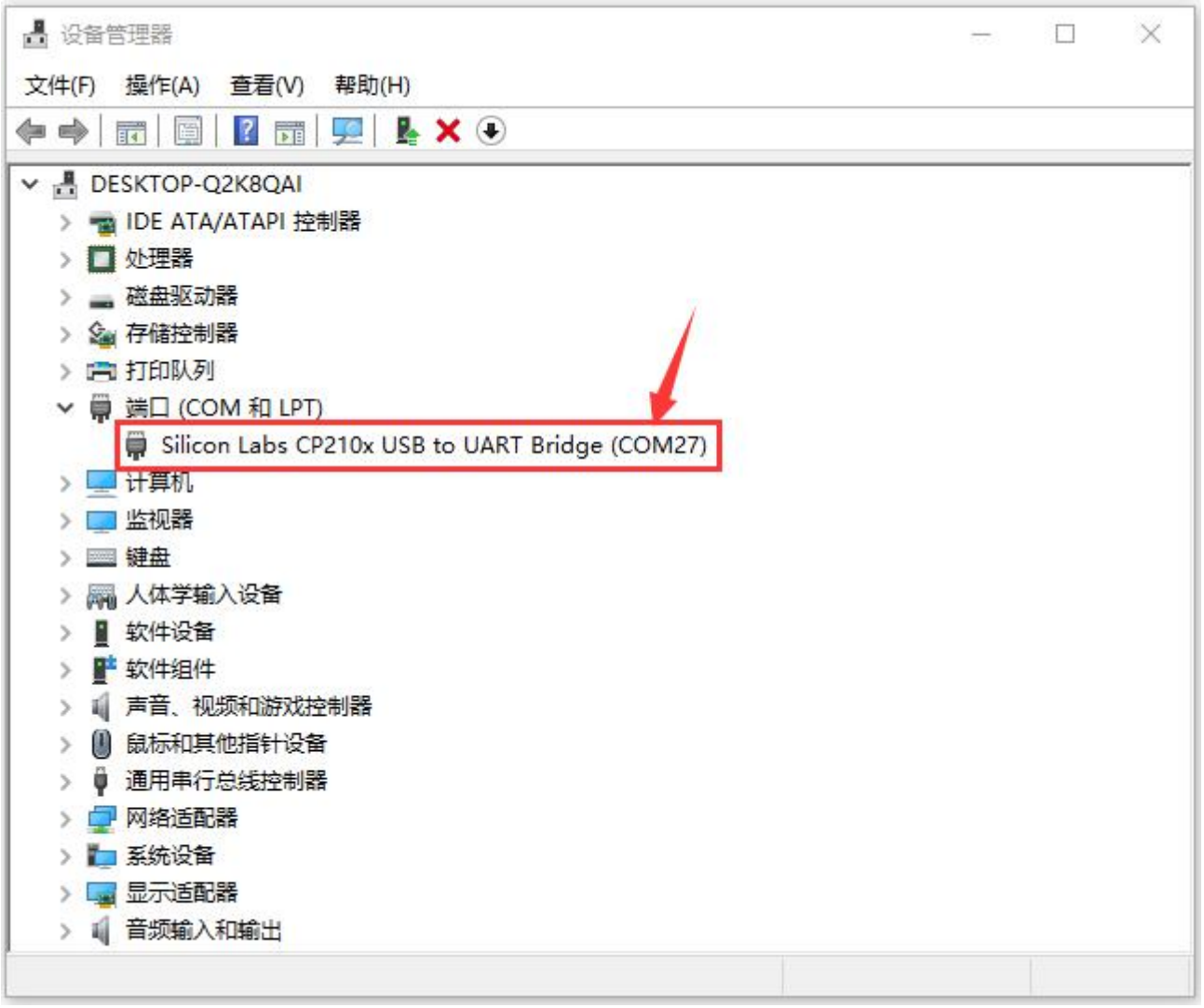


Click "Next step", as shown below

Install successfully, click to "Finish" to end installation, as shown below.

Then you can right click "Computer" —>"Properties"—>"Device manager", you

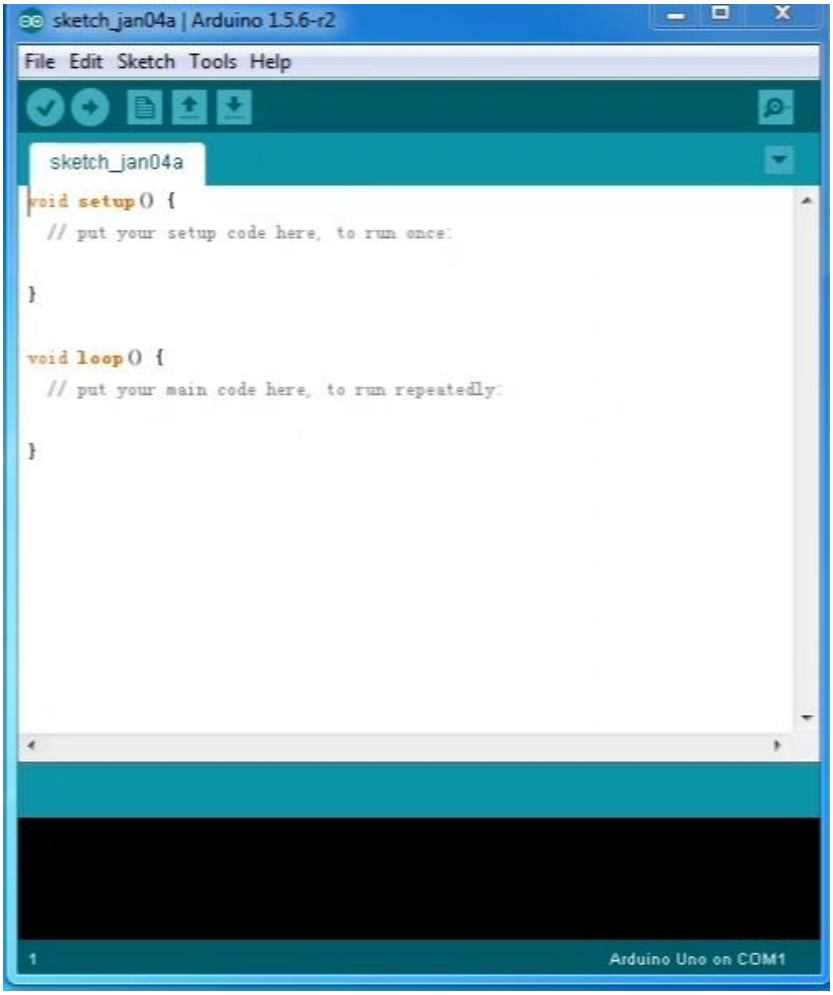
should see the corresponding COM port as the figure shown below.



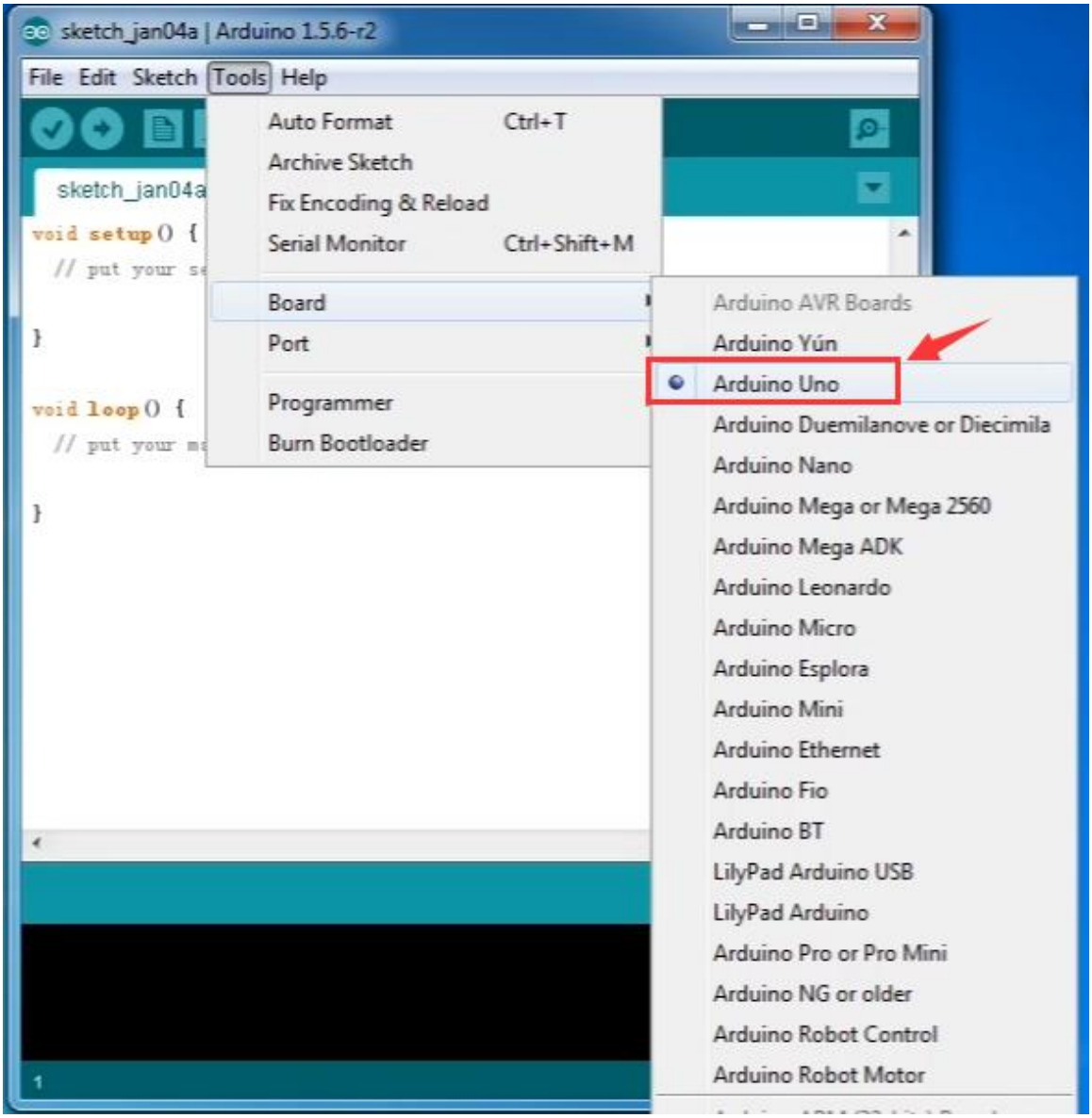
3.3 Arduino IDE Setting

Click 

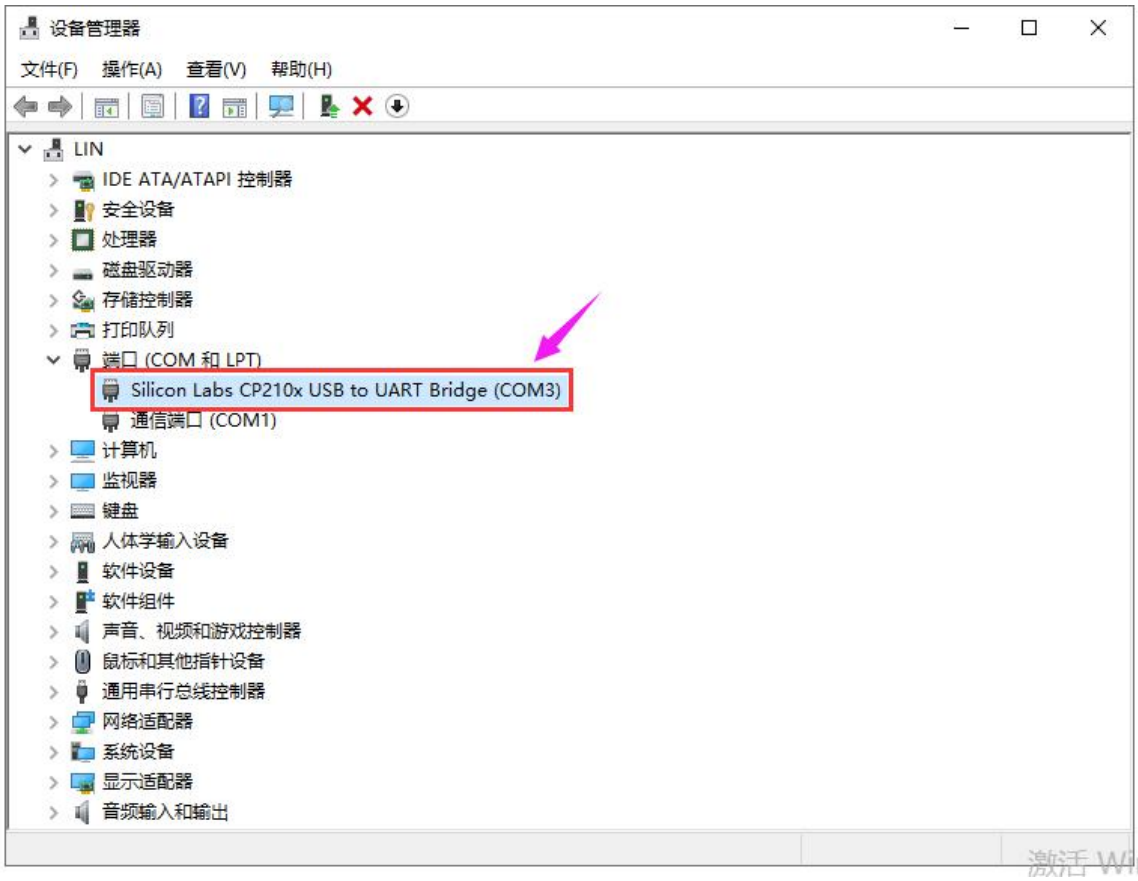
icon，open Arduino IDE.

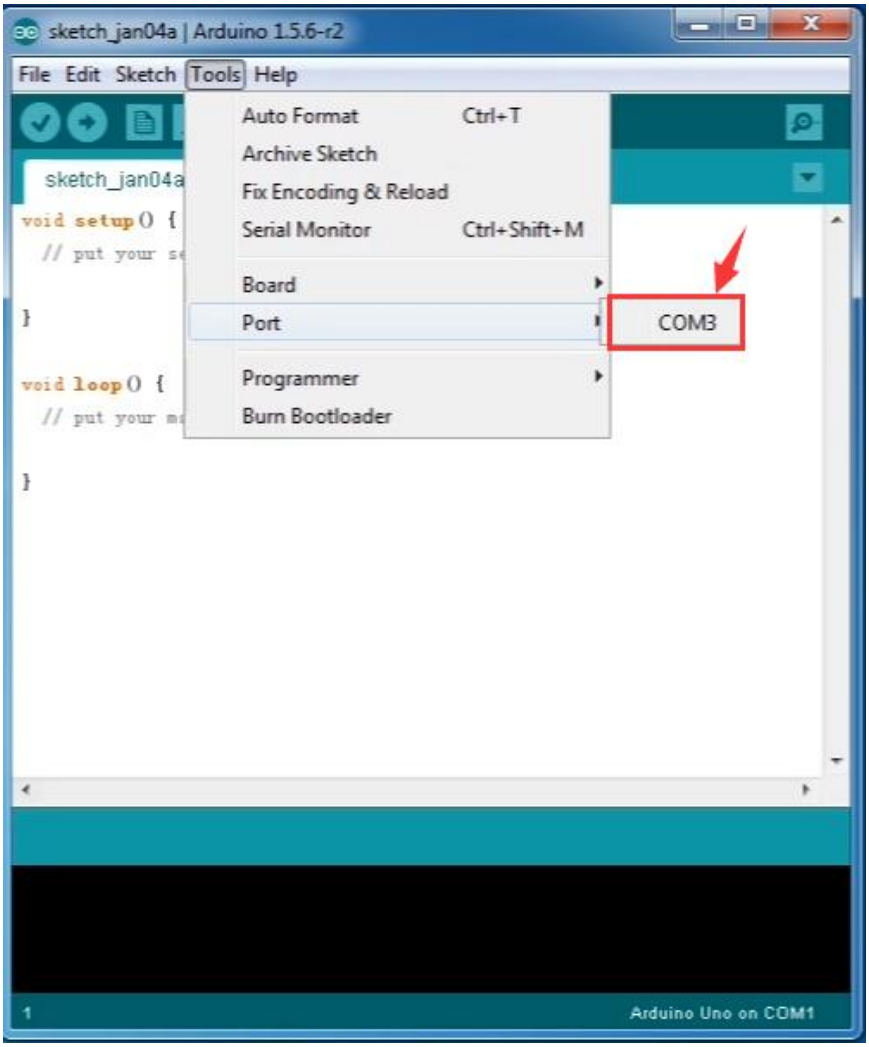


To avoid the errors when uploading the program to the board, you need to select the correct Arduino board that matches the board connected to your computer. Then come back to the Arduino software, you should click Tools→Board, select the board. (as shown below)

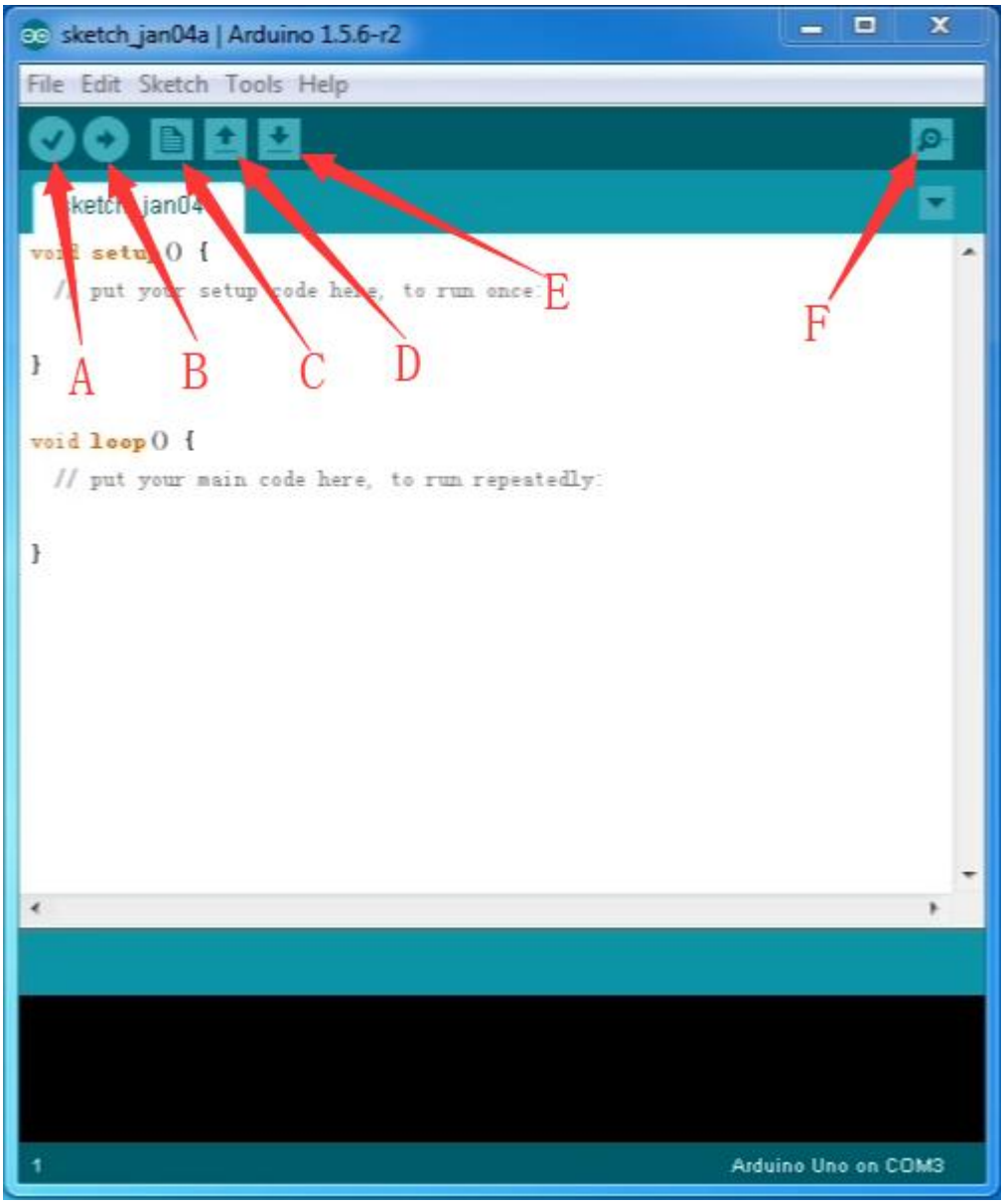


Then select the correct COM port (you can see the corresponding COM port after the driver is successfully installed)


设备管理器
文件(F) 操作(A) 查看(V) 帮助(H)
LIN
IDE ATA/ATAPI 控制器
安全设备
处理器
磁盘驱动器
存储控制器
打印队列
端口 (COM 和 LPT)
Silicon Labs CP210x USB to UART Bridge (COM3)
通信端口 (COM1)
计算机
监视器
键盘
人体学输入设备
软件设备
软件组件
声音、视频和游戏控制器
鼠标和其他指针设备
通用串行总线控制器
网络适配器
系统设备
显示适配器
音频输入和输出



Before uploading the program to the board, let's demonstrate the function of each symbol in the Arduino IDE toolbar.



A- Used to verify whether there is any compiling mistakes or not.

B- Used to upload the sketch to your Arduino board.

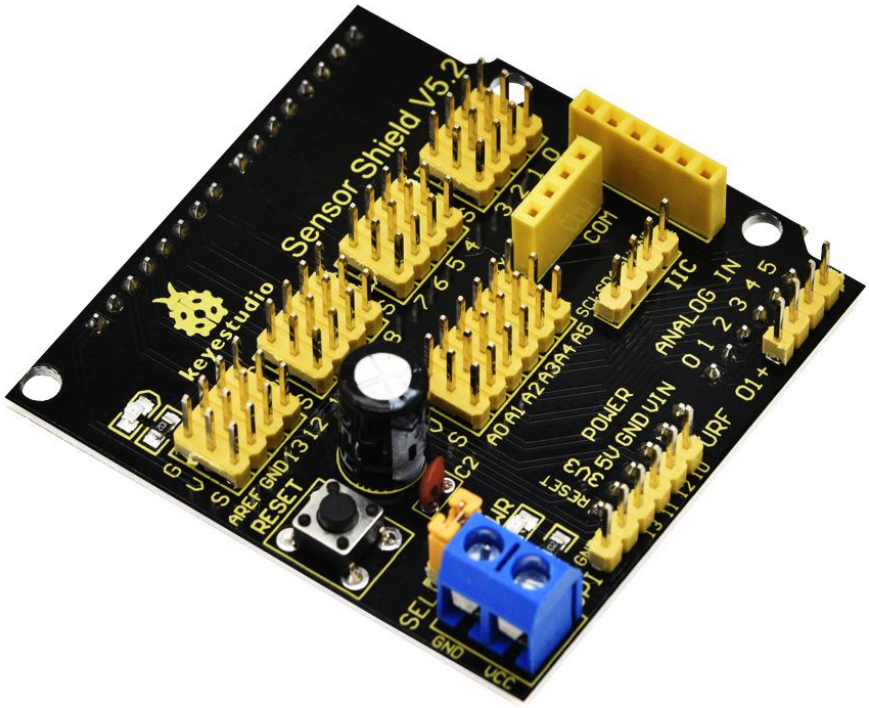
C- Used to create shortcut window of a new sketch.

D- Used to directly open an example sketch.

E- Used to save the sketch.

F- Used to send the serial data received from board to the serial monitor.

4. Keyestudio Sensor Shield V5.2



Description

In the experiments, you might often use ARDUINO control board and several sensor modules. If the interfaces of power output is not enough, you may need to add a breadboard and use many connection wires. Is it really troublesome？

But now, with this keyestudio sensor shield, you can easily solve that problem. This shield is fully compatible with keyestudio PLUS Control Board, so you can easily stack it onto PLUS Control Board.

This keyestudio sensor shield has extended the digital and analog ports out as 3PIN interface (G,V, S), which can directly connect 3PIN sensor modules.

It also breaks out some communication pins of 2.54mm pitch, like serial, IIC, and SPI communication.

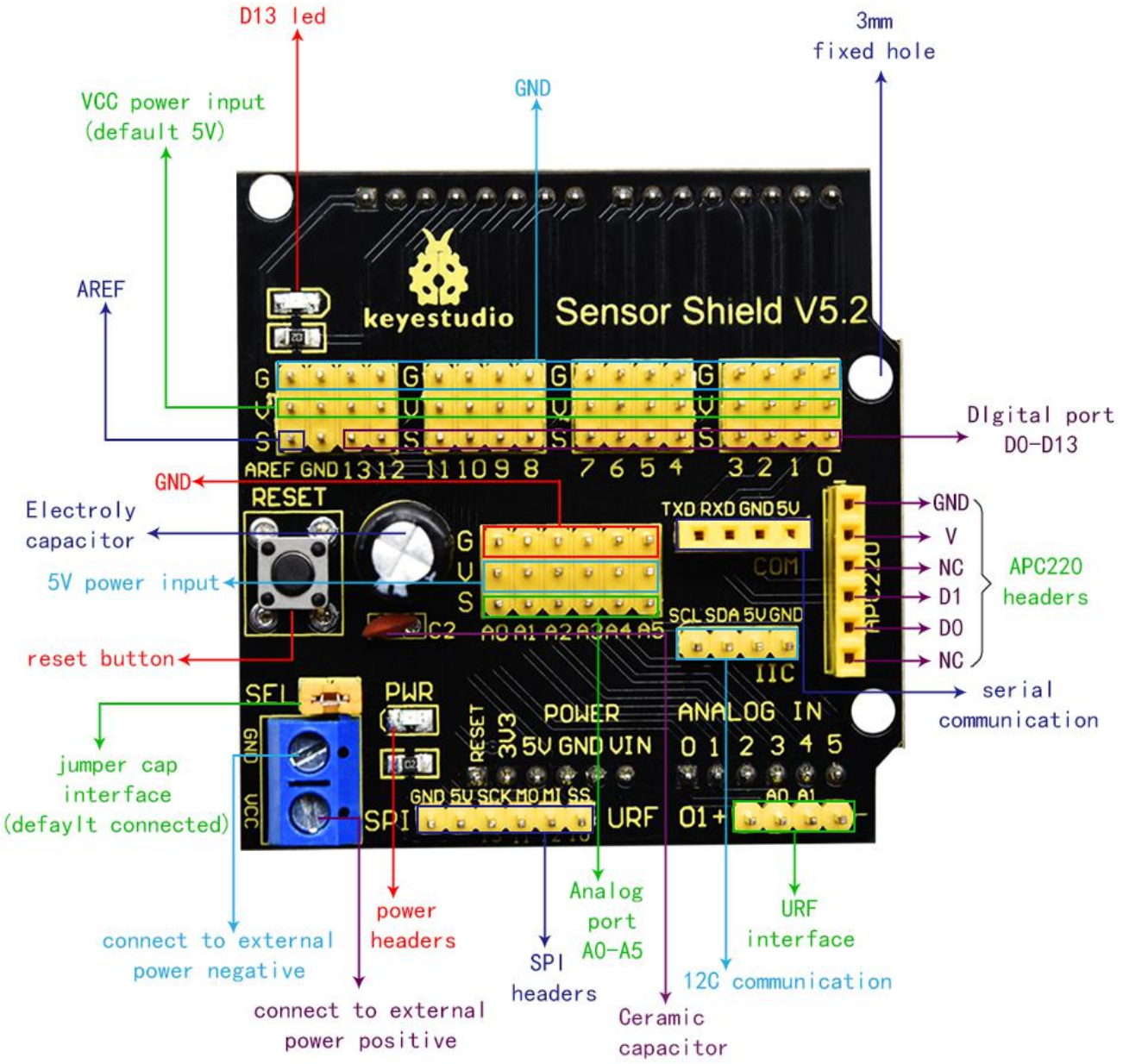
The shield comes with a reset button and 2 signal indicators as well.

Additionally, you can supply the voltage needed to the sensor modules through blue terminal blocks on the shield. Because some sensor modules is not used with 5V or 3.3V but with special voltage.

Parameter:

- Extends an Arduino Reset button
- Comes with 1 electrolytic capacitor and 1 ceramic capacitor
- Comes with a built-in power indicator and a D13 indicator
- Breakout all the digital and analog ports of keyestudio PLUS Control Board as 3PIN headers
- A serial communication interface
- A I2C communication interface
- A SPI communication interface
- Comes with a URF interface
- Comes with an APC220 interface
- You can supply the voltage needed for sensor modules via terminal blocks.

Pins Description:



When SE is connected with jumper cap, and input DC 7V to VCC /GND terminal block, so the voltage of V, V1 and + pins are 7V.

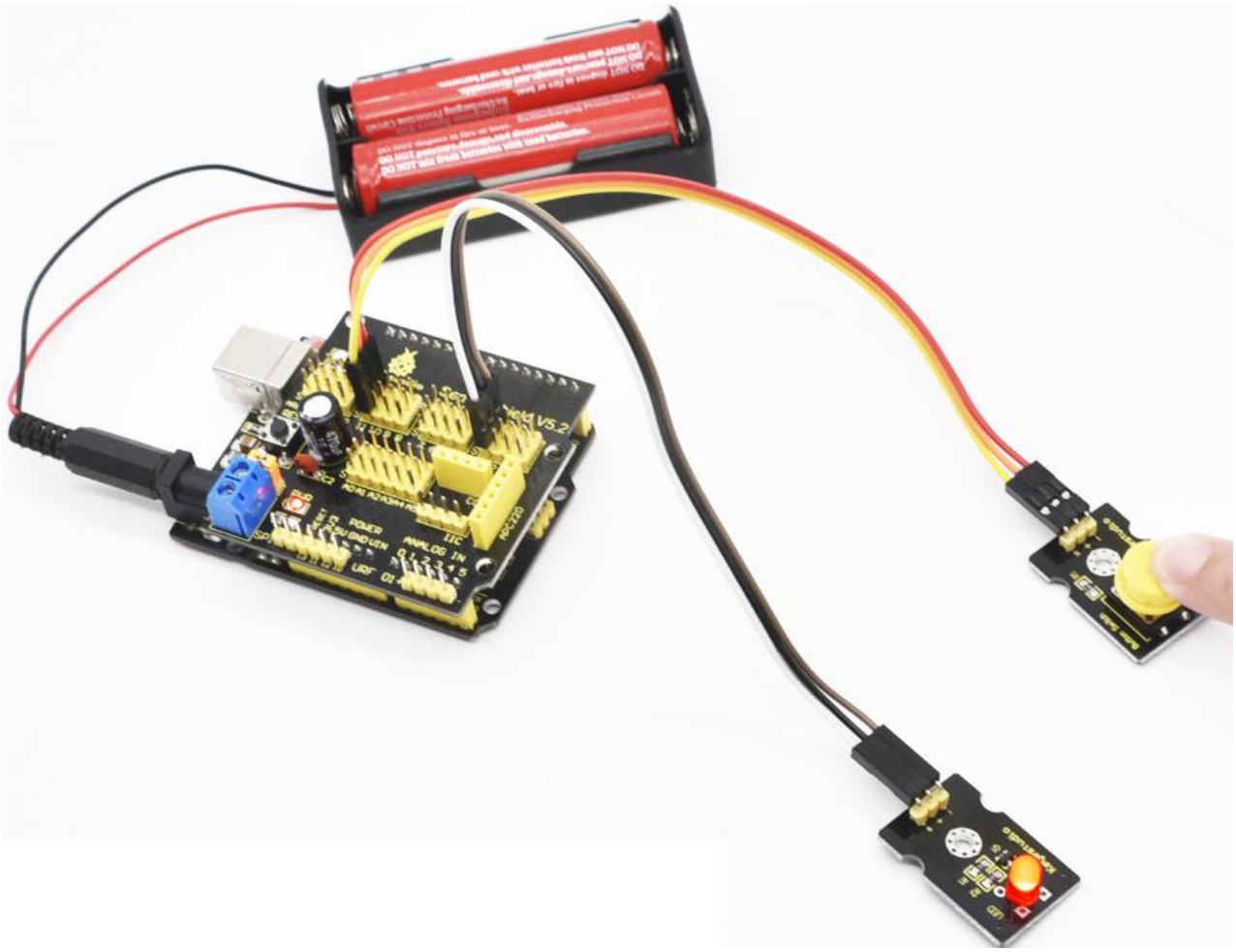
When SE is connected with jumper cap, and VCC /GND terminal block without voltage input, shield is powered via PLUS control board, so the voltage of V, V1 and + pins are 5V.

When SE is disconnected, input DC 7V to VCC /GND terminal block, so the voltage of V pin is 7V, the voltage of V1 and + pins are 0V.

When SE is disconnected, and VCC /GND terminal block without voltage input, shield is powered via PLUS control board, so the voltage of V pin is 0V, the voltage of V1 and + pins are 5V.

Example Use

Stack the Keyestudio Sensor Shield V5 on the Keyestudio PLUS Control Board, and build the circuit on the shield, as shown below.



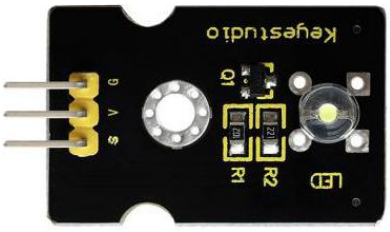
5. Projects

When you get this kit, there are 14 sensors/modules inside like keyestudio PLUS control board, sensor shield and dupont Lines.

We firstly stack Keyestudio Sensor Shield V5.2 on the keyestudio PLUS control board and connect 14 sensors to sensor shield via dupont lines, upload the test code and check the result.

Note: In this course, the interface of each sensor / module marked with (G,-, GND) indicates the negative pole, and G and GND are connected to Arduino board or the sensor shield; V, +, and VCC stands for positive pole, they are linked with V, VCC and 5V of Arduino board or sensor shield.

Project 1: LED Blink



Description:

In this kit, there are white and yellow LED modules. Their operating method is same. In this lesson, we directly connect the white LED module to the sensor

shield with three female-to-female DuPont cables. After wiring, we can upload code on the keyestudio PLUS control board to control white LED to display different colors.

After GND and VCC are powered on, the LED lights up when the signal terminal S is HIGH, and the LED turns off when signal end S is LOW.

Specifications:

Control interface: digital port

Working voltage: DC 3.3-5V

Pin pitch: 2.54mm

LED display color: white

Size: 30 * 20mm

Weight: 3g

Equipment:

White LED module * 1

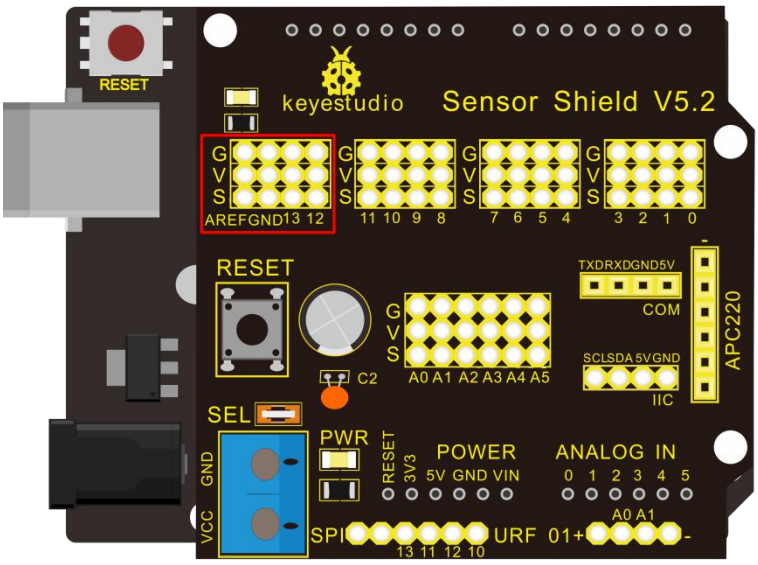
Keyestudio PLUS Control Board * 1

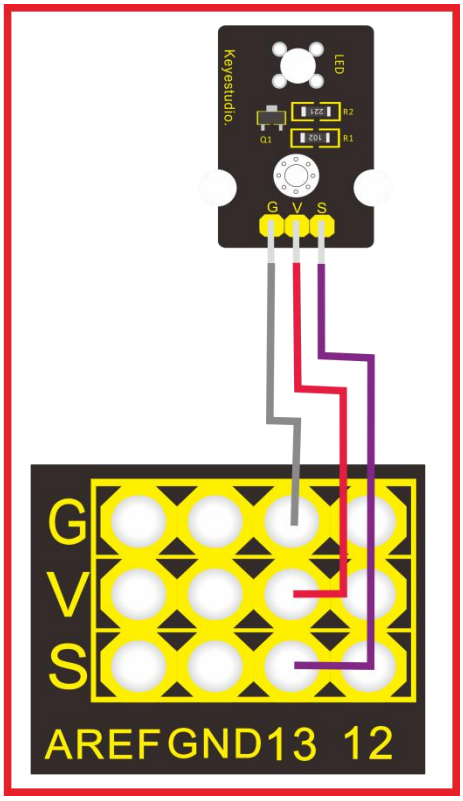
Sensor shield * 1

3pin female to female Dupont lines

USB cable * 1

Connection Diagram:





Note: The G, V, and S ports of the white LED module are separately connected to G, V, and 13, then power on.

Test Code:

```
**********************************************************************
**********************************
void setup()
{
  pinMode(13, OUTPUT);//set digital 13 to output}
```

```
void loop()
{
  digitalWrite(13,HIGH);//set digital 13 to HIGH level，LED lights up
  delay(1000);       //delay in 1000ms
  digitalWrite(13,LOW);//set digital13 to LOW level，LED lights off
  delay(1000);             //delay in 1000ms
}
```

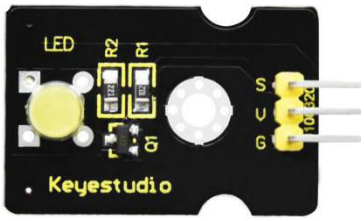
****

****

Test Result:

Upload the test code successfully. After power on, the white LED starts blinking, light up for 1,000ms, turn off for 1000ms,and alternately.

Project 2: LED Breathe



Description

In this project, we will control the red LED brightness via PWM.

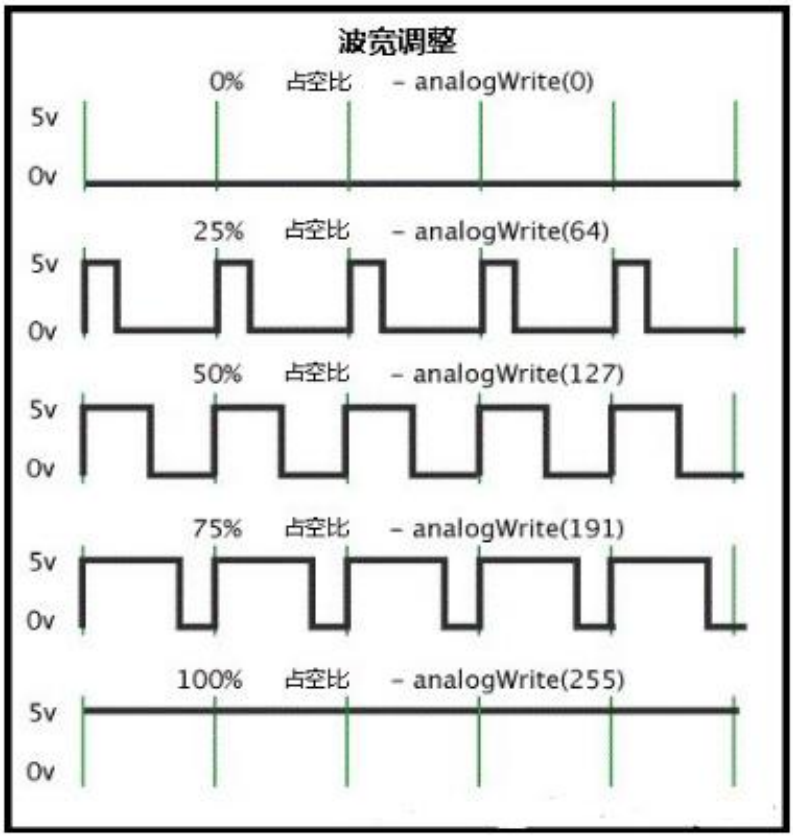
In the previous course, we introduced how to control the yellow LED on and off by code. Similarly, you can change the step length and delay in the code to set the red LED to achieve different breathing effects.

PWM is a means of controlling the analog output via digital means. Digital control is used to generate square waves with different duty cycles (a signal that constantly switches between high and low levels) to control the analog output. In general, the input voltage of port are 0V and 5V. What if the 3V is required? Or what if switch among 1V,3V and 3.5V? We can't change resistor constantly. For this situation, we need to control by PWM.

For the Arduino digital port voltage output, there are only LOW and HIGH, which correspond to the voltage output of 0V and 5V. You can define LOW as 0 and HIGH as 1, and let the Arduino output 500 0 or 1 signals within 1 second.

The output port is 2.5V, which is like showing movie. The movie we watch are not completely continuous. It actually outputs 25 pictures per second. In this case, the human can' t tell it. So does PWM. If want different voltage, need to control the ratio of 0 and 1. The more 0,1 signals output per unit time, the more accurate the control.

In the following figure, the green line represents a period, and its value is the reciprocal of the PWM frequency. In other words, if the frequency of the Arduino PWM is 500 Hz, then the period between the two green lines is 2 ms. The range that can be manipulated in the analogWrite () command is 0-255, analogWrite (255) means 100% duty cycle (normally open), and analogWrite (127) duty cycle is about 50% (half the time).



Specifications:

Control interface: digital port

Working voltage: DC 3.3-5V

Pin pitch: 2.54mm

LED display color: yellow

Size: 30 * 20mm

Weight: 3g

Equipment:

Yellow LED module * 1

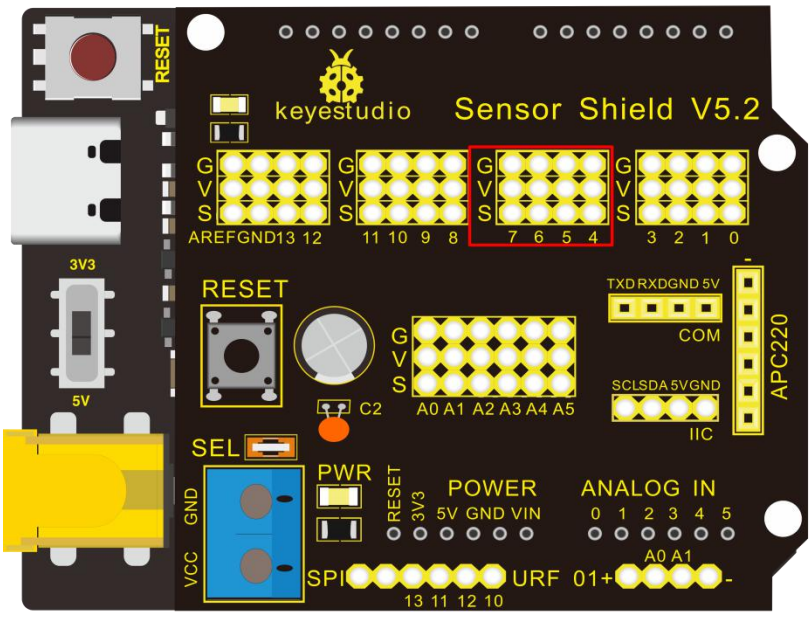
Keyestudio PLUS Control Board * 1

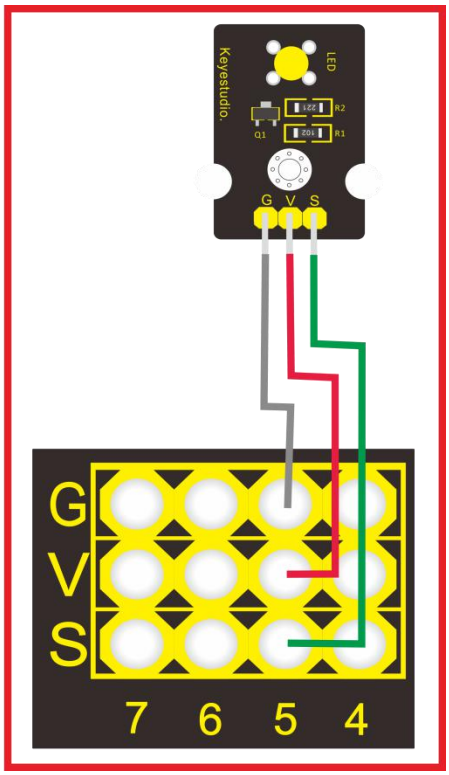
Sensor shield * 1

3pin female to female DuPont lines

USB cable * 1

Connection Diagram:





Note: The G, V, and S ports of the yellow LED module are separately

connected to G, V, and 13, then power on.

On the sensor expansion board, the G, V, and S pins of the yellow LED module are connected to G, V, and 5 respectively, and the power supply is connected.

Test Code:

```
**********************************************************************
**********************************
  void setup(){
  analogWrite(5,0);//set the PWM value of digital5 to 0
}

void loop(){
  for (int i = 0; i <= 255; i = i + (51))
// variable i increases from 0 to 255, add 51 each time
{
    analogWrite(5,i);//set variable i to the PWM value of digital5
    delay(500);//delay in 500ms
  }
  for (int i = 255; i >= 0; i = i + (-51))
```

```
//variable i reduces from 255 to 0，plus -51 each time
 {
    analogWrite(5,i);//set variable i to the PWM value of digital5
    delay(500);//delay in 500ms
  }}
```

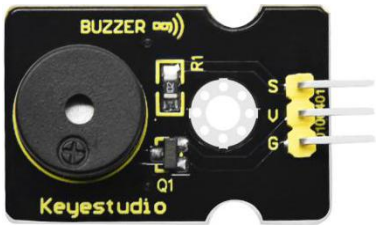
****

****

Test Result:

Upload test code successfully and power on, yellow LED gradually become brighter and darker to simulate the human breath.

Project 3: Passive Buzzer



1. Description

There are many interactive works that can be completed with Arduino. The most common element is buzzer and speaker. Buzzer is easy to use. And buzzer

concludes in active buzzer and passive buzzer. In this experiment, we make the buzzer sound and play music by test code.

Specifications:

Control Port: digital

Working Voltage: DC 3.3-5V

Equipment

Keyestudio PLUS Control Board * 1

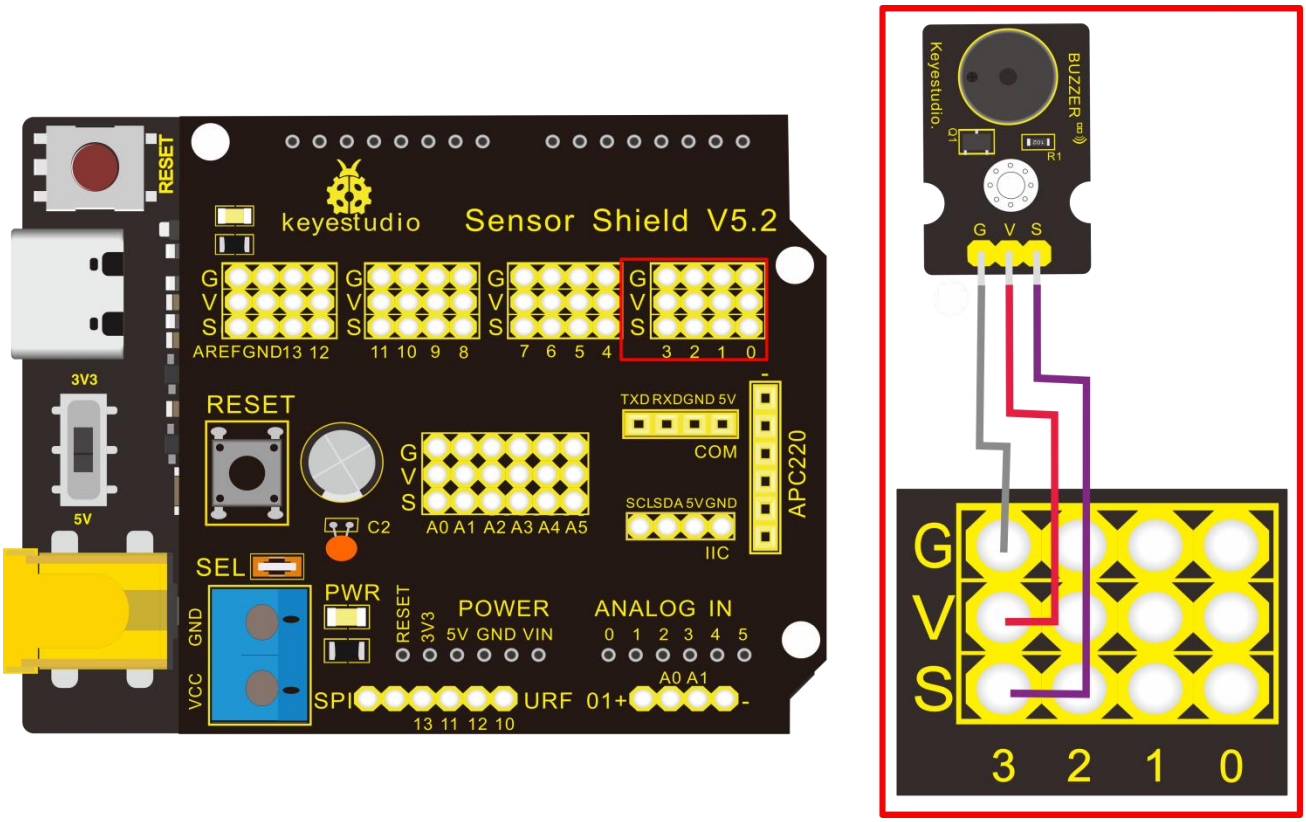
Sensor shield * 1

Passive buzzer module * 1

USB cable * 1

3pin female to female Dupont cable * 1

Connection Diagram:



Note: The G, V, and S port of the passive buzzer module are separately connected to G, V, and 3 on the shield, power up.

Test Code:

```
**********************************************************************
**********************************
```

```
    void setup(){
  pinMode(3, OUTPUT);//set digital3 to output
}
```

```
void loop(){
   tone(3,262);//digital3 outputs the sound of 262Hz
   delay(250);//delay in 250ms
   tone(3,294);;//digital3 outputs the sound of 294HZ
   delay(250);//delay in 250ms
   tone(3,330);
   delay(250);
   tone(3,349);
   delay(250);
   tone(3,392);
   delay(250);
   tone(3,440);
   delay(250);
   tone(3,494);
   delay(250);
   tone(3,532);
   delay(250);
   noTone(3);//digital3 turns off sound output
   delay(1000);
}
```

****

****

Test Result:

Upload test code on keyestudio PLUS Control Board, wire according to connection diagram, passive buzzer sounds "do re mi fa so la si do".

Project 4: Controlling LED By Button Module

Description:

In this project, we will control LED to light on and off via button module. When the button is pressed, the signal end outputs low level (0); when released, the signal end of sensor keeps high level(1).

Specifications:

Working voltage: DC 3.3-5V

Control signal: digital signal

Size: 34 * 22 * 15mm

Weight: 3.8g

Equipment

Keyestudio REV4 development board * 1

Sensor shield * 1

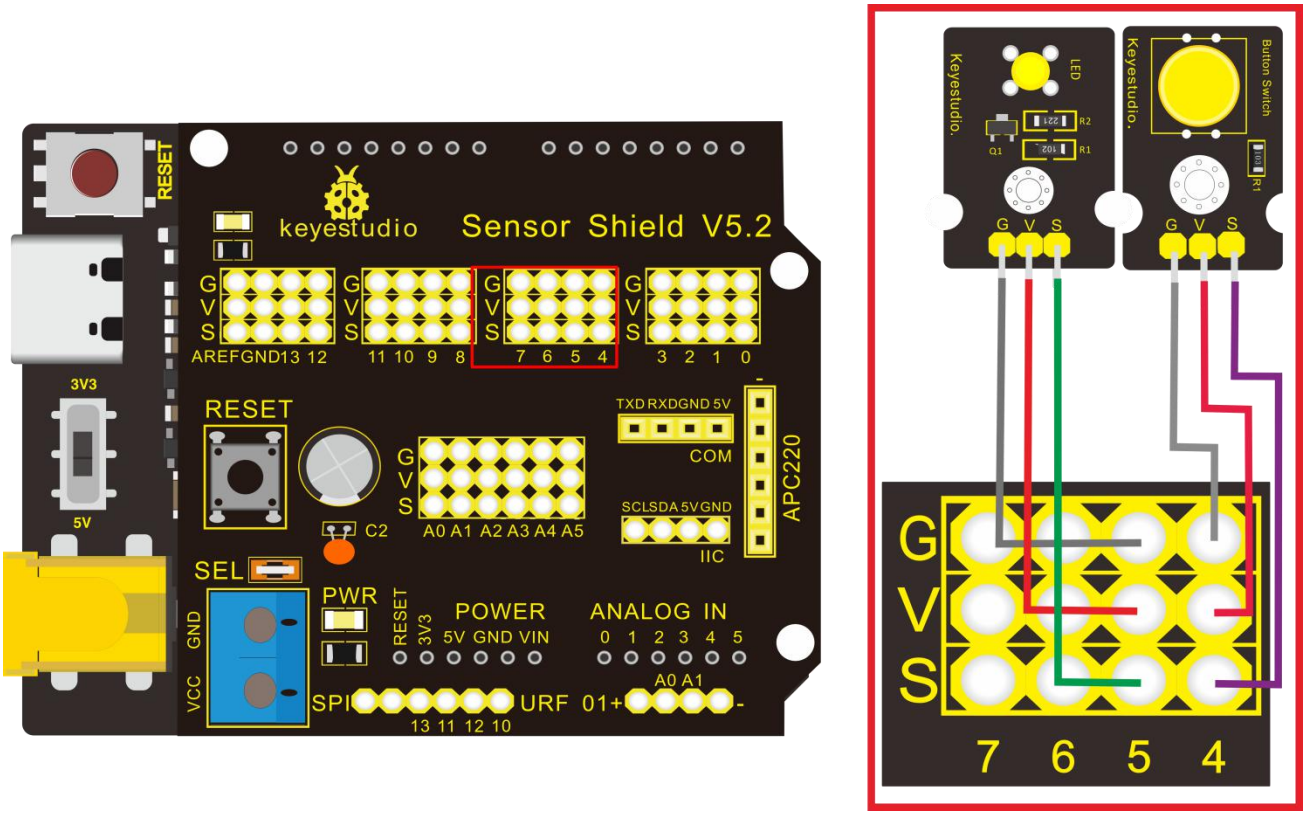
Yellow LED module * 1

Button module * 1

USB cable * 1

3pin female to female DuPont Lines *2

Connection Diagram:



Note: The G, V, and S pins of button sensor module are separately connected to G, V, and 4 on the shield, and the G, V, and S pins of the yellow LED module are connected to G, V, and 5 on the shield.

Test Code:

```
**********************************************************************
**********************************
```

```
void setup(){
   Serial.begin(9600);//set baud rate to 9600
   pinMode(4, INPUT);//set digital4 to input
   pinMode(5, OUTPUT);//set digital5 to output
}

void loop(){
   Serial.println(digitalRead(4));//wrap word and read the value of digital4
   delay(500);//delay in 500ms
   if (digitalRead(4) == 0)
//when read the value of digital4 to 0
 {
      digitalWrite(5,HIGH);//set digital5 to HIGH level, LED lights up
   }
else
//otherwise  (when the value of digital4 is 1)
 {
```

```
    digitalWrite(5,LOW);//set digital5 to LOW level, LED lights off
}}
**********************************************************************
*******************************
```

Test Result:

Upload test code, wire according to connection diagram, when the button is pressed, serial monitor displays 0 (low level), yellow LED is on; when released, serial monitor displays 1 (high level), yellow LED is off.

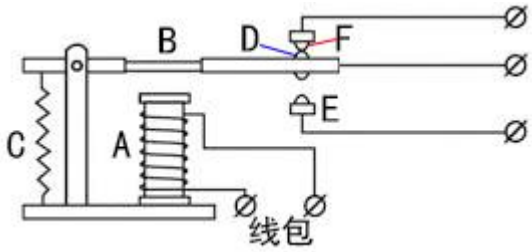
Project 5: 1-channel Relay Module

Description

This module is an arduino dedicated module, and compatible with arduino sensor expansion board. It has a control system (also called an input loop) and a controlled system (also called an output loop). Commonly used in automatic

control circuits, it is an "automatic switch" for high voltage which control a larger current and a lower voltage with a smaller current and a lower voltage.

Therefore, it plays the role of automatic adjustment, safety protection and conversion circuit in the circuit. It allows Arduino to drive loads below 3A, such as LED light strips, DC motors, miniature water pumps, solenoid valve pluggable interface.



The main internal components of the electromagnetic relay are electromagnet A, armature B, spring C, moving contact D, static contact (normally open contact) E, and static contact (normally closed contact) F, (as shown in the figure).

As long as a certain voltage is applied to both ends of the coil, a certain current will flow through the coil to generate electromagnetic effects, and the armature will attract the iron core against the pulling force of the return spring under the action of electromagnetic force attraction, thereby driving the moving contact and the static contact (normally open contact) to attract. When the coil is disconnected, the electromagnetic suction will also disappear, and the armature will return to the original position under the reaction force of the spring, releasing the moving contact and the original static contact (normally closed contact). This

pulls in and releases, thus achieving the purpose of turning on and off in the circuit. The "normally open, normally closed" contacts of the relay can be distinguished in this way: the static contacts on off state when the relay coil is not energized are called "normally open contacts"; the static contacts on state are called "normally closed contact". The module comes with 2 positioning holes for you to fix the module to other equipment.

Specifications:

Working voltage: 5V (DC)

Interface: G, V, S interface

Input signal: digital signal (high level 1, low level 0)

Contacts: static contacts (normally open contacts, normally closed contacts) and moving contacts

Rated current: 10A (NO) 5A (NC)

Maximum switching voltage: 150 V (AC) 24 V (DC)

Electric shock current: less than 3A

Weight: 15g

Contact action time: 10ms

Equipment:

keyestudio PLUS Control Board*1

Sensor expansion board * 1

1-channel relay module * 1

White LED * 1

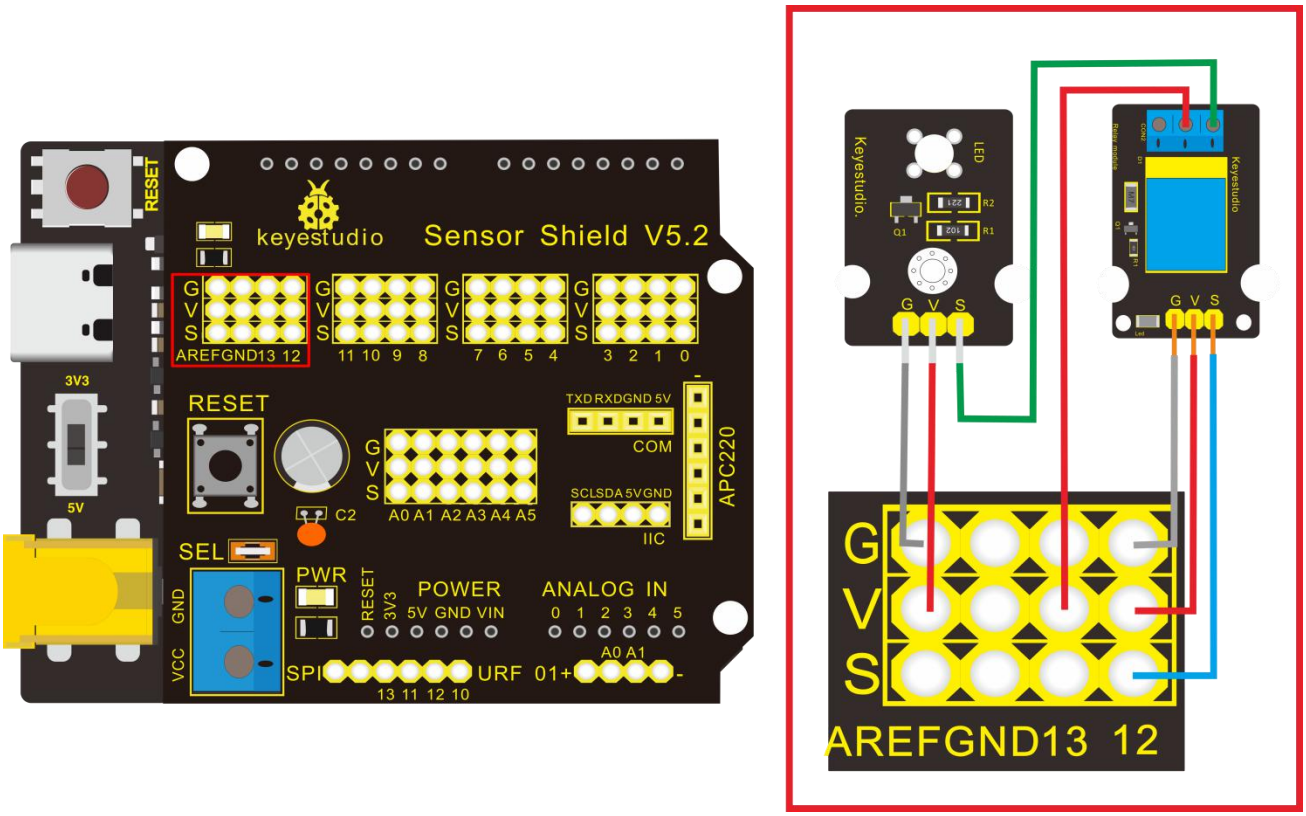
female to female Dupont line * 2

Male to female Dupont line * 2

3pin female to female Dupont cable * 1

USB cable * 1

Connection Diagram:



Note: On the shield, the G, V, and S pins of 1-channel relay module are connected to G, V, and 12 respectively. The NO is connected to V; the G, V, and S pins of white LED are respectively connected to G, V, and the NO of relay module.

Test Code:

```
**********************************************************************
**********************************
```

```
void setup(){
  pinMode(12, OUTPUT);//set digital12 to output}
```

```
void loop(){
```

digitalWrite(12,HIGH);//set digital12 to HIGH level, COM and NO are connected，LED is on

delay(500);//delay in 500ms

digitalWrite(12,LOW);//set digital12 to LOW level, COM and NO are disconnected，LED is off

delay(500);//delay in 500ms

}

Test Result:

Wire, power up and upload the code. The relay is connected("NO" is on , NC is off) for 0.5s, then disconnected for 0.5s (NC is on, NO is off), and alternately. When the relay is connected, the white LED is on, otherwise, the white LED is off.

Project 6：Photocell Sensor

Description

The photocell sensor (photoresistor) is a resistor made by the photoelectric effect of a semiconductor. It is very sensitive to ambient light, thus its resistance value vary with different light intensity.

We use its features to design a circuit and generate a photoresistor sensor module. The signal end of the module is connected to the analog port of the microcontroller. When the light intensity increases, the resistance decreases, and the voltage of the analog port rises, that is, the analog value of the microcontroller also goes up. Otherwise, when the light intensity decreases, the resistance increases, and the voltage of the analog port declines. That is, the analog value of the microcontroller becomes smaller. Therefore, we can use the photoresistor sensor module to read the corresponding analog value and sense the light intensity in the environment.

It is commonly applied to light measurement, control and conversion, light control circuit as well.

Specifications:

Working voltage: 3.3V-5V (DC)

Interface: 3PIN interface

Output signal: analog signal

Weight: 2.3g

Equipment:

keyestudio PLUS Control Board*1

Sensor expansion board * 1

Photocell sensor module * 1

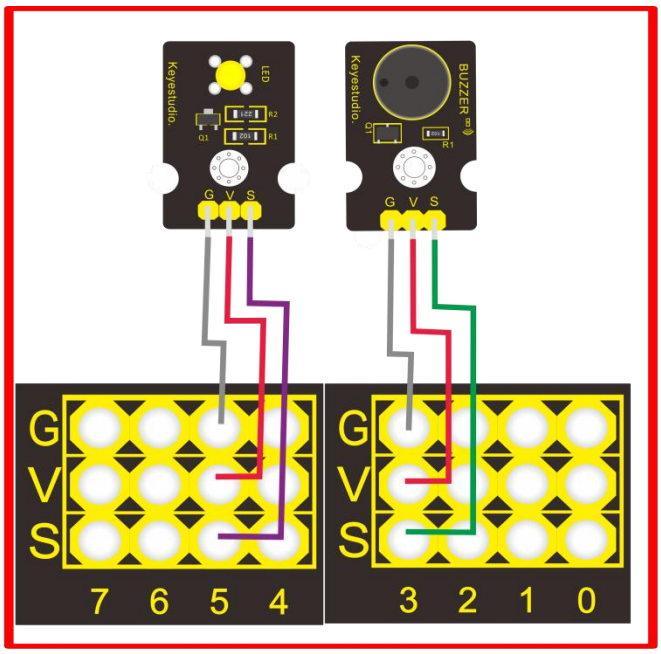
Passive buzzer module * 1

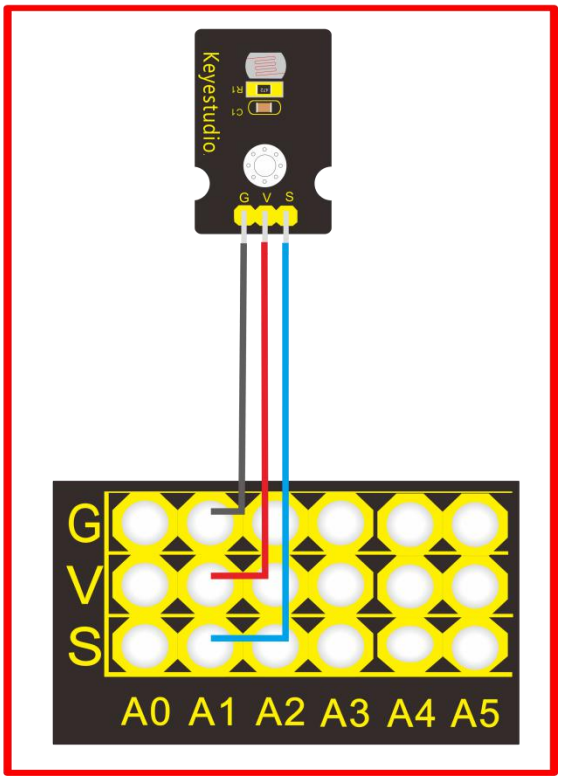
Yellow LED module * 1

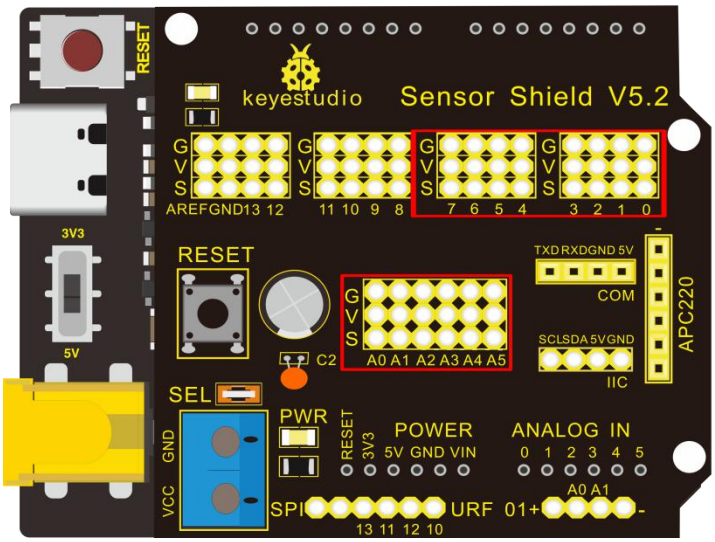
3pin female to female Dupont cable * 3

USB cable * 1

Connection Diagram:







Note: On the expansion board, the G, V, and S pins of the photocell sensor module are connected to G, V, and A1; the G, V, and S pins of the passive buzzer module are linked with G, V, and 3 respectively; the G, V, and S pins of the yellow LED module are connected to G, V, and 5 separately.

Test Code:

```
**********************************************************************
*********************************
```

```
void setup(){
  Serial.begin(9600);//set baud rate to 9600
  pinMode(A1, INPUT);//set A1 to input
  pinMode(3, OUTPUT);//set digital3 to output
  pinMode(5, OUTPUT);//set digital5 to output

}

void loop(){
  Serial.println(analogRead(A1));//wrap word and read the analog value of A1
delay(500);//delay in 500ms
  if (analogRead(A1) <= 600)
//if the analog value of A1 is less than or equal to 600
 {
    tone(3,262);//digital3 outputs the sound of 262Hz
    delay(250);//delay in 250ms
    digitalWrite(5,HIGH);//set digital5 to HIGH level，LED is on
    delay(500);//delay in 500ms
```

```
    tone(3,294);//digital3 outputs the sound of 294Hz
    delay(250);//delay in 250ms
  digitalWrite(5,LOW);//set digital5 to low level，LED is off
  delay(500);//delay in 500ms
    tone(3,330);
    delay(250);
    digitalWrite(5,HIGH);
    delay(500);
    tone(3,349);
    delay(250);
    digitalWrite(5,LOW);
    delay(500);
    tone(3,392);
    delay(250);
    digitalWrite(5,HIGH);
    delay(500);
    tone(3,440);
    delay(250);
    digitalWrite(5,LOW);
    delay(500);

}
```

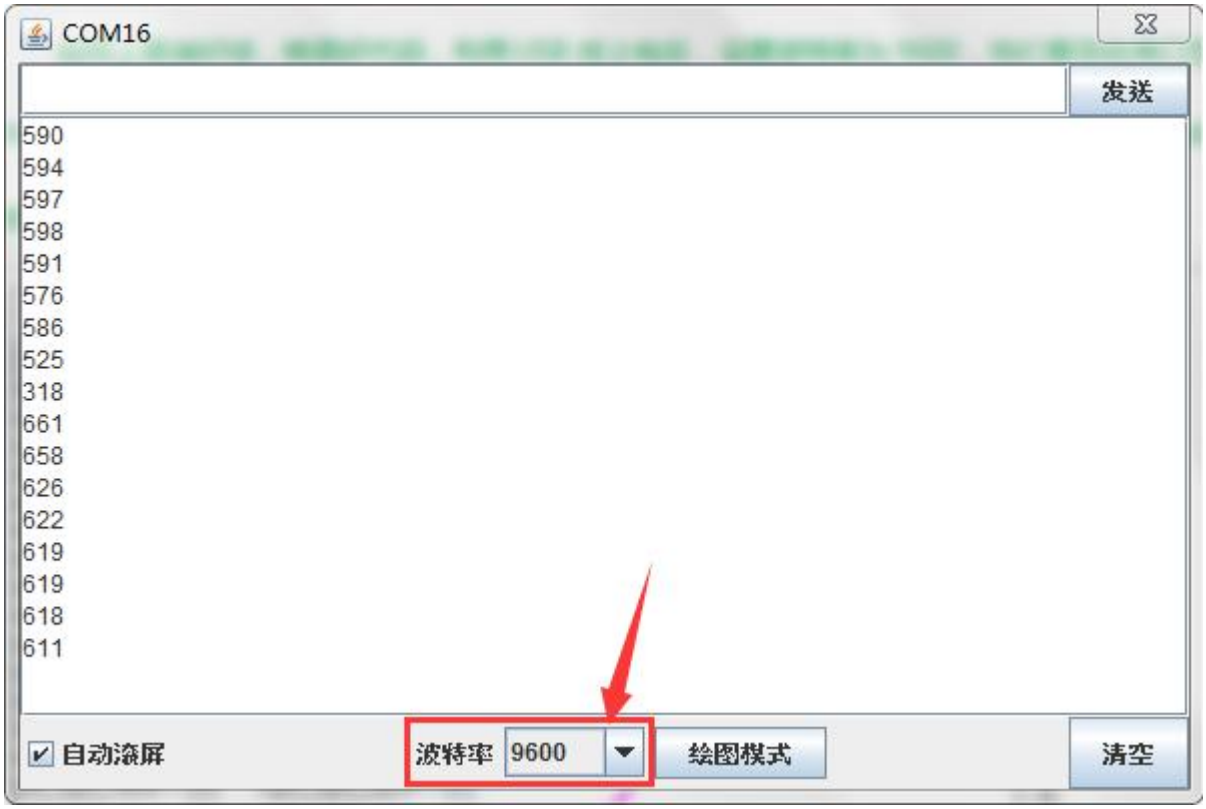
```
else
//otherwise  (if the analog value of A1 is greater than 600)
 {
     digitalWrite(5,LOW);//set digital5 to low level，LED is off
          noTone(3);//digital3 stops sounding
 }}
```

****

****

Test Result:

Wire according to connection diagram, burn the code, and set the baud rate to 9600 after powering on with USB cable. When the value displayed on serial monitor is less than or equal to 600, the yellow LED flashes and the passive buzzer sensor plays music; otherwise, the yellow LED is off, and the passive buzzer sensor stops playing.



Project 7：Adjusting Motor Servo Angle

Description

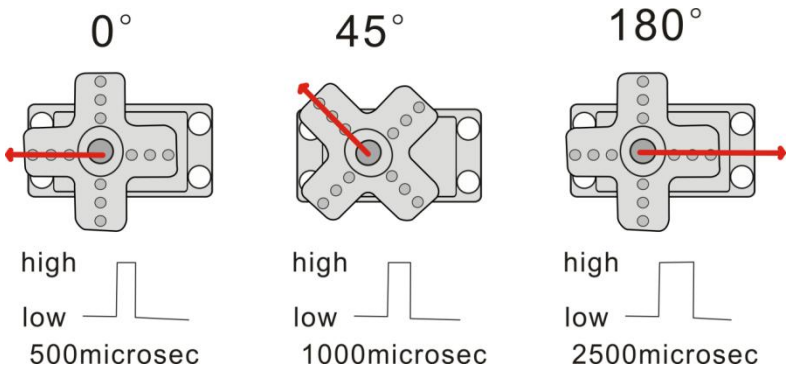
When we make this kit, we often control doors and windows with servos. In this course, we'll introduce its principle and how to use servo motors.

Servo motor is a position control rotary actuator. It mainly consists of housing, circuit board, core-less motor, gear and position sensor. Its working principle is that the servo receives the signal sent by MCU or receiver and produces a reference signal with a period of 20ms and width of 1.5ms, then compares the acquired DC bias voltage to the voltage of the potentiometer and outputs a voltage difference.

Servo motor comes with many specifications. But all of them have three connection wires, distinguished by brown, red, orange colors (different brand may have different color).

Brown one is for GND, red one for power positive, orange one for signal line.

The rotation angle of servo motor is controlled by regulating the duty cycle of PWM (Pulse-Width Modulation) signal. The standard cycle of PWM signal is 20ms (50Hz). Theoretically, the width is distributed between 1ms-2ms, but in fact, it's between 0.5ms-2.5ms. The width corresponds the rotation angle from 0° to 180°. But note that for different brand motor, the same signal may have different rotation angle.



There are two ways to control a servomotor with Arduino.

One is to use a common digital sensor port of Arduino to produce square wave with different duty cycle to simulate PWM signal and use that signal to control the positioning of the motor.

Another way is to directly use the Servo function of the Arduino to control the motor. In this way, the program will be easier but it can only control two-contact motor because for the servo function, only digital pin 9 and 10 can be used.

The Arduino drive capacity is limited. So if you need to control more than one motor, you will need external power.

Specifications:

Working voltage: DC 4.8V ~ 6V

Operating angle range: about 180 ° (at 500 → 2500 μsec)

Pulse width range: 500 → 2500 μsec

No-load speed: 0.12 ± 0.01 sec / 60 (DC 4.8V) 0.1 ± 0.01 sec / 60 (DC 6V)

No-load current: 200 ± 20mA (DC 4.8V) 220 ± 20mA (DC 6V)

Stopping torque: 1.3 ± 0.01kg · cm (DC 4.8V) 1.5 ± 0.1kg · cm (DC 6V)

Stop current: ≦ 850mA (DC 4.8V) ≦ 1000mA (DC 6V)

Standby current: 3 ± 1mA (DC 4.8V) 4 ± 1mA (DC 6V)

Lead length: 250 ± 5 mm

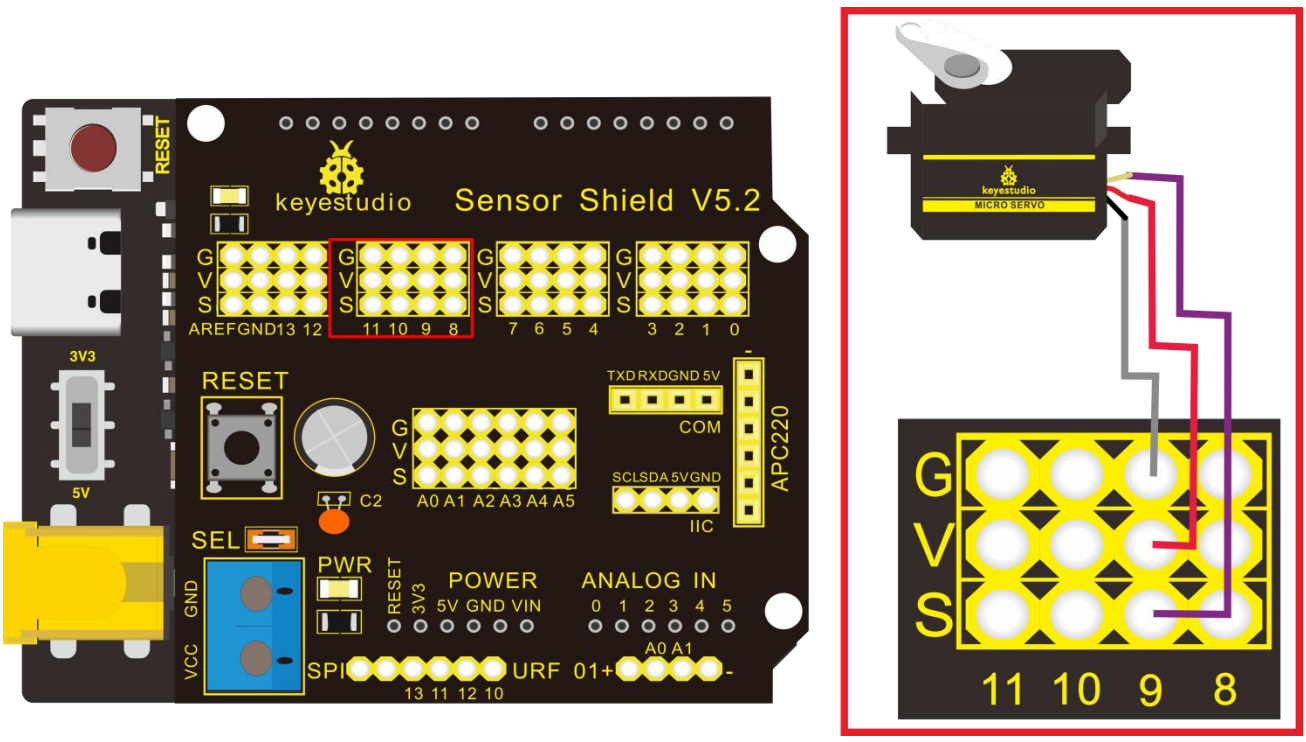
Appearance size: 22.9 * 12.2 * 30mm

Weight: 9 ± 1 g (without servo horn)

Storage temperature: -20 ℃ ~ 60 ℃

Operating temperature: -10 ℃ ~ 50 ℃

Connection Diagram:



Note: The servo is connected to G (GND), V (VCC), 9. The brown wire of the servo

is connected to Gnd (G), the red wire is connected to 5v (V), and the orange wire is connected to digital pin 9.

Test Code:

```
**********************************************************************
**********************************
```

```
volatile int pulsewidth;//set variable pulsewidth
//Set a subroutine to control the servo angle
void procedure(int servopin, int myangle) {
  for (int i = 0; i <= 50; i = i + (1)) {
    pulsewidth = myangle * 11 + 500;
   pinMode(servopin, OUTPUT);
   //varaible servopin  (means signal end of servo，set to output)
    digitalWrite(servopin,HIGH);
    delayMicroseconds(pulsewidth);//delay time is ms, the value is variable
pulsewidth
    pinMode(servopin, OUTPUT);
    digitalWrite(servopin,LOW);
    delay((20 - pulsewidth / 1000));
  }
```

```
}


void setup(){
   pulsewidth = 0;//set variable pulsewidth to 0
}
void loop(){
   procedure(9, 90);//set signal end of servo to digital9，the angle of servo is set
   to 90°
}
```

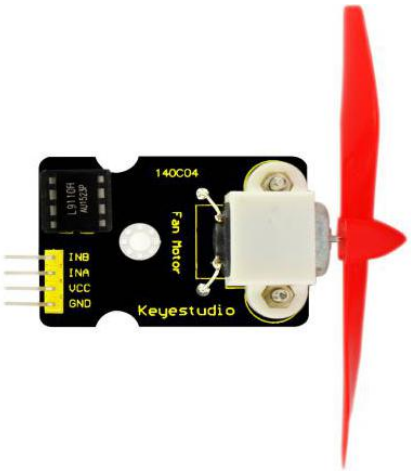
****

****

Test Result:

Upload code, wire according to connection diagram, and power on. The DIP switch is dialed to right side, the servo rotates to 90°.

Project 8：Fan Module



Description

The L9110 motor control chip is applied on Keyestudio L9110 motor control module. It can control the rotation direction and speed of the motor so as to control the fan.

The module is compatible with servo motor control. This module is efficient with good fan, commonly applied to fire robot.

Specifications:

Fan diameter: 75mm

Working voltage: 5V

Equipment:

keyestudio PLUS Control Board*1

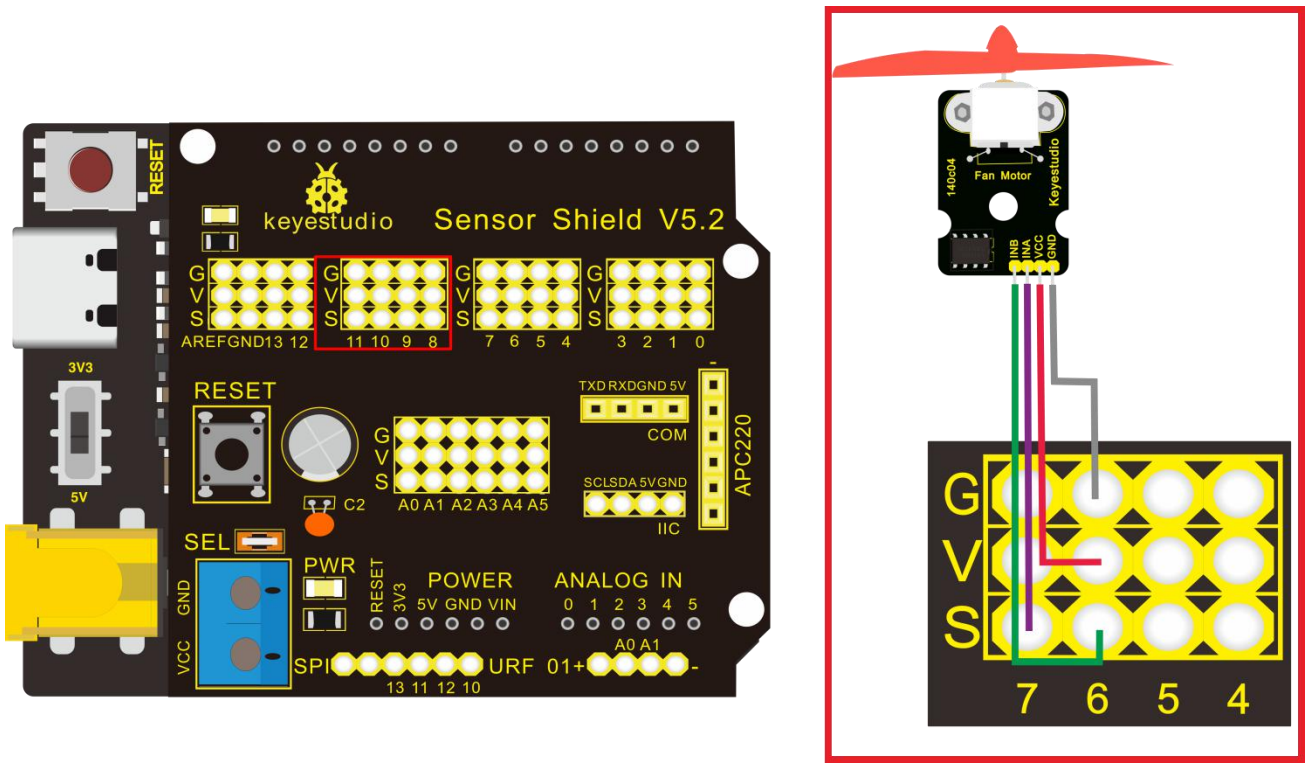
Sensor shield* 1

Small fan module * 1

USB cable * 1

Female to female DuPont Lines

Connection Diagram:



Note: On the shield, the GND, VCC, INA, and INB pins of the fan module are respectively connected to G, V, 7, 6.

Test Code:

****

****

```
void setup(){
   pinMode(7, OUTPUT);//set digital7 to output
   pinMode(6, OUTPUT);//set digital6 to output
}

void loop(){
   //set fan to rotate counterclockwise for 3000ms
   digitalWrite(7,LOW);
   digitalWrite(6,HIGH);
   delay(3000);
//set fan to stop rotating for 1000ms
   digitalWrite(7,LOW);
   digitalWrite(6,LOW);
   delay(1000);
   //set fan to rotate clockwise for 3000ms
   digitalWrite(7,HIGH);
   digitalWrite(6,LOW);
   delay(3000);
}
```

****

****

Test Result:

Upload test code, wire according to connection diagram, the DIP switch is dialed to right side and power on. The fan rotates counterclockwise for 3000ms, stops for 1000ms, then rotates clockwise for 3000ms.

Project 9：Steam Sensor

Description

This is a commonly used steam sensor. Its principle is to detect the amount of water by bare printed parallel lines on the circuit board. The more the water is, the more wires will be connected. As the conductive contact area increases, the output voltage will gradually rise. It can detect water vapor in the air as well. The steam sensor can be used as a rain water detector and level switch. When the humidity on the sensor surface surges, the output voltage will increase.

The sensor is compatible with various microcontroller control boards, such as

Arduino series microcontrollers. When using it, we provide the guide to operate steam sensor and Arduino control board. Connect the signal end of the sensor to the analog port of the microcontroller, sense the change of the analog value, and display the corresponding analog value on the serial monitor.

It comes with the 3pin with 2.54mm pitch. Connect it to control board with dupont lines. We particularly design the shield compatible with this sensor and keyestudio PLUS control board. You just need to stack shield to keyestudio PLUS control board and connect them together with 3pin dupont lines.

Note: the connect part is not waterproof, don't immerse it in the water please.

Specifications:

Working voltage: DC 3.3-5V

Working current: <20mA

Operating temperature range: -10 ℃ ～ ＋ 70 ℃;

Control signal: analog signal output

Interface: 2.54mm 3pin pin interface

Size: 35 * 20 * 8mm

Weight: 2.2g

S: signal output

V (+): Power supply (VCC)

G (-): Ground (GND)

Equipment:

keyestudio PLUS Control Board*1

Sensor shield * 1

Water vapor sensor * 1

Photocell sensor module * 1

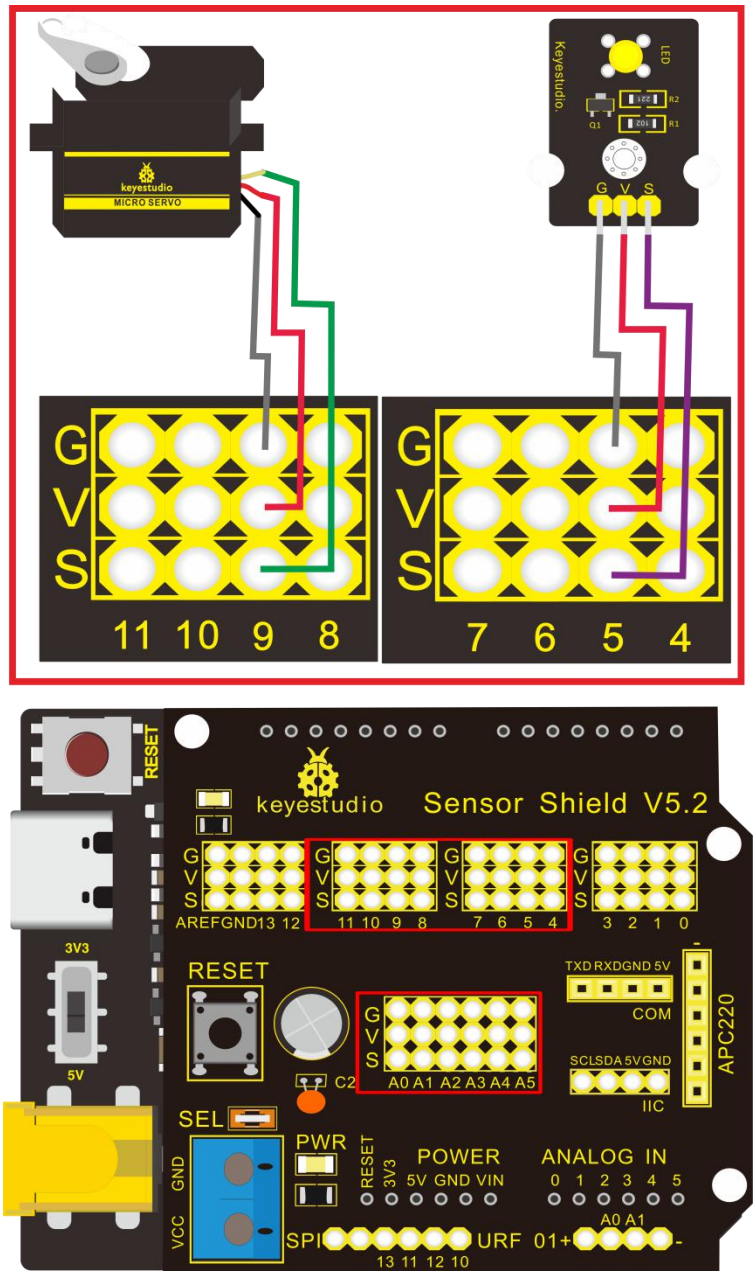
Servo * 1

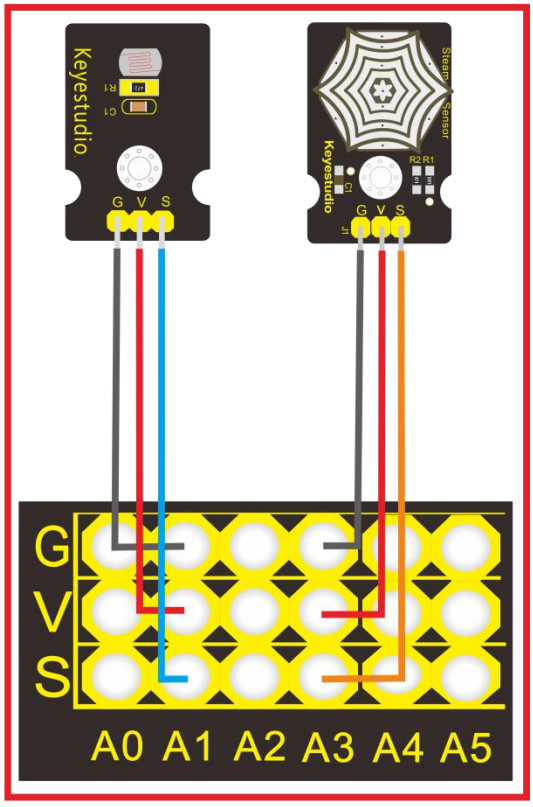
Yellow LED * 1

3pin female to female DuPont Lines

USB cable * 1

Connection Diagram:





Note: On the sensor shield, the pins G， V and S of steam sensor are connected to G, V and A3; the pins G, V and S of photocell sensor are connected to G, V and A1; the pins G, V and S of yellow LED are linked with G, V and 5; the brown line is linked with G, red wire to V, orange wire to 9.

Test Code:

```
**********************************************************************
*********************************
```

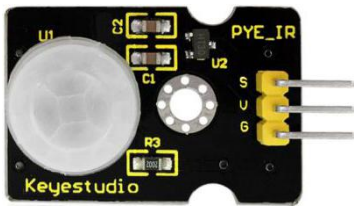
```
#include <Servo.h>//use the library of Servo.h

Servo servo_9;

void setup(){
  Serial.begin(9600);//set baud rate to 9600
  pinMode(A1, INPUT);//set A1 to input
  pinMode(A3, INPUT);//set A3 to input
  pinMode(5, OUTPUT);;//set digital5 to input
  servo_9.attach(9);//set signal end of servo to digital9
}

void loop(){
  Serial.print("illumination");
  Serial.print(analogRead(A1));//serial port outputs the analog value of A1
Serial.print("          ");
  Serial.print("humidity");
  Serial.println(analogRead(A3));//wrap word and output the analog value of
```

```
A3
delay(500);//delay in 500ms
   if (analogRead(A3) >= 500 && analogRead(A1) <= 450)
//if the analog value of A3≥500 and the analog value of  ≤  450
{
      digitalWrite(5,HIGH);//set digital5 to HIGH level，LED is off
      servo_9.write(0);//set servo angle to 0°
      delay(200);//delay in 200ms
   }
else
//otherwise
 {
    if (analogRead(A3) < 500 && analogRead(A1) > 450)
//if the analog value of A3 is less than 500 and the analog value of A3 is greater
   than 450
    {
        digitalWrite(5,LOW);//set digital5 to low level，LED is off
        servo_9.write(180);//set servo angle to 180°
        delay(200);//delay in 200ms
 }}}
**********************************************************************
*********************************
```

Test Result:

Upload test code successfully, wire according to connection diagram. When put a drop of water on steam sensor and cover the photocell sensor, the analog value of steam sensor will increase on serial monitor, the analog value of photocell sensor will gradually go down, the yellow LED will light on, servo will rotate to 0°. However, when wipe off the water drop and the photocell sensor is under strong light, the analog value on the serial monitor will gradually decline, the analog value of photocell sensor will gradually increase, yellow LED will be off, and servo will rotate to 180°

Project 10: PIR Motion Sensor



Description

The Pyroelectric infrared motion sensor can detect infrared signals from a moving person or moving animal, and output switching signals. It can be applied to a variety of occasions to detect the movement of human body. Conventional

pyroelectric infrared sensors are much more bigger, with complex circuit and lower reliability. Now we launch this new pyroelectric infrared motion sensor, specially designed for Arduino. This sensor integrates an integrated digital pyroelectric infrared sensor, and the connection pins. It features higher reliability, lower power consumption and simpler peripheral circuit.

Specifications:

Input voltage: DC 3.3V ~ 18V

Working current: 15uA

Working temperature: -20 ~ 85 degrees Celsius

Output voltage: high 3 V, low 0 V

Output delay time (high level): about 2.3 to 3 seconds

Detection angle: about 100 °

Detection distance: 3-4 meters

Output indicator LED (high-level light)

Pin limit current: 100mA

Special note：

1. The maximum distance is 3-4 meters during testing.

2. When testing, first open the white lens, you can see the rectangular sensing part. When the long line of the rectangular sensing part is parallel to the ground, the distance is the best.

3. When testing, the sensor needs to be covered with white lens, otherwise it will affect the distance.

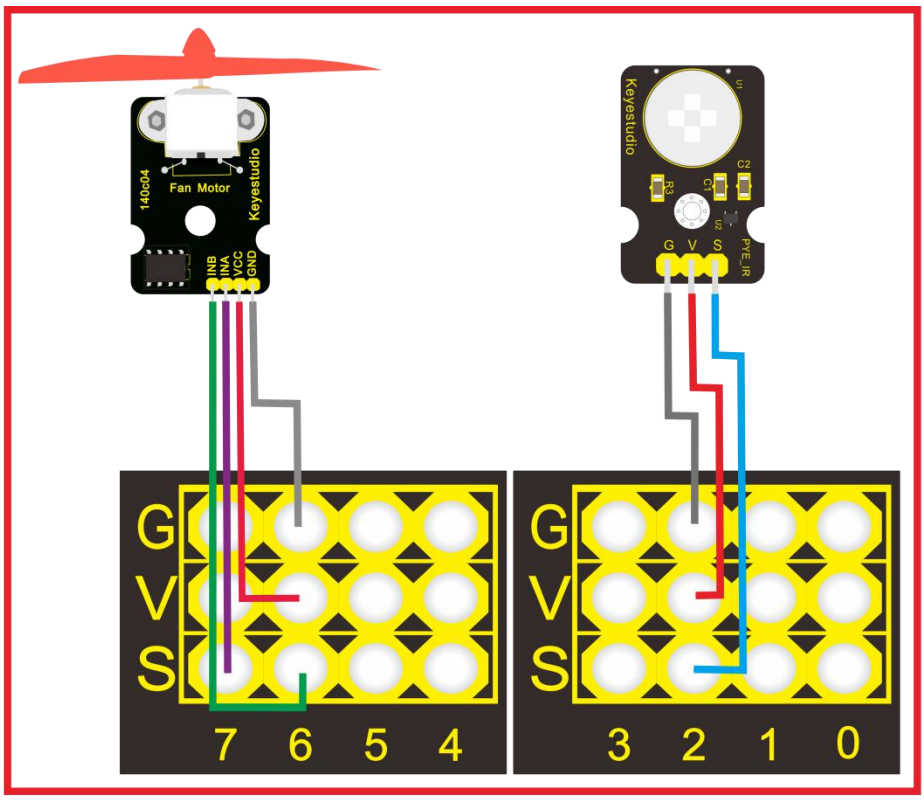
4. The distance is best at 25℃, and the detection distance is shortened when it exceeds 30℃.

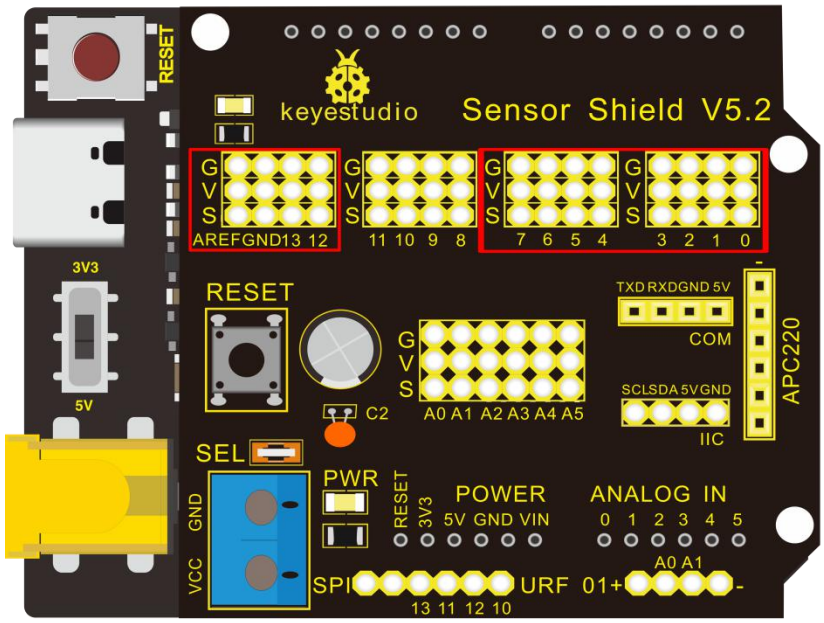
5. Done powering up and uploading the code, you need to wait 5-10 seconds then start testing, otherwise it is not sensitive.

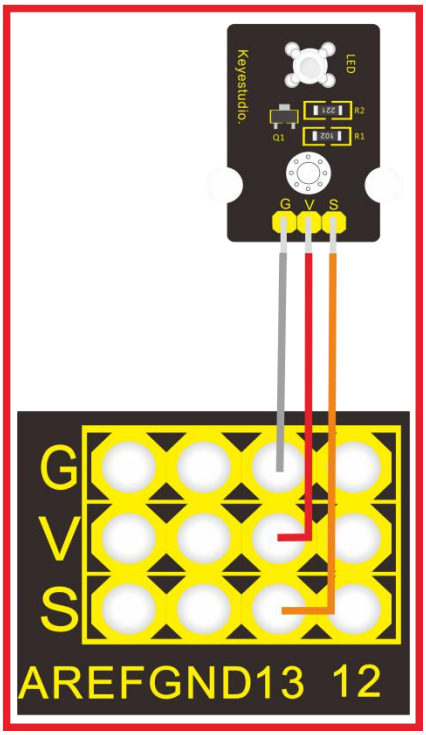
Equipment:

- keyestudio PLUS Control Board*1
- Sensor expansion board * 1
- PIR Motion sensor * 1
- Small fan module * 1
- White LED module * 1
- 3pin female to female Dupont cable * 2
- Mother to mother Dupont line * 4
- USB cable * 1

Connection Diagram:







Note: On the shield, the G, V and S of PIR motion sensor are connected to G, V and 2; the GND, VCC, INA and INB of fan module are separately connected to G,V,7,6. The pin G, V and S of LED module are linked with G, V and 13.

Test Code:

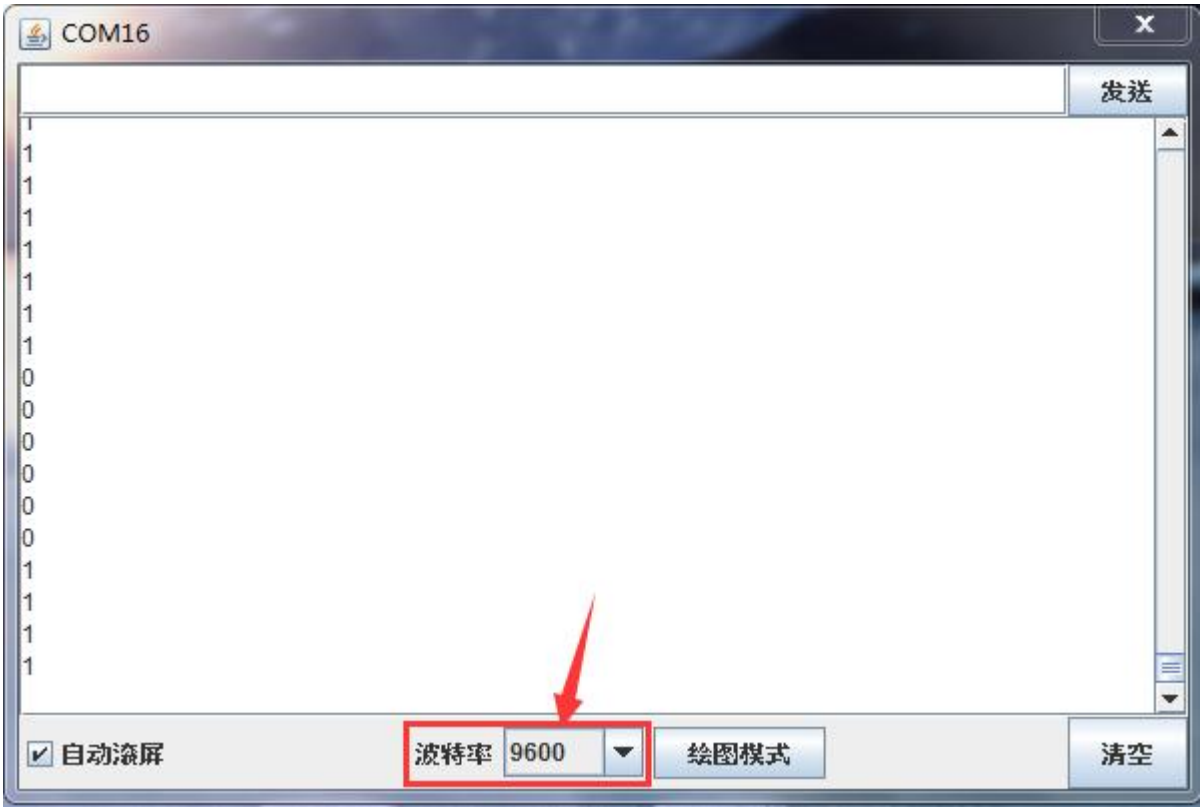
```
**********************************************************************
**********************************
```

```
void setup(){
   Serial.begin(9600);//set baud rate to 9600
   pinMode(2, INPUT);//set digital2 to input
   pinMode(13, OUTPUT);//set digital13 input
   pinMode(7, OUTPUT);//set digital7 input
   pinMode(6, OUTPUT);//set digital6 input}
void loop(){
   Serial.println(digitalRead(2));//wrap word and output the value of digital2
   delay(500);//delay in 500ms
   if (digitalRead(2) == 1)
//when there is person around，the value of digital2 is 1
 {
    digitalWrite(13,HIGH);//set digital13 to high level,LED lights up
    //set fan to rotate
     digitalWrite(7,HIGH);//set digital7 to high level
     analogWrite(6,150);//set the PWM value of digital6，the higher the value，
the slower the fan rotates
```

```
  } else
//otherwise  (when nobody is around, the value of digital2 is 0)
 {
    digitalWrite(13,LOW);//set digital13 to low level，LED lights on
    //fan stops rotating
      digitalWrite(7,LOW);
      analogWrite(6,0);
  }}
**********************************************************************
*******************************
```

Test Result:

Upload test code, open serial monitor, and set baud rate to 9600. If PIR motion sensor detects the people around, the serial monitor displays "1", the D13 and white LED light on at same time, fan rotates. If there is no person around, the serial monitor shows "0", the D13 indicator and white LED stops rotating.



Project 11: Analog（MQ-2）Sensor

Description

This gas sensor is used for household gas leak alarms, industrial combustible gas alarms and portable gas detection instruments. And it is suitable for the detection of liquefied gas, benzene, alkane, alcohol, hydrogen, etc., and widely used in various fire alarm systems. The MQ-2 smoke sensor can be accurately a multi-gas detector, and has the advantages of high sensitivity, fast response, good stability,

long life, and simple drive circuit.

It can detect the concentration of flammable gas and smoke in the range of 300~10000ppm.Meanwhile, it has high sensitivity to natural gas, liquefied petroleum gas and other smoke, especially to alkanes smoke.

It must be heated for a period of time before using the smoke sensor, otherwise the output resistance and voltage are not accurate. However, the heating voltage should not be too high, otherwise it will cause my internal signal line to blow.

It is belongs to the tin dioxide semiconductor gas-sensitive material, and belongs to the surface ion type N-type semiconductor. At a certain temperature, tin dioxide adsorbs oxygen in the air and forms negative ion adsorption of oxygen, reducing the electron density in the semiconductor, thereby increasing its resistance value. When in contact with flammable gas in the air and smog, if the potential barrier at the grain boundary is adjusted by the smog, it will cause the surface conductivity to change. With this, information about the presence of smoke or flammable gas can be obtained. The greater the concentration of smoke or flammable gas in the air, the greater the conductivity, and the lower the output resistance, the larger the analog signal output. The sensor comes with a positioning hole, which is convenient for you to fix the sensor to other devices. In addition, the sensitivity can be adjusted by rotating the potentiometer.

Specifications:

Working voltage: 3.3-5V (DC)

Interface: 4 pins (VCC, GND, D0, A0)

Output signal: digital signal and analog signal

Weight: 7.5g

Equipment:

keyestudio PLUS Control Board*1

Sensor shield* 1

MQ-2 gas sensor * 1

Passive sensor*1

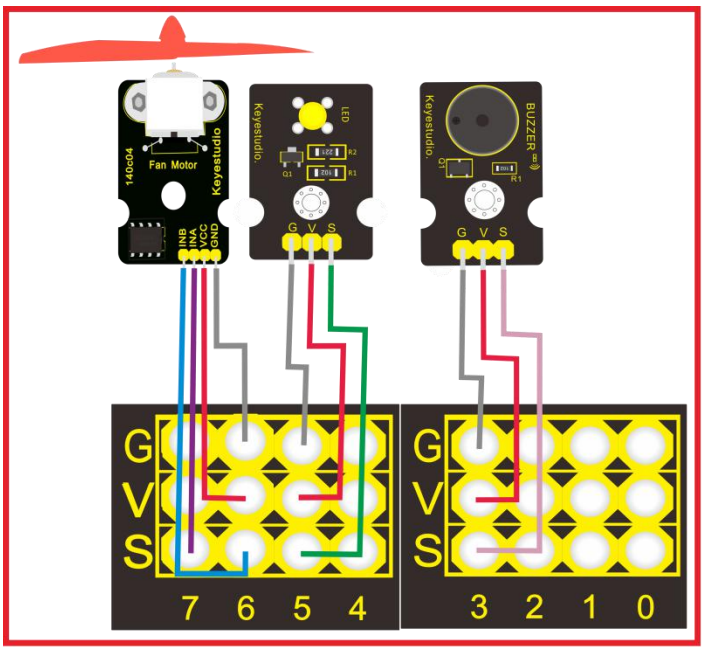
fan module * 1

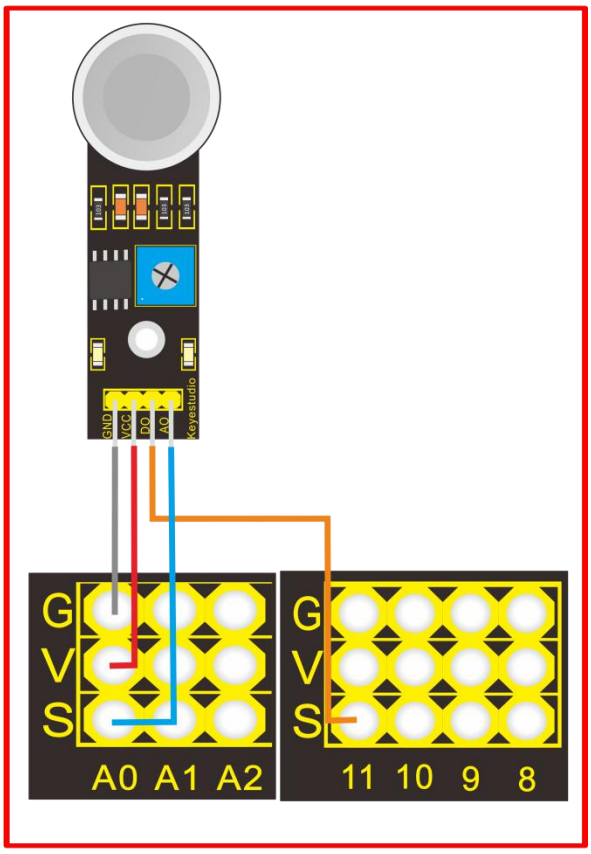
Yellow LED module * 1

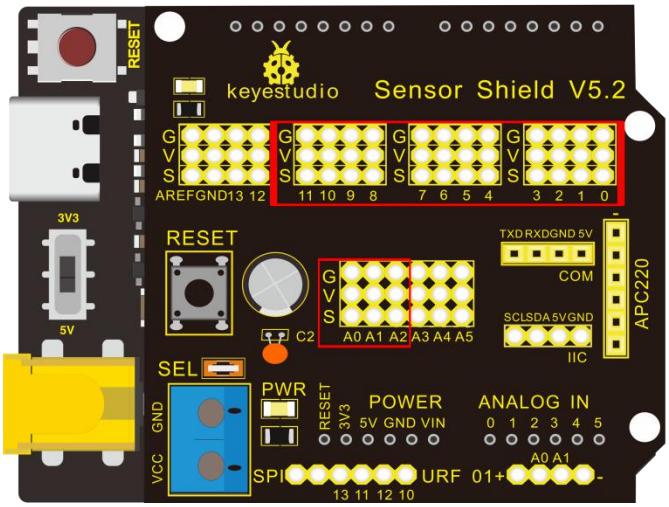
3pin female to female DuPont Lines

USB cable * 1

Connection Diagram:







Note: On the shield, the pin GND, VCC, D0 and A0 of gas sensor are linked with pin GND, VCC, D0 and A0. Pin GND, VCC, INA and INB of fan module are connected to G,V, 7 and 6. The pin G,V and S of passive buzzer are connected to G,V and 3; the pin G, V and S of yellow LED are connected to G,V and 5.

Test Code:

****

****

```
void setup(){
   Serial.begin(9600);//set baud rate to 9600
   pinMode(11, INPUT);//set digital11 to input
   pinMode(A0, INPUT);//set A0 to input
   pinMode(3, OUTPUT);//set digital3 to output
   pinMode(7, OUTPUT);//set digital7 to output
   pinMode(6, OUTPUT);//set digital6 to output
   pinMode(5, OUTPUT);//set digital5 to output
}

void loop(){
   Serial.println(digitalRead(11));//wrap word and read the analog value of digital11
   Serial.println(analogRead(A0));//wrap word and read the analog value of A0
   delay(500);//delay in 500ms
   if (analogRead(A0) > 45 && digitalRead(11) == 0) {
//if the analog value of A0 is greater than 45 and the digital value of digital11 is 0
      tone(3,131);//digital3 outputs the sound of 131HZ
     delay(125);//delay in 125ms
     //set fan to rotate
      digitalWrite(7,LOW);
```

```
    analogWrite(6,200);//set the PWM value of digital6 to 200, the bigger the the
value of PWM is，the faster the fan rotates
        digitalWrite(5,HIGH);//set digital5 to high level，LED lights up
    delay(200);//delay in 200ms
    digitalWrite(5,LOW);//set digital5 to low level，LED is off
        delay(200);//delay in 200ms
  } else
//otherwise
 {
    noTone(3);//digital3 stops sounding
   digitalWrite(5,LOW);//set digital5 to low level，LED is off
   //fan stops rotating
    digitalWrite(7,LOW);
    analogWrite(6,0);
  }}
```

Test Result:

Upload test code, wire according to connection diagram and power on. When gas sensor detects the flammable gas, passive buzzer will sound, fan will rotate and yellow LED will be on; when there is no flammable gas, the passive buzzer won't sound, the fan won't rotate and yellow LED will be off.

Project 12: 1602 LCD Display

Description

keyestudio 1602 I2C module is a 16 character by 2 line LCD display with Blue background and White backlight.

The original 1602 LCD needs 7 IO ports to be up and running, but ours is built with Arduino IIC/I2C interface, saving you 5 IO ports.

This LCD is ready-to-use because it is compatible with the Arduino Liquid Crystal Library. LCDs are great for printing data and showing values. Adding an LCD to your project will make it super portable and allow you to integrate up to 32 characters (16x2) of information.

On the back of LCD display there is a blue potentiometer. You can turn the potentiometer to adjust the contrast.

Notice that the screen will get brighter or darker and that the characters become more visible or less visible.

Specifications:

I2C address: 0x27

Backlight (blue, white)

Power supply voltage: 5V

Adjustable contrast

GND: A pin that connects to ground

VCC: A pin that connects to a +5V power supply

SDA: A pin that connects to analog port A4 for IIC communication

SCL: A pin that connects to analog port A5 for IIC communication

Equipment:

keyestudio PLUS Control Board*1

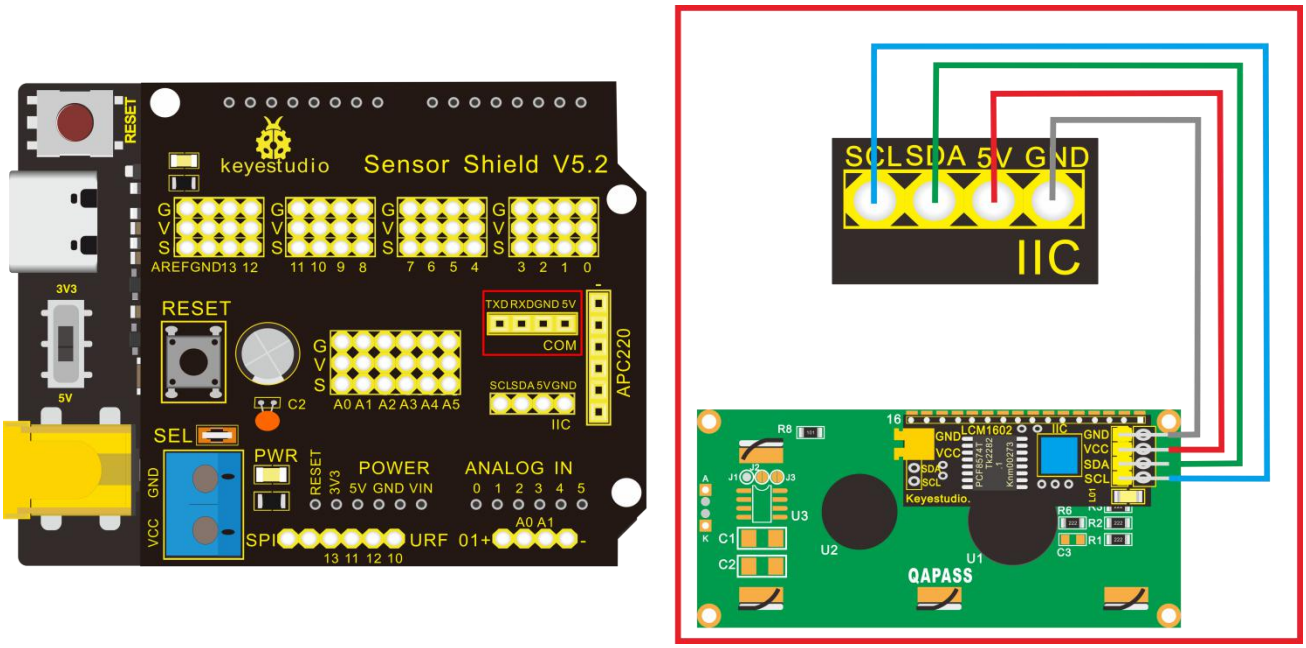
Sensor expansion board * 1

1602 LCD module * 1

4pin female to female DuPont line * 1

USB cable * 1

Connection Diagram:



Note: there are pin GND, VCC, SDA and SCL on 1602LCD module. GND is linked with GND（-）of IIC communication, VCC is connected to 5V（+）, SDA to SDA, SCL to SCL.

Test Code:

****

****

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
//use relevant library file
LiquidCrystal_I2C mylcd(0x27,16,2);
//set the communication address of I2C to 0x27， set to 1602 for display （16
```

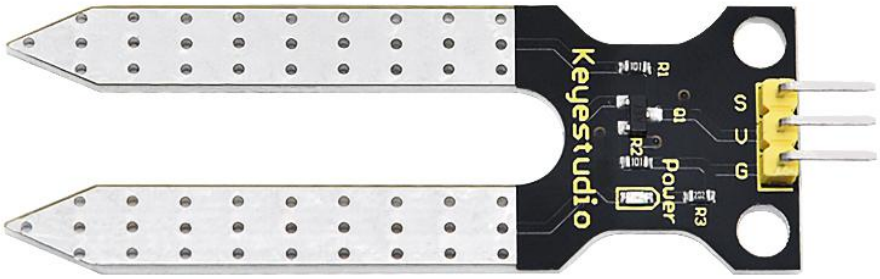
```
characters per row, 2 rows in total
)
void setup(){
   mylcd.init();
   mylcd.backlight();
   mylcd.clear();//clear up the pattern displayed
}

void loop(){
//the first character of first row starts to show "Hello World!"
   mylcd.setCursor(0, 0);
   mylcd.print("Hello World!");
//the character isn't shown at second row
   mylcd.setCursor(0, 1);
   mylcd.print("");
   mylcd.scrollDisplayLeft();//the shown character rolls to the left
   delay(300);//delay in 300ms
}
**********************************************************************
*********************************
```

Test Result:

Upload the test code, wire according to connection diagram and power on. 1602 LCD shows "Hello World！ " at first line and rolls to the left.

Note: Wire according to connection diagram, upload the code, and after power-on, when the display doesn't show characters, you can adjust the potentiometer behind the 1602LCD and backlight to make the 1602LCD display the corresponding character string.

Project 13: Soil Humidity Sensor



Description

This is a simple soil humidity sensor aims to detect the soil humidity.

If the soil is in lack of water, the analog value output by the sensor will decrease; otherwise, it will increase. If you use this sensor to make an automatic watering device, it can detect whether your botany is thirsty to prevent it from withering when you go out.

Using the sensor with Arduino controller makes your plant more comfortable and your garden smarter. The soil humidity sensor module is not as complicated as

you might think, and if you need to detect the soil in your project, it will be your best choice. The sensor is set with two probes inserted into the soil, then with the current go through the soil, the sensor will get resistance value by reading the current changes between the two probes and convert such resistance value into moisture content. The higher moisture (less resistance), the higher conductivity the soil has. Insert it into the soil and then use the AD converter to read it. With the help of this sensor, the plant can remind of you: I need water.

Specification

Power Supply Voltage: 3.3V or 5V

Working Current: ≤ 20mA

Output Voltage: 0-2.3V (When the sensor is totally immersed in water, the voltage will be 2.3V) the higher humidity, the higher the output voltage

Sensor type: Analog output

Interface definition: S- signal, G- GND, V - VCC

Packaging : Electrostatic bag sealing

Size: 63 * 20 * 8mm

Weight: 2.5g

Equipment

keyestudio PLUS Control Board * 1

Sensor expansion board * 1

Soil humidity sensor module * 1

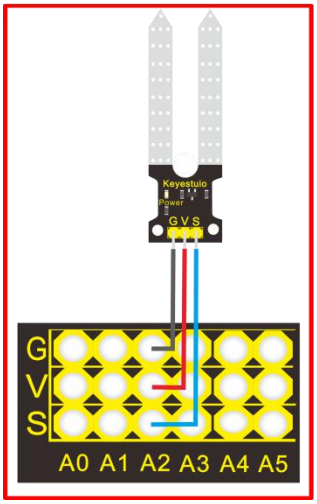
1602 LCD * 1

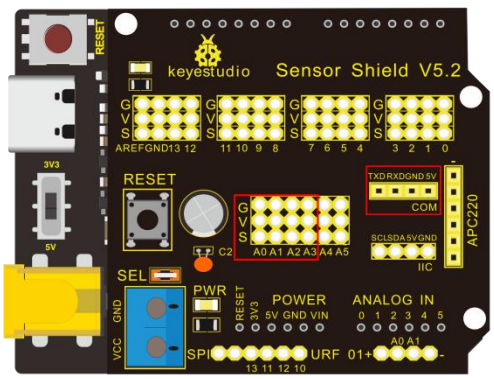
3pin female to female DuPont line * 1

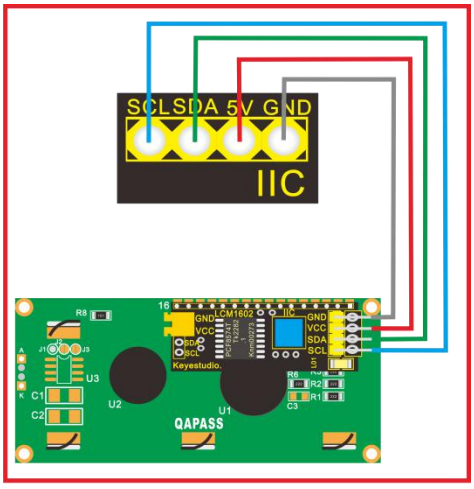
4pin female to female DuPont line * 1

USB cable * 1

Connection Diagram







Note: On the shield, the pin G, V and S of soil humidity sensor are connected to G, V and A2; GND of 1602LCD is linked with GND of ICC communication, VCC is connected to 5V（+）, SDA to SDA, SCL to SCL.

Test Code:

```
**********************************************************************
**********************************
```

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
//Use relevant library functions
volatile int value;//set variable "value"
LiquidCrystal_I2C mylcd(0x27,16,2);
//Set the I2C communication address to 0x27 and the display to 1602 (16 characters per line, 2 lines total)
void setup(){
  Serial.begin(9600);//set baud rate to 9600
  value = 0;//set value to 0
  mylcd.init();
  mylcd.backlight();
  mylcd.clear();//clear up the displayed information on LCD
  pinMode(A2, INPUT);//set A2 to input
```

```
}

void loop(){
  Serial.print("Soil moisture value:");
  Serial.print("        ");
  Serial.println(value);//wrap word and output the value of variable value
delay(500);//delay in 500ms
  value = analogRead(A2);//assign the analog value of A2 to variable value
  if (value < 300) {
  //if the variable value is less than 300
    mylcd.clear();//clear up the displayed information on LCD
    // Display the "value" character from the first character in the first line
          mylcd.setCursor(1-1, 1-1);
    mylcd.print("value:");
    //Display the value of variable value from the seventh character at the first line
          mylcd.setCursor(7-1, 1-1);
    mylcd.print(value);
    //Display the "dry soil" character from the first character in the second line
      mylcd.setCursor(1-1, 2-1);
      mylcd.print("dry soil");
      delay(300);//delay in 300ms
        }
```

```
else if (value >= 300 && value <= 700)
   //else if the variable value is greater than or equal to 300 and less than or equal
        to 700
{
    mylcd.clear();//clear up the displayed information on the LCD
    //Display the "value" character from the first character in the first line
     mylcd.setCursor(1-1, 1-1);
    mylcd.print("value:");
    //Display the value of variable value from the seventh character of the first line
     mylcd.setCursor(7-1, 1-1);
    mylcd.print(value);
    //Display the "humid soil" character from the first character on the second line
     mylcd.setCursor(1-1, 2-1);
     mylcd.print("humid soil");
     delay(300);//delay in 300ms
  }
else if (value > 700)
   //else if variable value is greater than 700
 {
    mylcd.clear();//clear up the displayed information on the LCD
    //Display the "value" character from the first character in the first line
     mylcd.setCursor(1-1, 1-1);
```

```
    mylcd.print("value:");
    //Display the value of variable value from the seventh character of the first line
      mylcd.setCursor(7-1, 1-1);
    mylcd.print(value);
    //Display the "in water" character from the first character in the second line
      mylcd.setCursor(1-1, 2-1);
      mylcd.print("in water");
      delay(300);//delay in 300ms
  }
}
```

****

****

Test Result:

Connect according to wiring diagram, and burn the program. After powering on, open the serial monitor and insert the soil sensor into the soil. The greater the humidity is, the bigger the number, in the range of 0-1023. The soil sensor is inserted into the soil and water with different humidity, and the 1602LCD displays the corresponding value.

Project 14: Bluetooth Test

In 21th century, technology has changed our life. People can work at home with wireless device like mouse, earphone, printer and speaker, which highly enhances our life standard.

Bluetooth can make work at home easily, as well as the entertainment. Users can control wirelessly the audio file from PC or Apple iPod within 30 inches. Bluetooth technology can also be used in adapters, allowing people to share their daily life with friends from internet and social media.

➢ Bluetooth Remote Control

Bluetooth technology is a wireless standard technology that enables short-distance data exchange between fixed devices, mobile devices, and building personal area networks (using UHF radio waves in the ISM band of 2.4 to 2.485 GHz).

This kit is equipped with the HM-10 Bluetooth module, which is a master-slave machine. When use as the Host, it can send commands to the slave actively; when use as the Slave, it can only receive commands from the host.

The HM-10 Bluetooth module supports the Bluetooth 4.0 protocol, which not only supports Android mobile, but also supports iOS system.

In the experiment, we default use the HM-10 Bluetooth module as a Slave and the cellphone as a Host. We install the Bluetooth APP on the mobile phone, connecting the Bluetooth module; finally use the Bluetooth APP to control the parts of smart home kit.

We also provide you with 2 types of mobile APP, for Android and iOS system.

Parameters of HM-10 Bluetooth Module:

Pins	Description
BRK	As input pin, short press control, or input single pulse of 100ms low level to achieve the following functions: 1. When module is in sleep state: Module is activated to normal state, if open AT+NOTI, serial port will send OK+WAKE. 2. When in connected state: Module will actively request to disconnect When in standby mode: Module will be in initial state

RXD	Serial data inputs
TXD	Serial data outputs
GND	ground lead
VCC	Positive pole of power, input 5V
STATE	As output pin, show the working state of module Flash slowly in standby state——repeat 500ms pulse; Always light in connected state——high level You could set to no flashing in standby state, always light in connected state

Parameters:

Bluetooth protocol: Bluetooth Specification V4.0 BLE

No byte limit in serial port Transceiving

In open environment, realize 100m ultra-distance communication with iphone4s

USB protocol: USB V2.0

Working frequency: 2.4GHz ISM band

Modulation method: GFSK(Gaussian Frequency Shift Keying)

Transmission power: -23dbm, -6dbm, 0dbm, 6dbm, can be modified by AT command.

Sensitivity: ≤-84dBm at 0.1% BER

Transmission rate: Asynchronous: 6K bytes ; Synchronous: 6k Bytes

Security feature: Authentication and encryption

Supporting service: Central & Peripheral UUID FFE0, FFE1

Power consumption: Auto sleep mode, stand by current 400uA~800uA, 8.5mA during transmission.

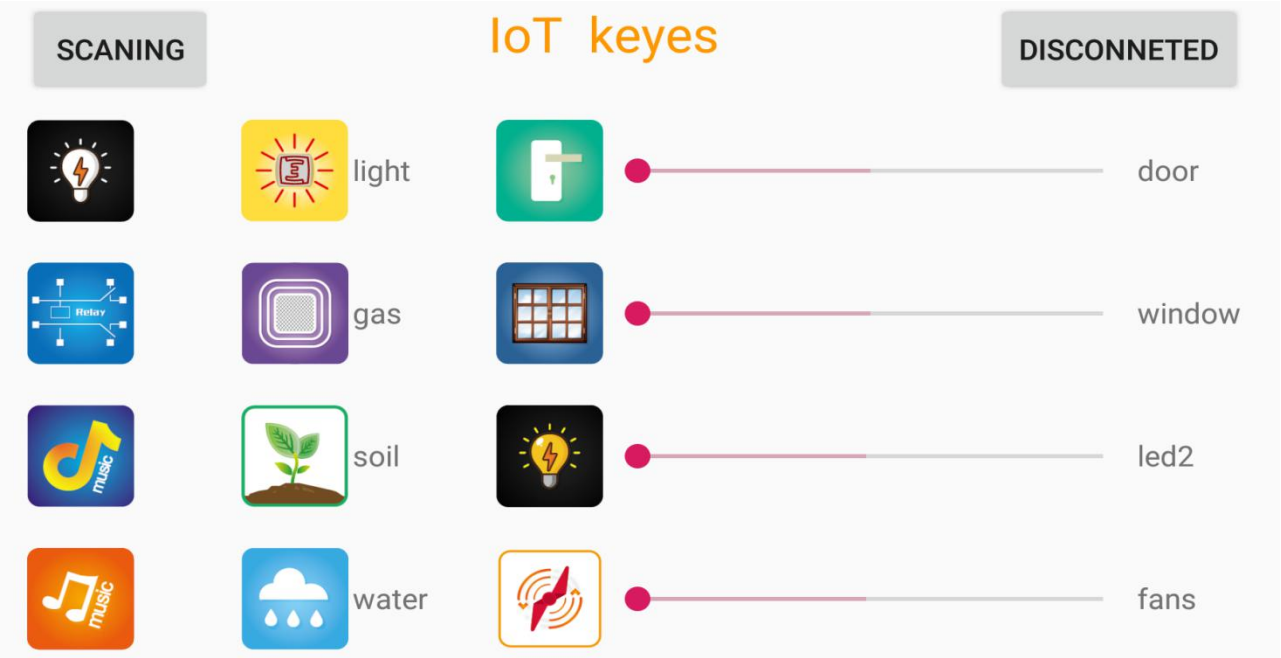
Power supply: 5V DC

Working temperature: –5 to +65 Centigrade

➢ Using Bluetooth APP

Description:

In the previous lesson, we've introduced the basic parameter principle of HM-10 Bluetooth module. In this project, let's show you how to use the HM-10 Bluetooth module. In order to efficiently control this kit by HM-10 Bluetooth module, we specially designed an APP, as shown below.



There are 16 control buttons in the app. When we connect the HM-10 Bluetooth module and app, only press control button of APP, and the Bluetooth of cellphone sends a control character. The Bluetooth module will receive a corresponding control character. When programming, we set the corresponding function of each sensor or module according to the corresponding key control character. Next, let's test 16 buttons on app.

APP for Android mobile(overseas):

Enter google play，search "keyes IoT"，if you can't search it on app store, please download app in the following link:

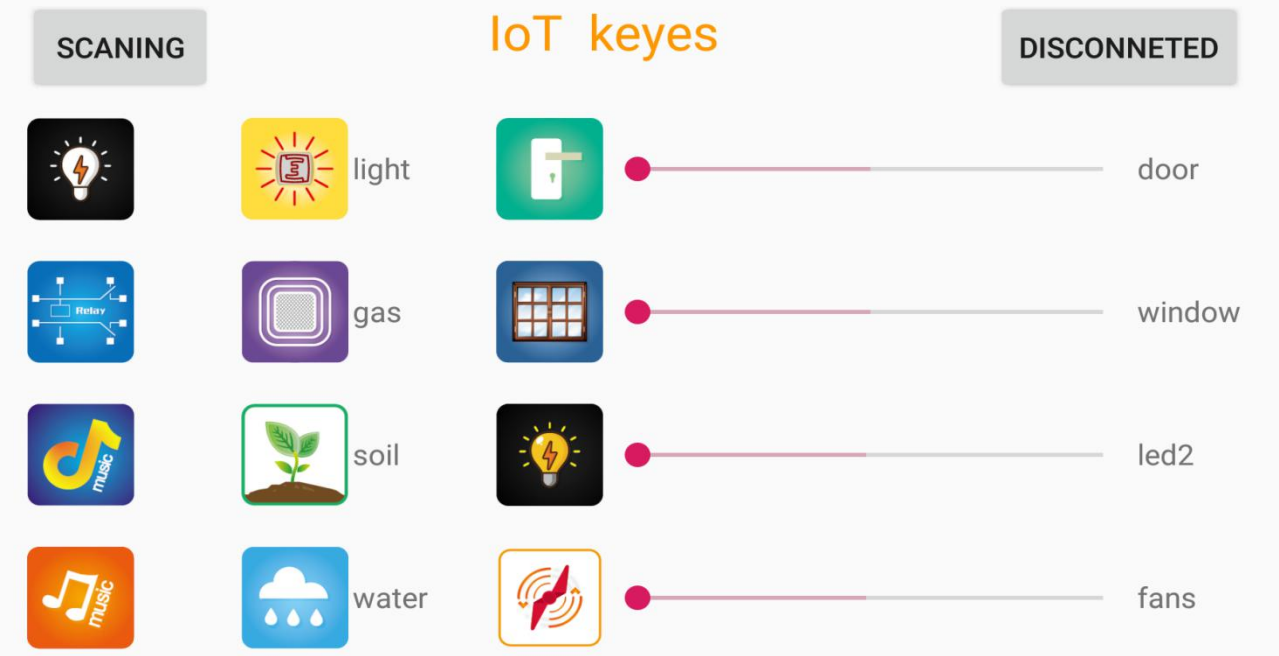
https://play.google.com/store/apps/details?id=com.keyestudio.iot_keyes

APP for Android mobile（domestic）：

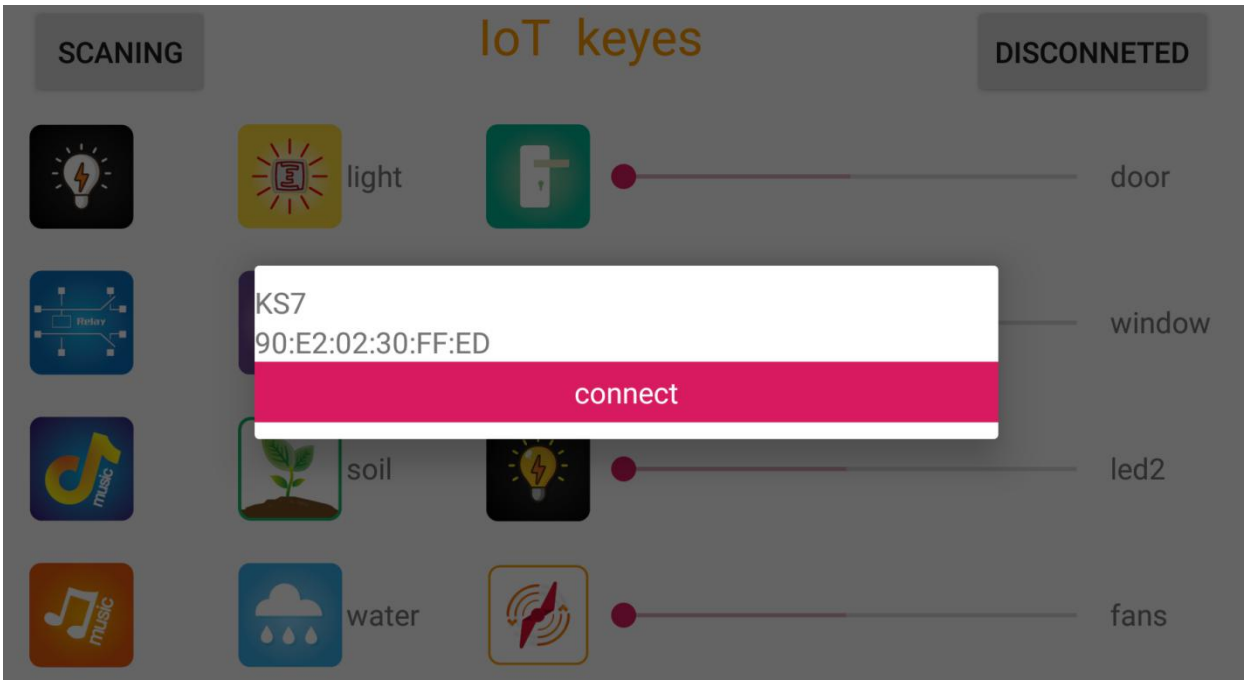
Download Link：https://pan.baidu.com/s/1xE1kGjG3CroPNH3-HgF6pQ

After installing and open the app 

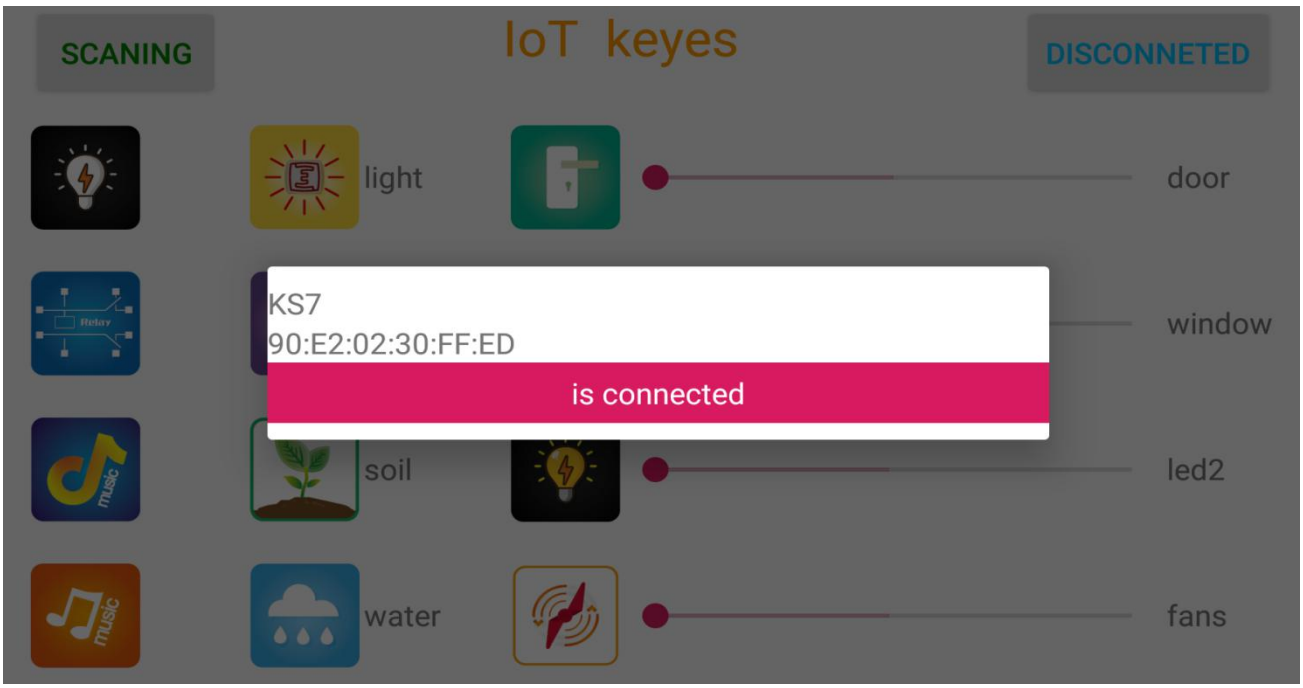
，the interface pops up as below:



Upload code and power on, Led of Bluetooth module blinks. Start Bluetooth and open App to click "CONNECT" to connect.

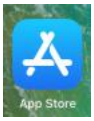


Click to "Connect", Bluetooth is connected successfully. As shown below, the LED of Bluetooth module is normally on.



3. For IOS system:

(1) Open App Store



(2) Search "IoT keyes"on APP Store, then click "downlaod".

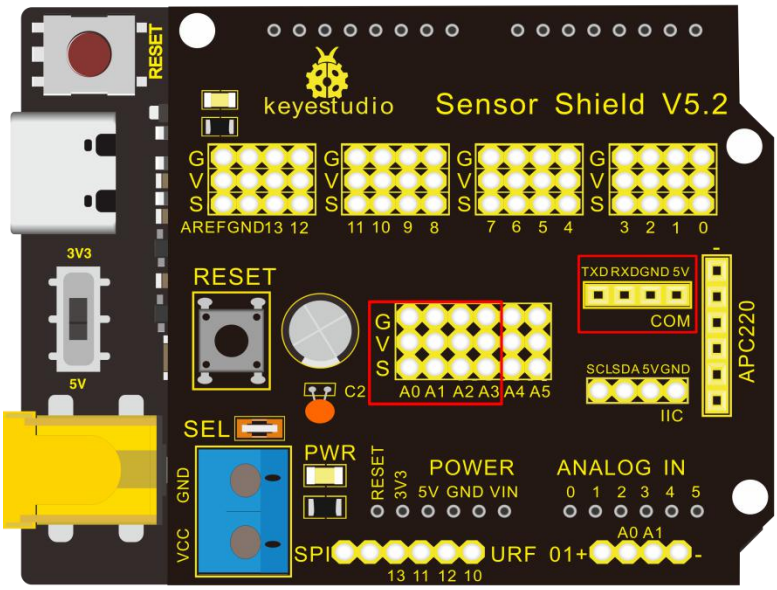
(3) After installing successfully and open 

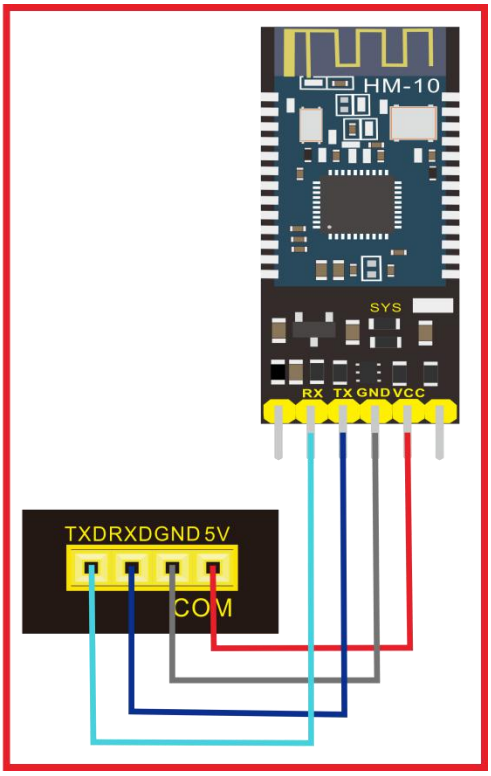
, the interface is shown below:

(4) Upload the test code successfully, insert the Bluetooth module and power on. LED of Bluetooth module is flashing. Start Bluetooth on cellphone, then click "connect" on the left to search Bluetooth and pair. After paring successfully, the LED of Bluetooth module is on.

Note: Remove the Bluetooth module please, when uploading the test code. Otherwise, the program will fail to upload. Connect the Bluetooth and Bluetooth module to pair after uploading the test code.

4. Connection Diagram





Note: On the sensor expansion board, the RXD, TXD, GND, and VCC of the Bluetooth module are respectively connected to TXD, RXD, GND, and 5V, and the STATE and BRK pins of the Bluetooth module do not need to be connected. Connect the power supply.

Test Code:

```
**********************************************************************
**********************************
char val;//set a variable val
void setup(){
   Serial.begin(9600);//set baud rate to 9600
}
```

```
void loop(){
  if (Serial.available()>0)//read the data on serial port
  {
    val = Serial.read();//assign the character read by serial port to val
        Serial.print(val);//serial port outputs val
  }}
```

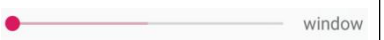
****

****

Key function on app:

SCANING	Function: pair and connect HM-10 Bluetooth module	
DISCONNECT	Function: disconnect Bluetooth	
	Press to send "a"; Press again to send "b"	Turn on the white light; Turn off light
	Press to send "c"; Press again to send "d"	Start relay Turn off relay
	Press to send "e"; Release to send "g"	Play music

	Press to send "f"; Release to send "g"	Play another music
	Press to send "h"; Press again to send "S"	receive the light intensity detected by photocell sensor, "light" changes into data value; Turn off photocell sensor
	Press to send "i"; Press again to send "S"	receive the gas or smoke data detected by gas sensor, "gas" changes into data value; Turn off the gas sensor
	Press to send "j"; Press again to send "S"	Receive the humidity data detected by soil humidity sensor detects humidity, "soil" changes into value; Turn off soil humidity

		sensor
	Press to send "k"; Press again to send "S"	Receive the water capacity detected by steam sensor, "water" changes into data value; Turn off steam sensor
	Press to send "l"; Release to send "m"	Open the door, close the door
Progress bar 	The value stands for the rotation angle of servo1	Control the door to move, the bigger the value, the servo1 rotates clockwise, otherwise, it rotates counterclockwise
	Press to send "n"; Release to send "o"	Open the window, close the window
Progress bar 	The value stands for the rotation angle of servo2	Control the window and angle of servo2, when value is 180, the

		window is fully open, when value is 0, the window is closed
	Press to send "p"; Release to send "q"	Turn on yellow LED; Turn off LED
Progress bar 	The value stands for light intensity, when value is 255, the led2 is brightest, when value is 0, the led2 is off	Control the brightness led2 stands for PWM value, the bigger the value, the brighter the led2
	Press to send "r"; Release to send "s"	Turn on fan module, turn off fan module
Progress bar 	The value represents the rotation speed, when value is 255, the fan rotates fastest, when value is 0, fan stops	Control the rotation speed of fan , the bigger the value is, the faster the fan rotates

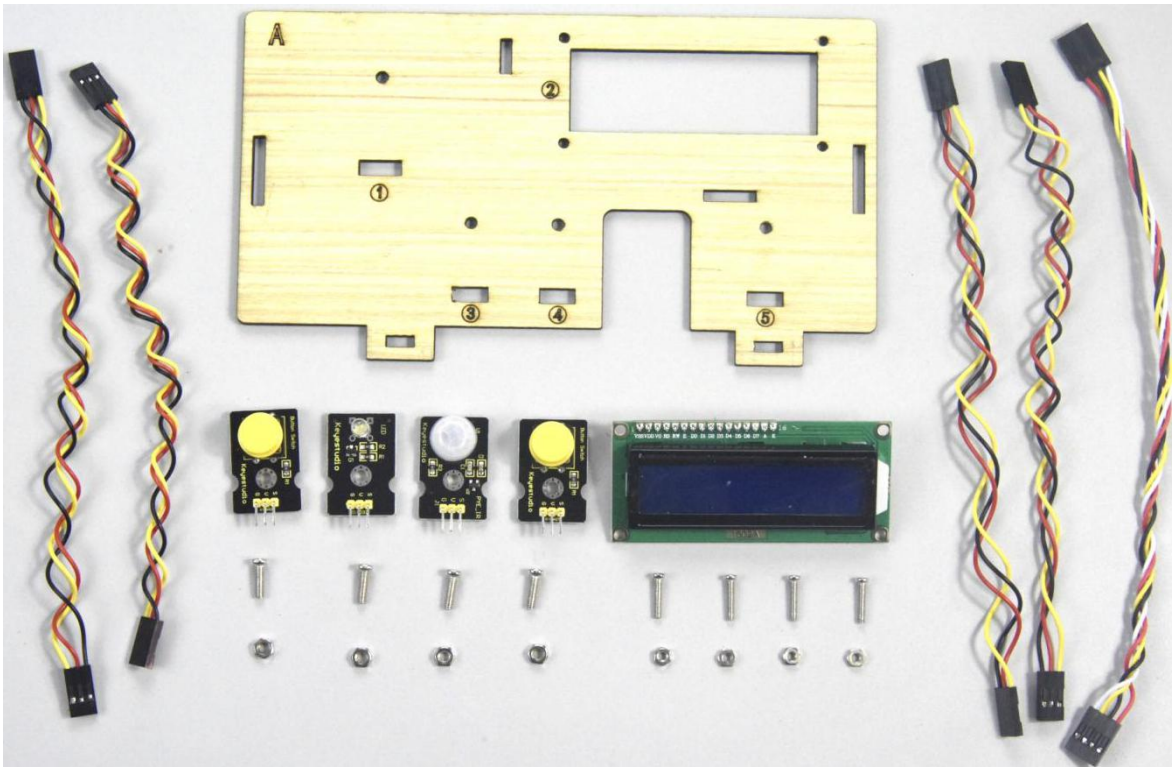
6. Assembled Guide

Check the board A~I and parts firstly

Step 1: Install sensors of A board

- 1*A board
- 4*M3*10MM round head screws
- 4*M3 nuts
- 4*M2.5*10MM round head screws

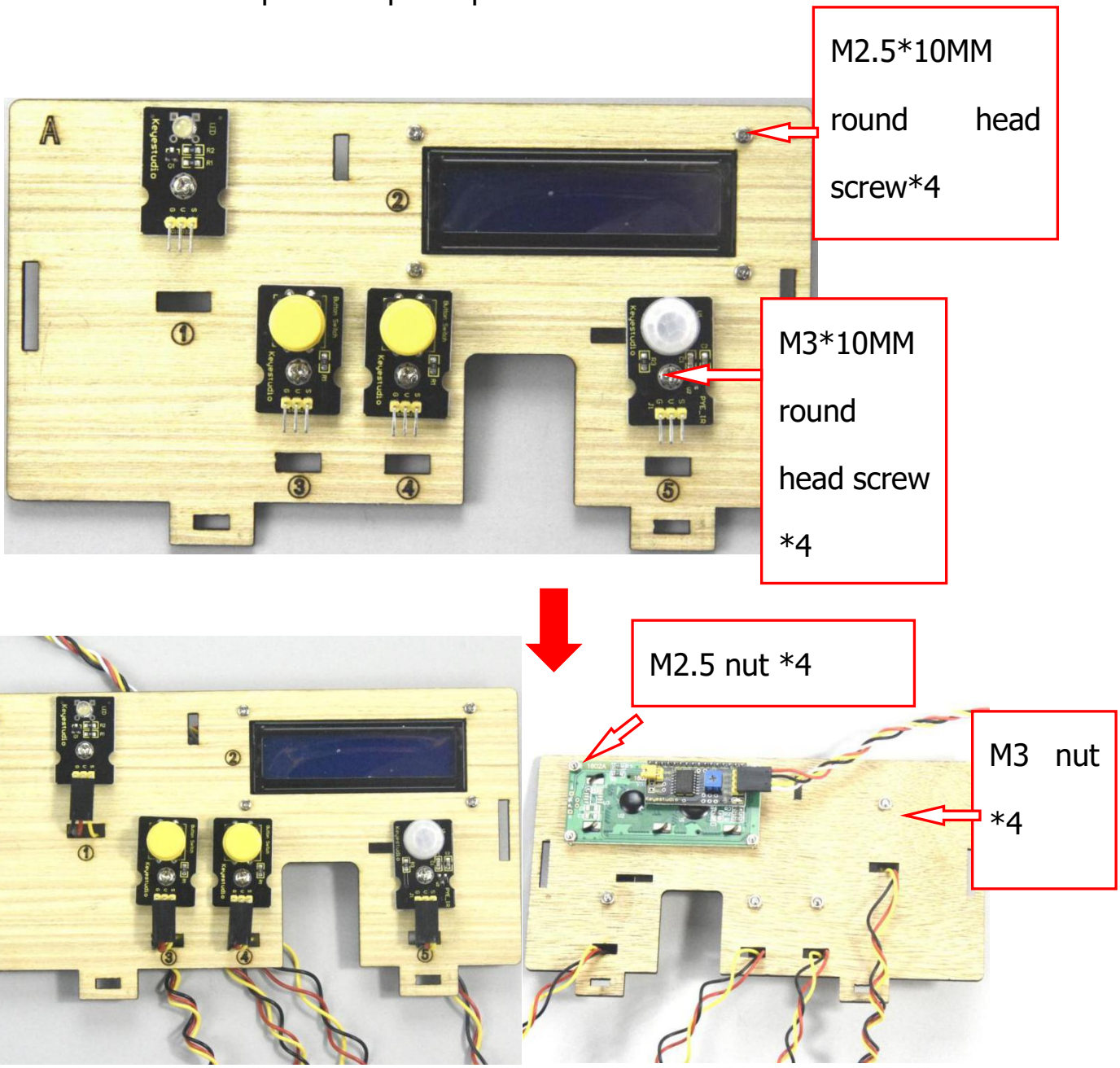
- 4*M2.5 nuts
- 1*White LED module
- 1*PIR motion sensor
- 2* Button sensor
- 1*LCD1602 display
- 4*3pin female to female dupont lines
- 1*4pin female to female dupont line



a. Fix white LED, 2 button sensors and PIR motion sensor on the corresponding area of the A board with 4pcs M3*10MM round head screws and 4pcs M3 nuts.

b. Then install LCD1602 display on A board with 4pcs M2.5*10MM round head screws and 4pcs M2.5 nuts.

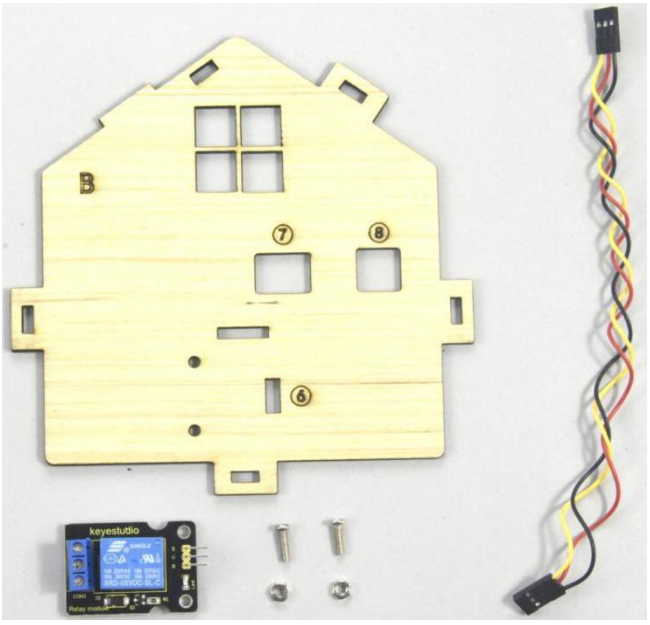
c. Connect them with 3pin and 4pin dupont lines.



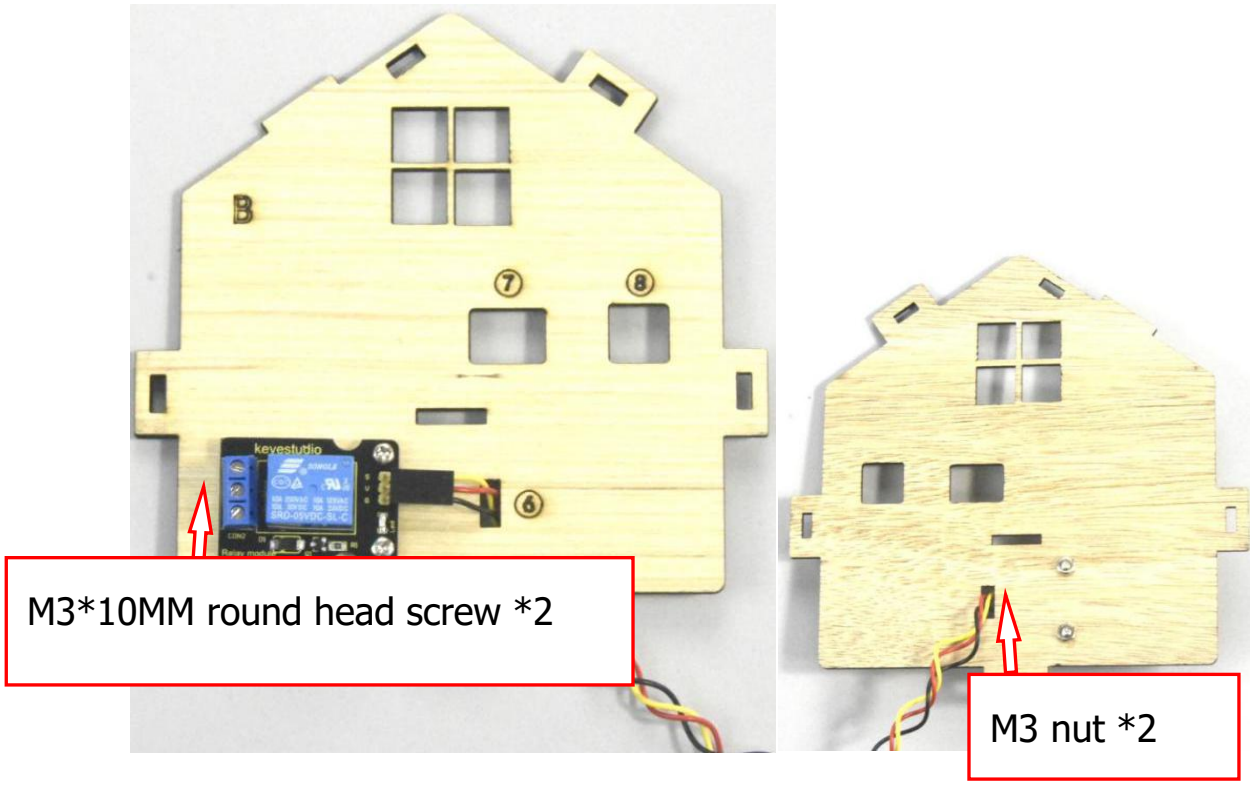
Step 2: Install the sensor of B board

- B board*1
- M3*10MM round screw*4

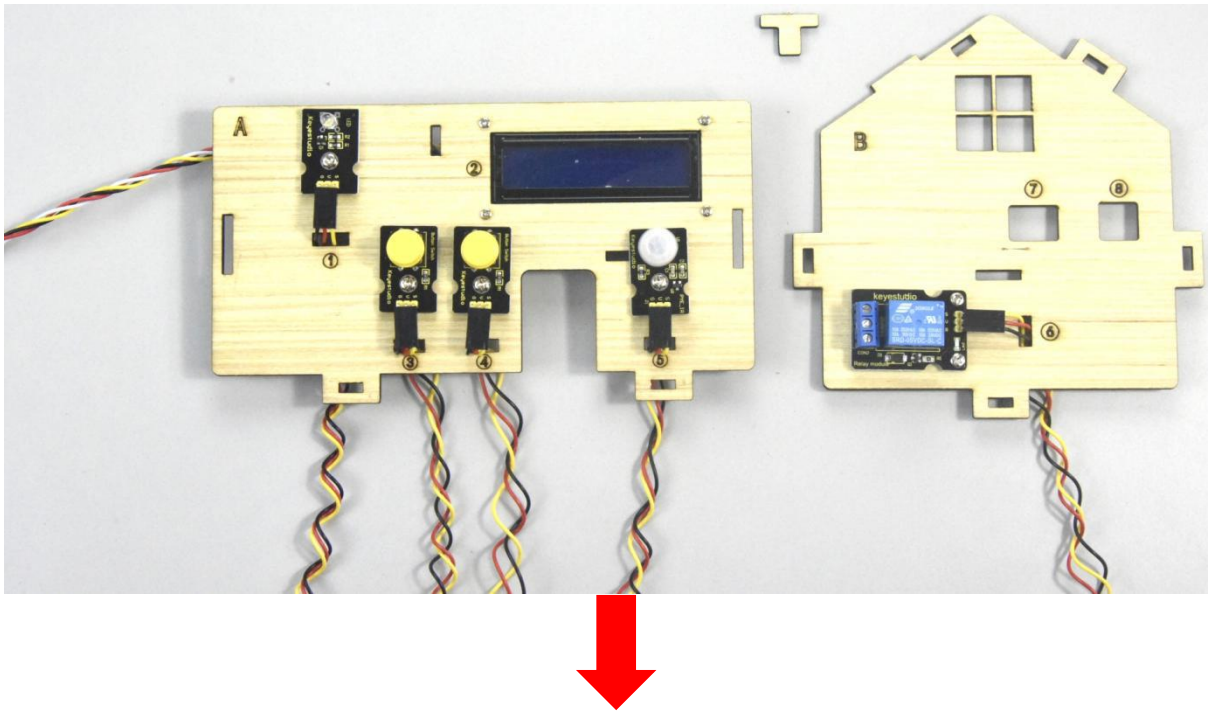
- M3 nut*4
- Relay module*1
- 3pin female to female dupont line *1

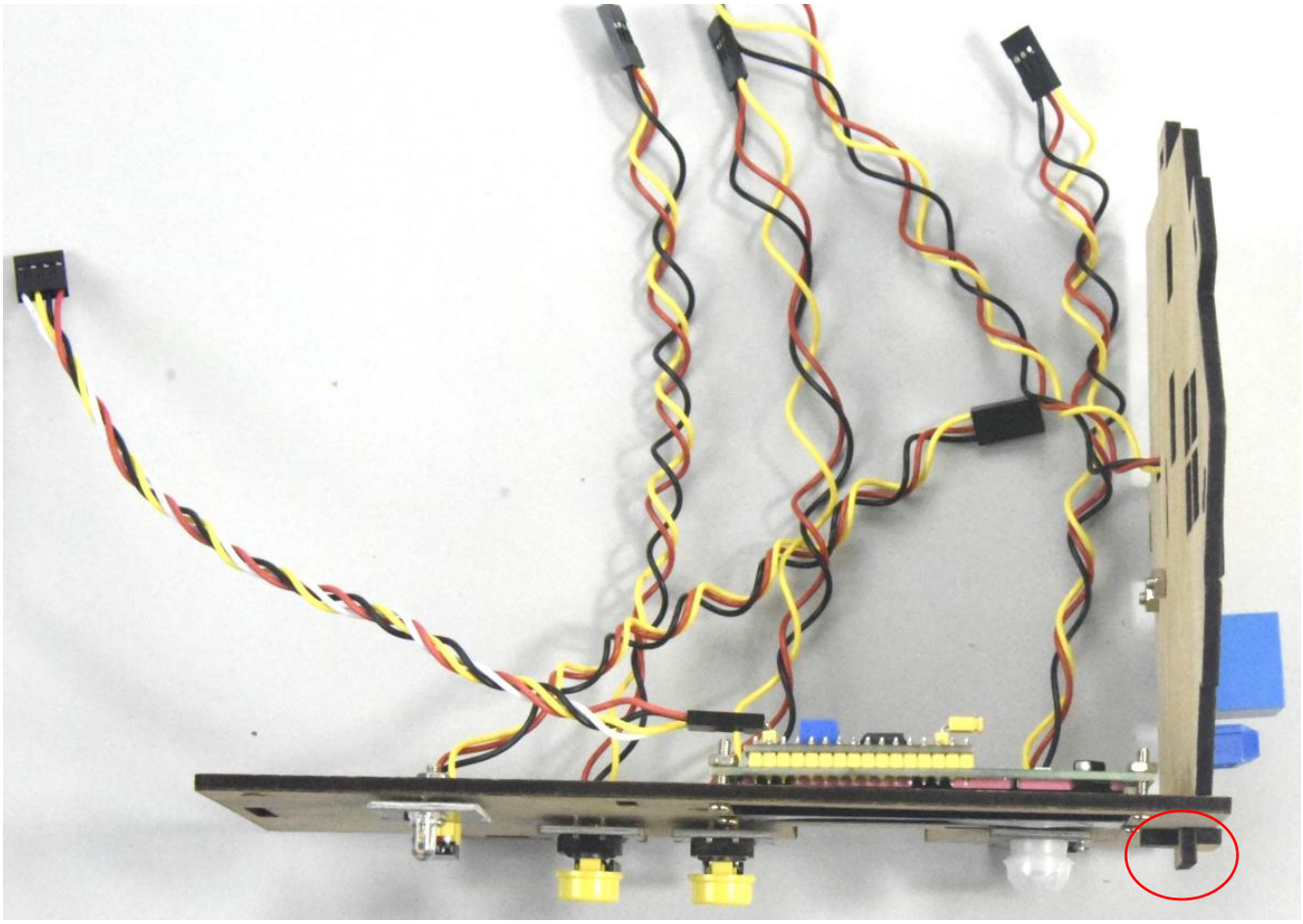


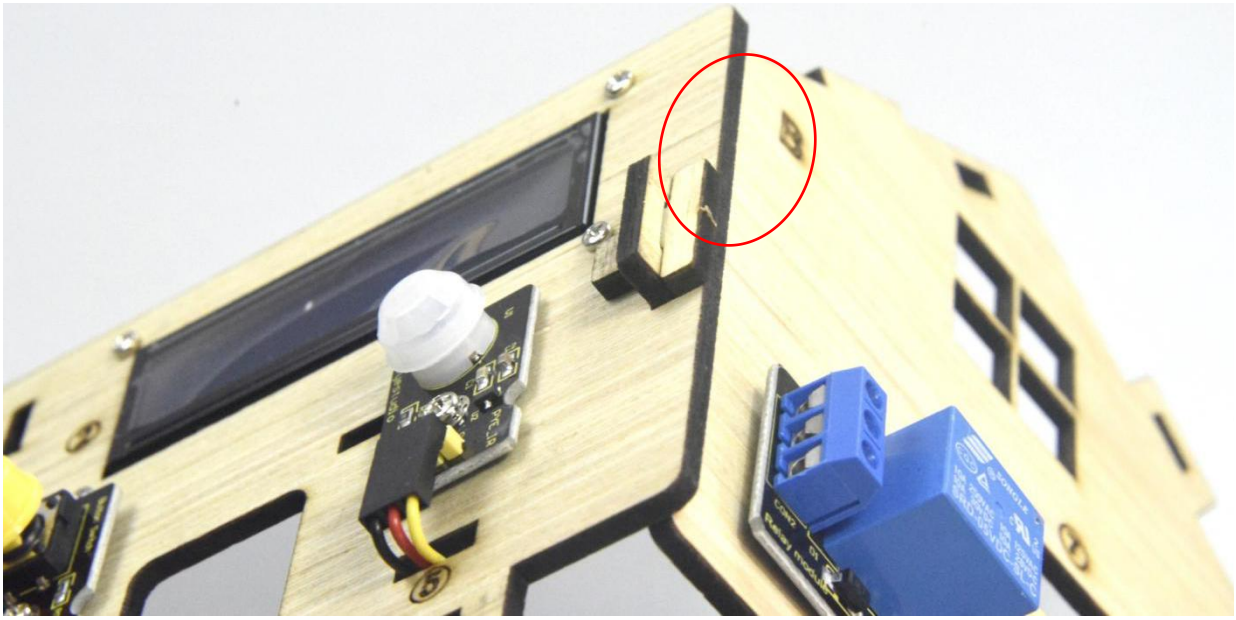
Assemble the relay module on B board with 2 pcs M3*10MM screws and 2pcs M3 nuts, link them together with 3pin dupont line



Step 3: Fix A board and B board together with a “T” bolt

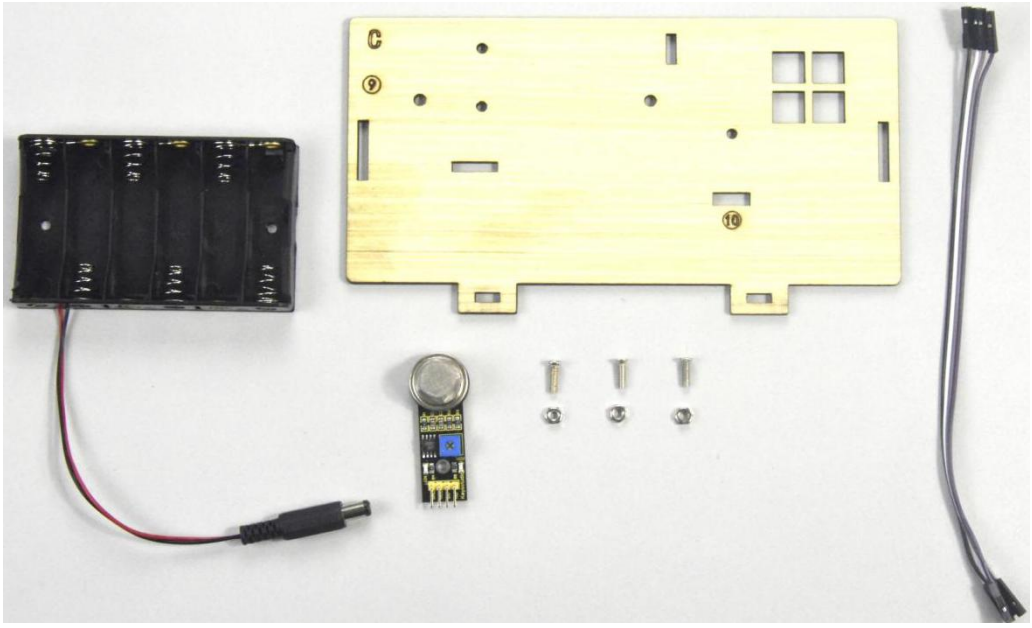






Step 4: Assemble the sensors and battery holder of C board

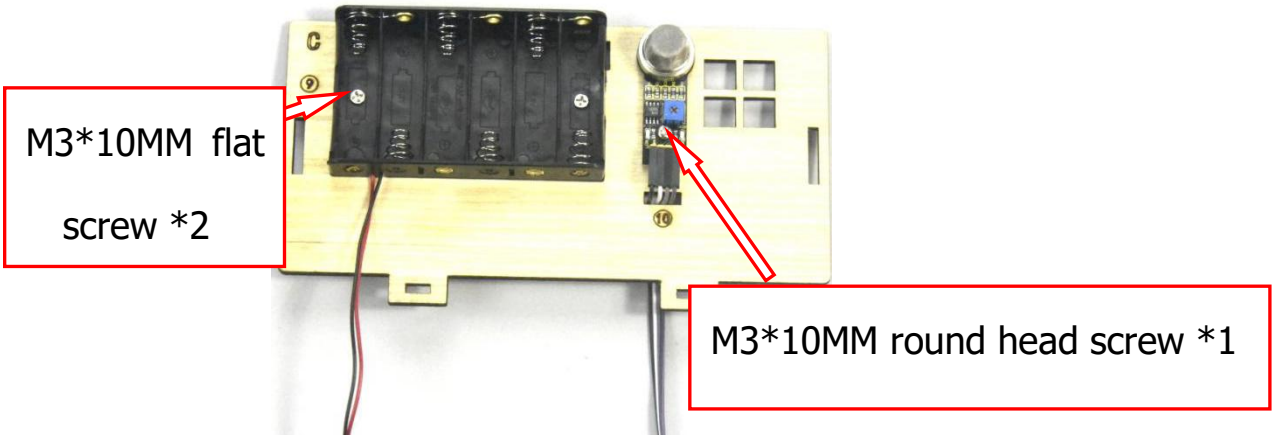
- C board*1
- MQ-2 gas sensor
- Battery holder
- M3*10MM flat screw*2
- M3*10MM round screw*1
- M3 nut*3
- 4 female to female dupont lines

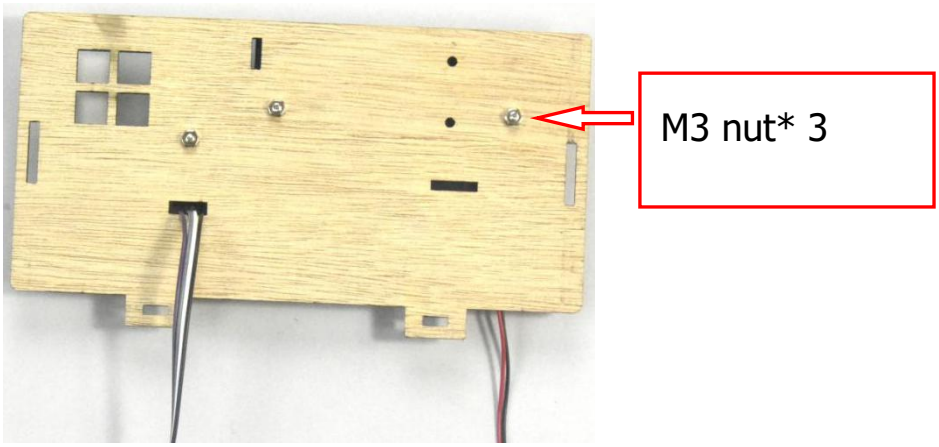


A. Fix the battery holder on C board with 2pcs M3*10MM flat head screws and 2 pcs M2 nuts

B. Then install the MQ-2 gas sensor on the corresponding area of C board with a M3*10MM round head screw and a M3 nut.

C. Connect them with 4 female to female dupont lines

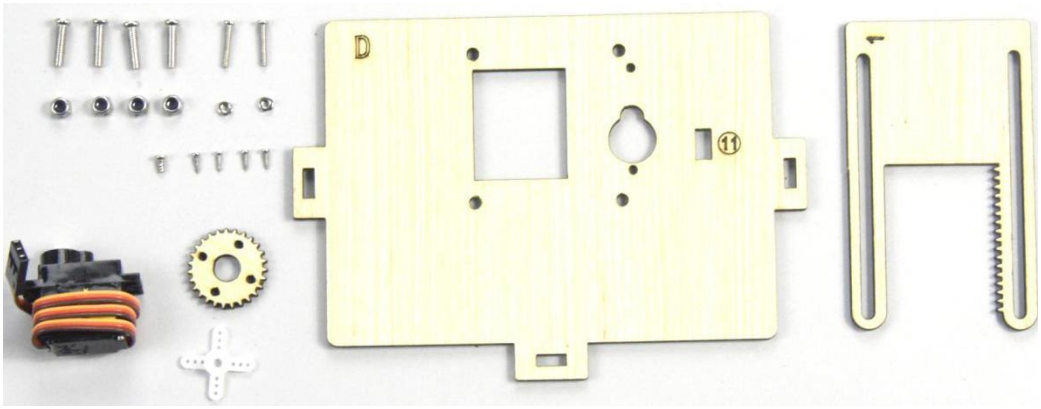




Step 5: Install the sensors and parts of D board

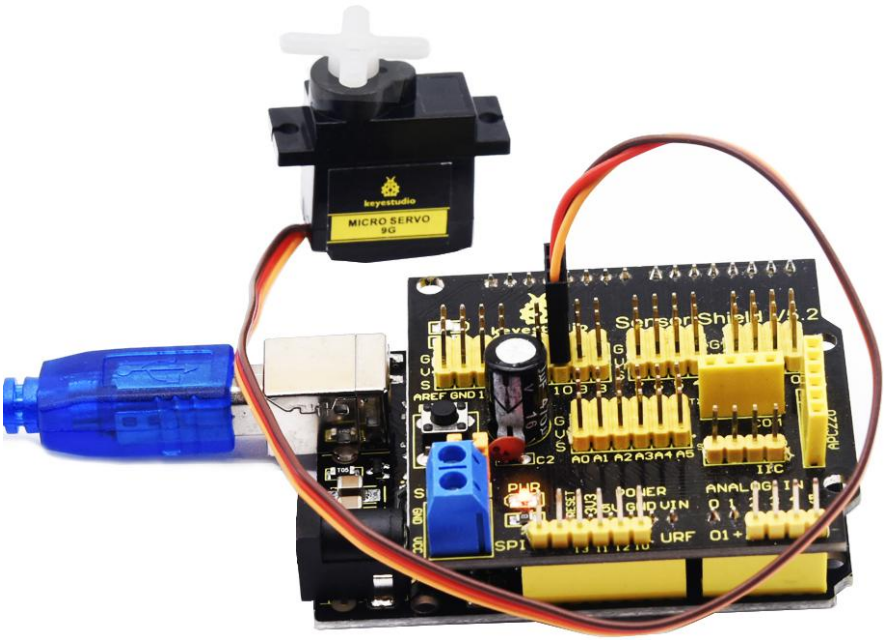
- Black servo*1
- M3*10MM self-tapping screw*4
- Cross white mount*1 (included in servo)
- M2*5 round head screw*1 (included in servo)
- M2*12MM round screw*2
- M2 nut*2
- M3*12MM round head screw*4
- M3 self-locking nut*4
- D board*1
- Gear*1

- Board 1*1



Rotate servo to 90° before installing, connect servo to keyestudio PLUS control board; upload test code on control board and make servo rotate to 90°

Servo Motor	
Brown wire	GND
Red wire	5V
Orange wire	S （10）



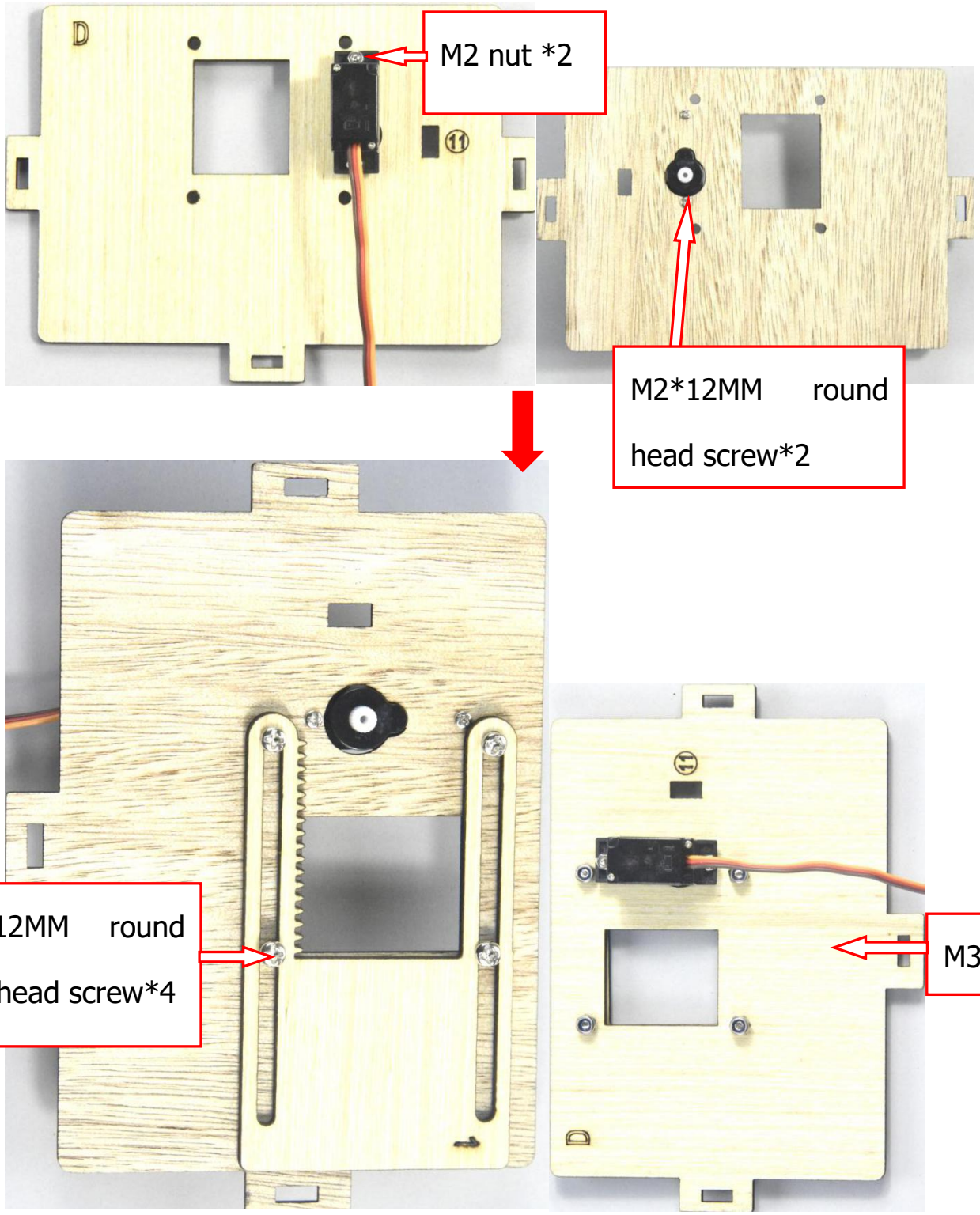
Test Code:

```
#include <Servo.h>
Servo servo_10;


void setup(){
   servo_10.attach(10);
}
void loop(){
   servo_10.write(90);
   delay(500);}
```

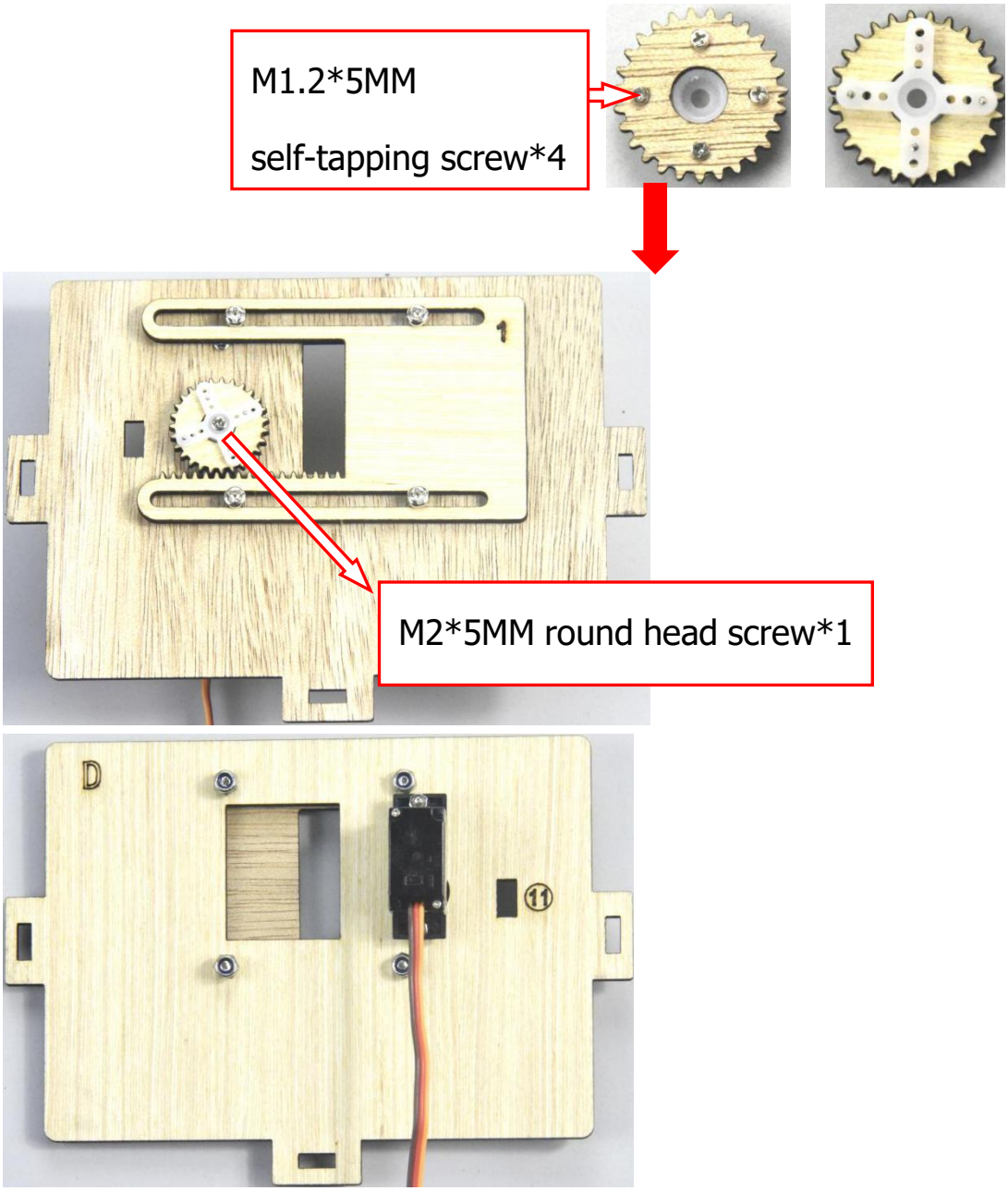
Upload the test code successfully, the servo rotates to 90°

A. Fix servo on the corresponding area on D board with 2pcs M2*12MM round head screws and 2 M2 nuts.

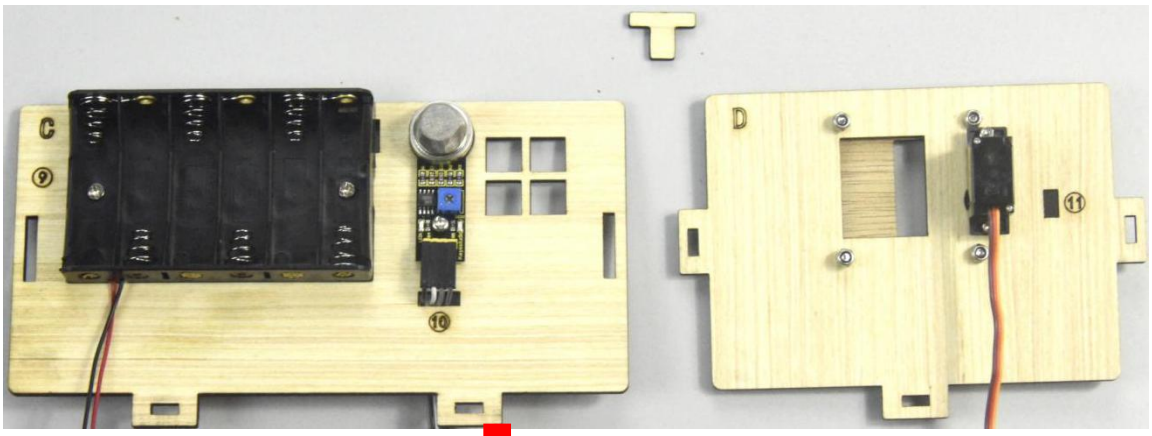
B. Then install the square board 1 on the D board with 4pcs M3*12MM round head screws and 4 M3 self-locking nuts.

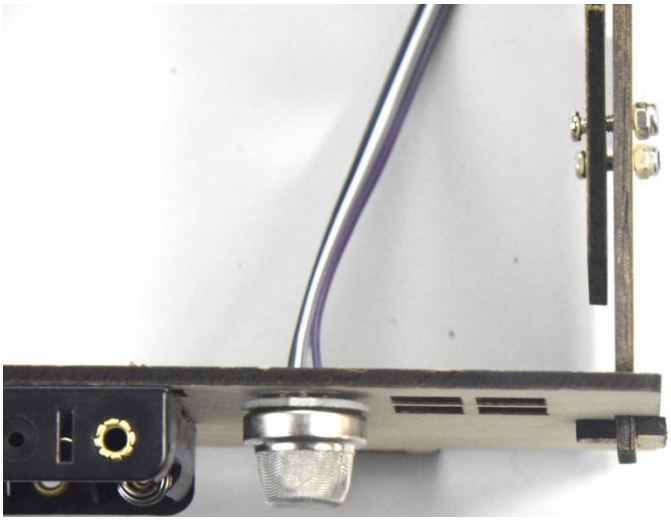


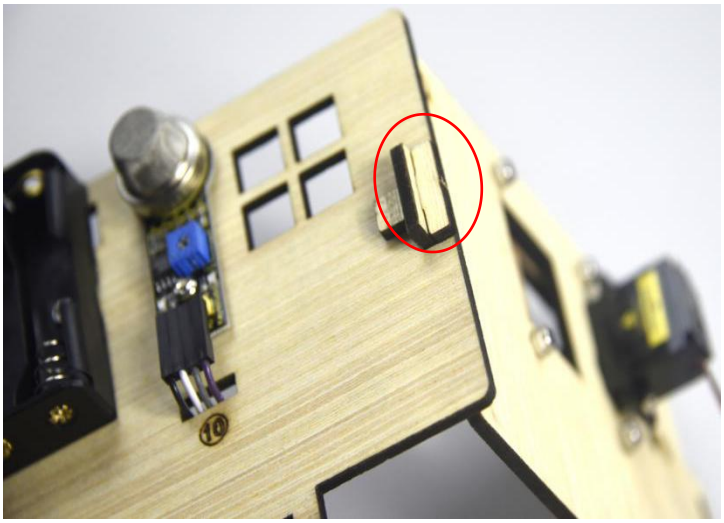
Fix the white cross mount on the gear with 4pcs M1.2*5MM self-tapping screws, and mount the gear on the servo motor with 1 M2*5MM round head screw.



Step 6: Assemble C board with D board by a “T” type bolt.



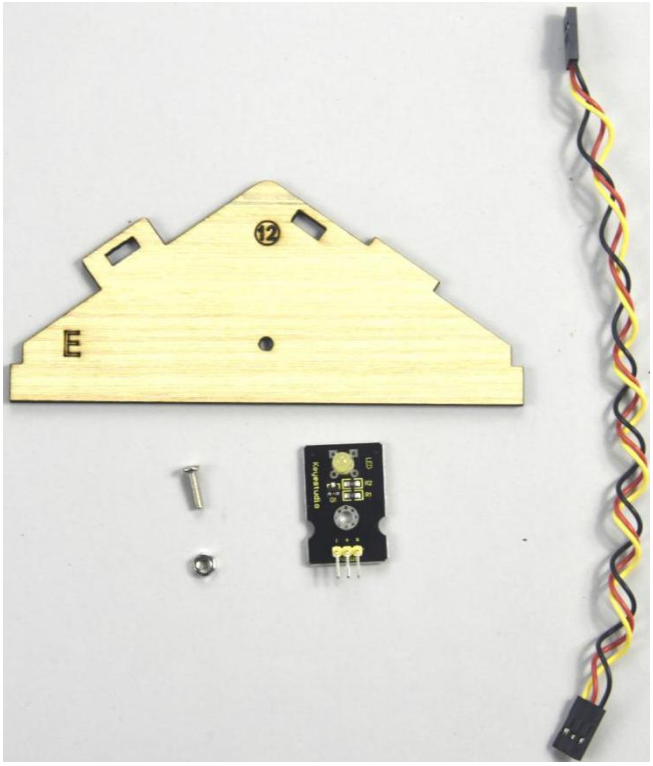




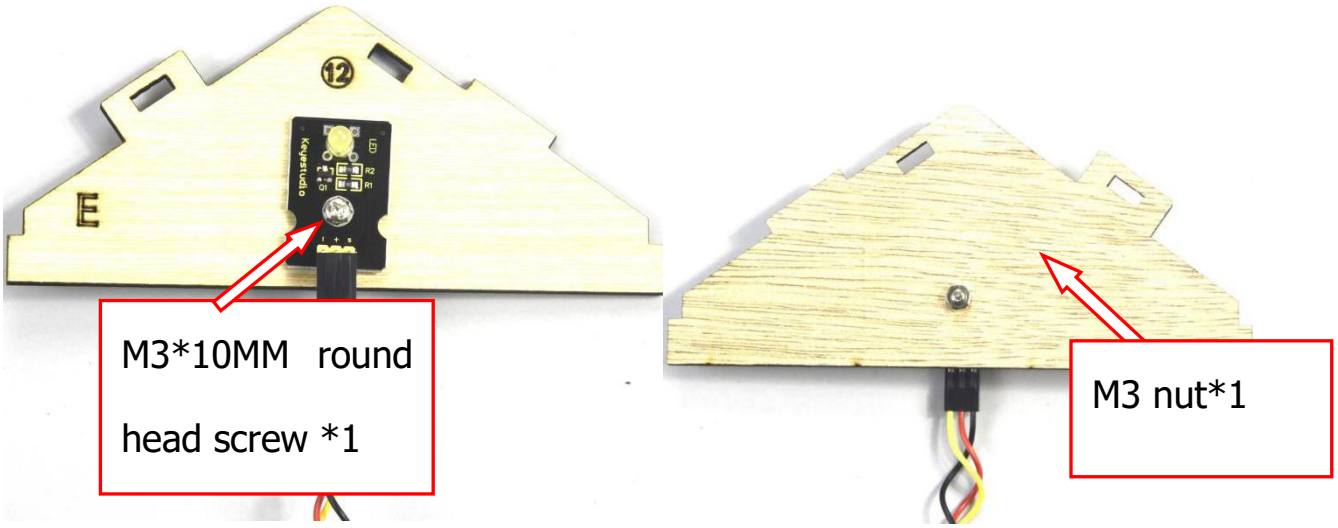
Step 7: install the sensor of E board

- Yellow LED*1
- E board*1

- M3*10MM round head screw*1
- M3 nut*1
- 3pin female to female dupont line*1



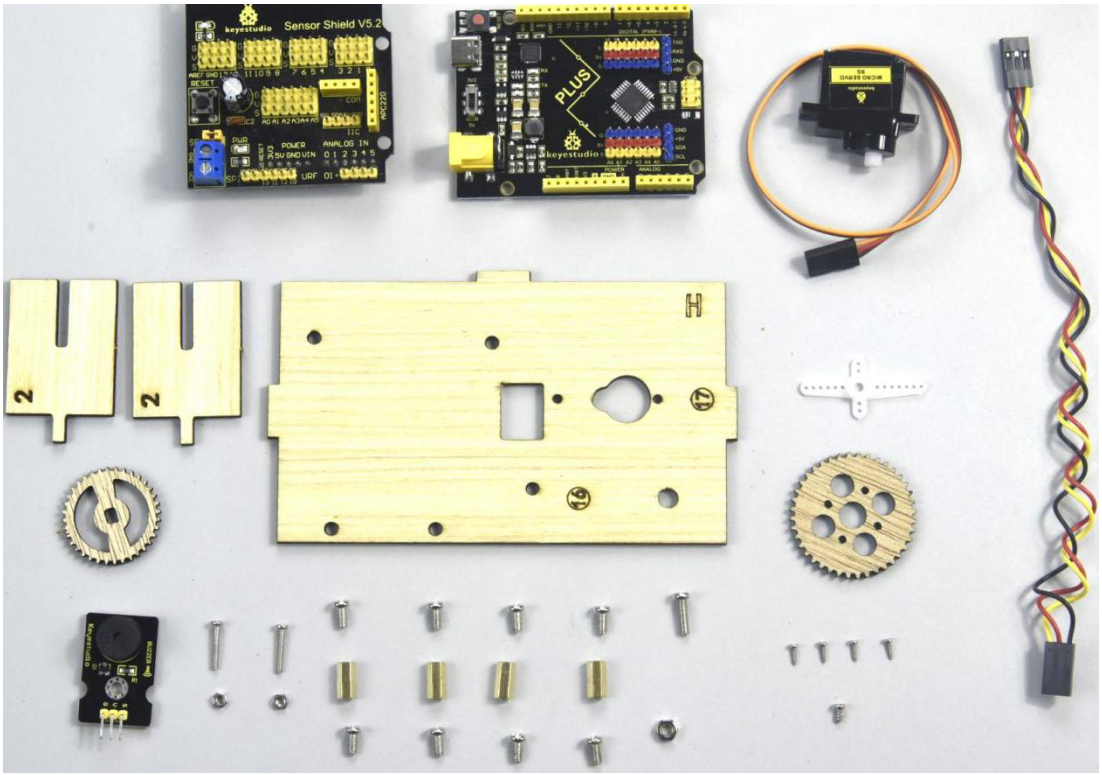
Assemble the yellow LED on the corresponding area of E board with 1 M3*10MM round head screw and 1 M3 nut，then connect them with a 3pin female to female dupont line.



Step 8: Install control board, sensors and parts of H board

- Black servo*1
- Passive buzzer*1
- M1.2*5 self-tapping screw*4
- White cross mount (included in servo)
- M2*5 screw *1 (included in servo)
- M2*12MM round head screw*2
- M2 nut*2
- M3*10MM round head screw*1
- M3 nut*1
- M3*6MM round head screw*8
- M3*10MM dual-pass copper pillar*4
- keyestudio PLUS Control Board
- Sensor expansion board
- 3pin female to female dupont line*1

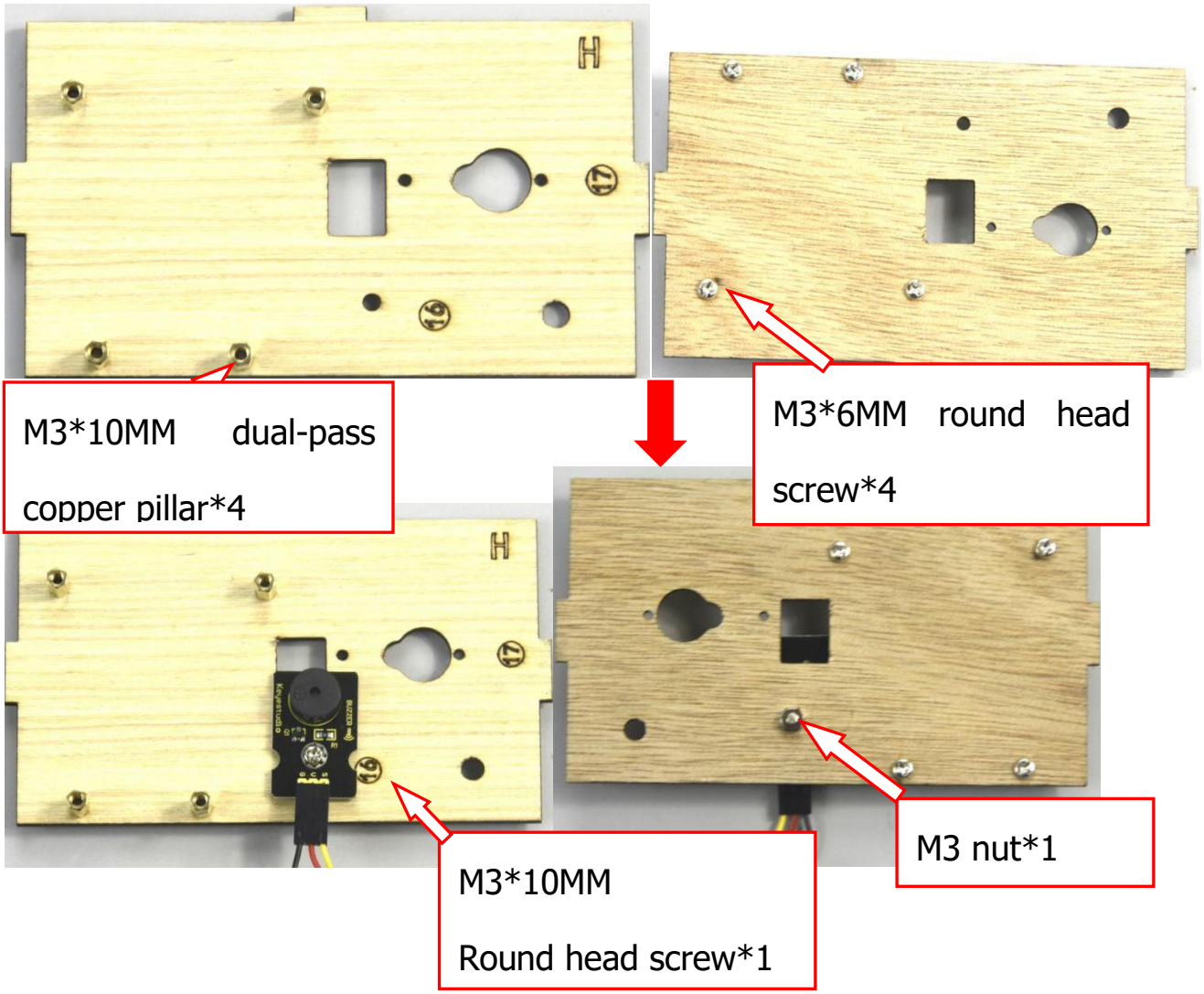
- Board E
- Gear *2
- Board 2*2



A. Mount 4pcs dual-pass copper pillars on the H board with 4pcs M3*6MM screws

B. Then fix the passive buzzer on H board with 1 M3*10MM round head screw and 1 MS nut.

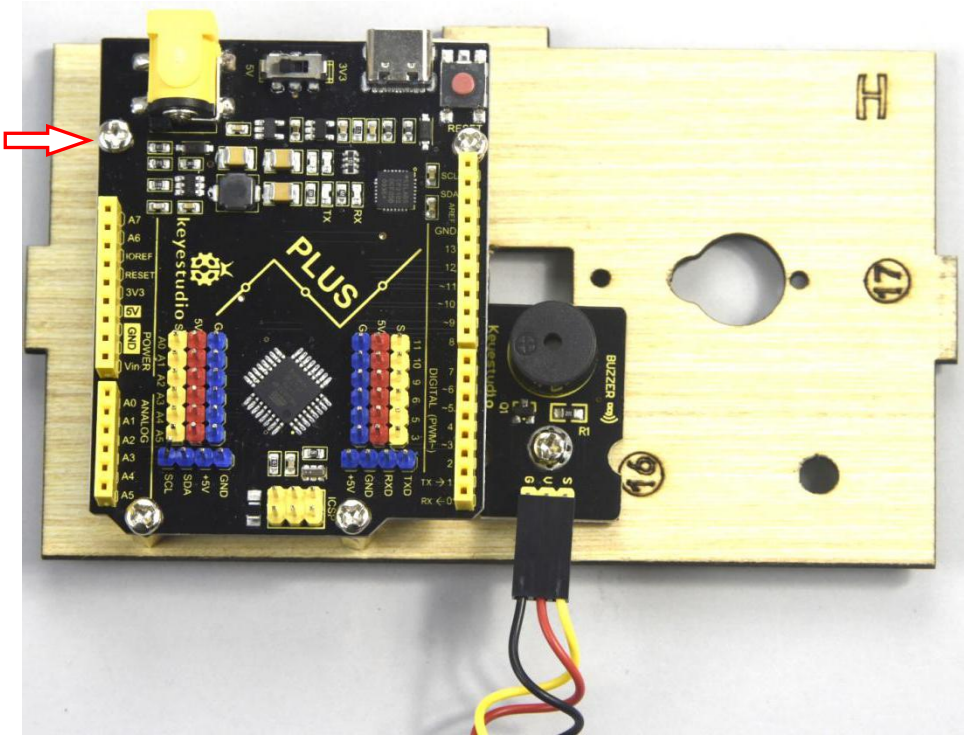
C. Connect them with a 3pin female to female dupont wire

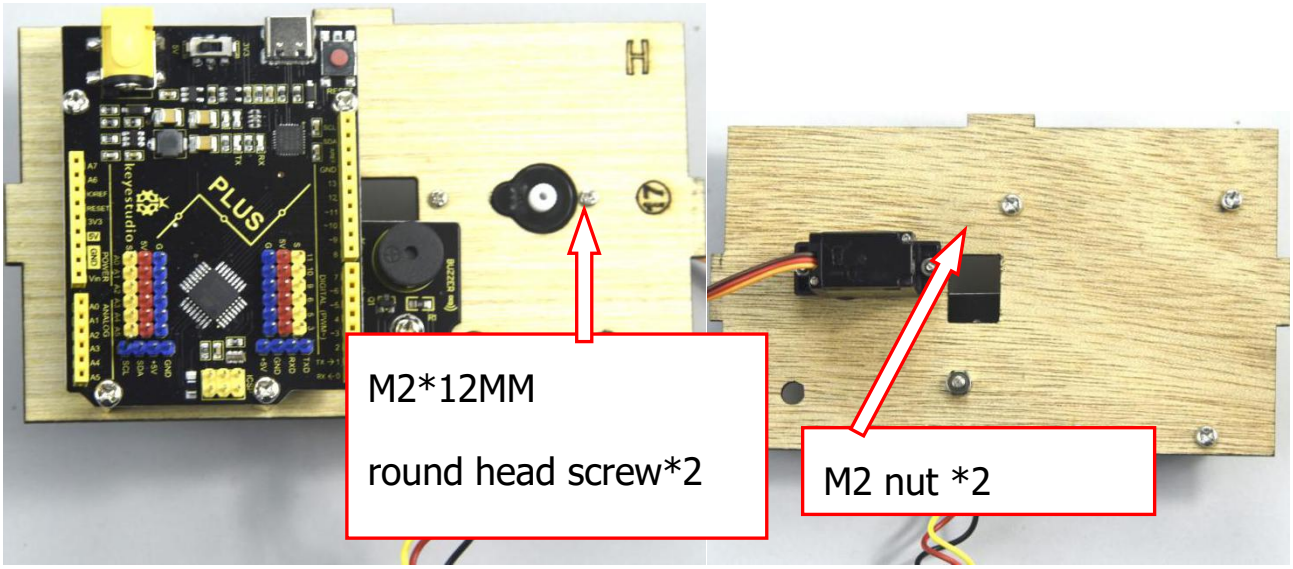


Rotate the servo to 90° before installing, the method is same as the step 6.

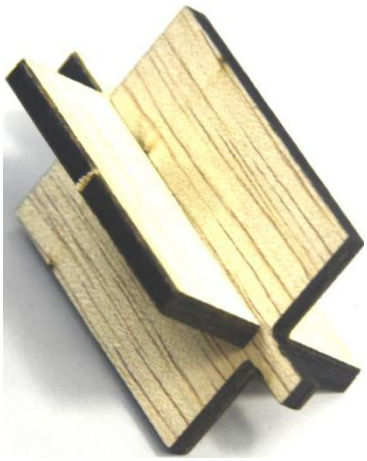
Assemble the 4pcs M3*10MM copper pillars on the keyestudio PLUS control board with 4 M3*6MM round head screws, then fix the servo on the corresponding area of H board with 2 M2*12MM round head screws and 2 M2 nuts.

M3*6MM round head screw *4

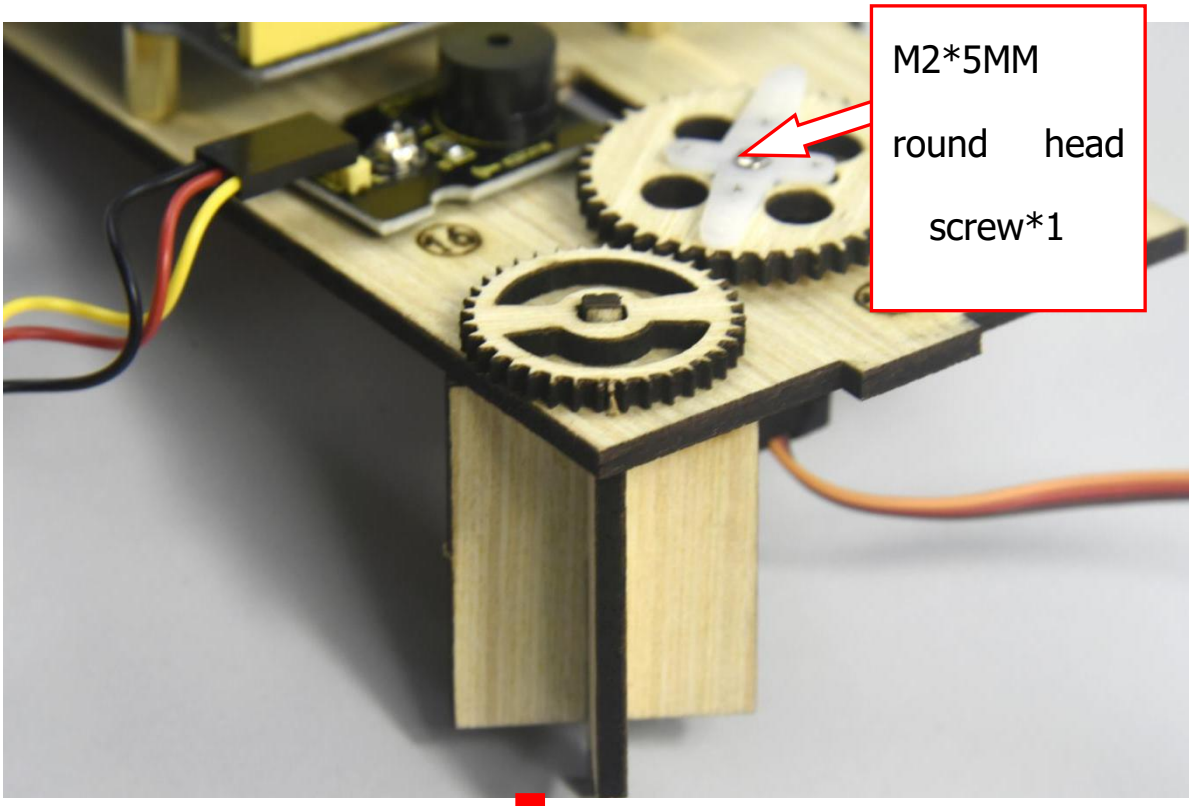


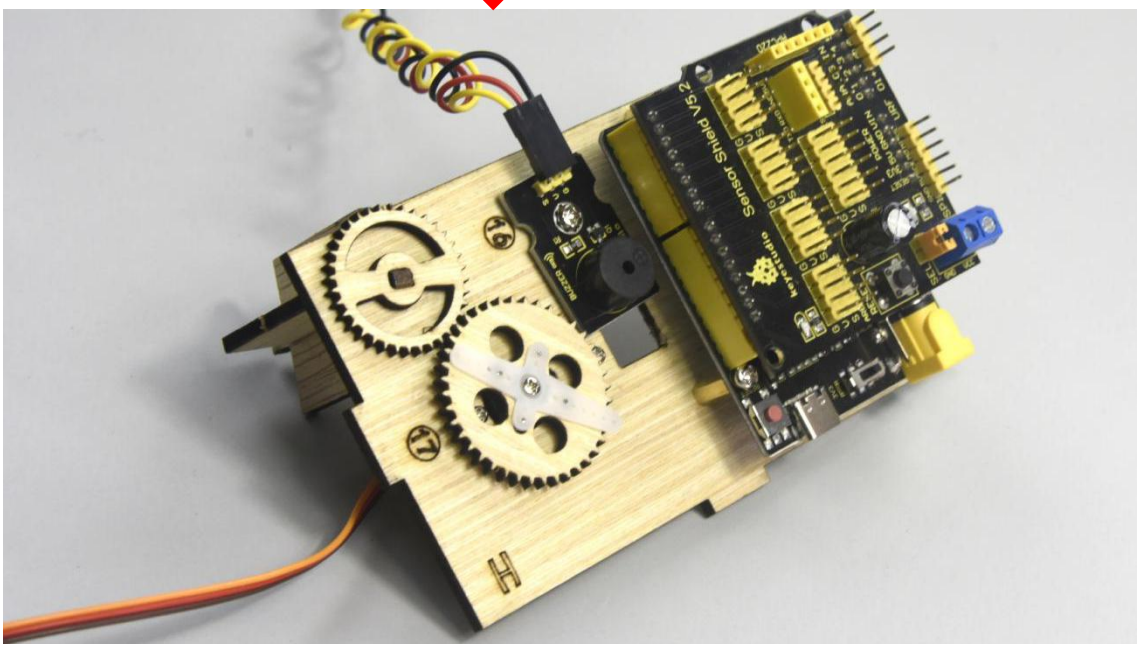


Assemble 2pcs board 2 together, then fix white cross mount on the gear with 4pcs M1.2*5 self-tapping screws

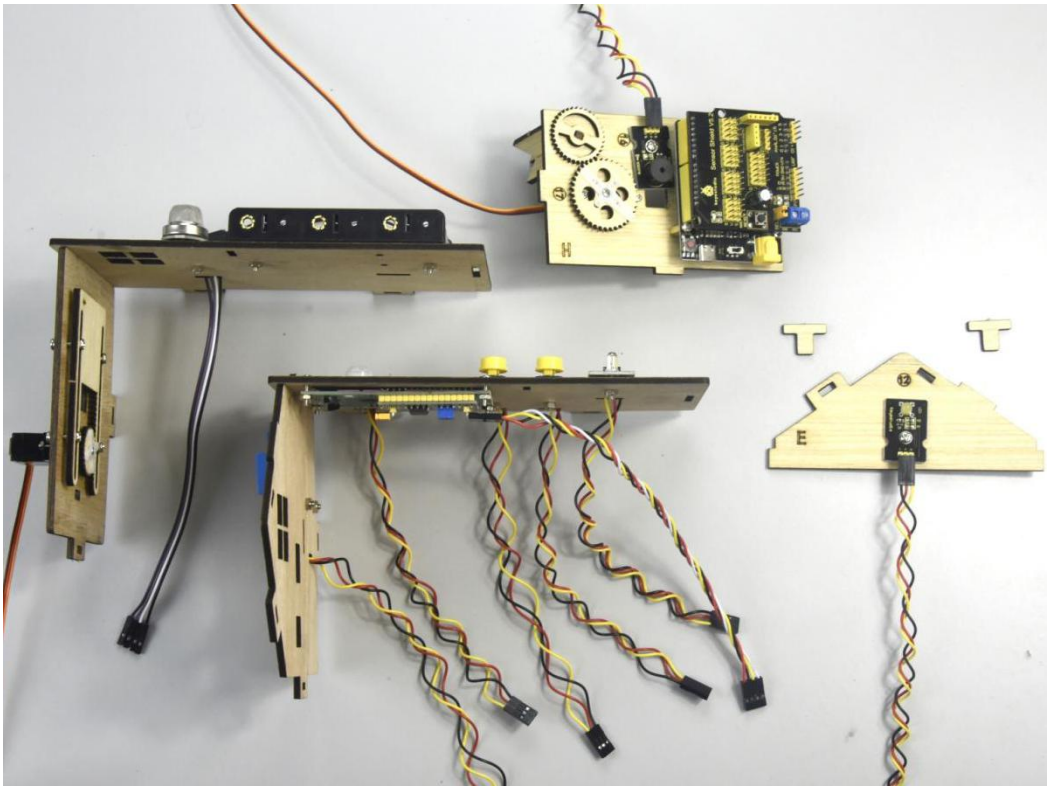


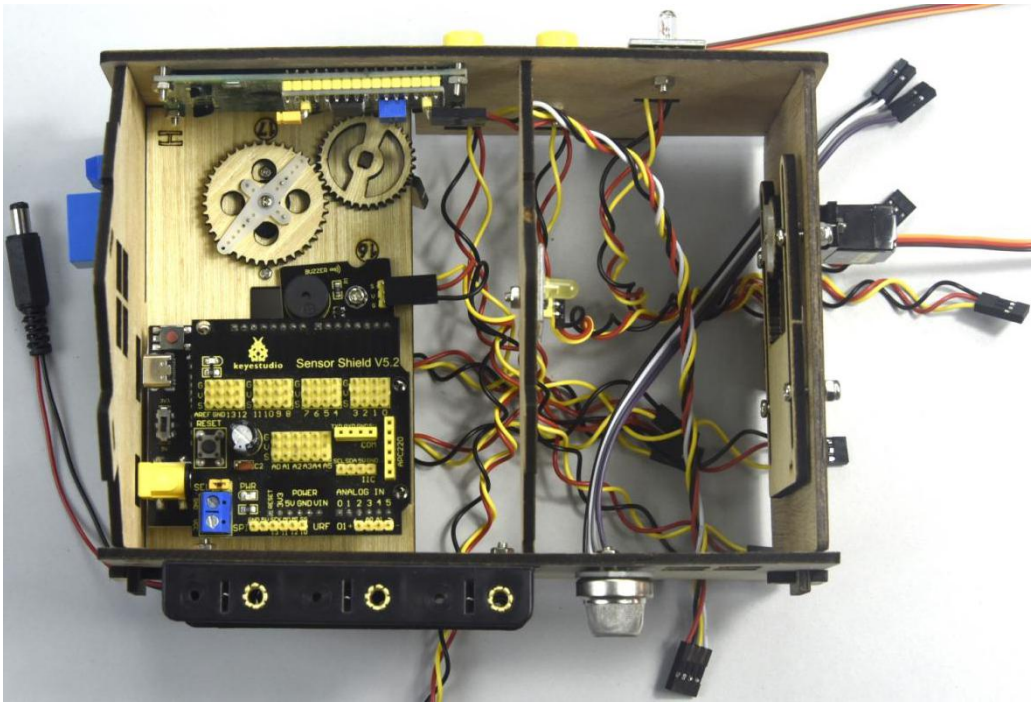
Fix the gear with white cross mount on the black servo by 1 M2*5MM screw(included in servo), then install the combination of 2pcs board 2 and another servo on the corresponding area of H board, finally stack the sensor shield on the keyestudio PLUS control board.

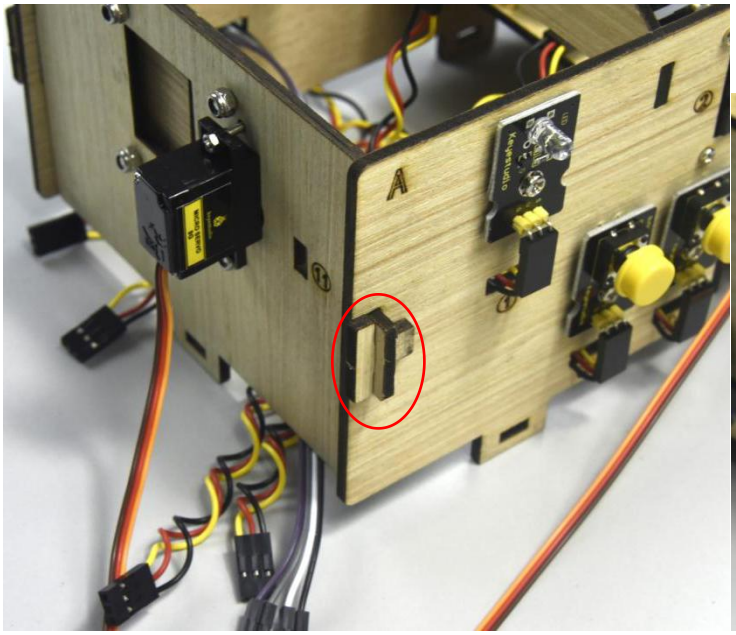

M2*5MM
round head
screw*1

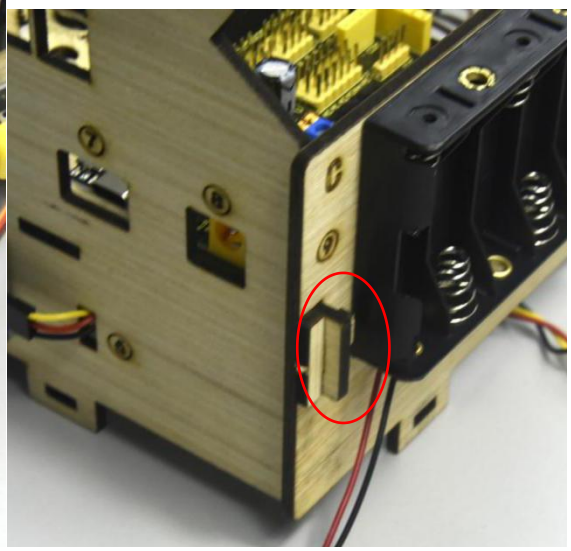


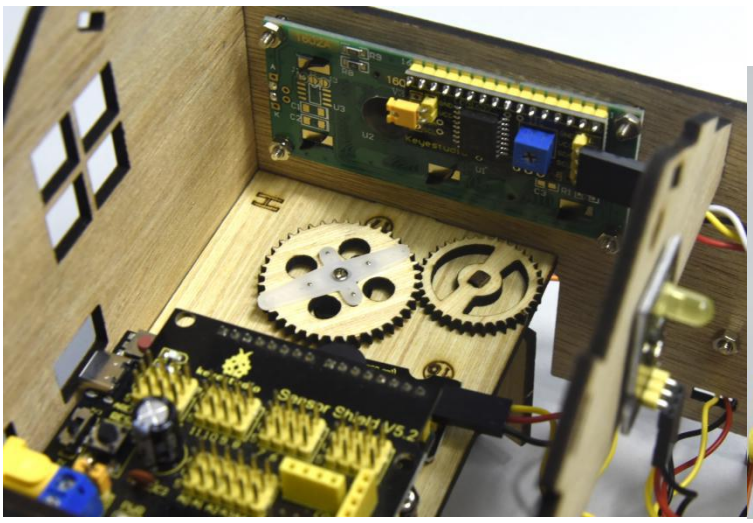
Step 9: Assemble A, B, C, D, E and H board together, then fix them with 2 "T" type bolts. (Note: the power interface of PLUS control board is aligned with the hole ⑧ on board B, and the interface of USB cable is aligned with the hole ⑦ on board B)

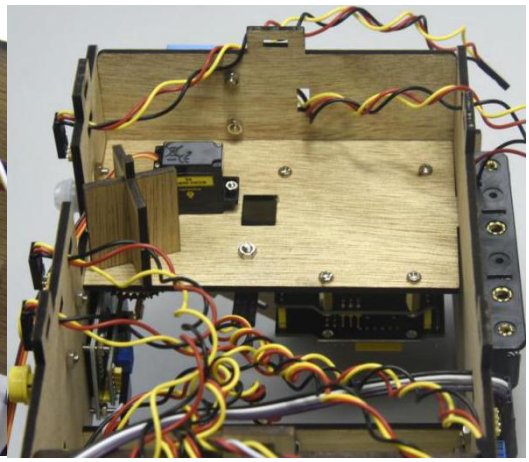



keyestudio
Sensor Shield V5.2



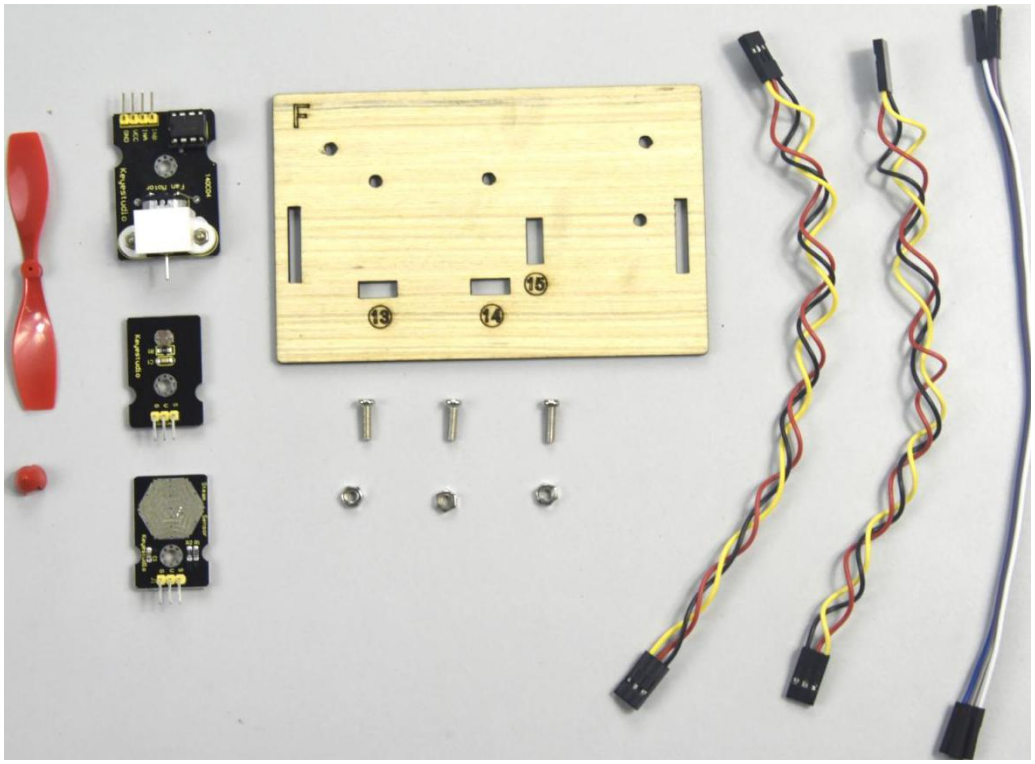




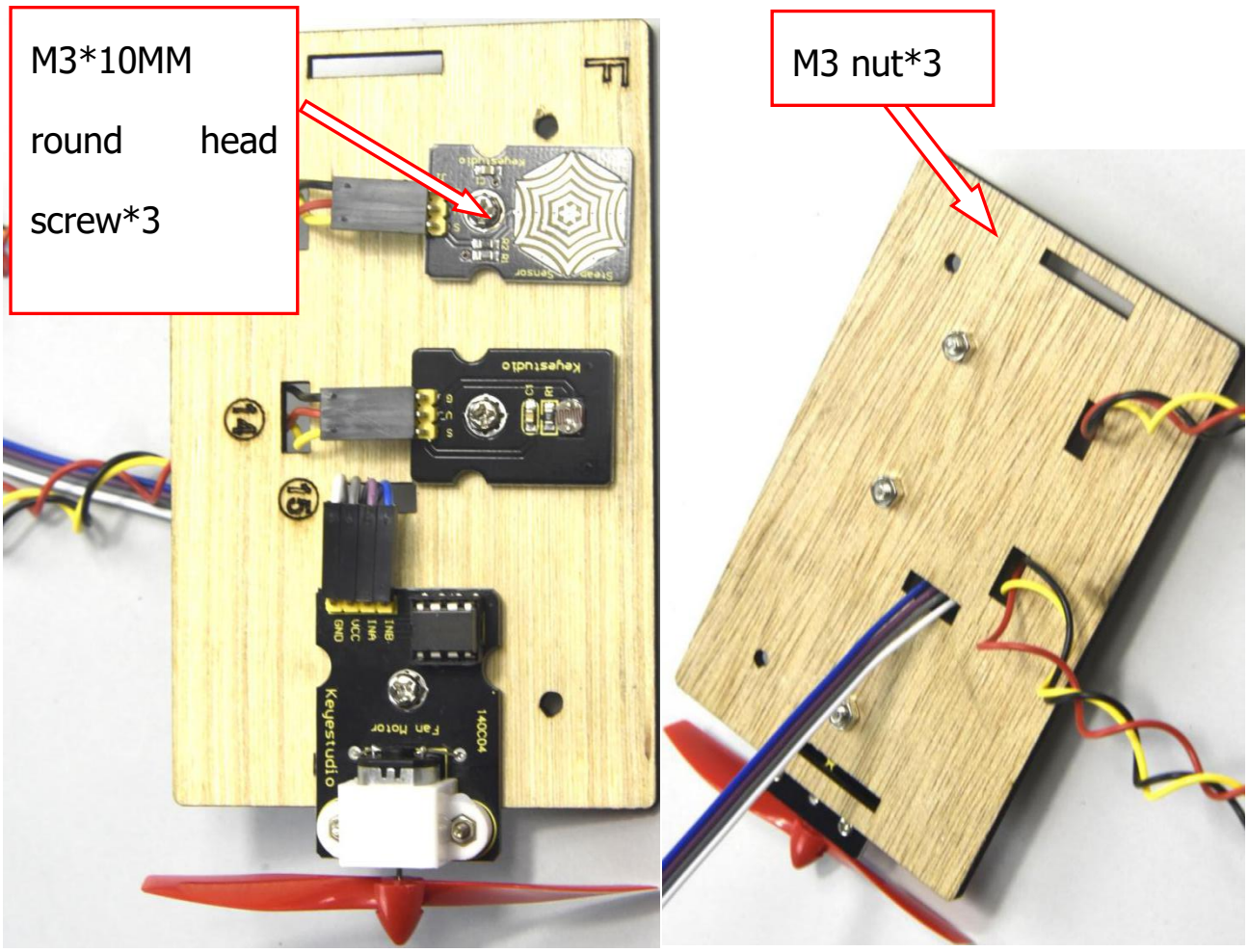


Step 10: Install the sensor of F board

- Steam sensor*1
- Photocell sensor*1
- Fan module*1
- F board*1
- 3pin female to female dupont line*2
- Female to female dupont line*4
- M3*10MM round head screw*3
- M3 nut*3

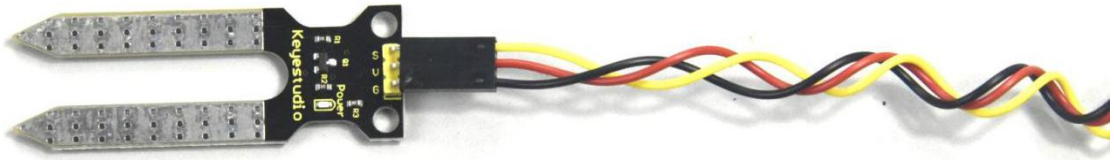


Separately fix steam sensor, photocell sensor and fan module on the F board with 3pcs M3*10MM round head screws and 3pcs M3 nuts, then connect them with 3pin and 4pin dupont lines.



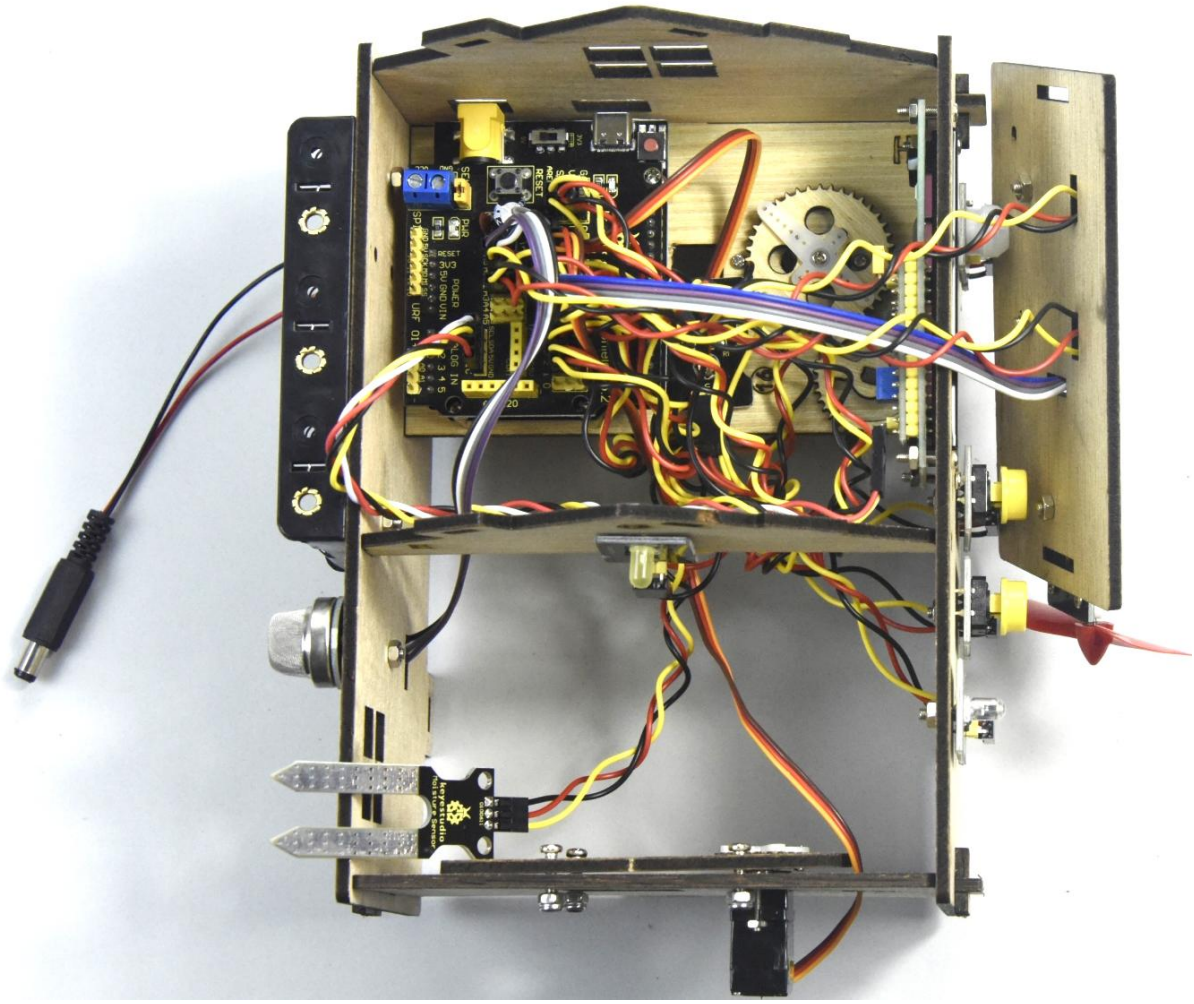
Step 11: Connect sensor/module

Connect one end of 3pin dupont line to the pin of soil humidity sensor, then link all sensors to sensor shield. (fix 2 servo and make dupont wire go through the holes of board)



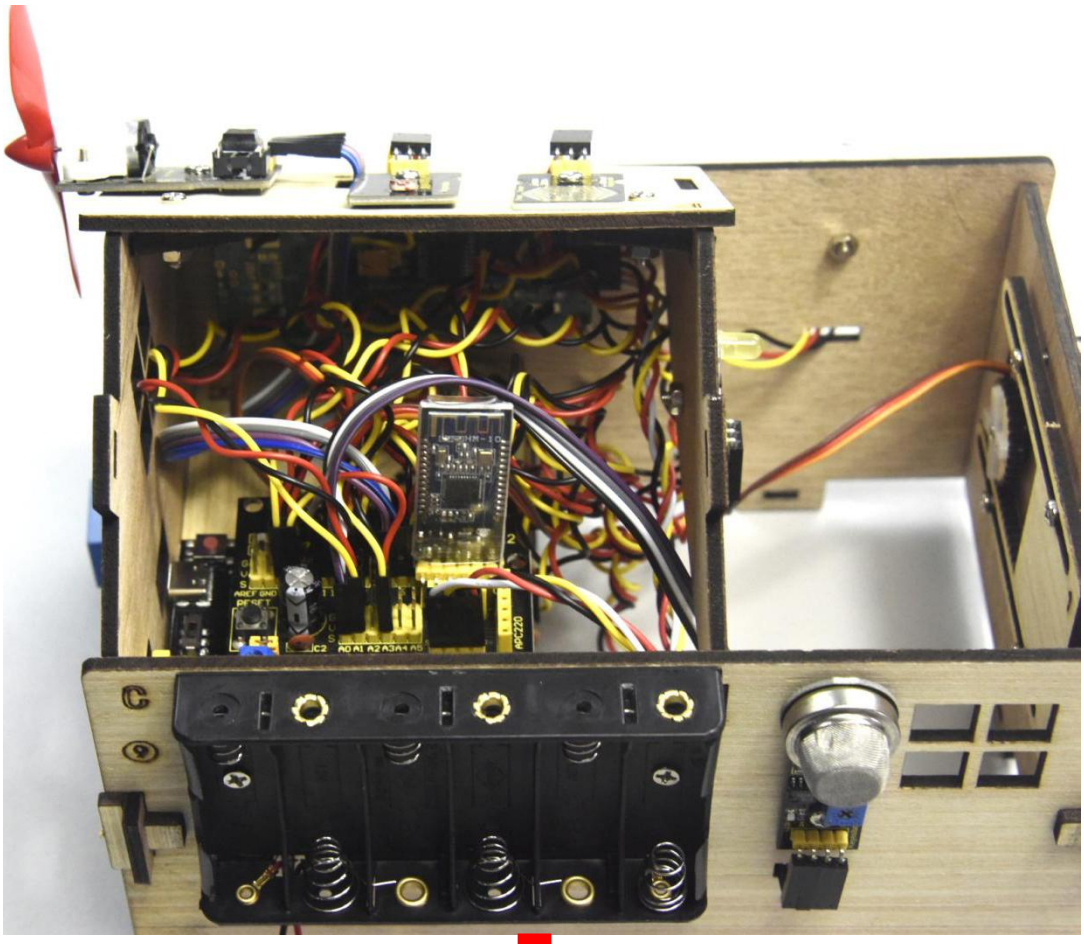
Name	The corresponding interfaces of sensors and sensor shield		The corresponding installed area on the board
PIR Motion Sensor	G/V/S	G/V/2	⑤
Passive buzzer	G/V/S	G/V/3	⑯
Button module 1	G/V/S	G/V/4	③
Yellow LED	G/V/S	G/V/5	⑫
Fan module	GND/VCC/INA/INB	G/V/7/6	⑮
Button module 2	G/V/S	G/V/8	④
Servo 1 controlling the door	Brown/Red/Orange wire	G/V/9	⑰
Servo 2 controlling the windows	Brown/Red/Orange wire	G/V/10	⑪
MQ-2 Gas Sensor	GND/VCC/D0/A0	G/V/11/A0	⑩
Relay Module	G/V/S	G/V/12	⑥
White LED	G/V/S	G/V/13	①
LCD1602 Display	GND/VCC/SDA/SCL	GND/5V/SDA/SCL	②

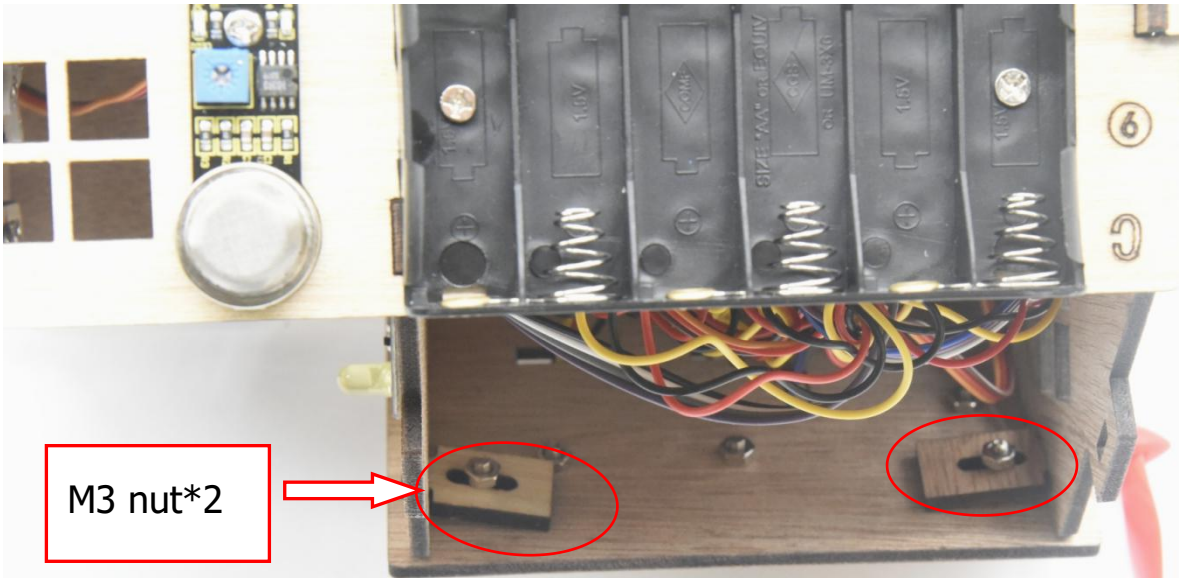
Photocell Sensor	G/V/S	G/V/A1	⑭
Soil humidity sensor	G/V/S	G/V/A2	
Steam sensor	G/V/S	G/V/A3	⑬

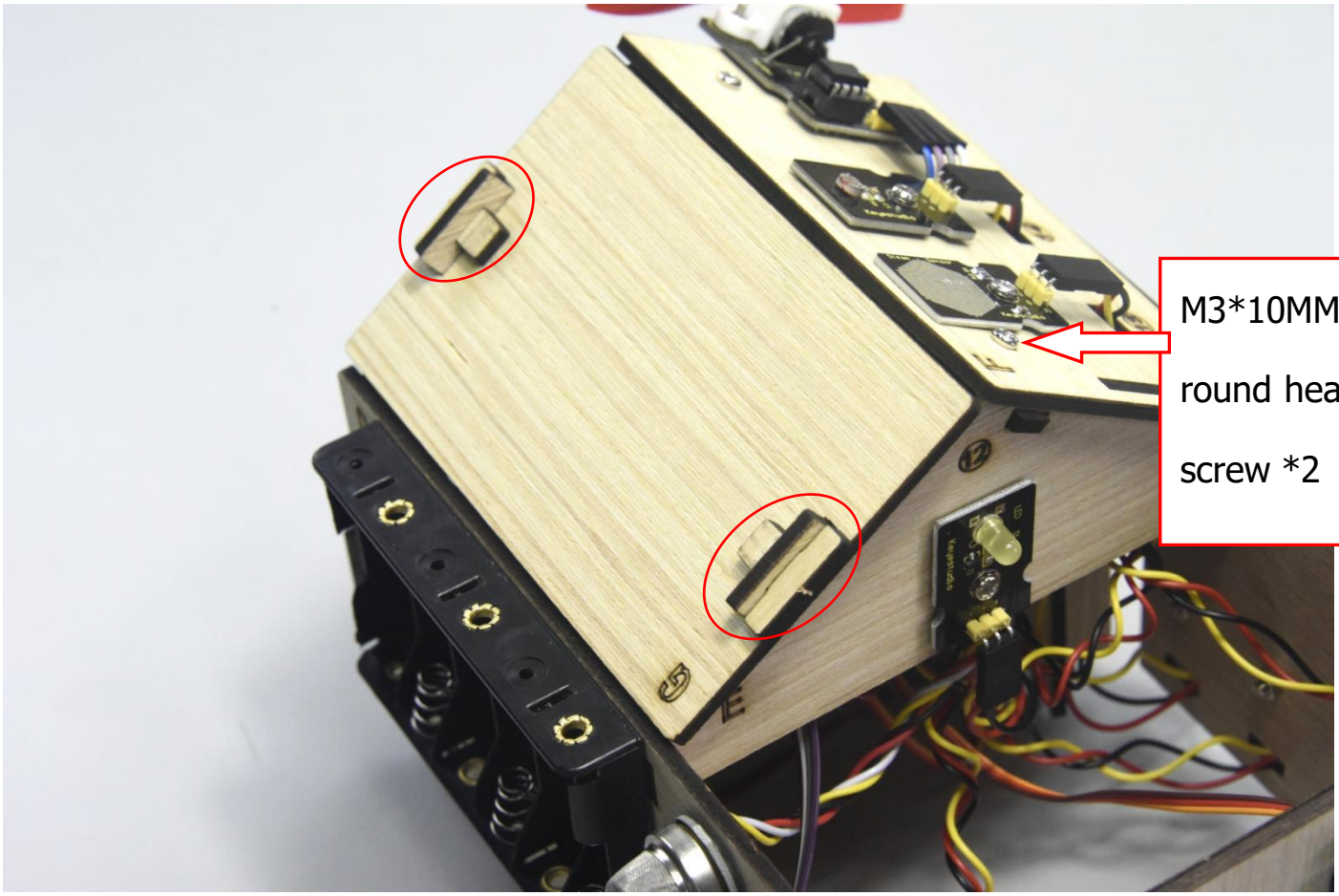


Insert the Bluetooth module into sensor shield, then fix the F board with 2 M3*10MM round head screws, 2 M3 nuts and 2 pcs parts with holes in the middle, mount G board well with 2 “T” type bolts.

Bluetooth Module	Sensor Expansion Board
VCC	5V
GND	GND
TXD	RXD
RXD	TXD

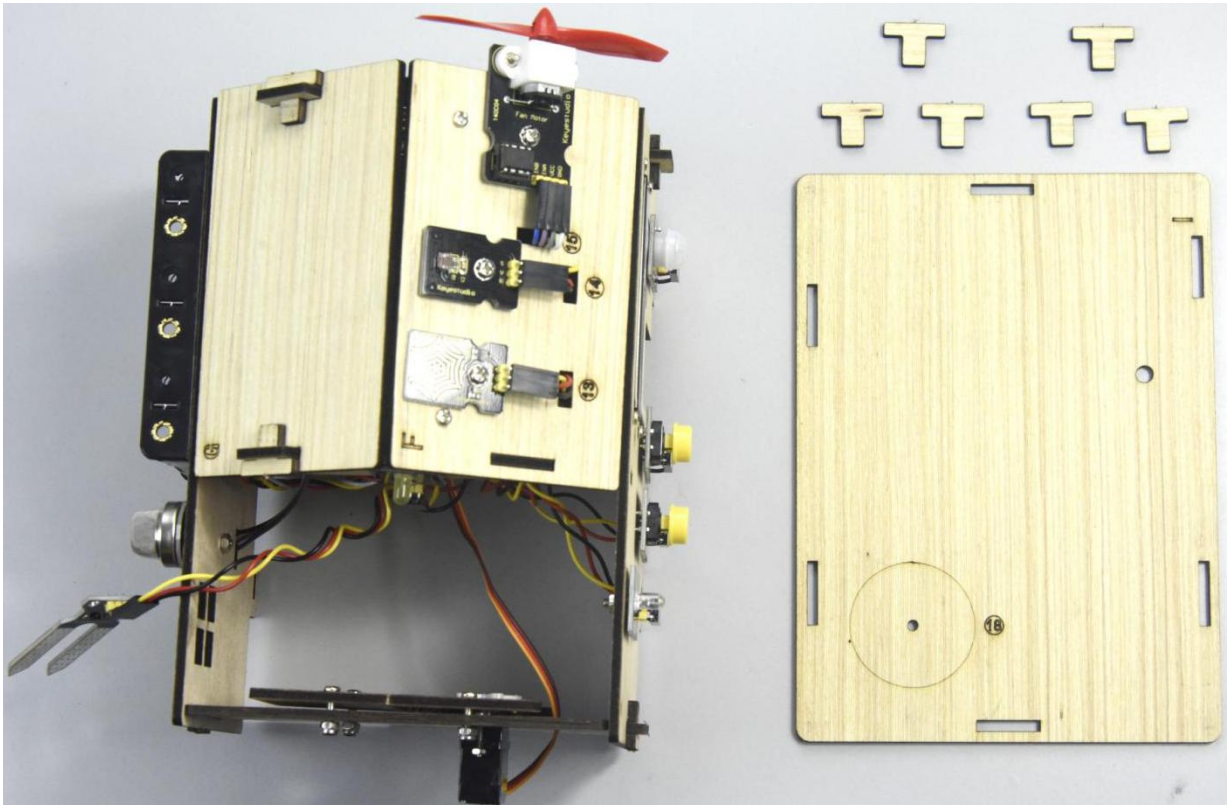


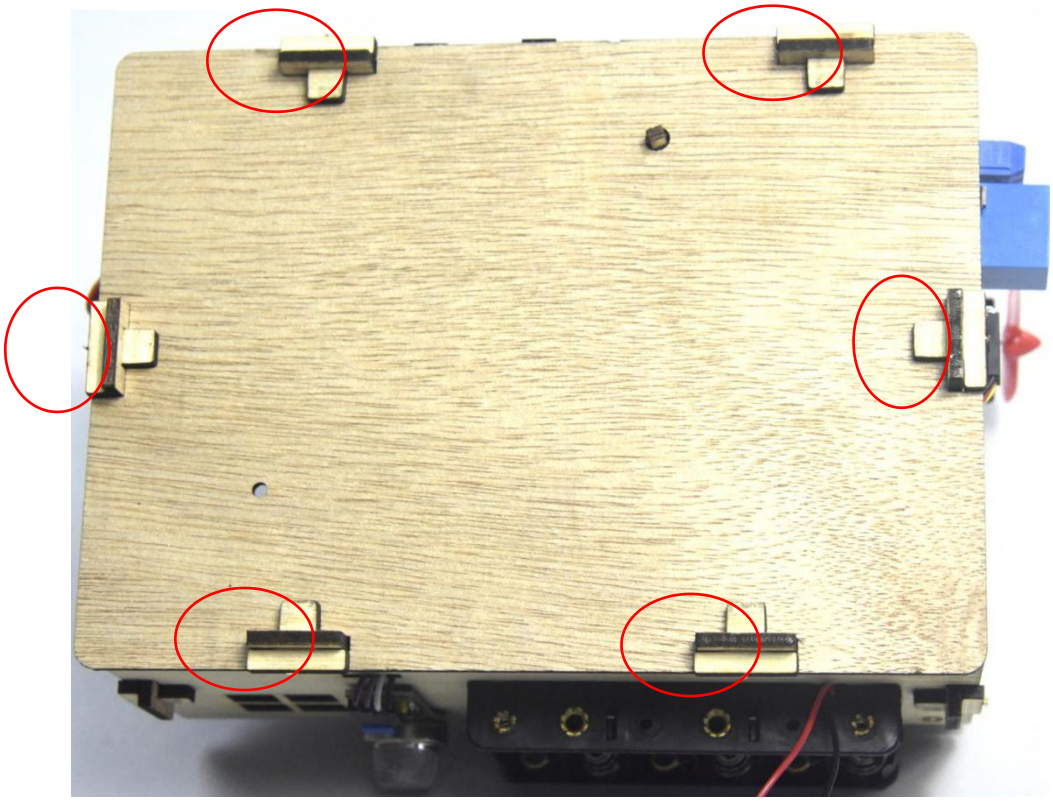

M3 nut*2

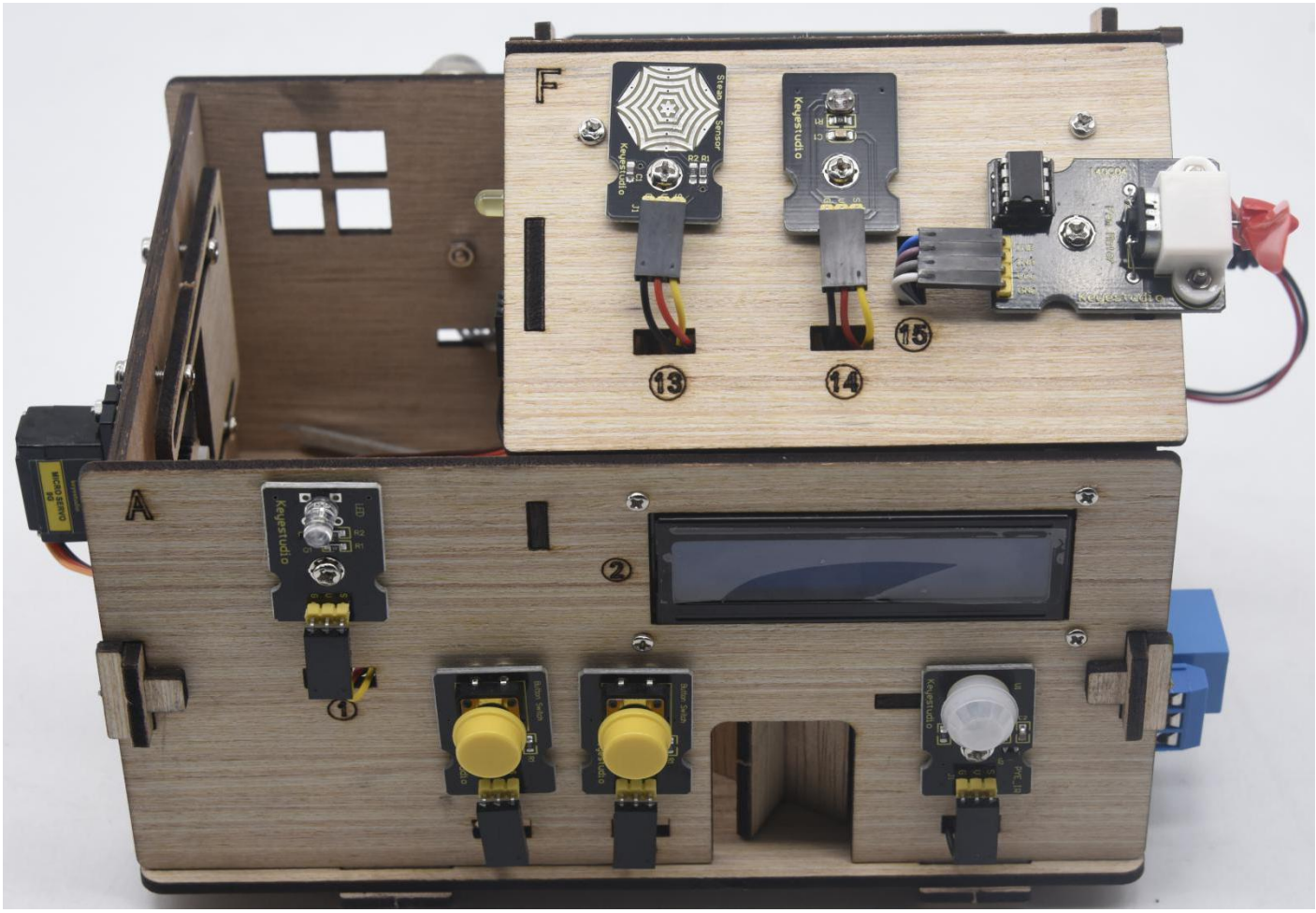

M3*10MM
round head
screw *2

Step 12: Assemble the kit

Fix the board I with 6 “T” bolts







Project 15：Multi-purpose Smart Home

Description

In the previous projects, we introduce all buttons on app. In this lesson, we will control the smart home by Bluetooth.

We will achieve the effect as follow:

（1）When at night, someone is passing, the LED lights on, otherwise, the LED lights off.

（2）There are 1602LCD display, 2 buttons, 1 servo on the board. Press button1 to enter the password(you can set password in the test code), the 1602LCD will show "*", then press button2 to "ensure". Then 1602LCD will show "open", the door will be open. If the password is wrong, the "error" pops up , after 2s, "error" will change into "again" , you can enter password again.

The door will be closed when PIR motion sensor doesn't detect people around. What's more, press and hold button2, buzzer will sound, LCD displays "wait".

（If the password is right, the servo will rotate to 180°, otherwise，the servo don't rotate)

（3）Insert soil humidity into plant pot, when the soil is too dry, buzzer will alarm and you will get the notification on app.

（4）When the gas sensor detects the gas with high concentration, the buzzer emits a "tick,tick" alarm sound.

（5）When steam sensor detects rains, the servo 2 will be activated, the window will be closed automatically, otherwise, the window will be open.

Equipment:

keyestudio PLUS Control Board * 1

Sensor expansion board * 1

Bluetooth module * 1

PIR motion sensor * 1

Photocell sensor * 1

Button module * 2

White LED module * 1

Yellow LED module * 1

Relay module * 1

Passive buzzer module * 1

Small fan module * 1

Steam sensor * 1

Servo module * 2

LCD1602 display module * 1

Soil humidity sensor * 1

MQ-2 gas sensor * 1

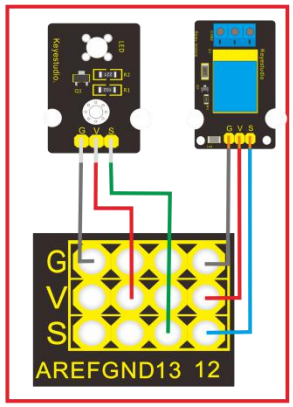
3pin female to female DuPont line * 10

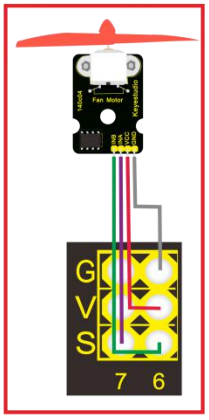
4pin female to female DuPont line * 1

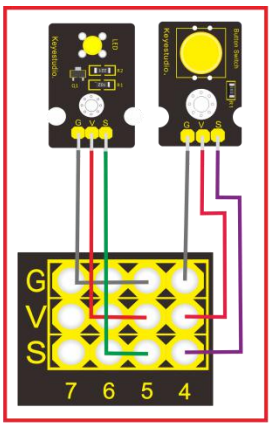
Several female to female dupont lines

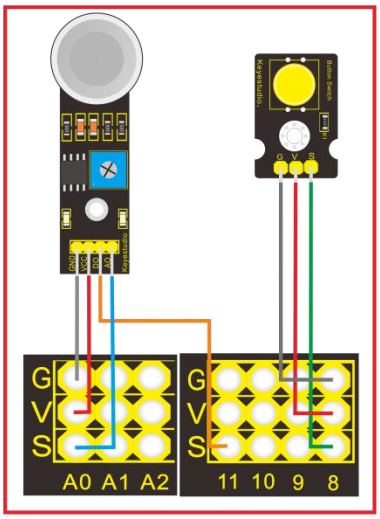
USB cable * 1

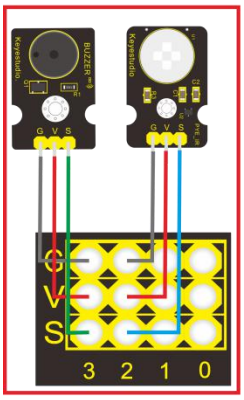
Connection Diagram:

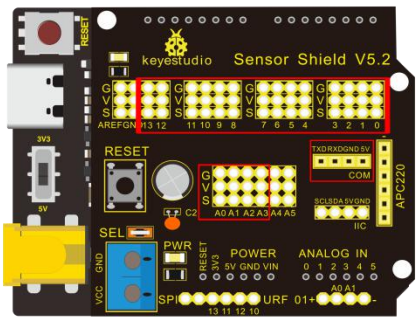


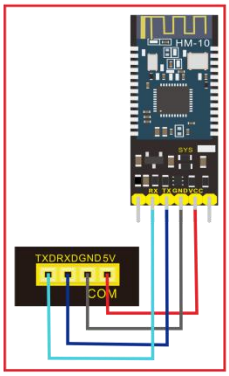


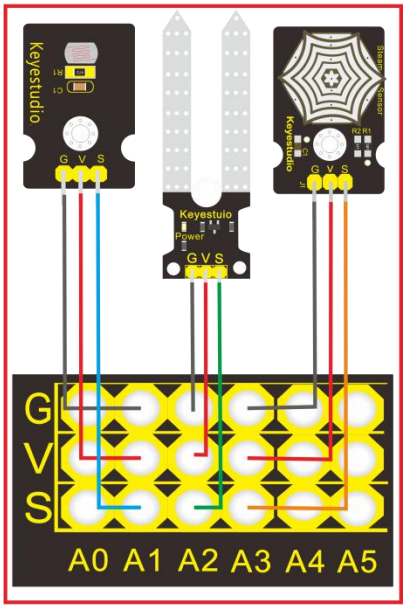


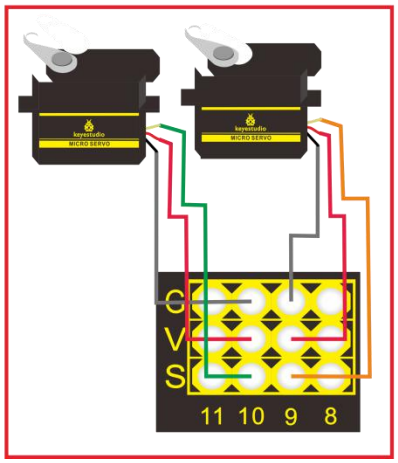


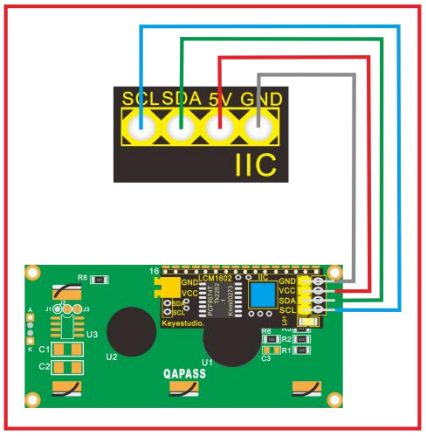












Name	The corresponding interfaces of sensors and sensor shield	The corresponding installed area on the board

PIR Motion Sensor	G/V/S	G/V/2	⑤
Passive Buzzer	G/V/S	G/V/3	⑯
Button sensor 1	G/V/S	G/V/4	③
Yellow LED Module	G/V/S	G/V/5	⑫
Fan Module	GND/VCC/INA/INB	G/V/7/6	⑮
Button Module 2	G/V/S	G/V/8	④
Servo 1 controlling the door	Brown/Red/Orange Wire	G/V/9	⑰
Servo 2 controlling the window	Brown/Red/Orange Wire	G/V/10	⑪

MQ-2 Gas Sensor	GND/VCC/D0/A0	G/V/11/A0	⑩
Relay Module	G/V/S	G/V/12	⑥
White LED	G/V/S	G/V/13	①
LCD1602 Display	GND/VCC/SDA/SCL	GND/5V/SDA/SCL	②
Photocell Sensor	G/V/S	G/V/A1	⑭
Soil Humidity Sensor	G/V/S	G/V/A2	
Steam Sensor	G/V/S	G/V/A3	⑬

Test Code:

****

****

```
#include <Servo.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
//use relevant library file
```

```
volatile int btn1_num;//set variable btn1_num
volatile int btn2_num;//set variable btn2_num
volatile int button1;//set variable button1
volatile int button2;//set variable button2
String fans_char;//string type variable fans_char
volatile int fans_val;//set variable fans_char
volatile int flag;//set variable flag
volatile int flag2;//set variable flag2
volatile int flag3;//set variable flag3
volatile int gas;//set variable gas
volatile int infrar;//set variable infrar
String led2;//string type variable led2
volatile int light;//set variable light
String pass;//string type variable pass
String passwd;//string type variable passwd
String servo1;//string type variable servo1
volatile int servo1_angle;//set variable light
String servo2;//string type variable servo2
volatile int servo2_angle;//set variable servo2_angle
volatile int soil;//set variable soil
volatile int val;//set variable val
volatile int value_led2;//set variable value_led2
```

```
volatile int water;//set variable water


void birthday()//set happy birthday song
{
   tone(3,294);//set digital3 to output the sound of 294HZ
   delay(250);//delay in 250ms
   tone(3,440);
   delay(250);
   tone(3,392);
   delay(250);
   tone(3,532);
   delay(250);
   tone(3,494);
   delay(500);
   tone(3,392);
   delay(250);
   tone(3,440);
   delay(250);
   tone(3,392);
   delay(250);
   tone(3,587);
   delay(250);
```

```
tone(3,532);
delay(500);
tone(3,392);
delay(250);
tone(3,784);
delay(250);
tone(3,659);
delay(250);
tone(3,532);
delay(250);
tone(3,494);
delay(250);
tone(3,440);
delay(250);
tone(3,698);
delay(375);
tone(3,659);
delay(250);
tone(3,532);
delay(250);
tone(3,587);
delay(250);
```

```
    tone(3,532);
    delay(500);
}

//define the name of every sound
#define D0 -1
#define D1 262
#define D2 293
#define D3 329
#define D4 349
#define D5 392
#define D6 440
#define D7 494
#define M1 523
#define M2 586
#define M3 658
#define M4 697
#define M5 783
#define M6 879
#define M7 987
#define H1 1045
#define H2 1171
```

```
#define H3 1316
#define H4 1393
#define H5 1563
#define H6 1755
#define H7 1971

#define WHOLE 1
#define HALF 0.5
#define QUARTER 0.25
#define EIGHTH 0.25
#define SIXTEENTH 0.625
//set the ports of two servos to digital9 and digital10
Servo servo_10;
Servo servo_9;
//set the frequency of sound
int tune[]=
{
  M3,M3,M4,M5,
  M5,M4,M3,M2,
  M1,M1,M2,M3,
  M3,M2,M2,
  M3,M3,M4,M5,
```

```
  M5,M4,M3,M2,
  M1,M1,M2,M3,
  M2,M1,M1,
  M2,M2,M3,M1,
  M2,M3,M4,M3,M1,
  M2,M3,M4,M3,M2,
  M1,M2,D5,D0,
  M3,M3,M4,M5,
  M5,M4,M3,M4,M2,
  M1,M1,M2,M3,
  M2,M1,M1
};
//set I2C communication address of LED to 0x27, display 16 characters each line,
2 lines in total
LiquidCrystal_I2C mylcd(0x27,16,2);
//set beats
float durt[]=
 {
  1,1,1,1,
  1,1,1,1,
  1,1,1,1,
  1+0.5,0.5,1+1,
```

```
  1,1,1,1,
  1,1,1,1,
  1,1,1,1,
  1+0.5,0.5,1+1,
  1,1,1,1,
  1,0.5,0.5,1,1,
  1,0.5,0.5,1,1,
  1,1,1,1,
  1,1,1,1,
  1,1,1,0.5,0.5,
  1,1,1,1,
  1+0.5,0.5,1+1,
};

int length;
int tonepin=3;//set the signal end of passive buzzer to digital3

void Ode_to_Joy()//play “Ode_to_Joy”
{
  for(int x=0;x<length;x++)
  {
    tone(tonepin,tune[x]);
```

```
      delay(300*durt[x]);
    }
}
// gas detection
void auto_sensor() {
   gas = analogRead(A0);//assign the analog value of A0 to gas
   if (gas > 1000) {
   //if variable “gas” > 1000
      flag = 1;//set flag to 1
    while (flag == 1)
    //if flag is 1, the program is in loop
      {
         Serial.println("danger");//wrap word and output "danger"
         tone(3,440);
         delay(125);
         delay(100);
         noTone(3);
         delay(100);
         tone(3,440);
         delay(125);
         delay(100);
         noTone(3);
```

```
        delay(300);
        gas = analogRead(A0);//assign the analog value of A0 to gas
        if (gas < 10)   //if gas < 10
         {
           flag = 0;//set flag to 0
           break;//Exit from the loop
        }
      }

   } else
//otherwise
 {
      noTone(3);//digital3 stops sound
   }
   light = analogRead(A1);////assign the value of A1 to light
   if (light < 300)//if light is < 300
 {
      infrar = digitalRead(2);//assign the value of digital2 to infrar
      Serial.println(infrar);//wrap word and output the value of infrar
    if (infrar == 1)
    //if infra is 1
      {
```

```
        digitalWrite(13,HIGH);//set 13 to high level，LED lights off
    } else//otherwise
     {
        digitalWrite(13,LOW);//set digital13 to low level, LED is off
     }

  }
  water = analogRead(A3);//assign the analog value of A3 to variable water
  if (water > 800)
  //if variable water is greater than 800
{
     flag2 = 1;//set variable flag2 to 1
    while (flag2 == 1)
    //set flag2 to 1，the program is in loop
     {
        Serial.println("rain");//wrap word and output "rain"
        servo_10.write(180);//set servo motor connected to digital10 to 180°
        delay(300);//delay in 300ms
        delay(100);
        water = analogRead(A3);;//assign the analog value of A3 to water
        if (water < 30)//if water is less than 30
         {
```

```
            flag2 = 0;//set flag2 to 0
            break;// Exit from the loop
          }
        }

    } else//otherwise
  {
     if (val != 'u' && val != 'n')
     //if val isn't character 'u' or character 'n'
     {
         servo_10.write(0);//set servo motor connected to digital10 to 0°
         delay(10);

        }

    }
    soil = analogRead(A2);//assign the analog value of A2 to soil
    if (soil > 50)
//if the variable soil is greater than 50
{
       flag3 = 1;//set flag3 to 1
     while (flag3 == 1)
```

```
  //if flag3 is 1,  the program is in loop
   {
      Serial.println("hydropenia ");//wrap word and output "hydropenia "
      tone(3,440);
      delay(125);
      delay(100);
      noTone(3);
      delay(100);
      tone(3,440);
      delay(125);
      delay(100);
      noTone(3);//set digital3 to stop sounding
      delay(300);
      soil = analogRead(A2);//assign the analog value of A2 to variable “soil”
      if (soil < 10)//if soil is less than 10
     {
        flag3 = 0;//set flag3 to 0
        break;//exit the loop
      }
   }

} else//otherwise
```

```
 {
    noTone(3);//set digital3 to stop playing music
  }
  door();//Run subroutine
}

void door() {
  button1 = digitalRead(4);//assign the value of 4 to button1
  button2 = digitalRead(8);//assign the value of 8 to button2

  if (button1 == 0)//if button1 is 0
 {
    delay(10);//delay in 10ms
   while (button1 == 0) //if the variable button1 is 0, the program is in loop
   {
      button1 = digitalRead(4);//assign the value of 4 to button1
      btn1_num = btn1_num + 1;//btn1_num plus 1
      delay(100);//delay in 100ms
    }

  }
  if (btn1_num >= 1 && btn1_num < 5) //if variable btn1_num is greater than or
```

```
      equal to 1 and less than 5
{
      Serial.print(".");
      Serial.print("");
      passwd = String(passwd) + String(".");//set to passwd character
     pass = String(pass) + String("*");//set to pass character
     //LCD displays  “pass”  from the first character on the first line
      mylcd.setCursor(1-1, 2-1);
      mylcd.print(pass);
   }
   if (btn1_num >= 5)
//if variable btn1_num is greater than or equal to 5
 {
      Serial.print("-");
      passwd = String(passwd) + String("-");//set to passwd character
     pass = String(pass) + String("*");//set to pass character
     //LCD displays  “pass” from the first character on the first line
      mylcd.setCursor(1-1, 2-1);
      mylcd.print(pass);

   }
   if (button2 == 0) //if variable button2 is 0
```

```
{
      delay(10);
    if (button2 == 0)//if variable button2 is 0
     {
        if (passwd == ".--.-.")//if passwd is ".--.-."
 {
            mylcd.clear();// clears up the displayed information on LCD
    //LCD displays the "open!" character from the first character in the second line
            mylcd.setCursor(1-1, 2-1);
            mylcd.print("open!");
            servo_9.write(100);//set the servo connected to digital9 to 100°
            delay(300);
            delay(5000);
            passwd = "";
            pass = "";
            mylcd.clear();//clear up the displayed information on the LCD
            //LCD displays the "password:" character from first character on the first
line
            mylcd.setCursor(1-1, 1-1);
            mylcd.print("password:");

        } else //otherwise
```

```
    {
        mylcd.clear();//clear up the displayed information on the LCD
        //LCD displays the "error!" from the first character in the first line
        mylcd.setCursor(1-1, 1-1);
        mylcd.print("error!");
        passwd = "";
        pass = "";
        delay(2000);
        //LCD displays the "again" from the first character in the first line
        mylcd.setCursor(1-1, 1-1);
        mylcd.print("again");
      }
    }
  }
  infrar = digitalRead(2);//assign the value of digital2 to infrar
  if (infrar == 0 && (val != 'l' && val != 't'))
//if variable infrar is 0 and val is not 'l' or 't' character
 {
    servo_9.write(0);//set the servo connected to digital9 to 0°
    delay(50);
  }
  if (button2 == 0)//if variable button2 is 0
```

```
{
    delay(10);
  while (button2 == 0) //if variable button2 is 0,  the program is in loop
  {
      button2 = digitalRead(8);//assign the value of digital8 to variable button2
      btn2_num = btn2_num + 1;//variable btn2_num plus 1
      delay(100);
      if (btn2_num >= 15)//if variable btn2_num is greater than or equal to 15
       {
         tone(3,532);
         delay(125);
         mylcd.clear();//clear up the displayed information on LCD
         //LCD displays the "password:" from the first character in the first line
         mylcd.setCursor(1-1, 1-1);
         mylcd.print("password:");
         //LCD displays the "wait" from the first character in the first line
         mylcd.setCursor(1-1, 1-1);
         mylcd.print("wait");
       } else//otherwise
     {
         noTone(3);//set digital3 to stop playing music
       }
```

```
      }


   }
   btn1_num = 0;//set variable btn1_num to 0
   btn2_num = 0;//set variable btn2_num to 0
}
//Happy Birthday Song
void music1() {
   birthday();
}
//Ode_to_Joy Song
void music2() {
   Ode_to_Joy();
}
//PWM control
void pwm_control() {
   switch (val)
 {
    case 't'://when val is character 't',  the program is in loop
      servo1 = Serial.readStringUntil('#');
      servo1_angle = String(servo1).toInt();
      servo_9.write(servo1_angle);//set the angle of servo connected to digital9 to
```

```
servo1_angle
      delay(300);
      break;//exit the loop
    case 'u'://when val is character 'u', the program is in loop
      servo2 = Serial.readStringUntil('#');
      servo2_angle = String(servo2).toInt();
      servo_10.write(servo2_angle);//set the angle of servo connected to digital10
to servo2_angle
      delay(300);
      break;//exit the loop
    case 'v'://when val is character 'v', the program is in loop
      led2 = Serial.readStringUntil('#');
      value_led2 = String(led2).toInt();
      analogWrite(5,value_led2);//PWM value of digital5 is value_led2
      break;//exit the loop
    case 'w'://when val is character 'w', the program is in loop
      fans_char = Serial.readStringUntil('#');
      fans_val = String(fans_char).toInt();
      digitalWrite(7,LOW);
      analogWrite(6,fans_val);//set PWM value of digital6 to fans_val, the greater
      the value is, the faster the fan rotates.
      break;//exit the loop
```

```
    }
}

void setup(){
    Serial.begin(9600);//set baud rate to 9600
    mylcd.init();
    mylcd.backlight();// initially set LCD
    servo_9.attach(9);//connect servo to digital9
    servo_10.attach(10);//make servo connect to digital10
    pinMode(7, OUTPUT);//set digital7 to ounput
    pinMode(6, OUTPUT);//set 6 to ounput
    servo_9.write(0);//set the servo connected to digital9 to 0°
    delay(300);
    servo_10.write(0);//set the servo connected to digital10 to 0°
    delay(300);
//LCD displays the "password:" from the first character in the first line
    mylcd.setCursor(1-1, 1-1);
    mylcd.print("passcord:");
    digitalWrite(7,HIGH);//set digital7 to high level
    digitalWrite(6,HIGH);//set digital6 to high level
    pinMode(4, INPUT);//set digital4 to input
    pinMode(8, INPUT);//set digital8 to input
```

```
pinMode(2, INPUT);//set digital2 to input
pinMode(3, OUTPUT);//set digital3 to output
pinMode(A0, INPUT);//set A0 to input
pinMode(A1, INPUT);//set A1 to input
pinMode(13, OUTPUT);//set digital13 to output
pinMode(A3, INPUT);//set A3 to input
pinMode(A2, INPUT);//set A2 to input
val = 0;//set variable val to 0
button1 = 0;//set variable button1 to 0
button2 = 0;//set variablebutton2 to 0
btn1_num = 0;//set variable btn1_num to 0
btn2_num = 0;//set variable btn2_num to 0
passwd = "";
pass = "";
gas = 0;//set variable gas to 0
flag = 0;//set variable flag to 0
light = 0;//set variable light to 0
infrar = 0;//set variable infrar to 0
water = 0;//set variable water to 0
flag2 = 0;//set flag2 to 0
soil = 0;//set soil to 0
flag3 = 0;//set flag3 to 0
```

```
  servo1 = "";
  servo1_angle = 0;//set variable servo1_angle to 0
  servo2 = "";
  servo2_angle = 0;//set variable servo2_angle to 0
  led2 = "";
  value_led2 = 0;//set variable value_led2 to 0
  fans_char = "";
  fans_val = 0;//set variable fans_val to 0
  pinMode(12, OUTPUT);//set digital12 to output
  pinMode(5, OUTPUT);//set digital5 to output
  pinMode(3, OUTPUT);//set digital3 to output
  length=sizeof(tune)/sizeof(tune[0]);//set the value of length
}

void loop(){
  auto_sensor();
  if (Serial.available() > 0) //read the character on serial monitor
{
    val = Serial.read();//set val to the character read on serial port
Serial.println(val);//wrap word and output val
    pwm_control();
```

```
  }
  switch (val) {
    case 'a'://if set val to 'a',  the program is in loop
      digitalWrite(13,HIGH);//digital13 is high level,  LED lights on
           break;//exit the loop
    case 'b'://if val is 'b',  the program is in loop
      digitalWrite(13,LOW);//digital13 is low level,  LED is off
      break;//exit the loop
    case 'c'://if set val to 'c',  the program is in loop
      digitalWrite(12,HIGH);//set digital12 to high level, NO and COM ends of relay
module are connected
      break;//exit the loop
    case 'd'://if val is 'd' character,  the program is in loop
      digitalWrite(12,LOW);//set digital12 to low level,  NO and COM ends of relay
module are disconnected
      break;//exit the loop
    case 'e'://if set val to 'e',  the program is in loop
      music1();//play happy birthday song
      break;//exit the loop
    case 'f'://if set val to 'f',  the program is in loop
      music2();//play Ode_to_Joy song
      break;//exit the loop
```

```
    case 'g'://if val is 'g',  the program is in loop
      noTone(3);//set digital3 to stop playing music
      break;//exit the loop
    case 'h'://if val is 'h',  the program is in loop
      Serial.println(light);// wrap word and output the value of light
      delay(100);
      break;//exit the loop
    case 'i'://if val is 'i',  the program is in loop
      Serial.println(gas);//wrap word and output the value of gas
      delay(100);
      break;//exit the loop
    case 'j'://if val is ‘j’,  the program is in loop
      Serial.println(soil);//wrap word and output the value of soil
      delay(100);
      break;//exit the loop
    case 'k'://if val is ' k' character,  the program is in loop
      Serial.println(water);//wrap word and output the value of water
delay(100);
      break;//exit the loop
    case 'l'://if val is 'l' character,  the program is in loop
      servo_9.write(180);//set the servo connected to digital9 to 180°
      delay(500);
```

```
  break;//exit the loop
case 'm'://if val is 'm',  the program is in loop
  servo_9.write(0);;//set the servo connected to digital9 to 0°
  delay(500);
  break;//exit the loop
case 'n'://if set val to 'n',  the program is in loop
  servo_10.write(180);//set the servo connected to digital10 to 180°
  delay(500);
  break;//exit the loop
case 'o'://if val is 'o',  the program is in loop
  servo_10.write(0);/set the servo connected to digital10 to 0°
  delay(500);
  break;//exit the loop
case 'p'://if val is 'p' character,  the program is in loop
  digitalWrite(5,HIGH);//set digital5 to high level, LED lights on
        break;//exit the loop
case 'q'://if set val to 'q',  the program is in loop
  digitalWrite(5,LOW);//set digital5 to low level, LED is off
        break;//exit the loop
case 'r'://if set val to 'r',  the program is in loop
  digitalWrite(7,LOW);
  digitalWrite(6,HIGH);//the fan rotate anticlockwise at the fastest speed
```

```
      break;//exit the loop
    case 's'://if val is character 's' , the program is in loop
digitalWrite(7,LOW);
      digitalWrite(6,LOW);//the fan stops rotating
      break;//exit the loop
   }}
```

****

****

Upload the code and see the result!

Note: Remove the Bluetooth module please, when uploading the test code. Otherwise, the program will fail to upload. Connect the Bluetooth and Bluetooth module to pair after uploading the test code.

Test Result

Upload the test code, stack expansion board on PLUS control board, and power on. After pairing and connecting Bluetooth successfully, we can control the smart home through app.

7.Resources