kidsbits WiKi

kidsbits WiKi

Apr 25, 2024

KD3021 KIDSBITS SMART FARM KIT COMPATIBLE WITH LEGO


STEM Education Smart Farm Kit
for Arduino Compatible with Lego
kidsbits
STEM Education Smart Farm Kit
for Arduino Compatible with Lego
10
Projects
KD3021

CHAPTER

ONE

1. DOWNLOADING THE CODE:

- KidsBlock_Codes

CHAPTER

TWO

2. INTRODUCTION:

Based on the ESP32 IoT, the smart farm integrates multiple sensors such as a photoresistor, a soil moisture sensor, a water level sensor as well as a WiFi wireless communication module in a way that achieves automation, wireless operation and intelligent management. What's more, it is capable of realizing intelligent functions such as sensing, early warning, decision-making and analysis.

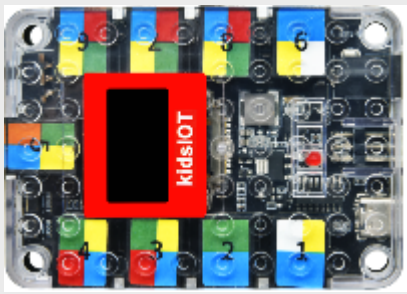
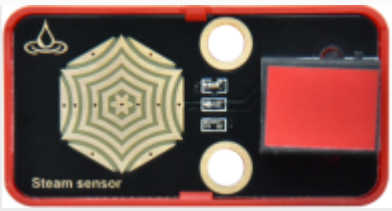
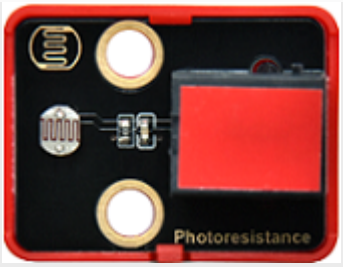
Therefore, the product contributes to helping you master how to use sensors to build an IoT system, and how to realize smart farm management via KidsBlock graphical programming.

By the way, detailed projects, sample code and LEGO building are provided in it, thus enhancing our hands-on ability, creativity, scientific and technological innovation awareness as well as problem-solving ability.

CHAPTER

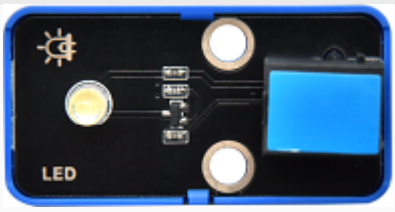
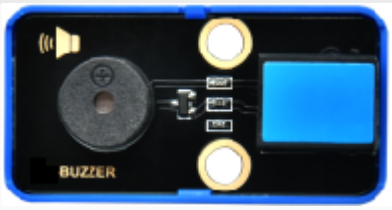
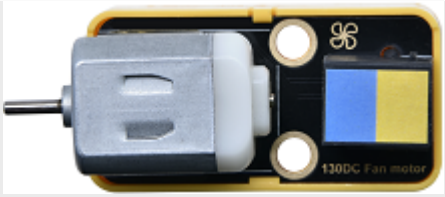
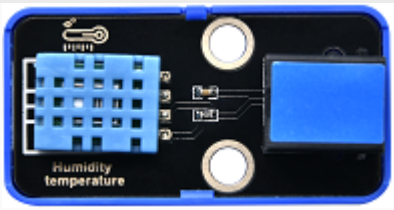
THREE

3. KIT LIST

#	Components	QTY	Picture
1	kidsIOT Mainboard	1	
2	Button Sensor	1	
3	Steam Sensor	1	
4	PIR Motion Sensor	1	
5	Photoresistor	1	

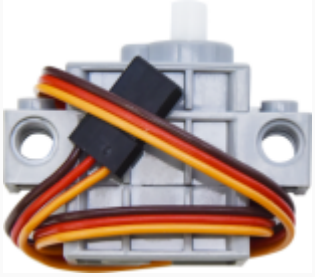
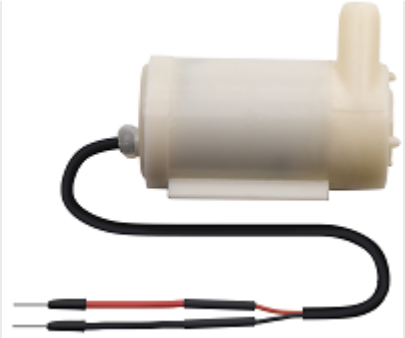
continues on ne

Table 1 – continued from previous page

#	Components	QTY	Picture
6	Ultrasonic Adapter Board	1	
7	Ultrasonic Sensor	1	
8	White LED Module	1	
9	Relay Module	1	
10	Passive Buzzer	1	
11	Motor	1	
12	Temperature and Humidity Sensor	1	

continues on ne

Table 1 – continued from previous page

#	Components	QTY	Picture
13	GPIO Shield	1	
14	Servo	1	
15	Water Level Sensor	1	
16	Soil Moisture Sensor	1	
17	Water Pump	1	
18	Water Pipe	1	
19	USB Cable	1	

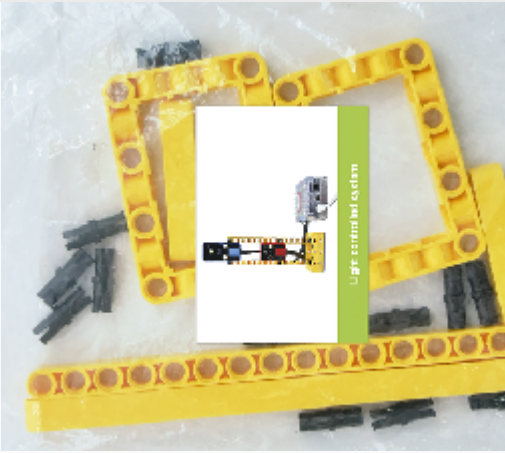
continues on ne

Table 1 – continued from previous page

#	Components	QTY	Picture
20	Battery Holder	1	
21	20cm Wire	3	
22	30cm Wire	8	
23	F-F DuPont Wire	1	
24	M-F DuPont Wire	1	
25	Slotted Screwdriver	1	
26	Sink	2	
27	Fan	1	
28	AA BatteryNot provide	6	

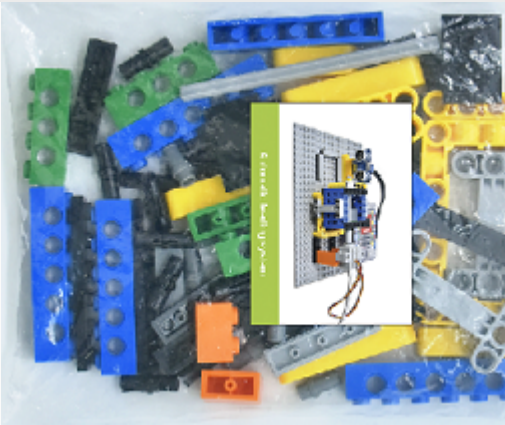
continues on ne

Table 1 – continued from previous page

#	Components	QTY	Picture
29	Lighting System LEGO Pieces	1	
30	Light Controlled System LEGO Pieces	1	
31	Anti-theft Alarm System LEGO Pieces	1	

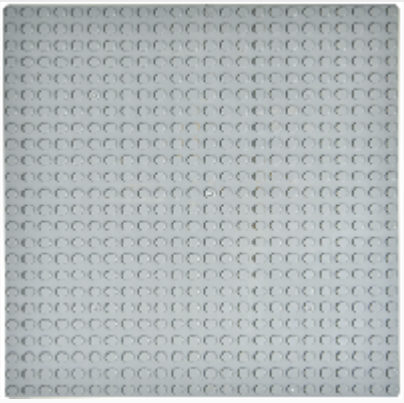
continues on ne

Table 1 – continued from previous page

#	Components	QTY	Picture
32	Automatic Feeding System LEGO Pieces	1	
33	Rainwater Control System LEGO Pieces	1	
34	Temperature and Humidity System LEGO Pieces	1	

continues on ne

Table 1 – continued from previous page

#	Components	QTY	Picture
35	Soil Moisture/Water Level/Automatic Irrigation System LEGO Pieces	1	
36	Lego Board	2	

CHAPTER

FOUR

KIDSBLOCK TUTORIAL

4.1 1. Mainboard_Introduction:

Refer to the linkhttp://kd2076-kidsbits-stem-electronic-building-block-programming.readthedocs.io/

4.2 2. KidsBlock Development Environment Configuration:

Please refer to the link to install and use the KidsBlock software https://kidsblocksite.readthedocs.io/en/latest/

Note: The control board used in this kit is the kidsIOT board. For importing the kidsIOT board, libraries and sample codes, please refer to the following content.

1. Click 

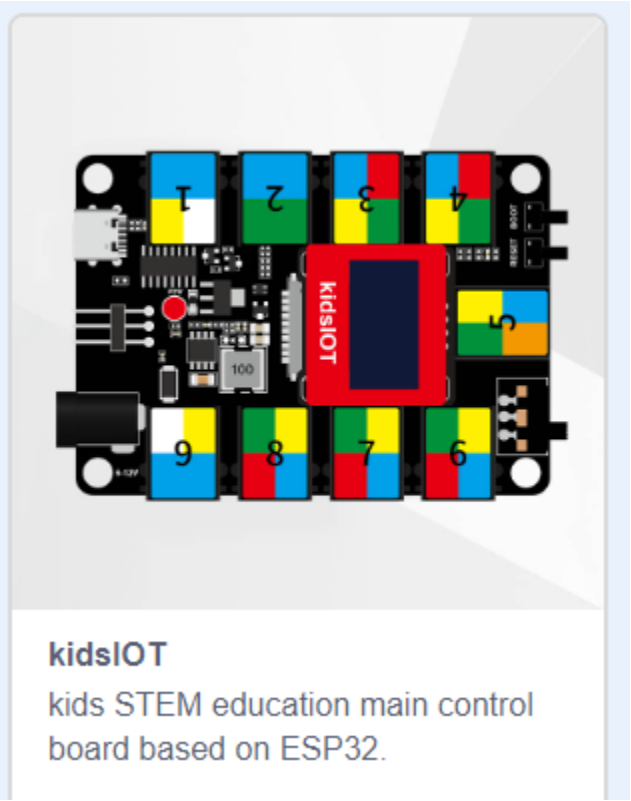
to enter the main page, and select the control board needed. In this project, we select the kidsIOT mainboard and click **Connect**, then it is connected. Click Go to Editor to return the code editor. Icon 

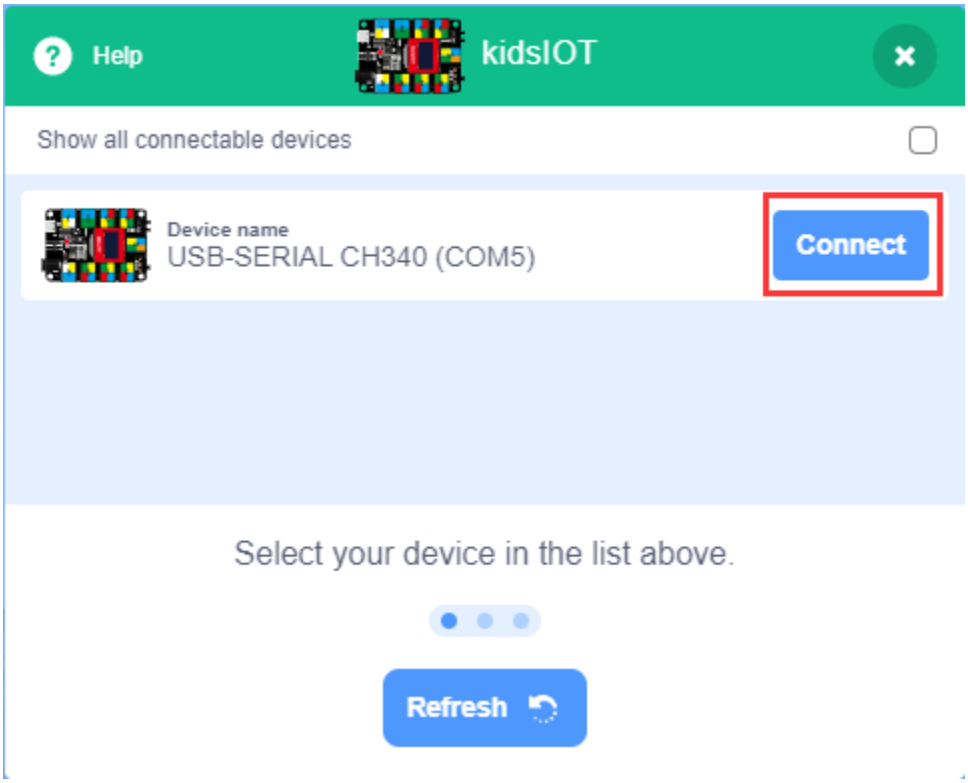
will change into 

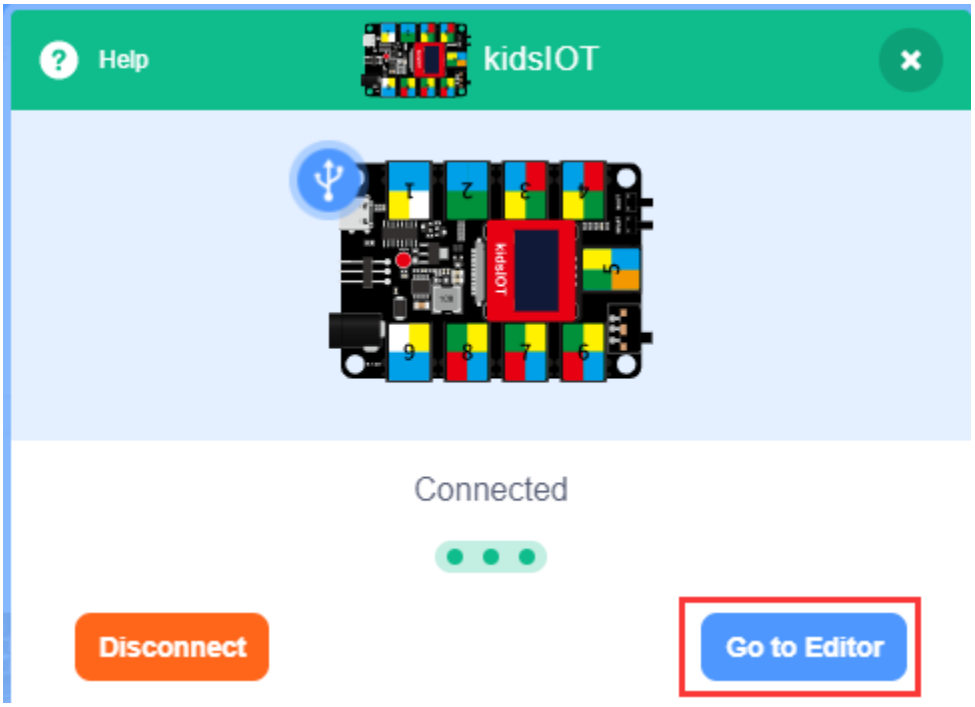
and 

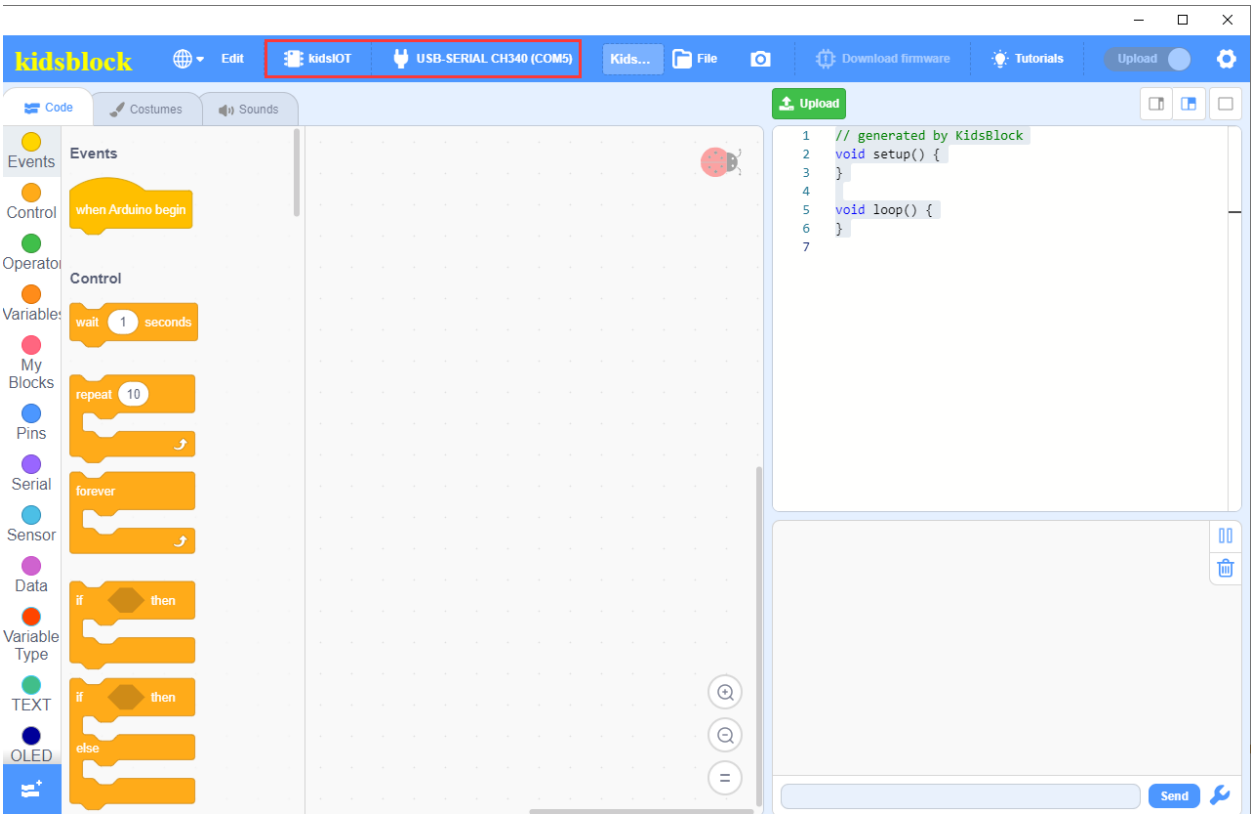
will change into 

. This means the kidsIOT mainboard and portsCOM)are connected.


kidsIOT
kids STEM education main control board based on ESP32.


Help
kidsIOT
Show all connectable devices
Device name
USB-SERIAL CH340 (COM5)
Connect
Select your device in the list above.
Refresh



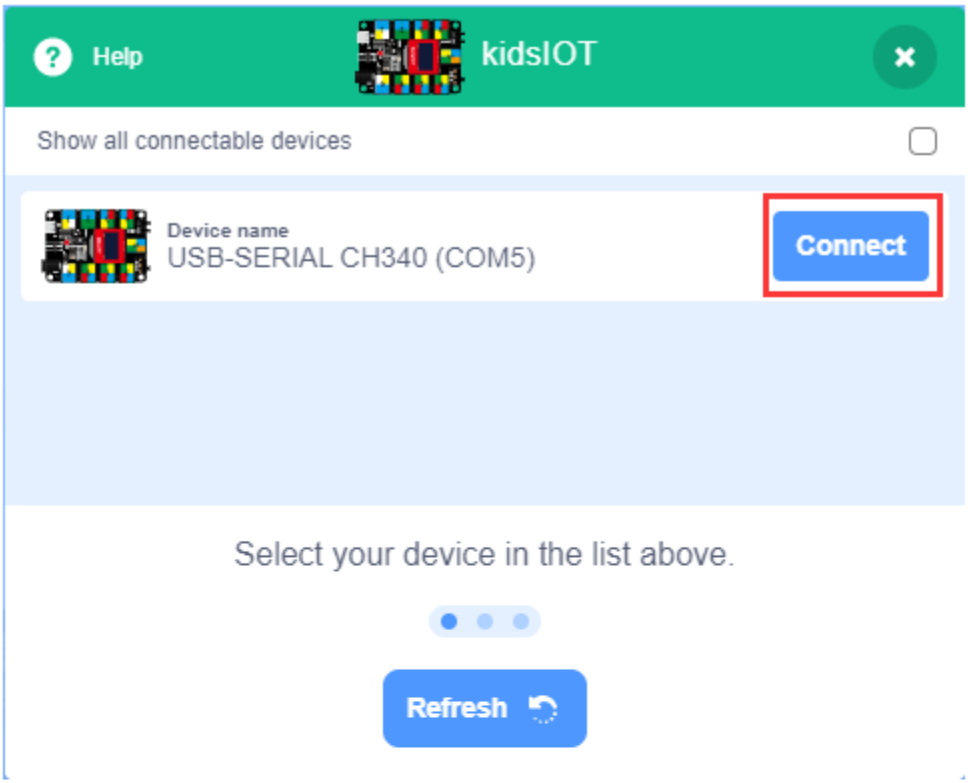


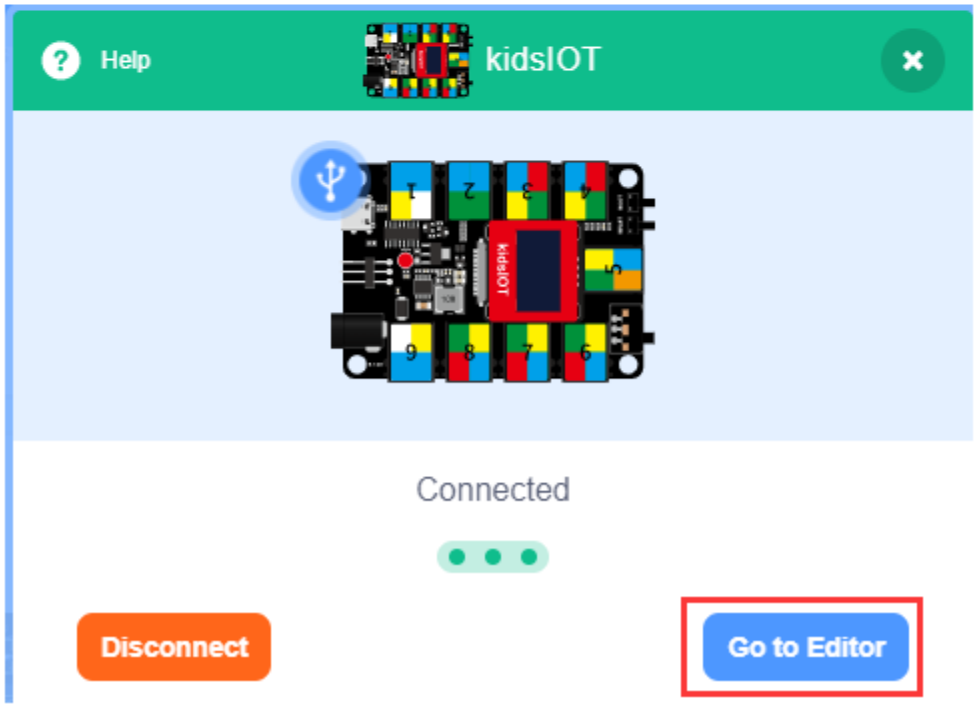
2. If the kidsIOT mainboard is connected , but icon 

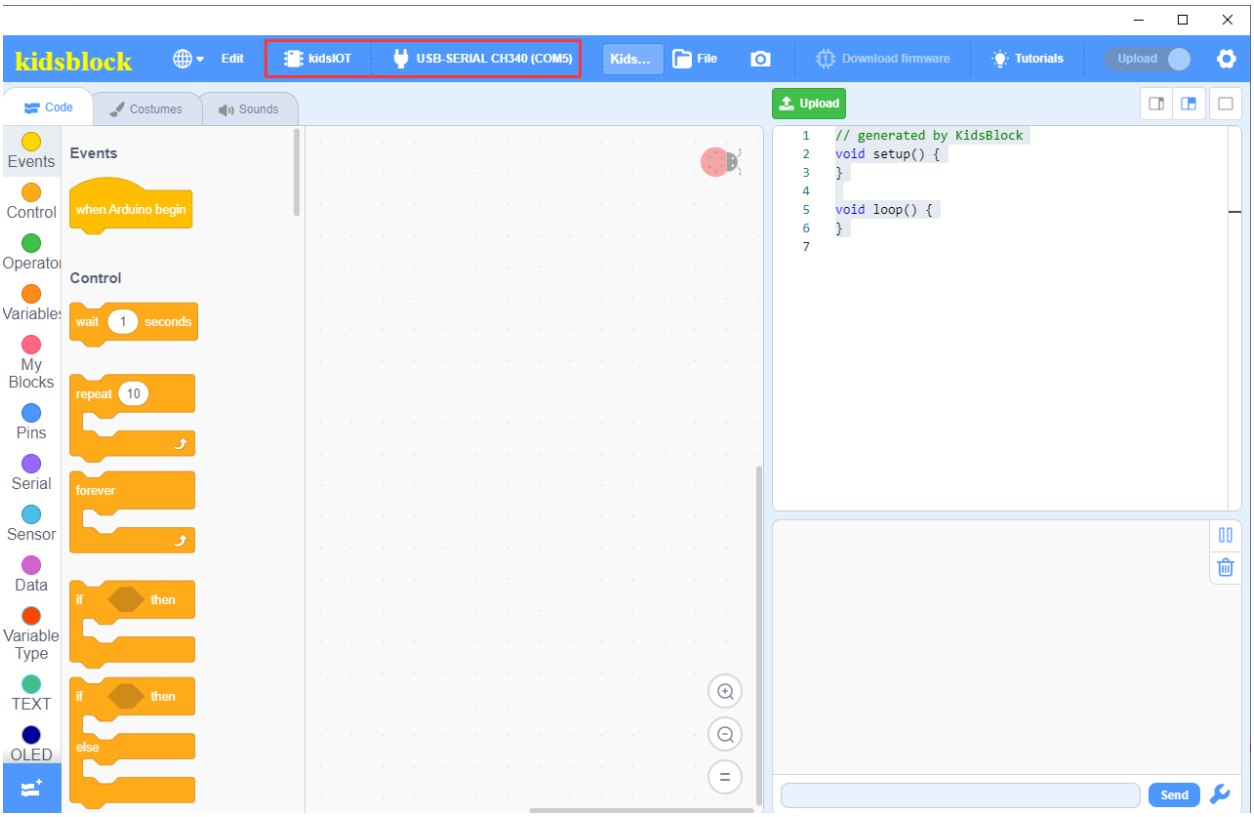
doesn't change into 

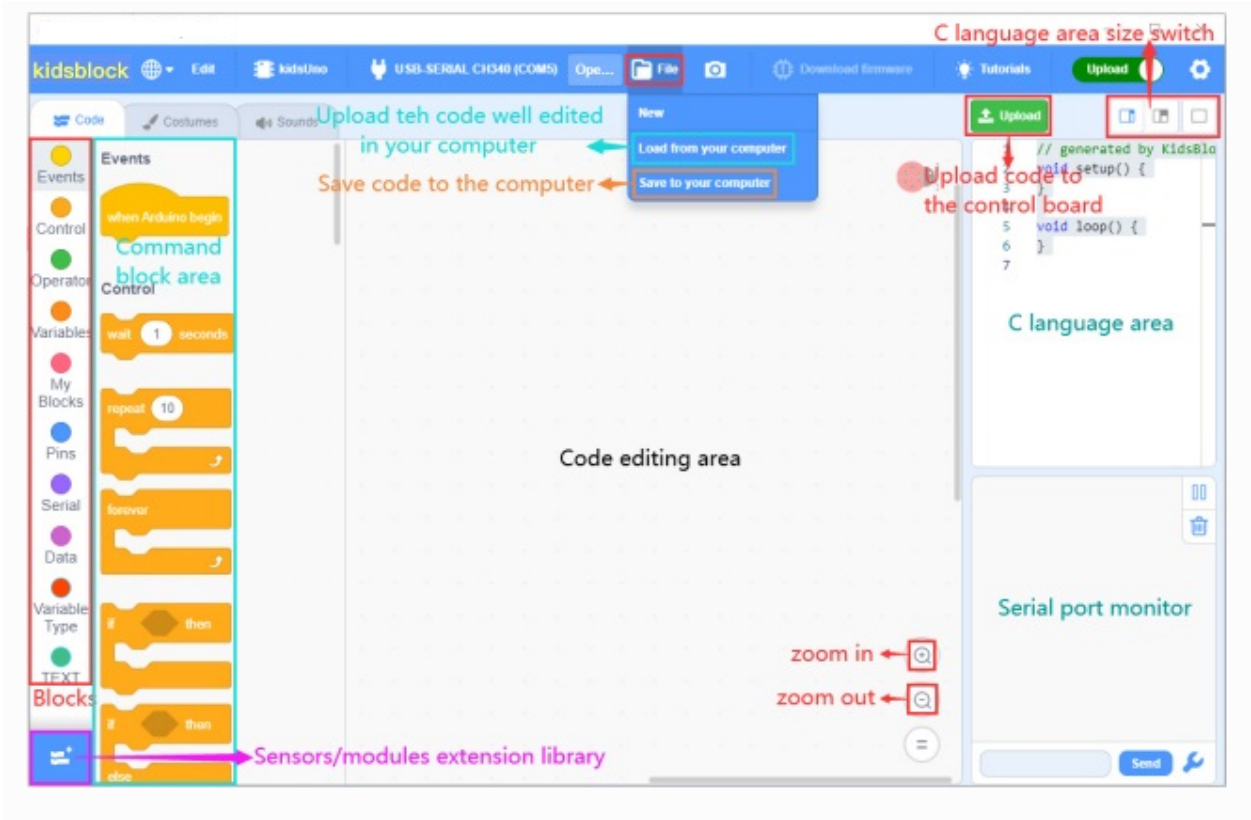
. You need to click to connect the COM port. Click 

. Then you will find a page pop up, showing Connected.


Help
kidsIOT
Show all connectable devices
Device name
USB-SERIAL CH340 (COM5)
Connect
Select your device in the list above.
Refresh

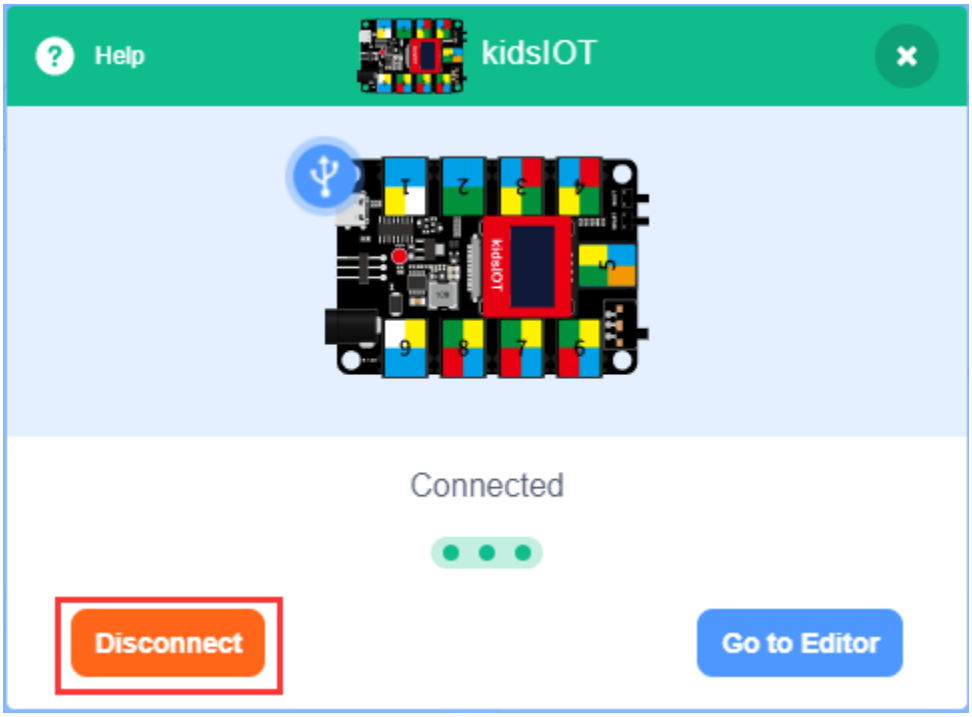

Help
kidsIOT
Connected
Disconnect
Go to Editor



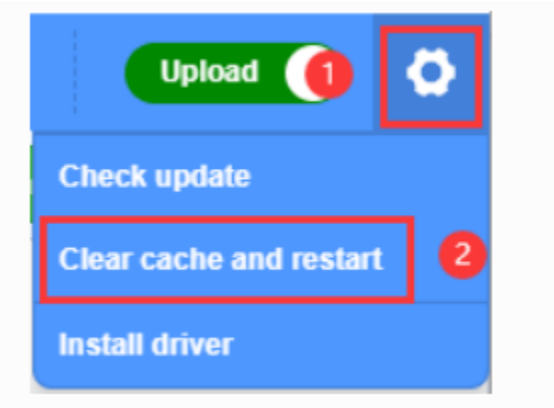


To disconnect the port, just click 

and Disconnect.



NoteIf you want to update libraries of KidsBlock, click then Clear cache and restart.



3.  stands for extension libraries of sensors and modules. Click  to enter the page of extension libraries, click a sensor or module to add. For example, if click the **esp32 passive buzzer** module,**Not loaded** will change into **Loaded**. Then the passive buzzer  is added.


Variable
Type
TEXT
OLED
if
then
else


S
V
G
P-BUZZER
esp32 Passive buzzer
esp32 Passive buzzer
Version
1.0.0
Author
keyes
Help
Not loaded

Click Back to return to the code editor. Then you can view the passive buzzer in the blocks area.

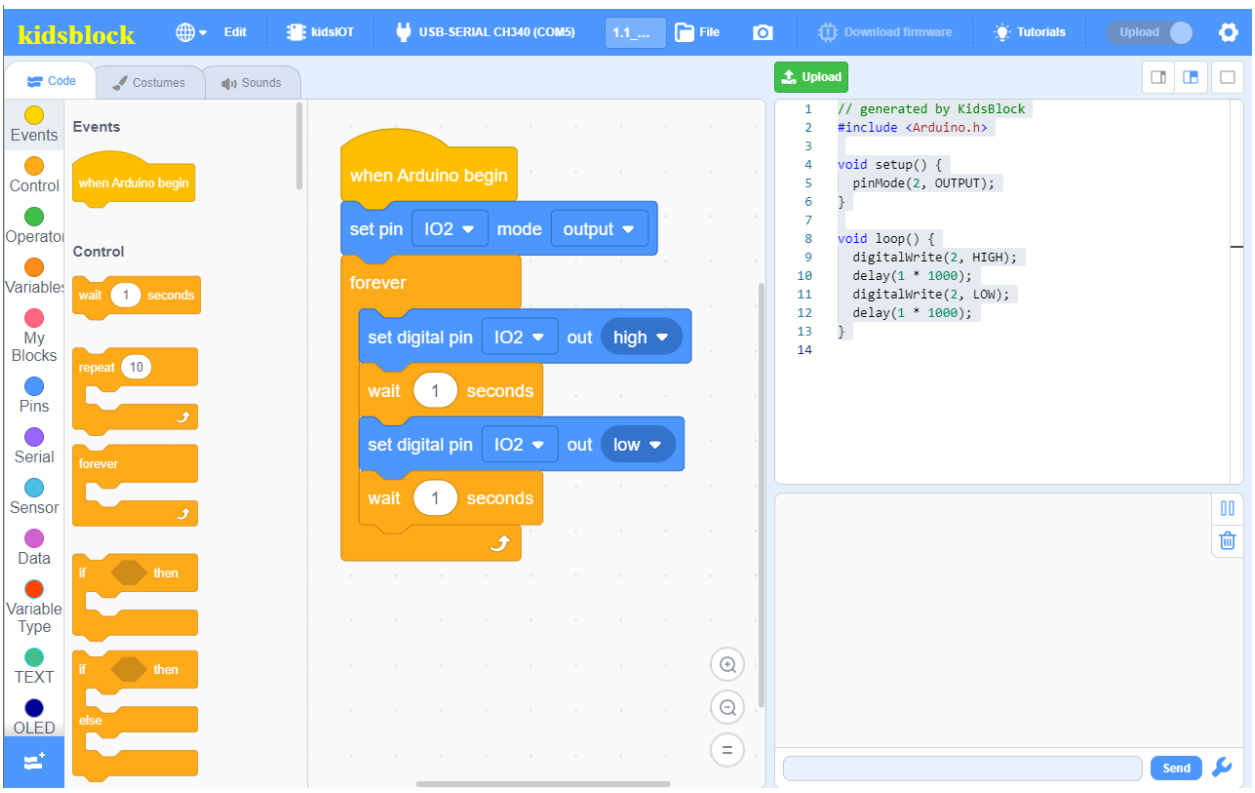
If you want to delete the **passive buzzer**, click  to select the "**esp32 Passive buzzer**"  . Then **Loaded** will change into **Not loaded**. Then the passive buzzer is deleted.


S
V
G
P-BUZZER
esp32 Passive buzzer
esp32 Passive buzzer
Version
1.0.0
Author
keyes
Help
Loaded

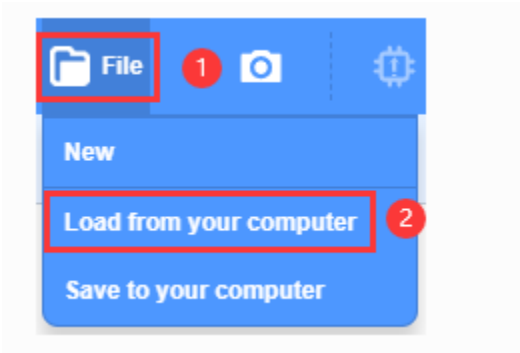
The way of deleting other sensors or modules is as same as the passive buzzer.

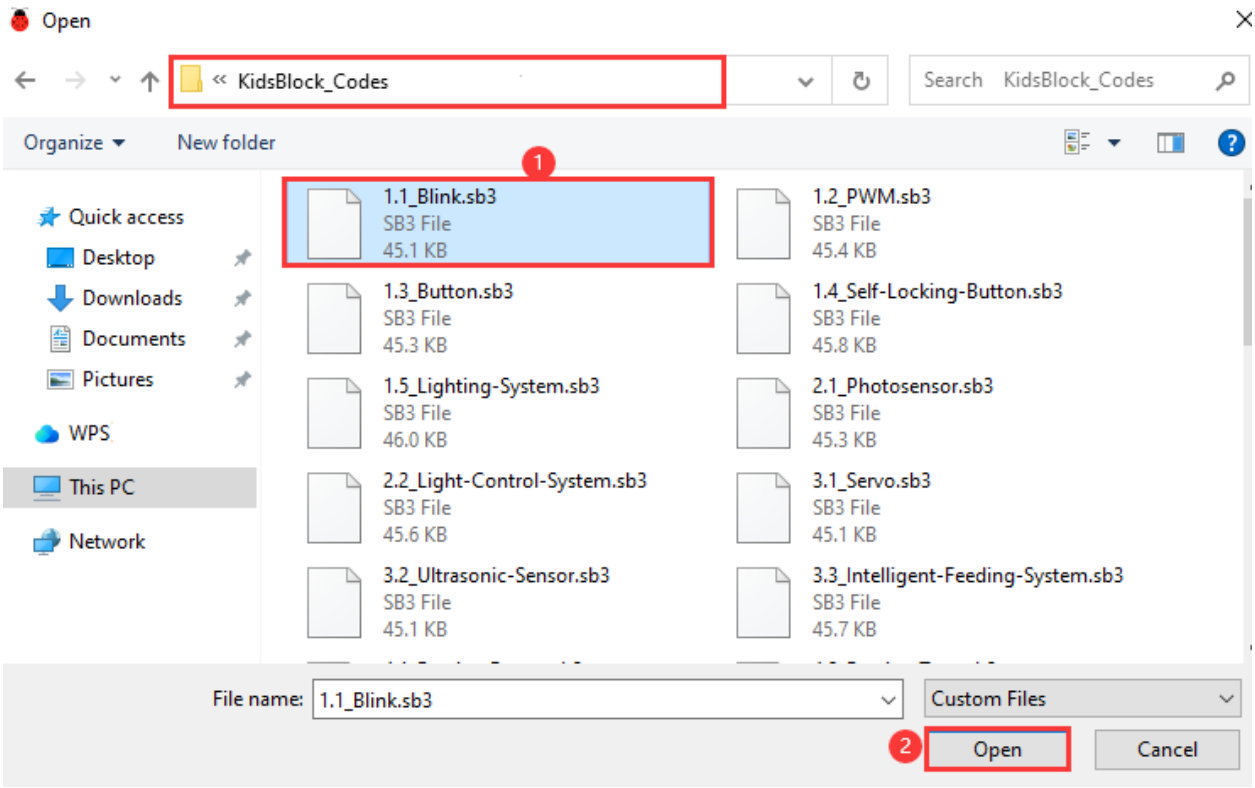
4. How to open SB3 type files

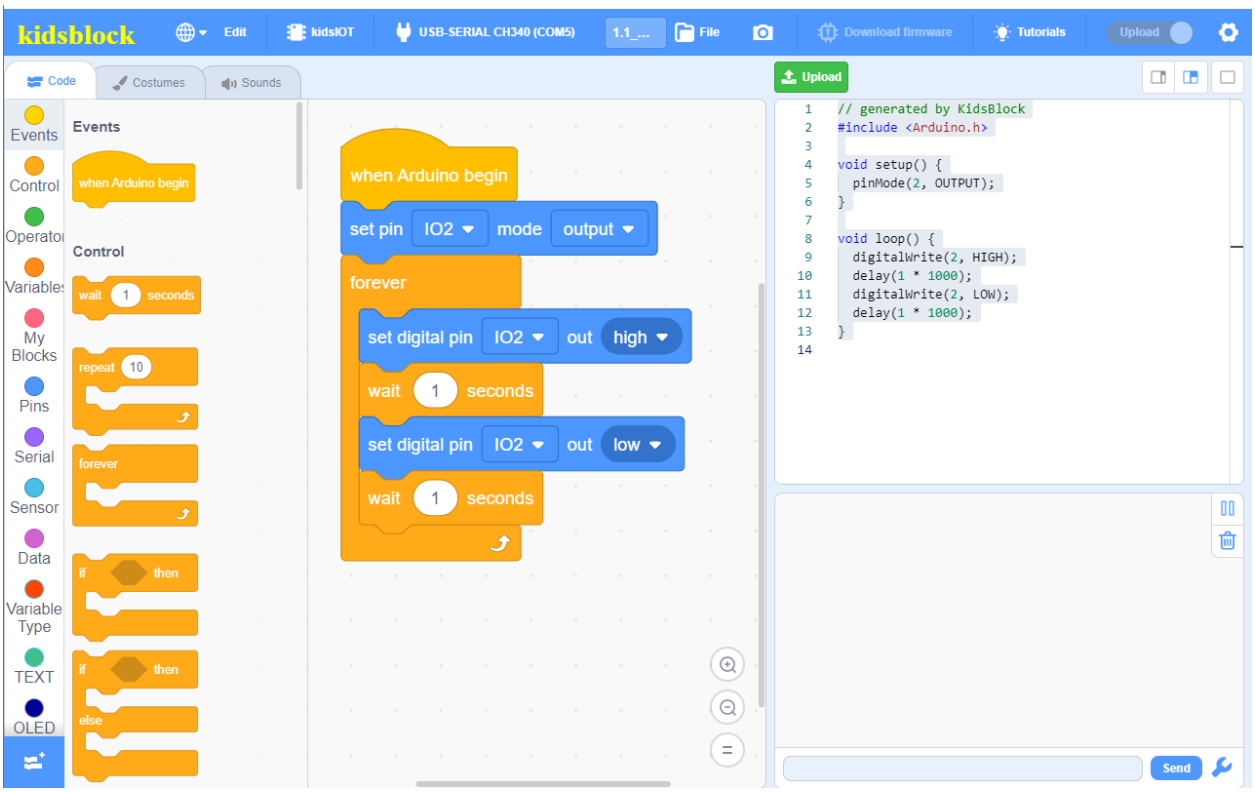
The first method Double-click SB3 type files to open them. For instance, open 1.1_Blink, then we need to double-click it.



The second method: Open Kidsblockclick **file** and **Load from your computer**then select the SB3 type file on the computer.for example 1.1_Blink)




Open
KidsBlock_Codes
Search KidsBlock_Codes
Organize
New folder
Quick access
Desktop
Downloads
Documents
Pictures
WPS
This PC
Network
1.1_Blink.sb3
SB3 File
45.1 KB
1.2_PWM.sb3
SB3 File
45.4 KB
1.3_Button.sb3
SB3 File
45.3 KB
1.4_Self-Locking-Button.sb3
SB3 File
45.8 KB
1.5_Lighting-System.sb3
SB3 File
46.0 KB
2.1_Photosensor.sb3
SB3 File
45.3 KB
2.2_Light-Control-System.sb3
SB3 File
45.6 KB
3.1_Servo.sb3
SB3 File
45.1 KB
3.2_Ultrasonic-Sensor.sb3
SB3 File
45.1 KB
3.3_Intelligent-Feeding-System.sb3
SB3 File
45.7 KB
File name: 1.1_Blink.sb3
Custom Files
Open
Cancel


kidsblock
Edit
kidsIOT
USB-SERIAL CH340 (COM5)
File
Download firmware
Tutorials
Upload
Code
Costumes
Sounds
Events
Control
Operator
Variables
My Blocks
Pins
Serial
Sensor
Data
Variable Type
TEXT
OLED
when Arduino begin
wait 1 seconds
repeat 10
forever
if then
else
when Arduino begin
set pin IO2 mode output
forever
set digital pin IO2 out high
wait 1 seconds
set digital pin IO2 out low
wait 1 seconds
// generated by KidsBlock
#include <Arduino.h>
void setup() {
pinMode(2, OUTPUT);
}
void loop() {
digitalWrite(2, HIGH);
delay(1 * 1000);
digitalWrite(2, LOW);
delay(1 * 1000);
}
Send

4.3 3. Projects:

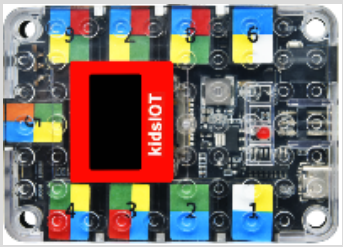
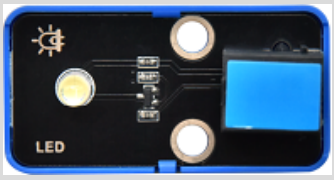
4.3.1 Project 01: Lighting System

1. Description

As an introductory project for smart farms, lighting up LED is one of the most basic KidsBlock (based on Scratch) practical projects. It is designed to let beginners understand the hardware and software programming of kidsIOT board (based on ESP32) and master basic circuit and programming knowledge.

In this project, you will learn the basic connections and settings of the kidsIOT board in the KidsBlock graphical programming environment, as well as control the digital pin to output level to control the state of LEDs, LED breathing lights, and button control of LEDs, and you can also apply them in your home or lounge.

2. Components

		
kidsIOT Mainboard×1	White LED Module×1	Button Module×1
		
USB Cable×1	Wire×2	Lighting System LEGO Pieces×1

About White LED and Button Module:

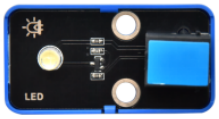
White LED: It is a device that can convert electrical energy into visible light. When current passes through the LED, it emits light.

Parameters:

Working voltage: DC 3.3V-5V

Working current: (Max) 1.5mA@5V

Maximum power: 0.07W



Button Module: It can output a digital signal 0 or 1. When the button is pressed, it will output a low level 0, otherwise it will output a high level 1. It is widely used in doorbells, desk lamps, air conditioner remote controls and fire alarms.

Parameters:

Working voltage: DC 3.3V-5V

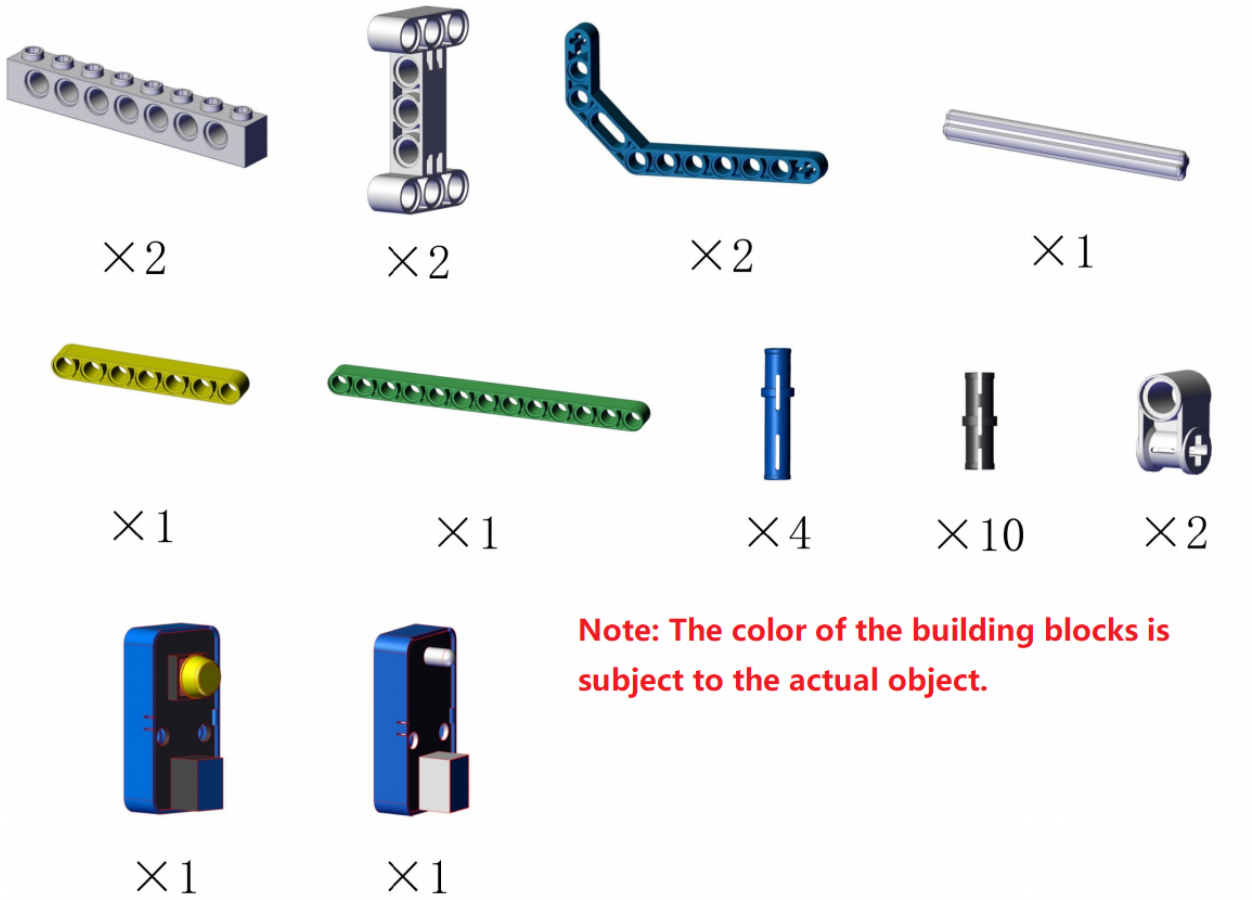
Working current: (Max) 1.1mA@5V

Maximum power: 5.5mW

Signal type: digital signal（0 or 1）

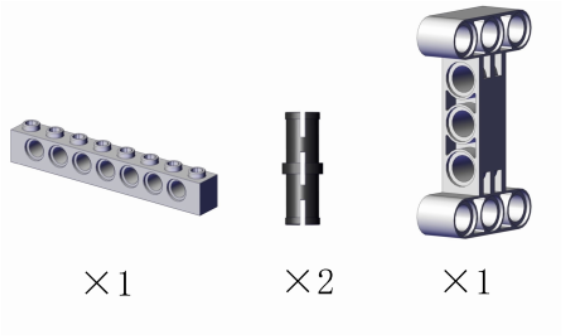
3. Assembly Steps

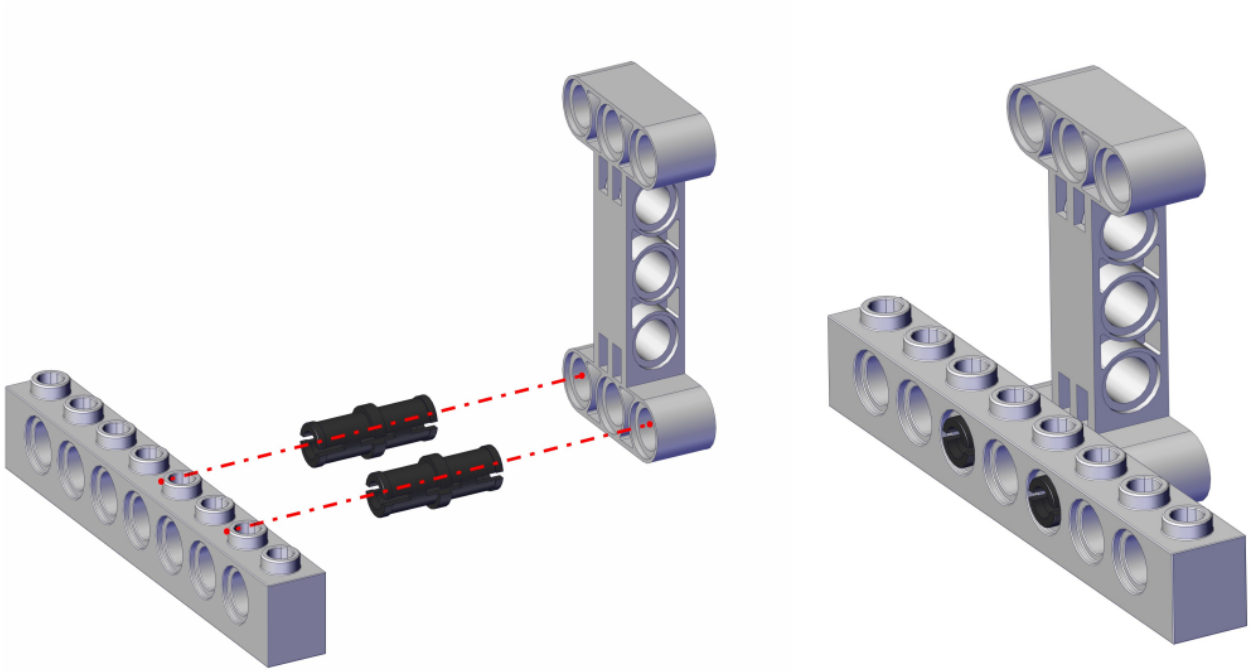
Step 1:Components Needed



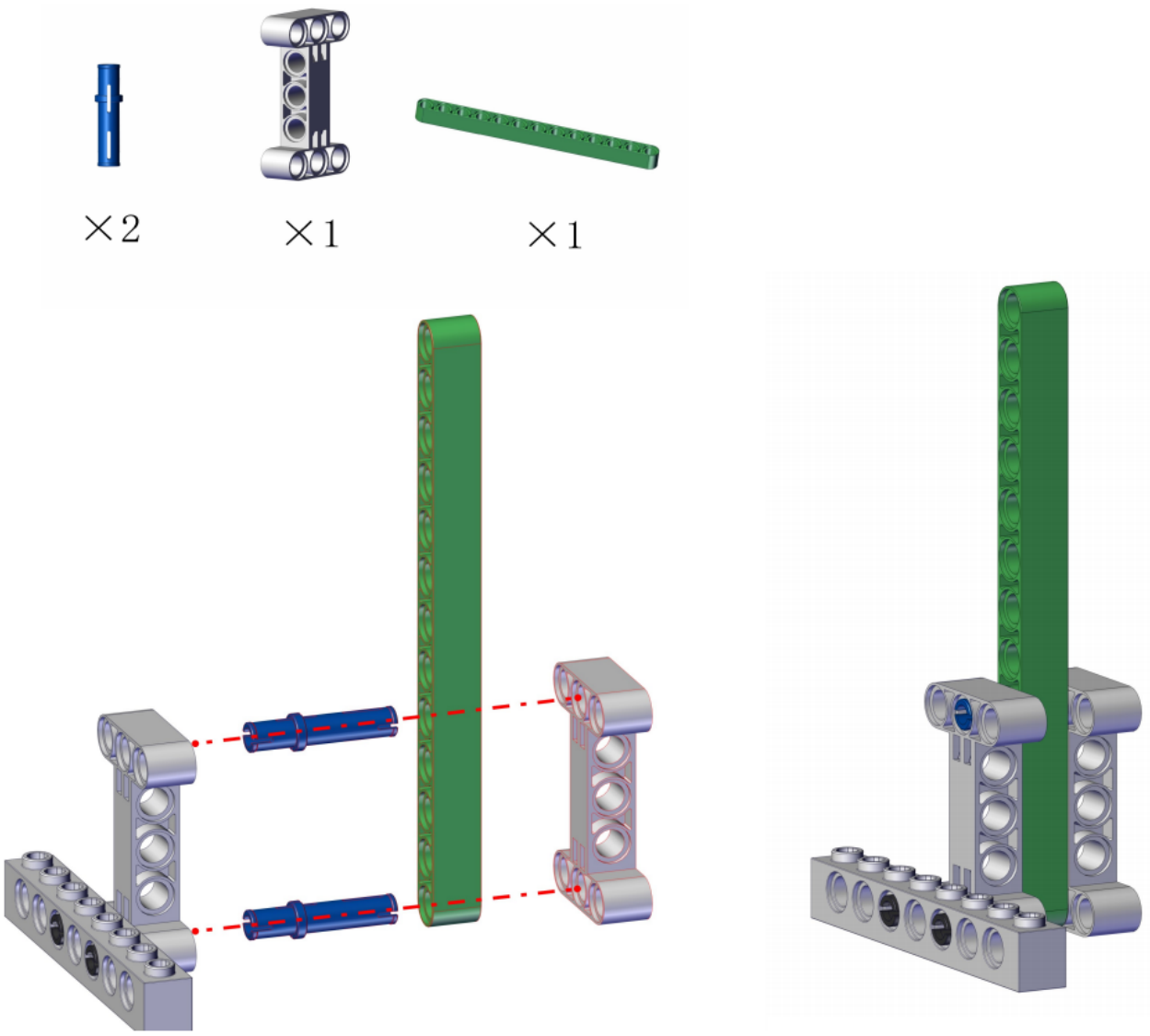
Step 2: Process

Process 1

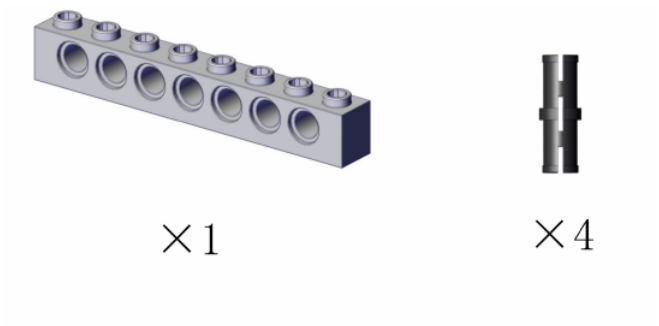


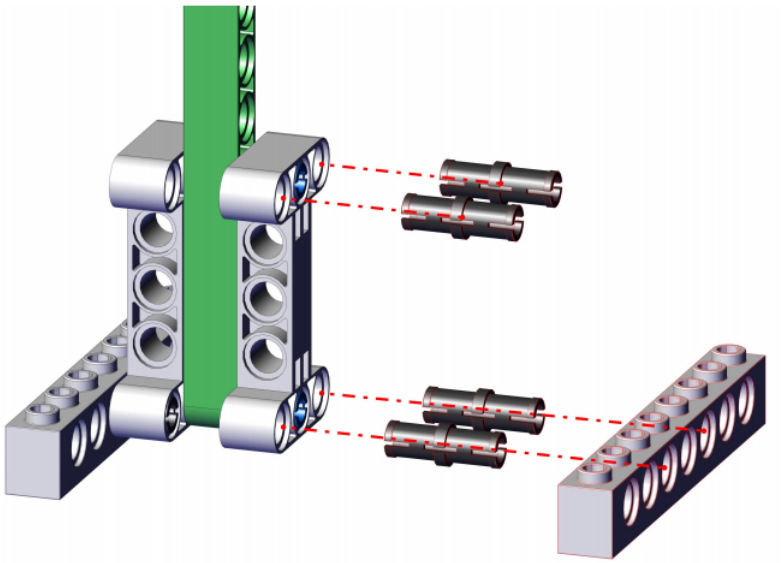


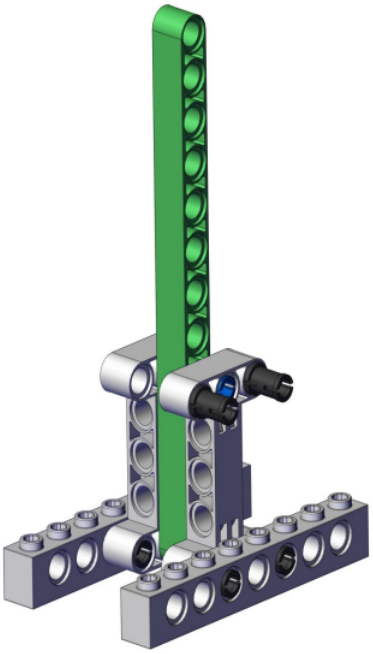
Process 2



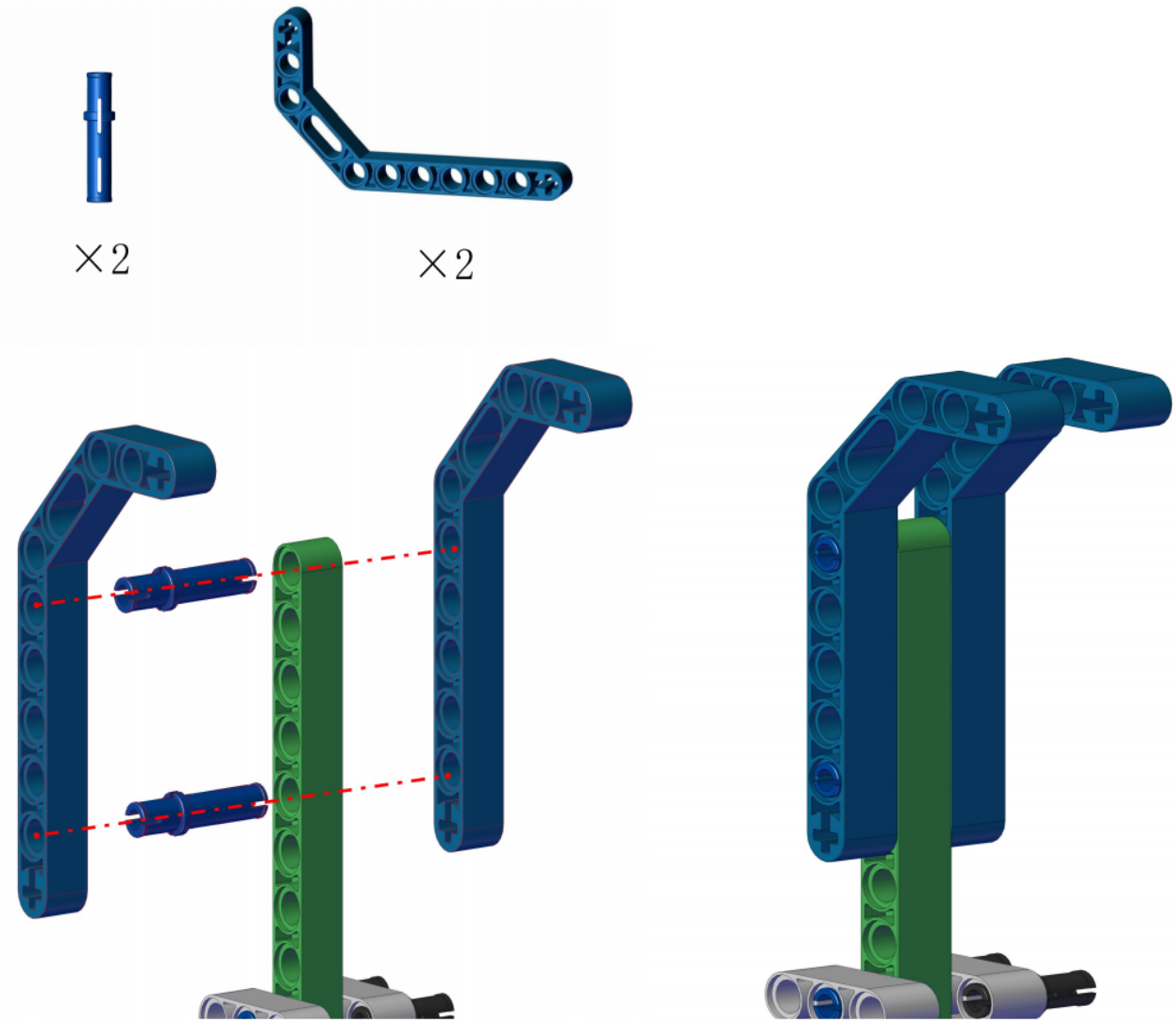
Process 3



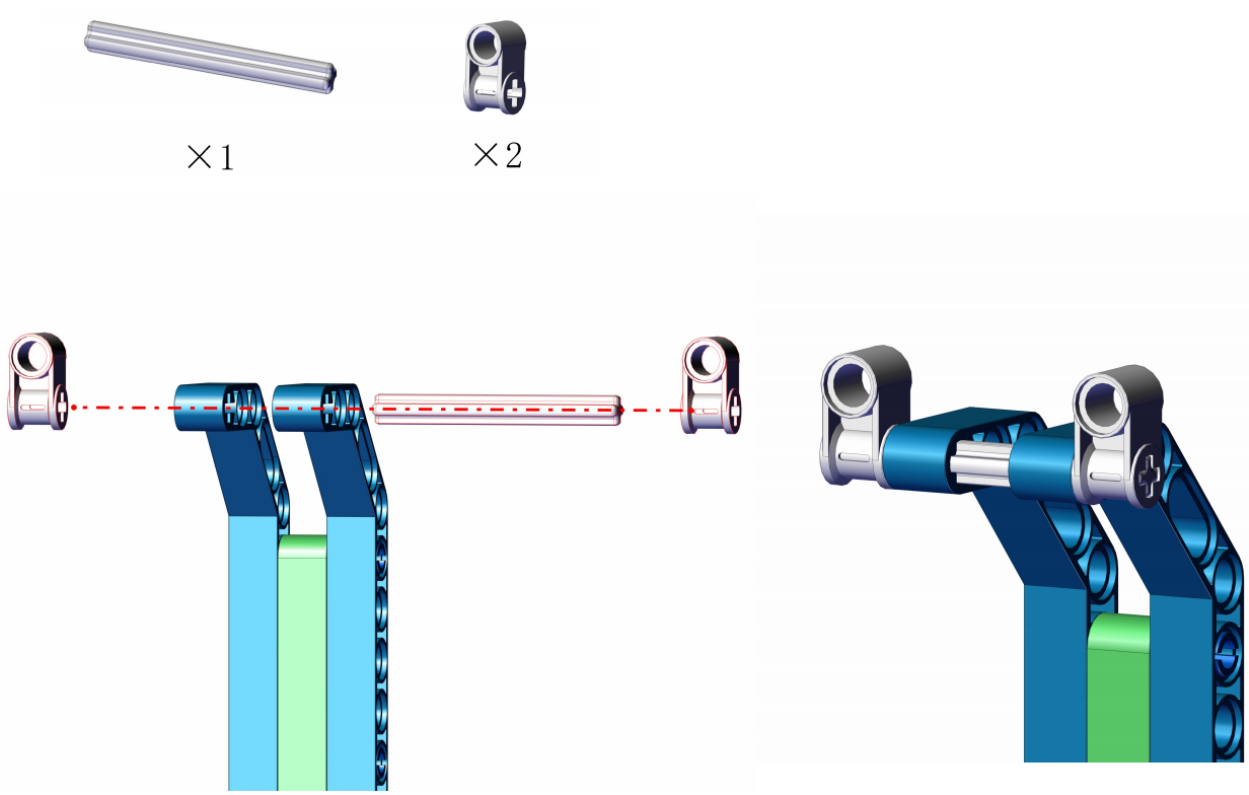




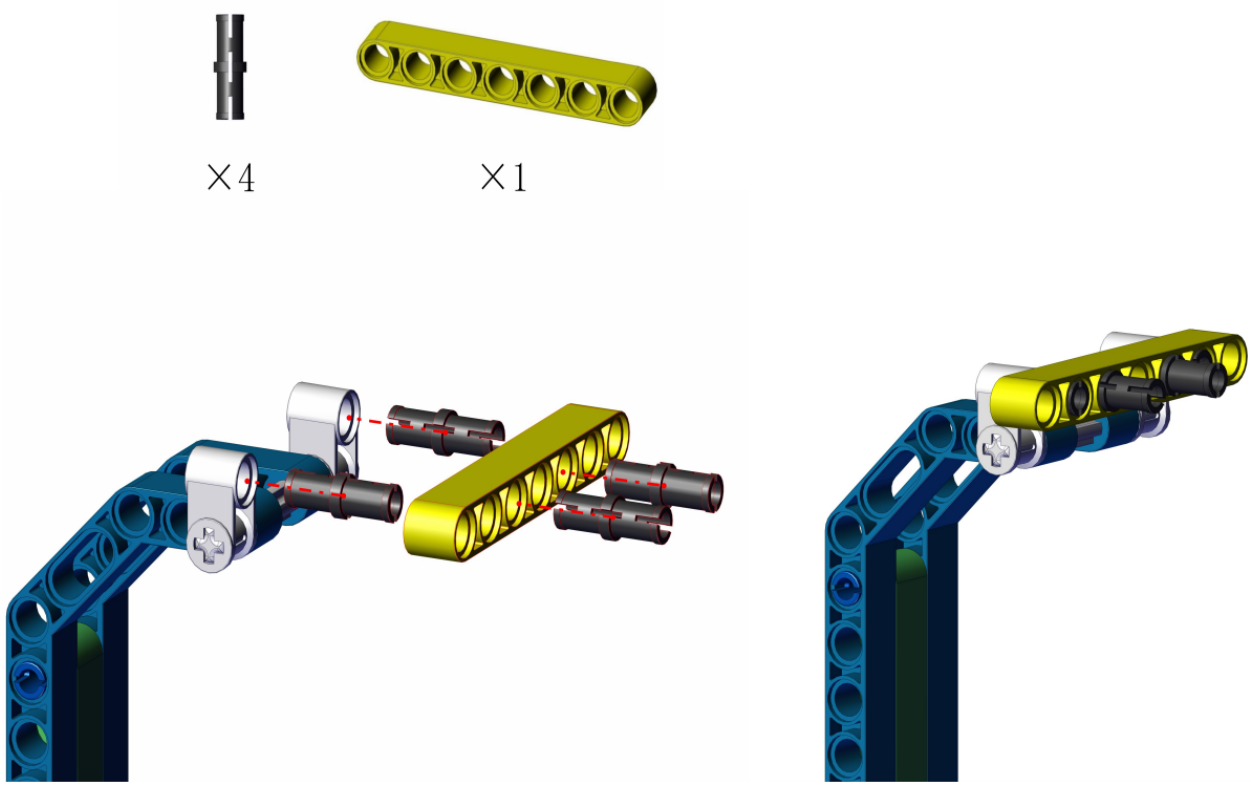
Process 4



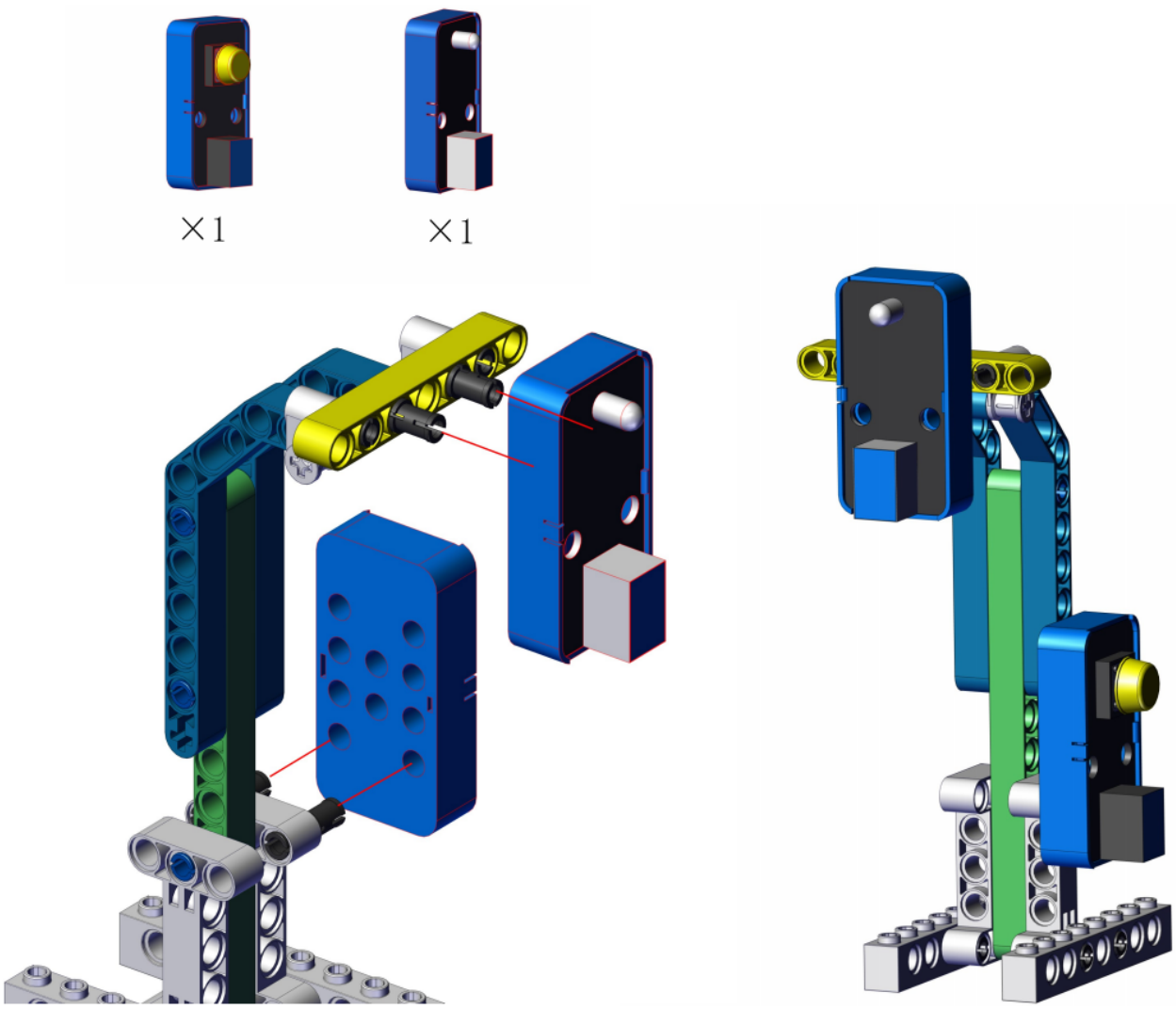
Process 5



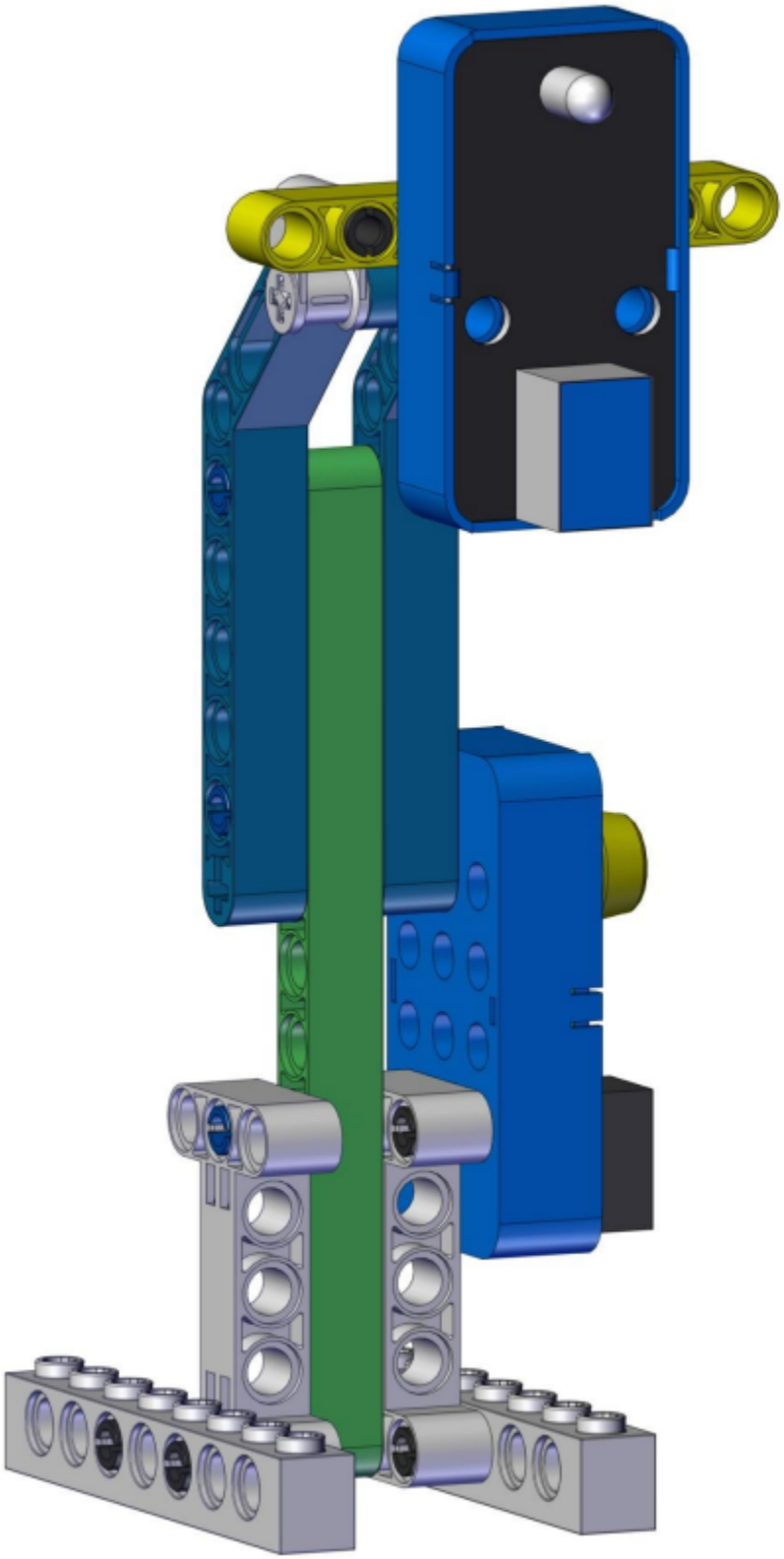
Process 6



Process 7



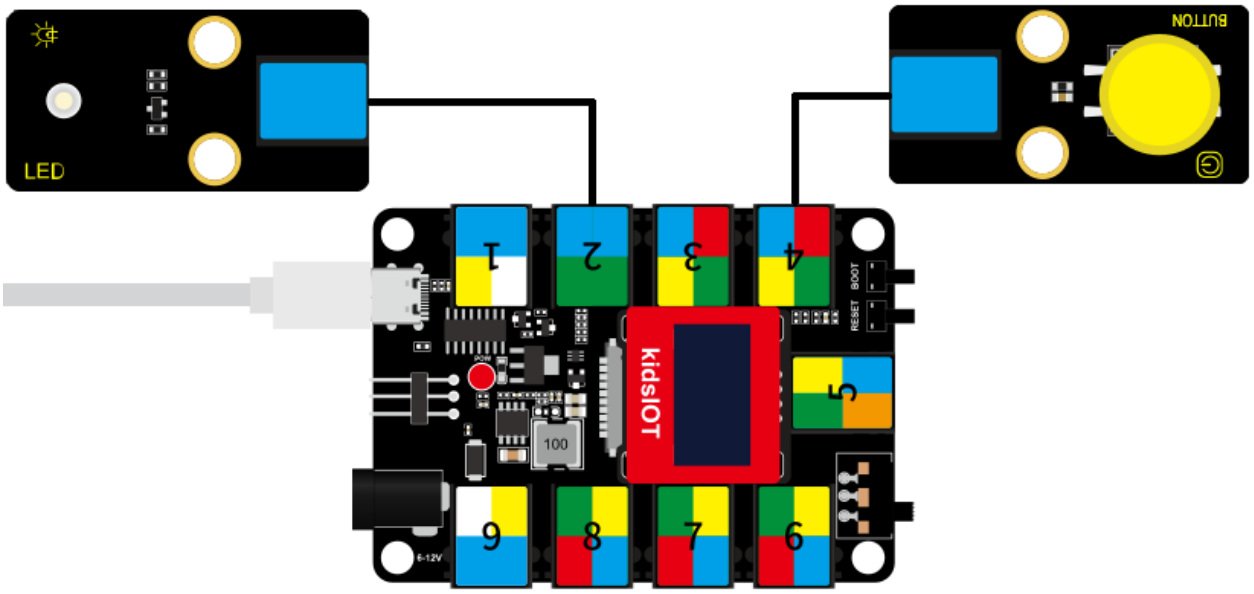
Complete



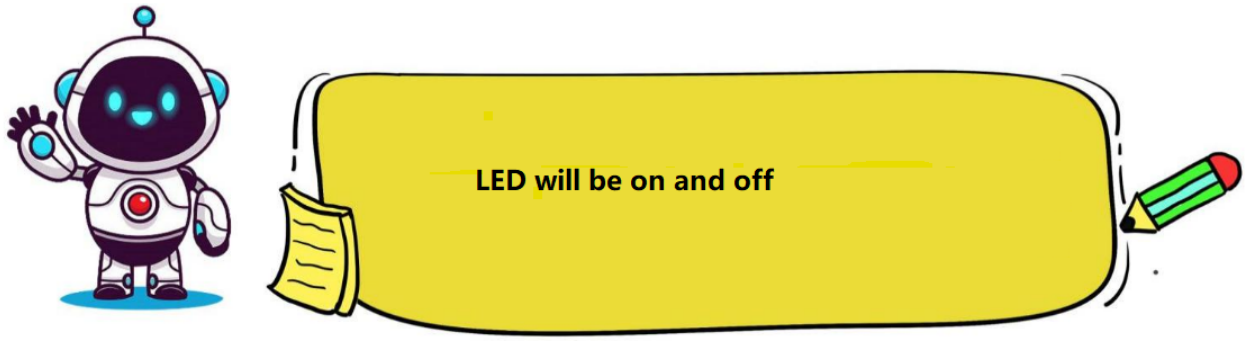
4. Wiring Diagram

Module	kidsIOT Mainboard
White LED Module	No.2 portcontrol pin is io2
Button Module	No.4 portcontrol pin is io27

Connect the kidsIOT mainboard to your computer via USB cable.

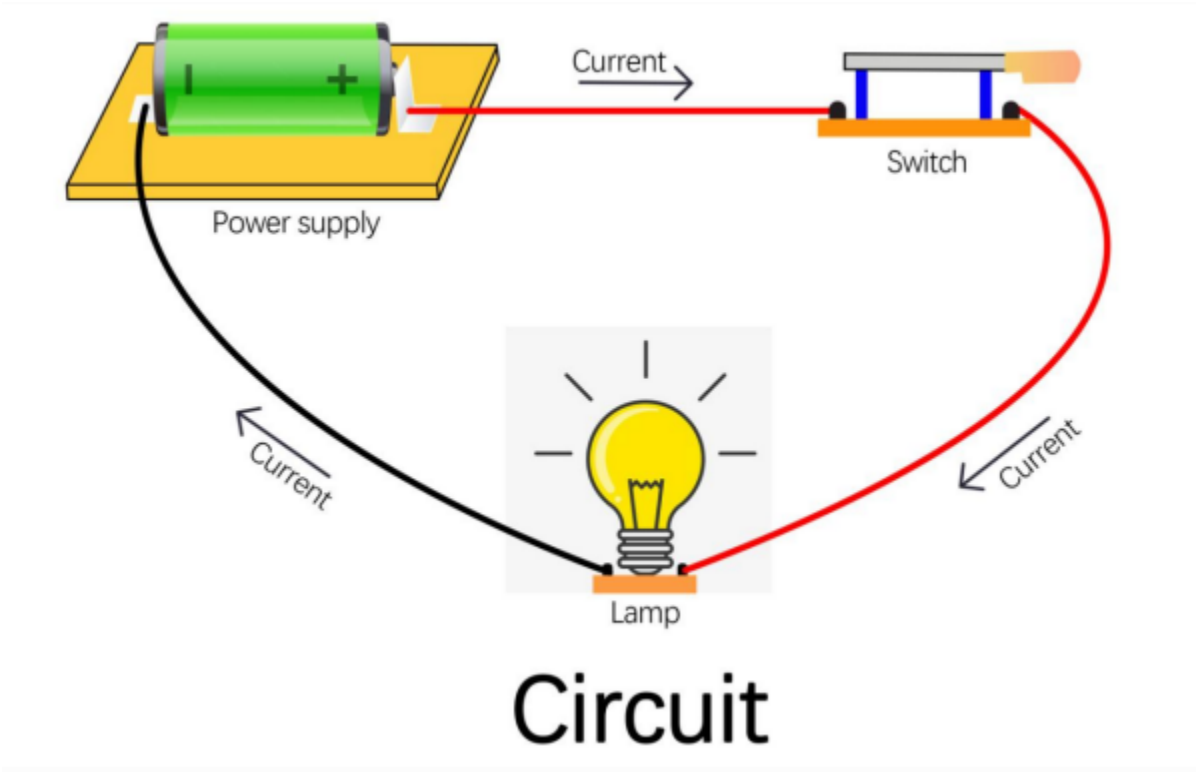


5. State of LED



(1). Knowledge

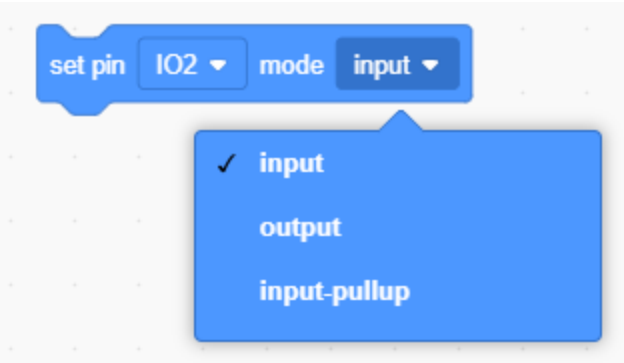
1To keep the light on, the electricity is needed. When we say that there is electricity, we mean that there is current flowing through an electrical appliance like a light. Current comes to our home from the power station via wires. And the generator of a power station is the power supply, which enables to provide voltage and current. The battery we usually use is also the power supply. Wires can be used to conduct electricity, which connect a path for the current to flow. This path is called a circuit. If we want to make a lamp emit light, both a power supply and a complete circuit are needed.



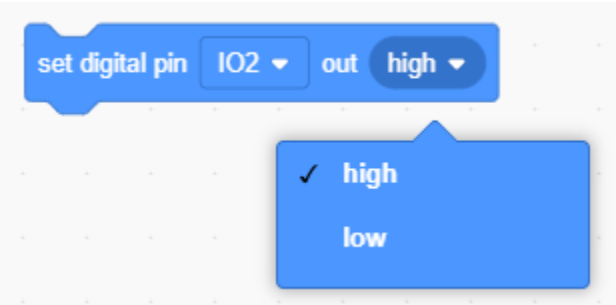
(2). Programming Steps

Step 1Description of the Building Blocks

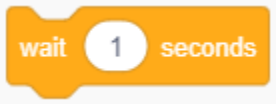
This block indicates that when the kidsIOT board is started, the code will be run.



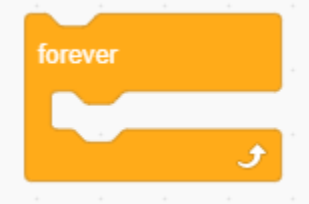
Set **input** or **output** to the specified pin. **input** means input mode,**output** means output mode. Select **input-pullup** can set the input mode for the pin and make it become high level.



Set **high** or **low** to the specified pin. Select **high** means to set high level for the pin. If there is voltage and current, the LED will be on. Select **low** means to set low level for the pin. If there is no voltage and current, the LED will be off.



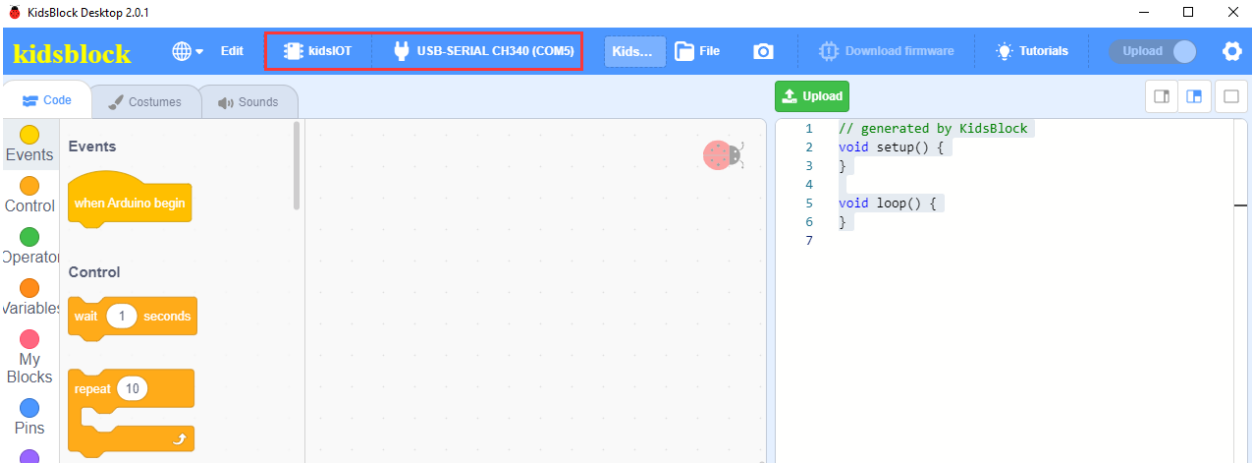
This is a delay block. The number 1 can be changed to whatever number of seconds it is delayed.



It will do one thing forever.

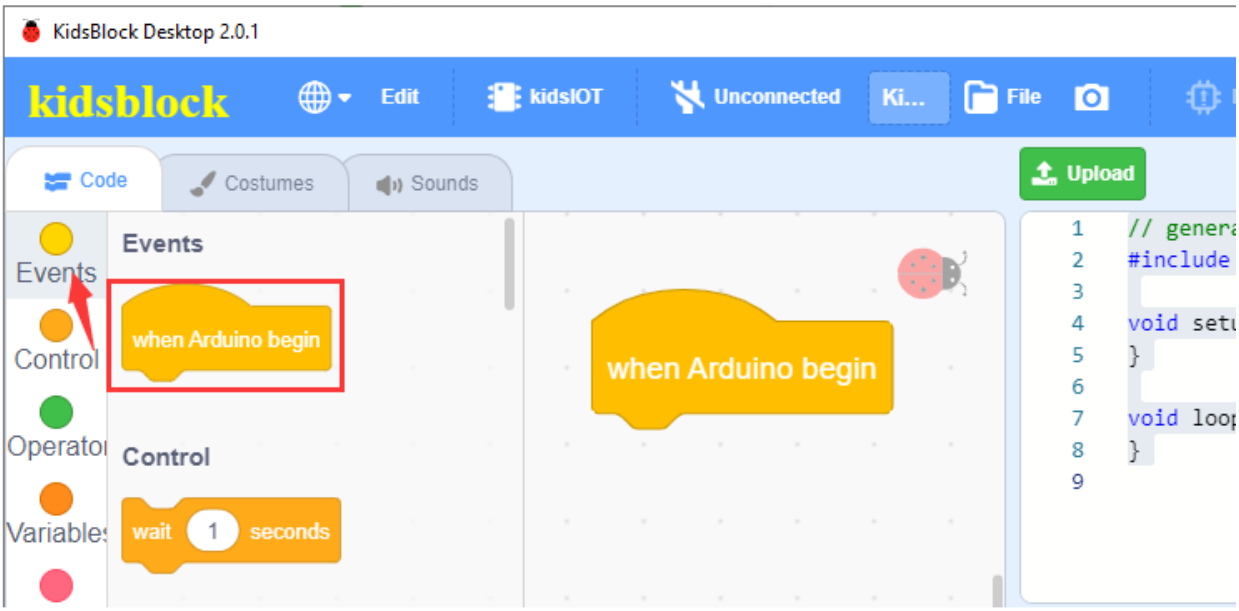
Step 2Write the Program

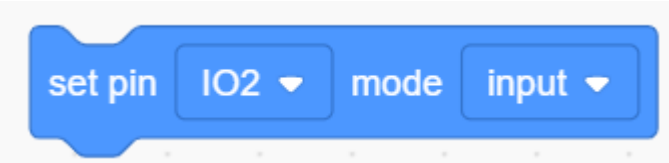
Open the KidsBlock(based on Scratch)software to select the kidsIOT board and port(COMx).



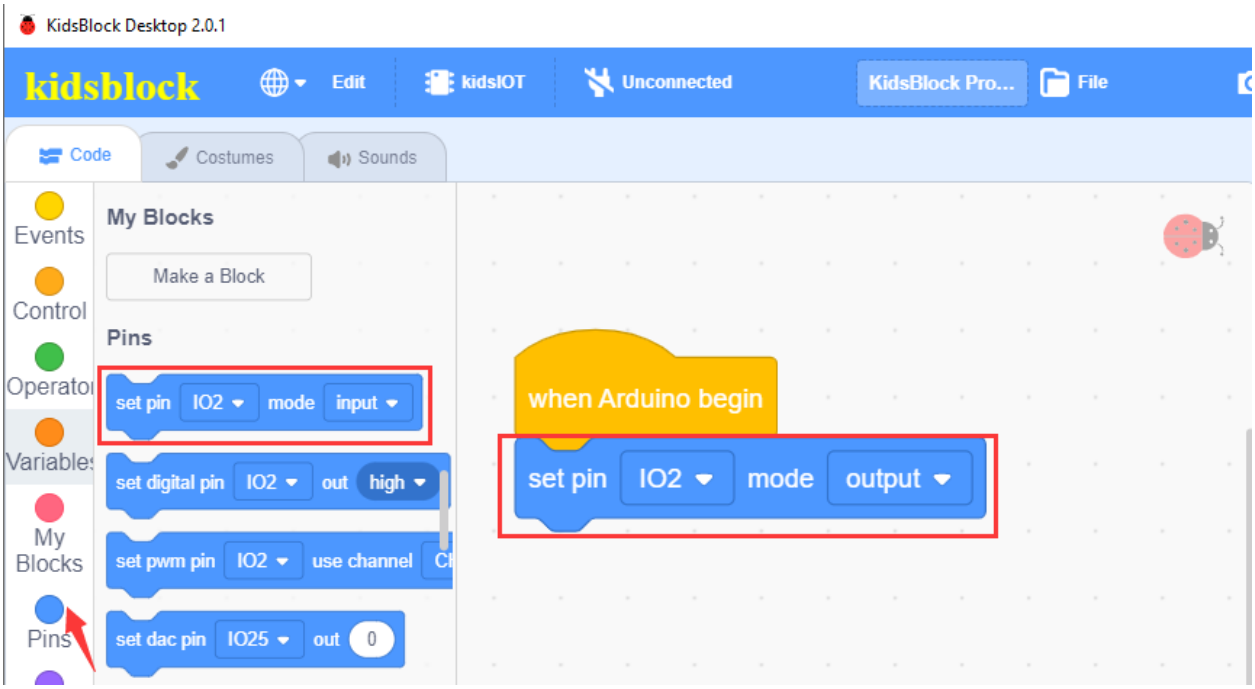
Drag the instruction block 

in the **Events** module to the script area.



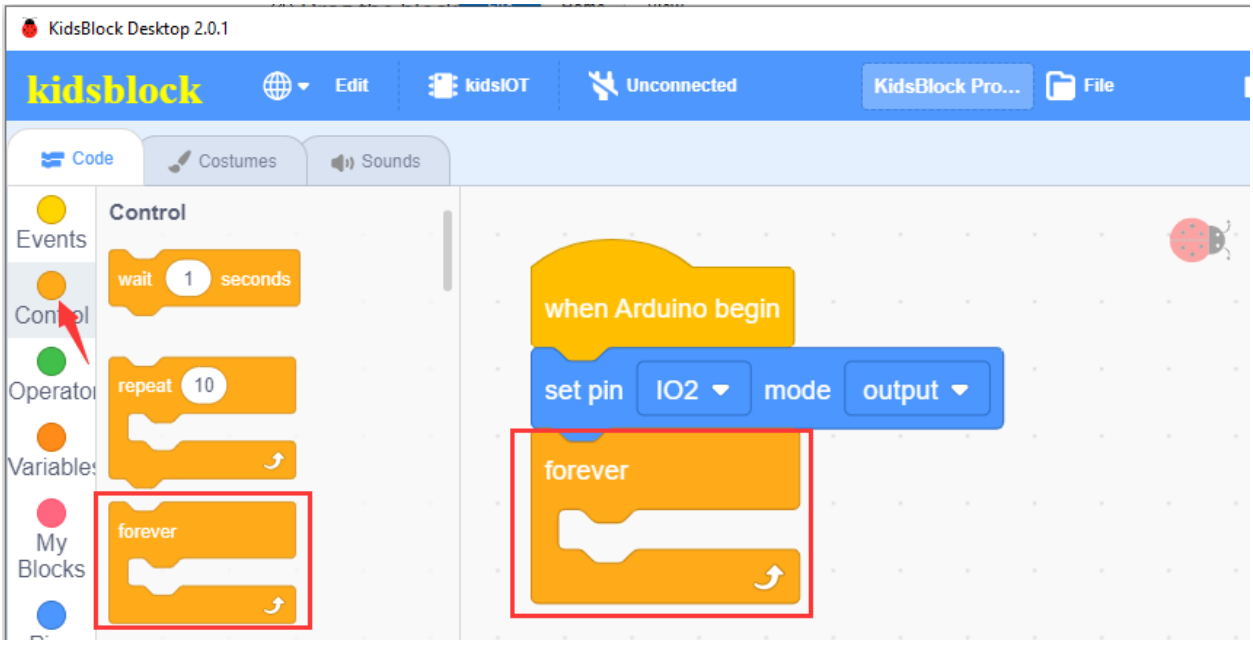
Drag the instruction block 

in the "**Pin**" module to the script area. Since the white LED module is connected to the No. 2 interface on the mainboard (The control pin is io2) and it is in output mode, so change "**input**" to "**output**".



Drag the block 

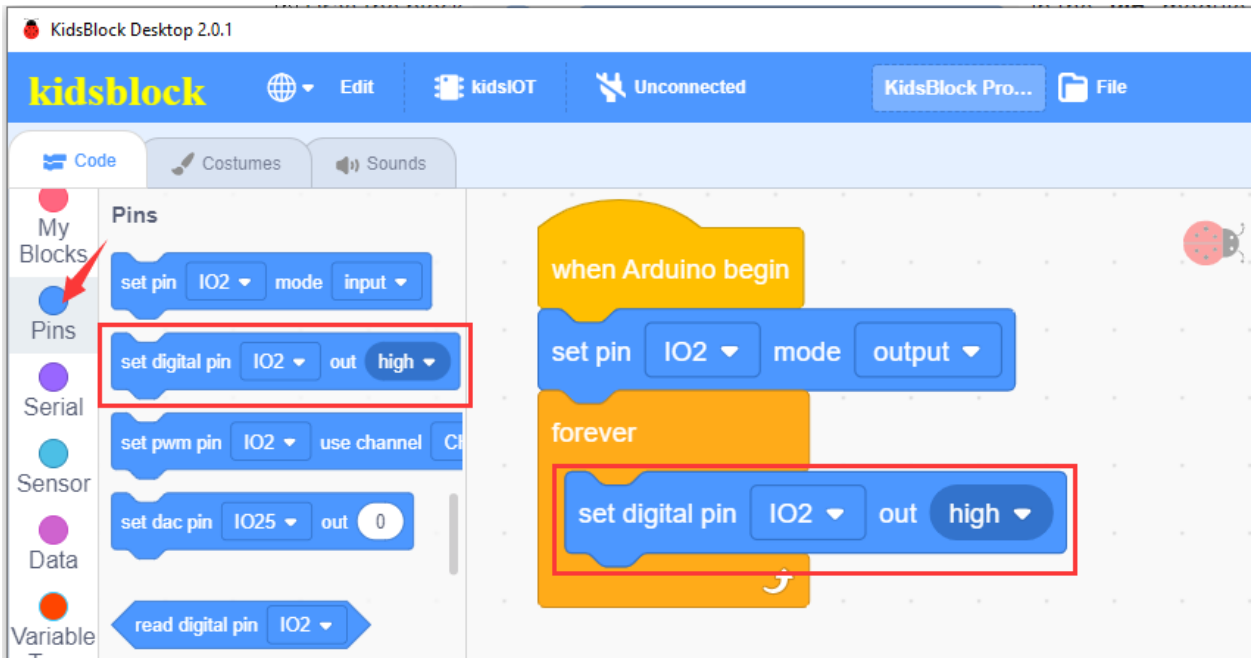
in the "**Control**" module to the script area.

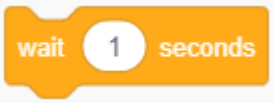


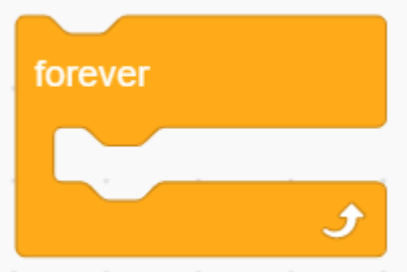
Drag the block 

in the "**Pin**" module to 

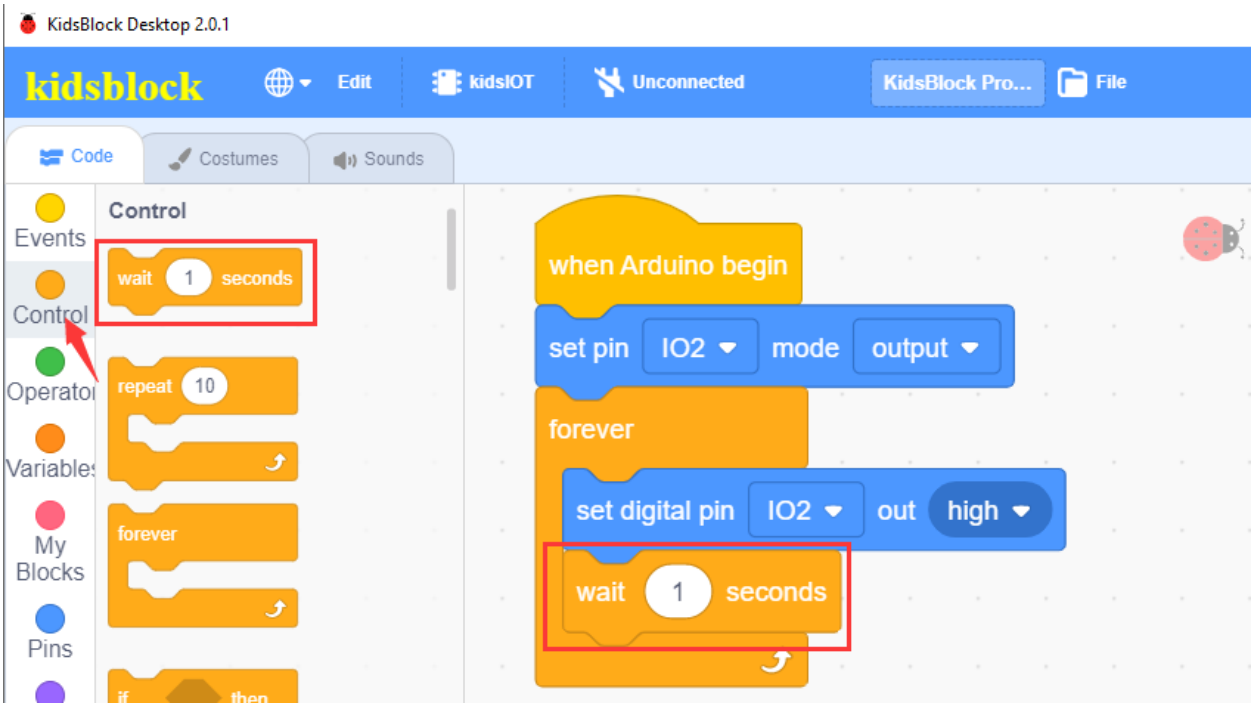
.

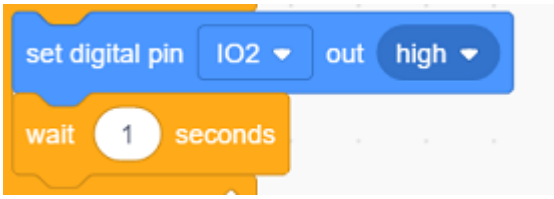


Drag the block 

in the "**Control**" module to 

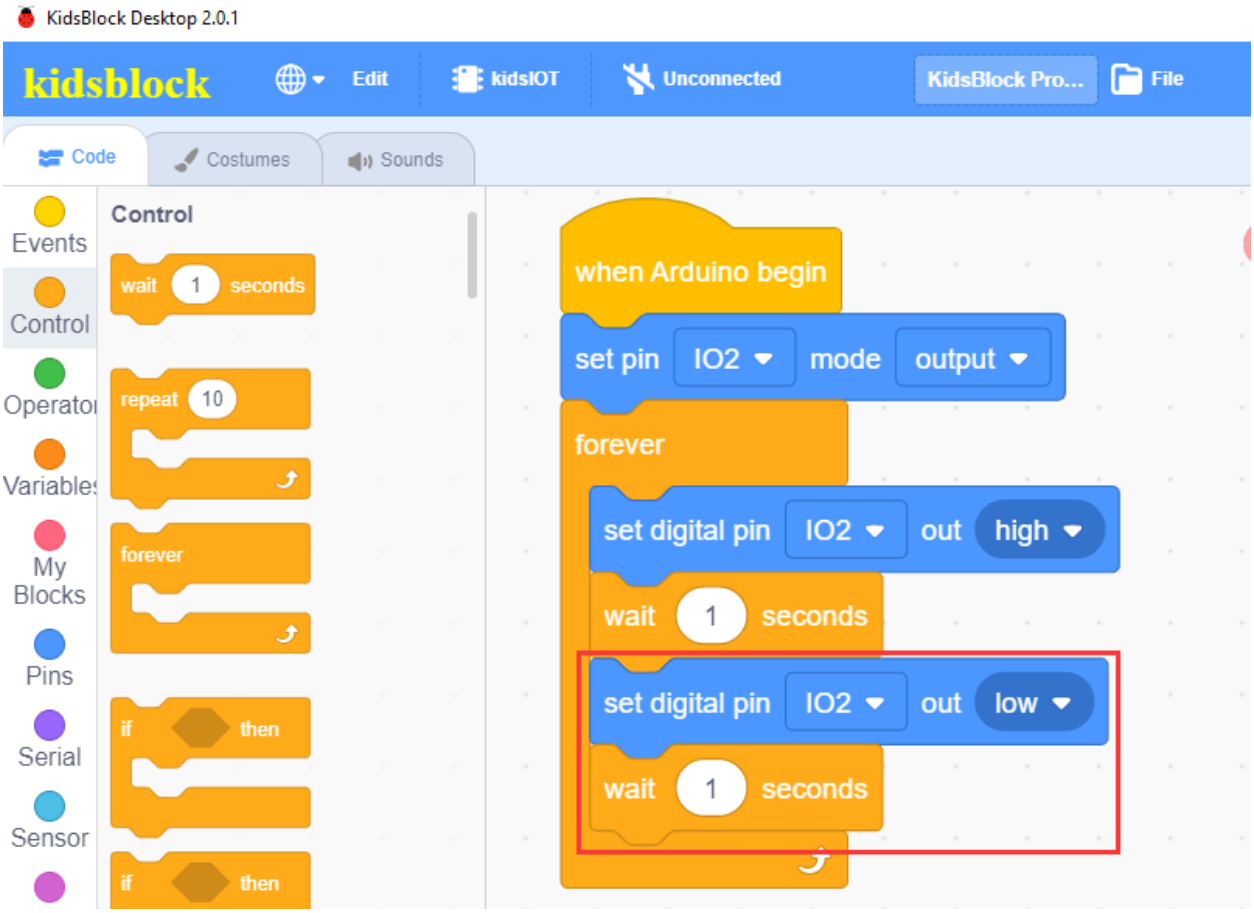
and set to delay **1** second.



Copy the code block 

into 

, and change "**High**" to "**Low**".



(3). Test Result

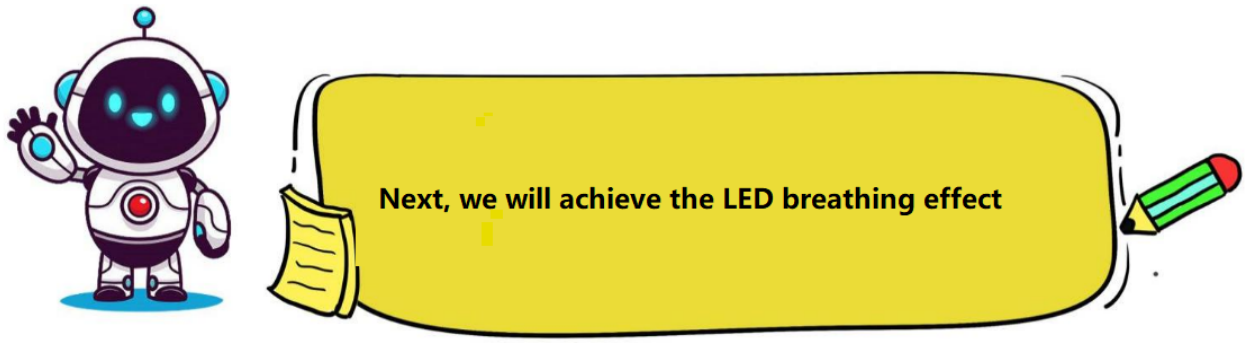
Click 

to upload the above complete code to the kidsIOT mainboard. After powering up via the USB cable, the white LED will be on and off for 1s.

We can also realize breathing light effect, flowing water light effect and police light effect via LEDs.

Level	Function
high	LED lights up
low	LED lights off

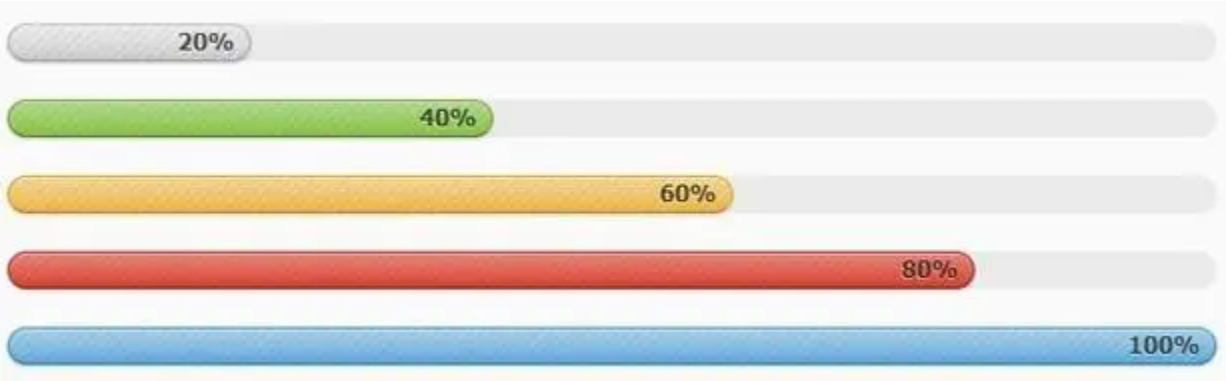
6. LED Breathing LED



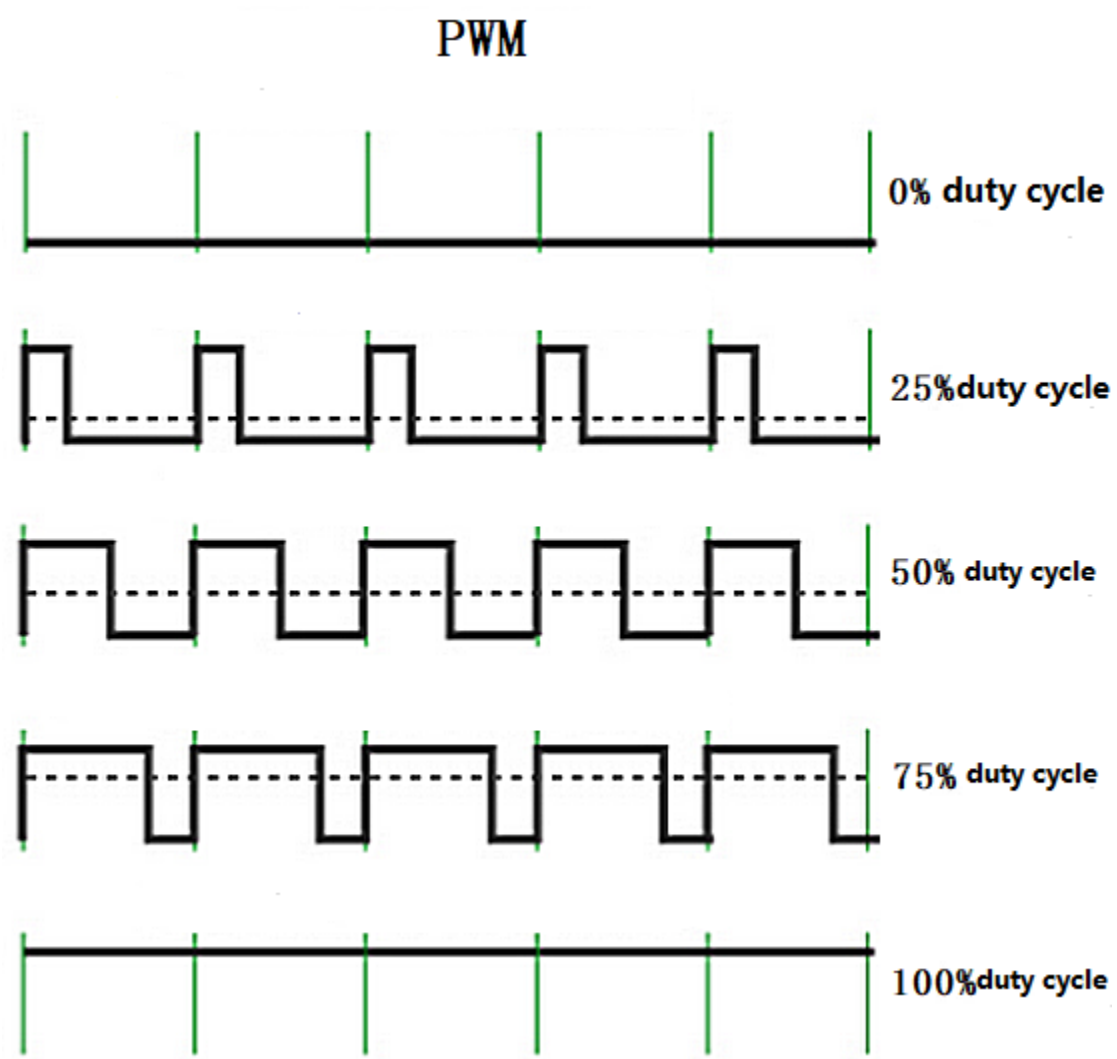
(1). Knowledge

The IO port on the kidsIOT mainboard outputs digital signals, which can only output high level and low level. For example, in the lighting up LEDs project, the digital output of the ESP32 is used, which has only two levels: high (3.3V) and low (0V).

Assuming that the high level of the kidsIOT board is 3.3V and the low level is 0V, then if you want to output a voltage between 0 and 3.3V, you need to use PWM (Pulse Width Modulation). PWM can output different voltage values, like a progress bar, which is analog output.

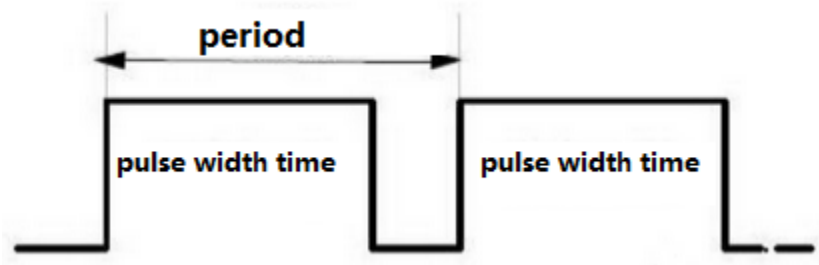


PWM uses digital control to generate square waves with different duty cycles (a signal that constantly switches between high level and low level) to control the analog output.



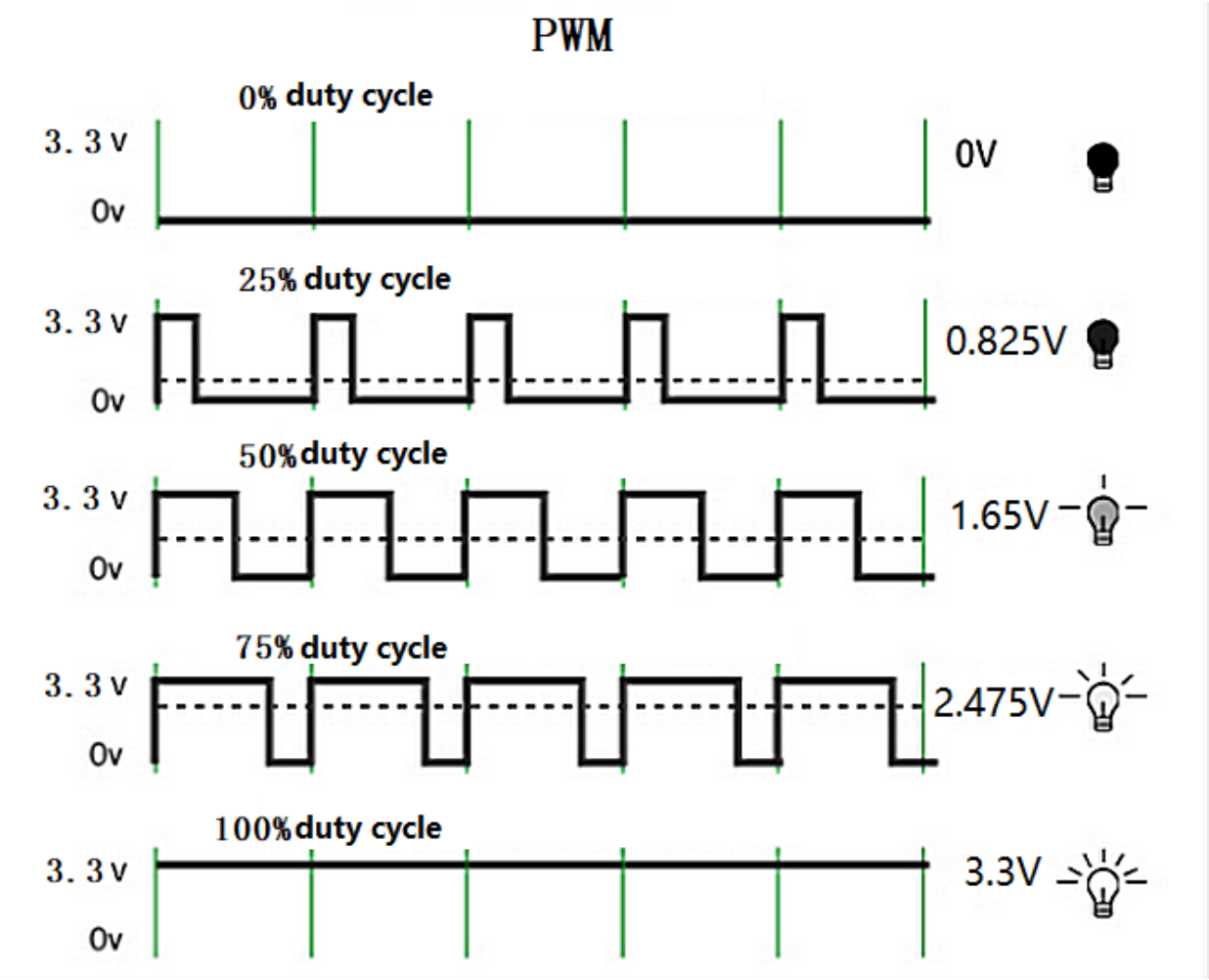
PWM has three elements: frequency (Hz), period (s) and duty cycle (%)

- **PWM frequency (f):** It refers to the number of PWM cycles in one second.
- **PWM period (T):** Period=1/frequency (T=1/f, where 1 is 1 second), for example: the frequency is 50Hz, which means that one period is 20ms, then one second is 50 PWM cycles.
- **PWM duty cycle:** It refers to the ratio of high level time to the entire cycle time within a pulse cycle. For example: the cycle time is 10ms, the pulse width time is 8ms, then the low level time is 2ms, and the total duty cycle is 8/(8+2)= 80%.



PWM can change the effective output voltage by changing the duty cycle in one cycle under the appropriate signal frequency. Among the levels output by the IO port at the specified time, the more high levels, the greater the PWM

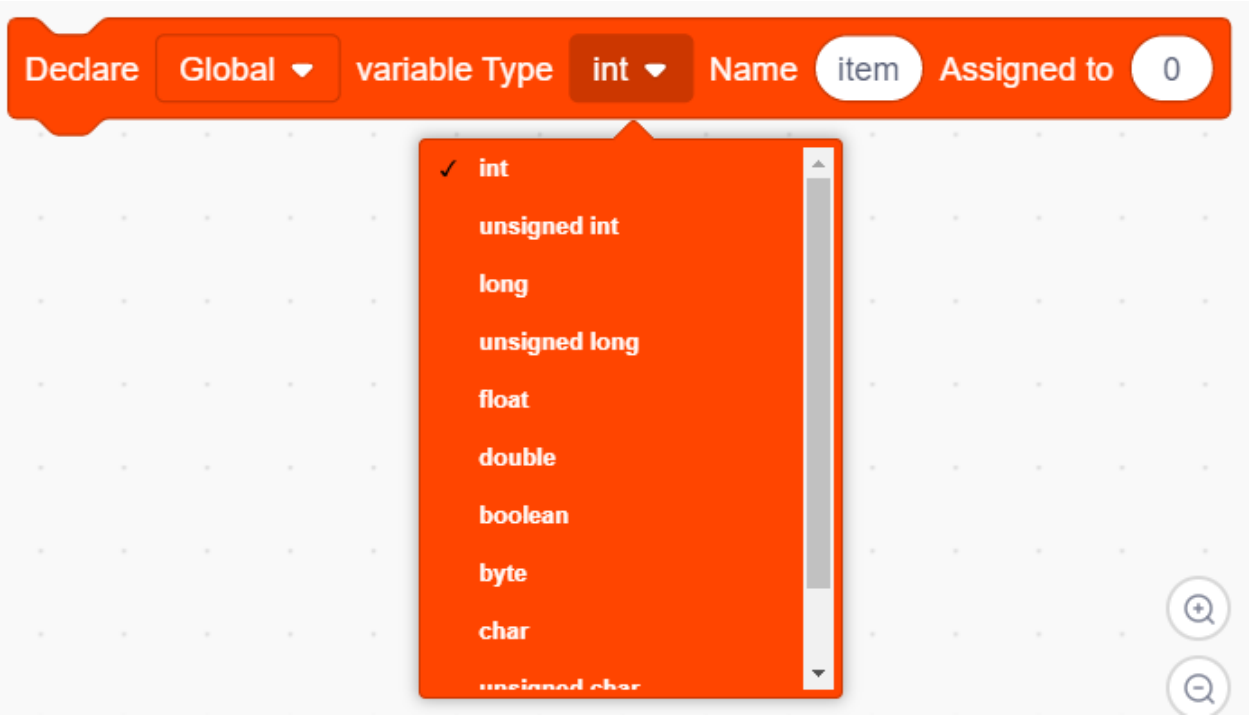
value and the brighter the LED.



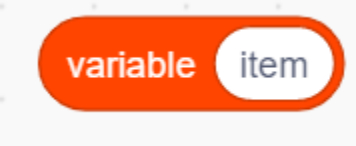
(2). Programming Steps

Step 1Description of the Building Blocks

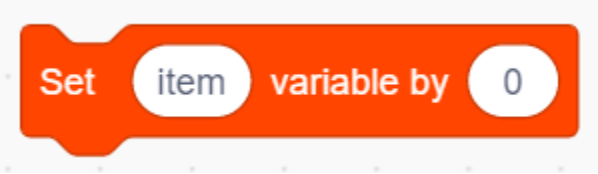
The following are "**Variable**" command blocks.



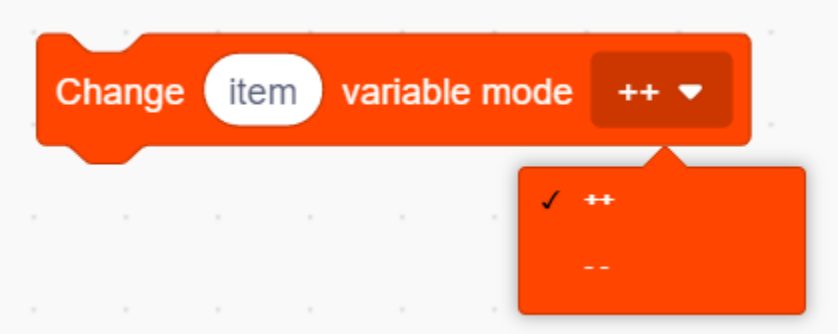
This block is used to create "Variable". You can declare "global" or "local", or set the type, name and value of the variable, item is the variable name.



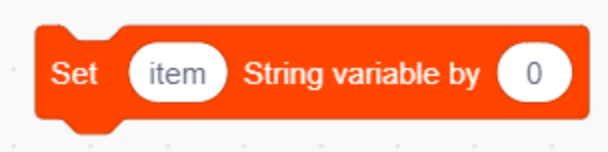
Get variable item.



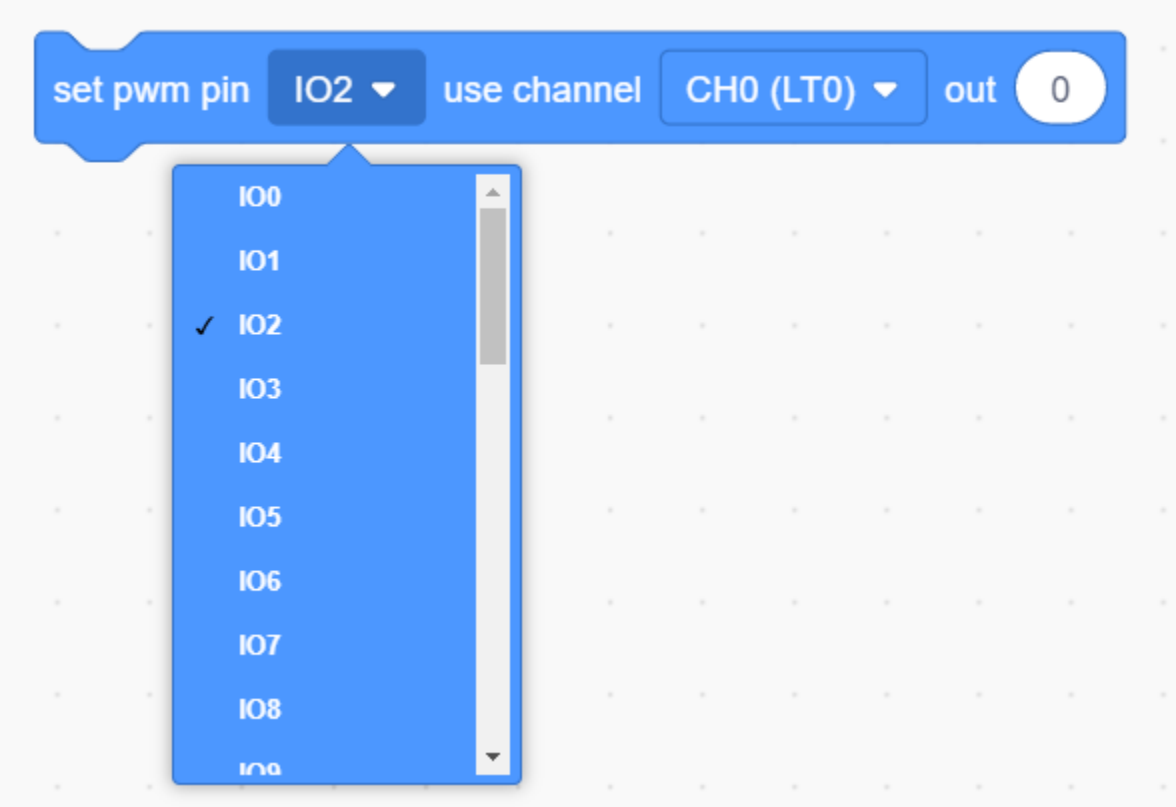
Set the value of variable item.

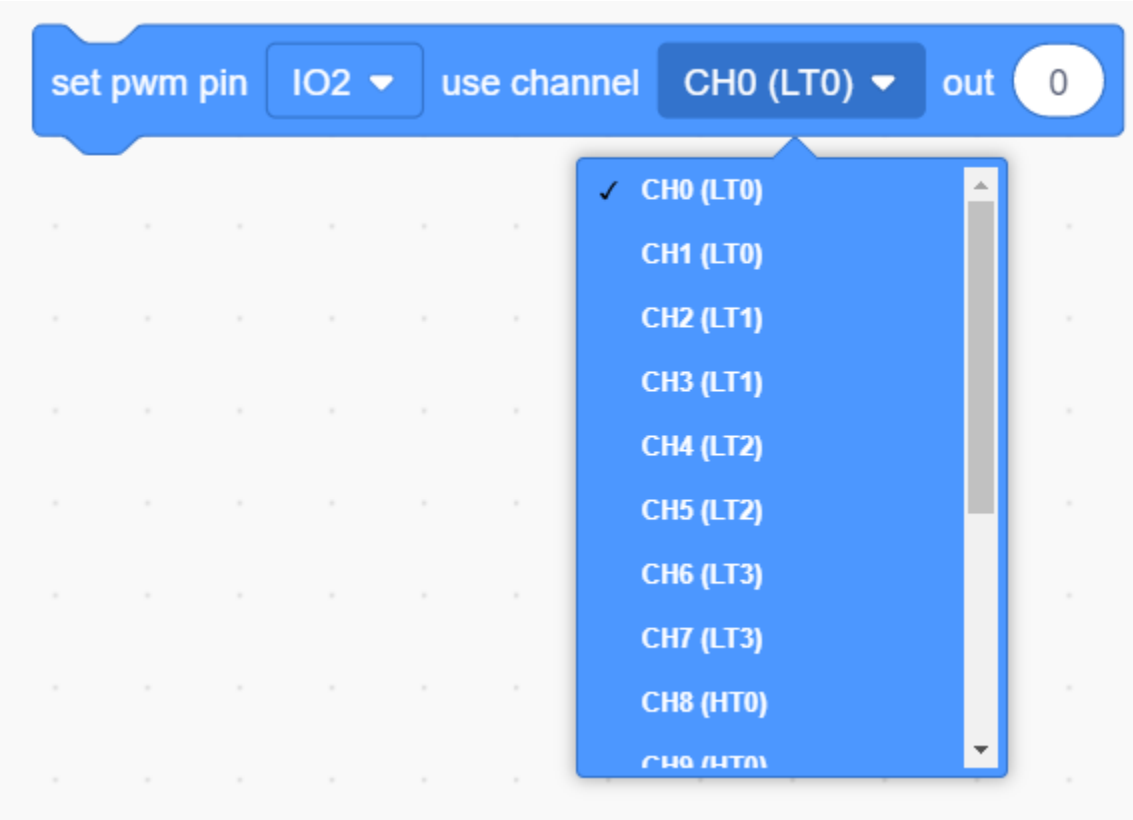


Set the variable item mode to increase item by 1 or decrease item by 1 every time the loop is executed.

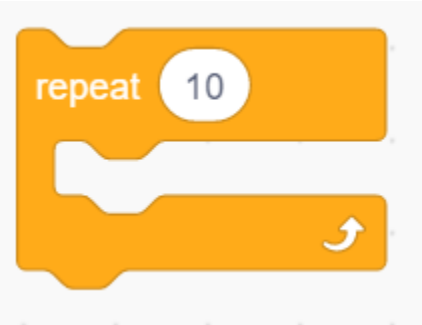


Set the string variable item.





This block is used to set the PWM. You need to set the corresponding pin via the channel (a total of 16 channels (0~15)) and the output value, so that the PWM value can be output.

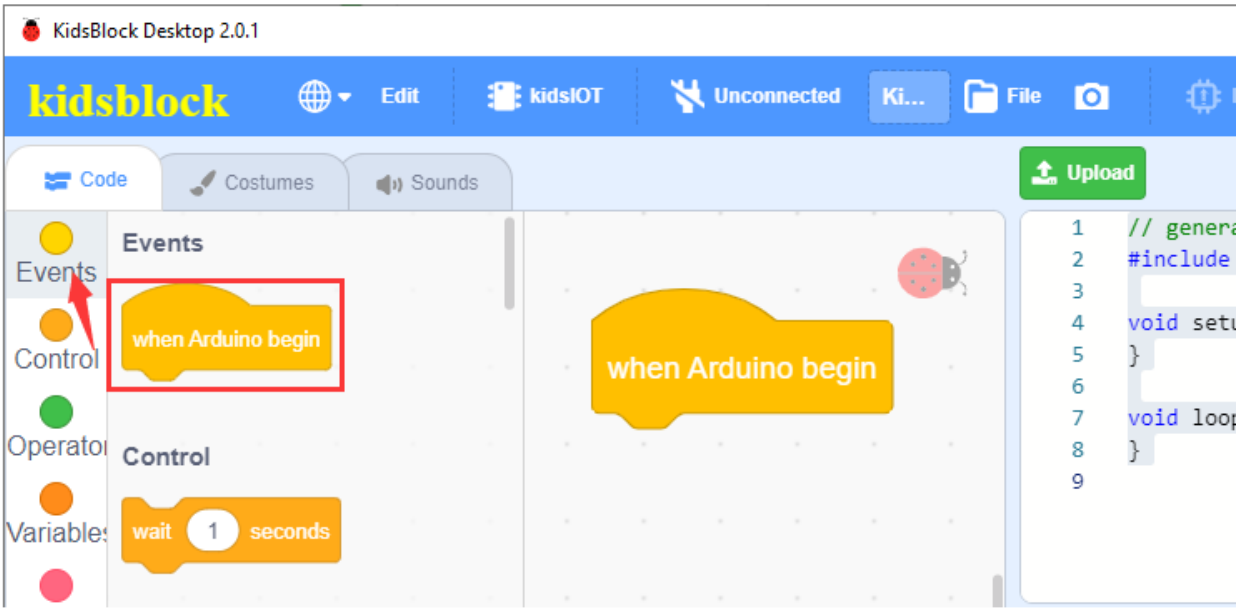


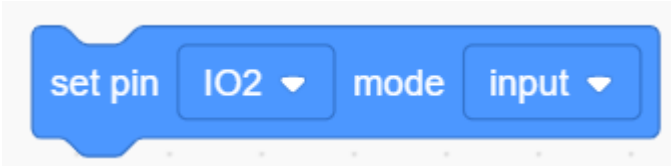
This is a conditional loop control statement that exit the loop when the number of loops is met. For example: 10 means that the loop is executed 10 times. The number 10 can be changed to other numbers.

Step 2Write the Program

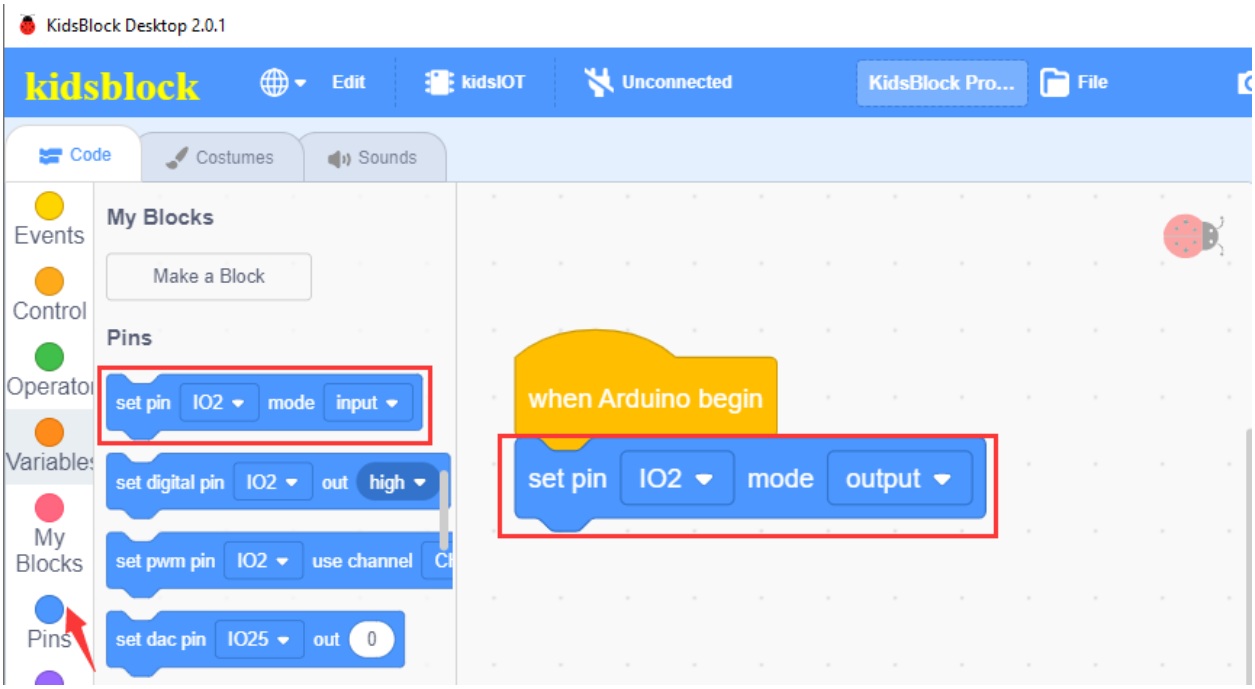
Drag the instruction block 

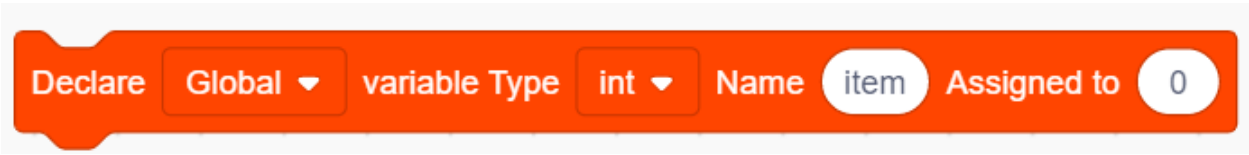
in the **Events** module to the script area.



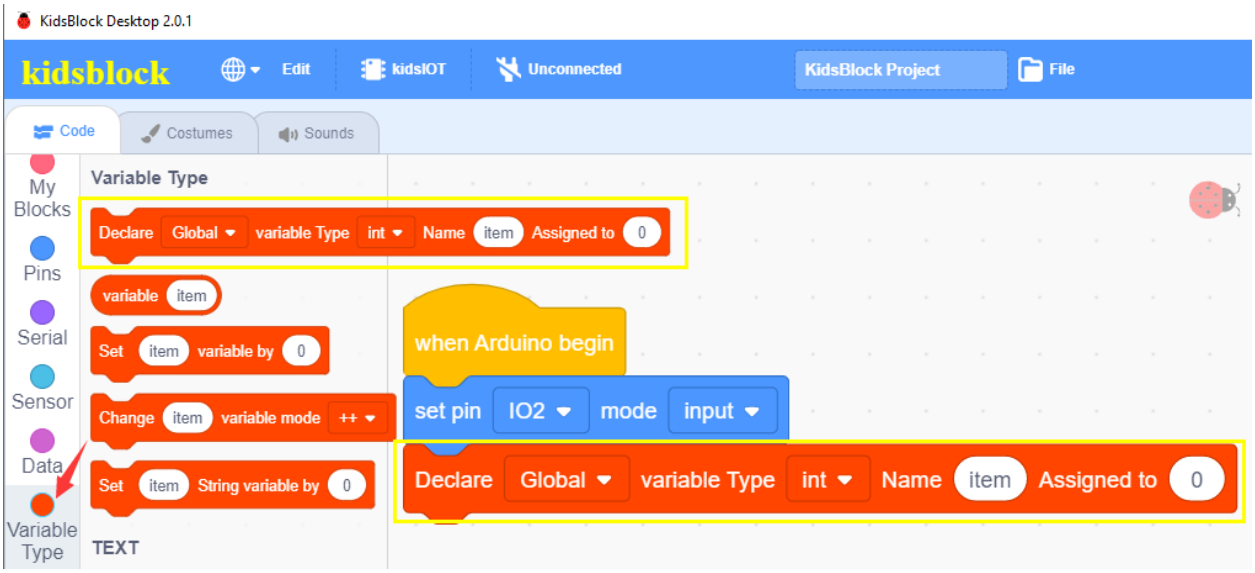
Drag the instruction block 

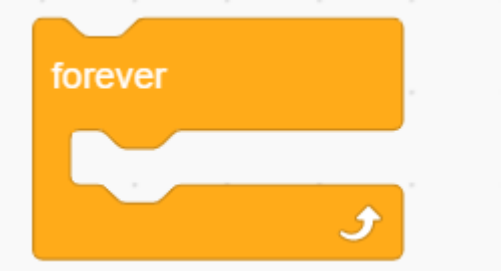
in the "**Pin**" module to the script area, change "**input**" to "**output**".

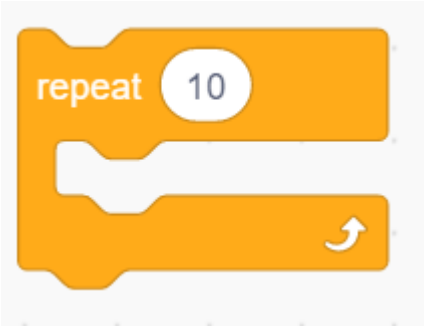


Drag the block 

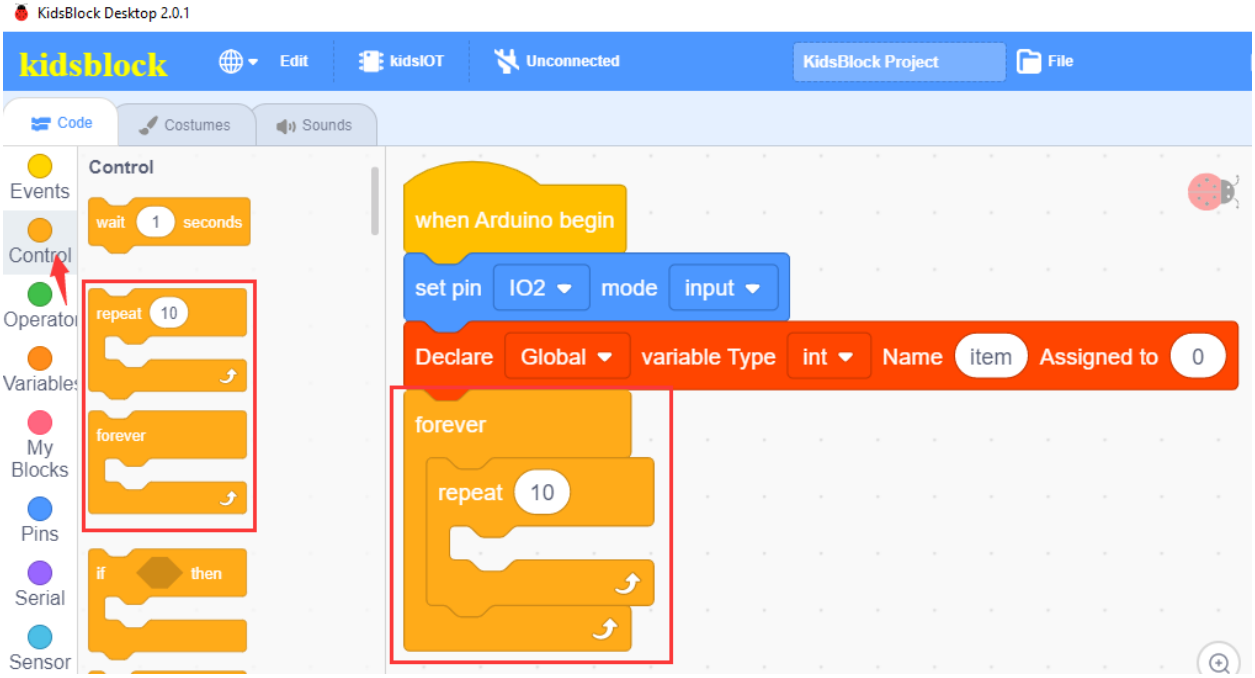
in the "**Variable Type**" module to the script area.

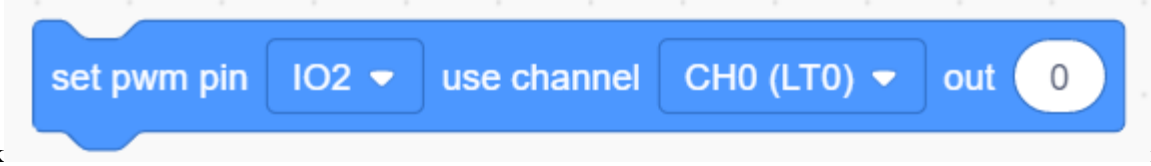


Drag blocks 

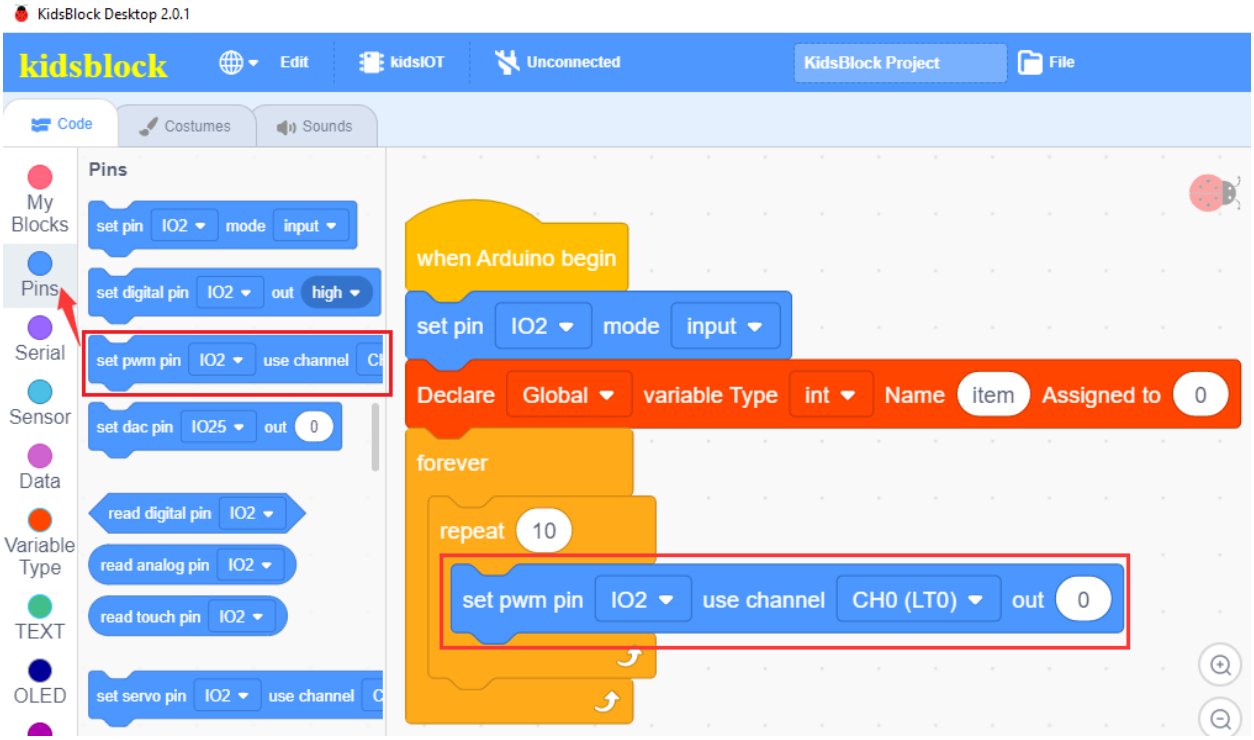
and 

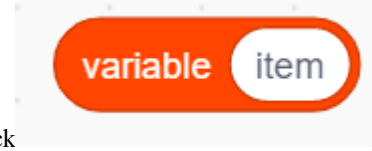
in the "**Control**" module to the script area.



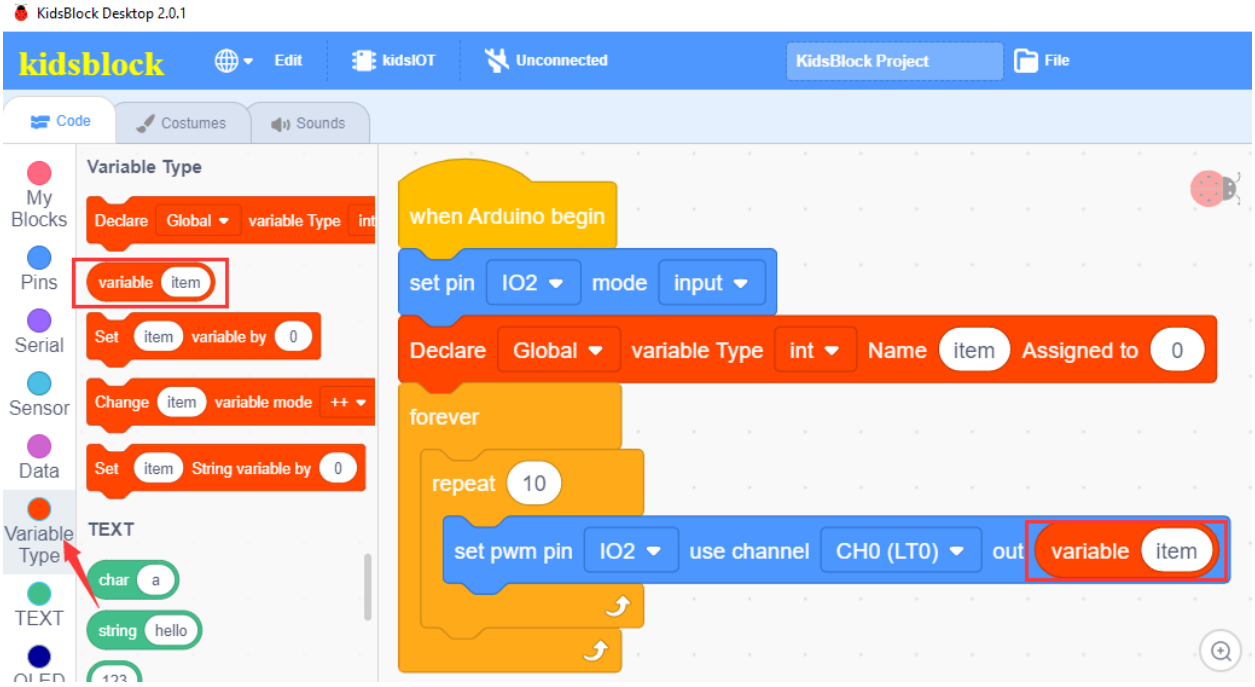
Drag the instruction block 

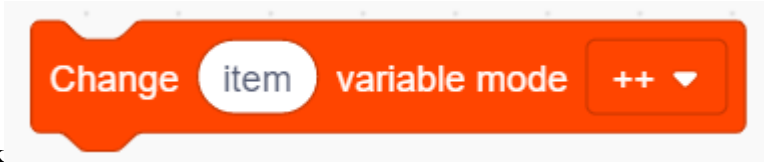
in the "**Pin**" module to the script area.



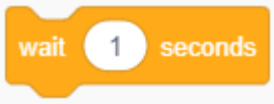
Drag the block 

in the "**Variable Type**" module to the script area.

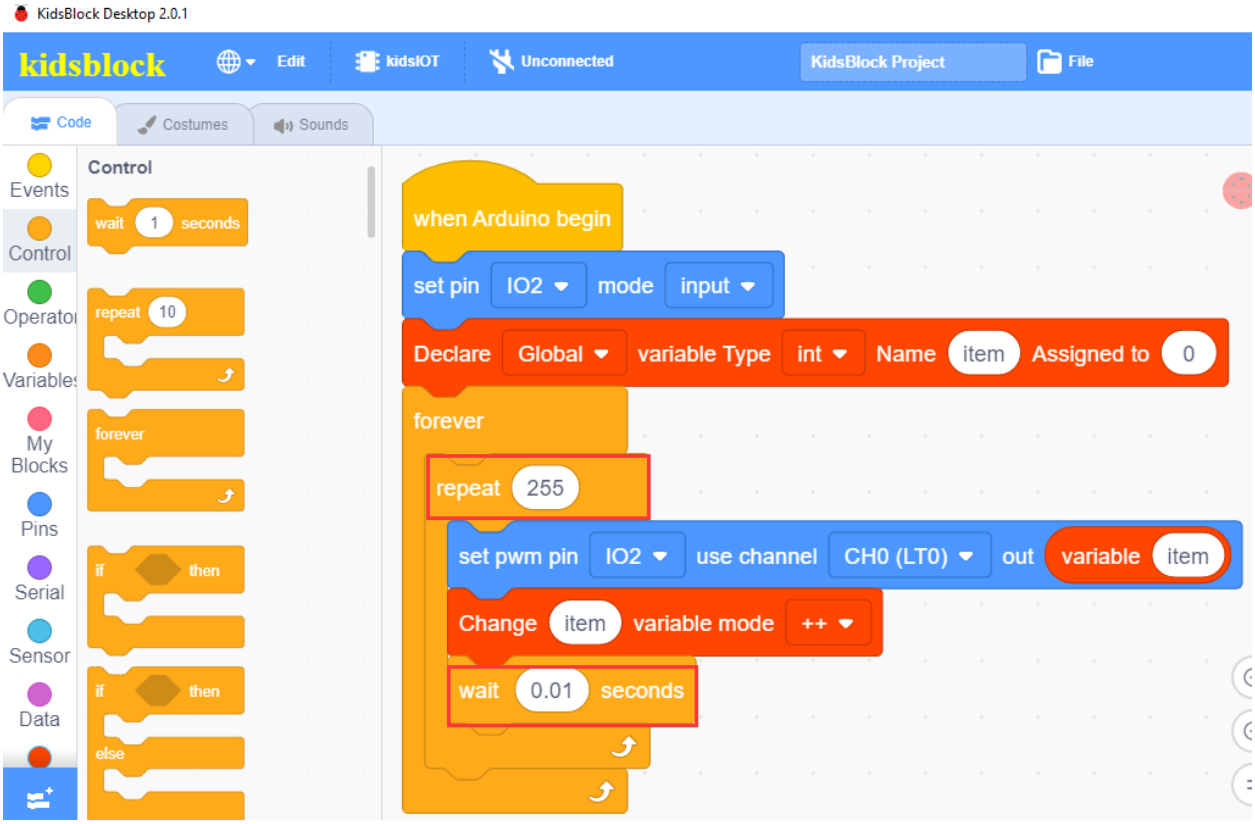


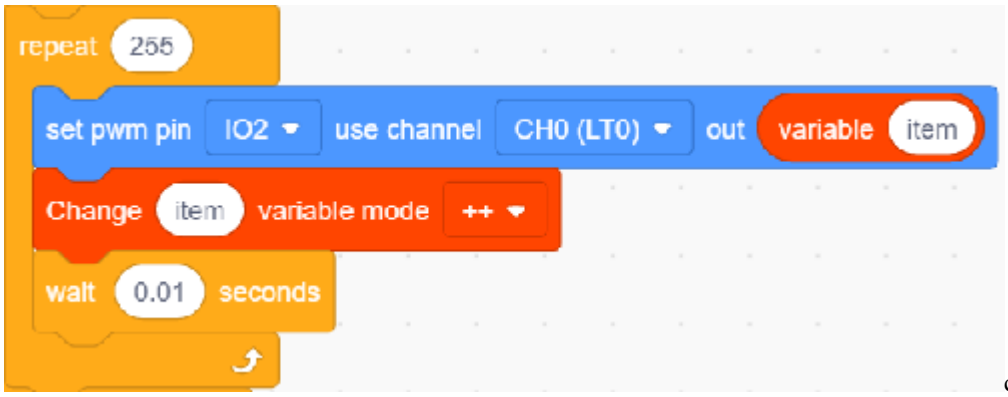


Drag the block in the "**Variable Type**" module to the script area"**++**" means that each time the loop is executed, **item** will be increased by **1**.

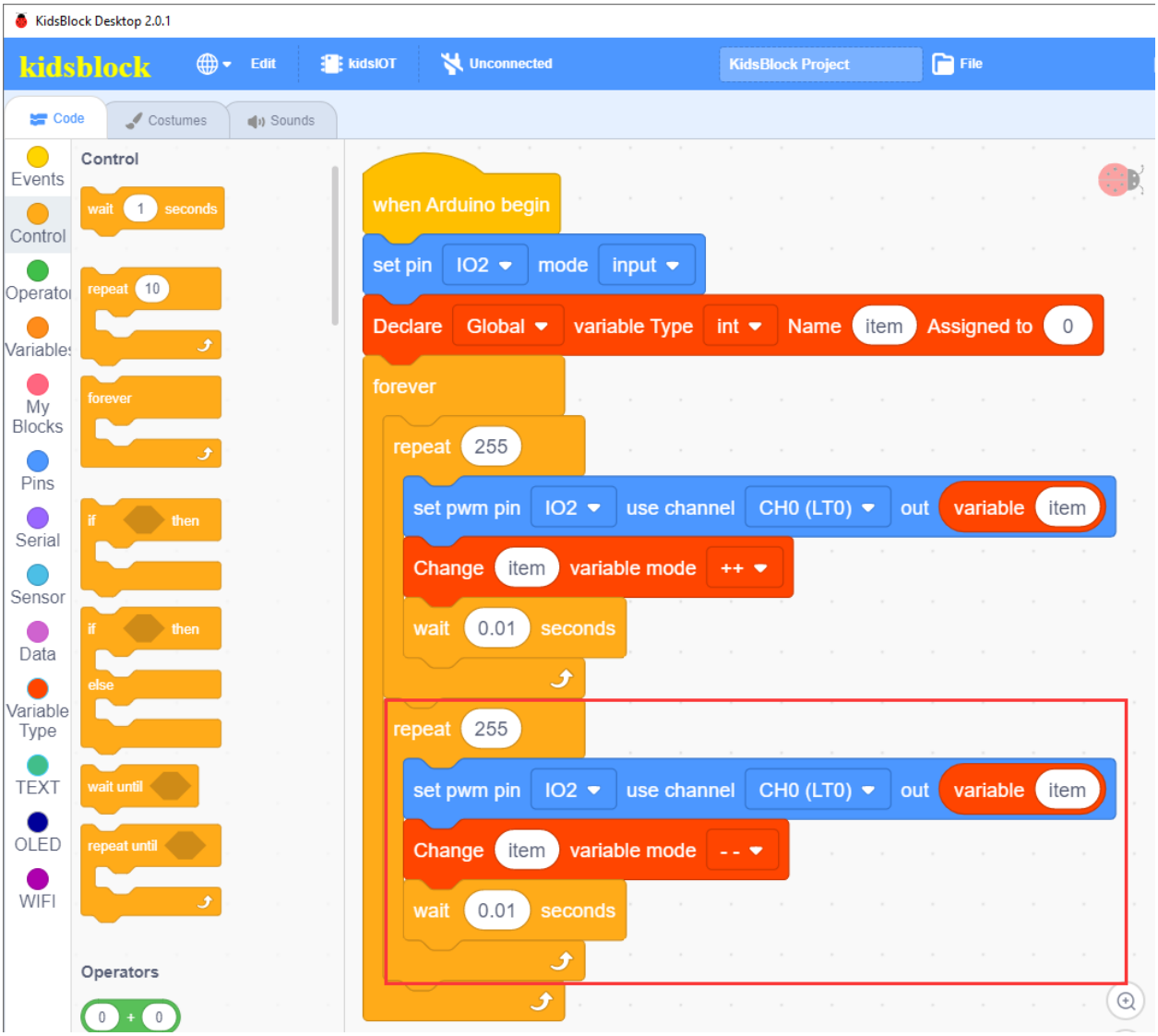
Drag the block 

in the "**Control**" module to the script area and set the delay to **0.01** second, the repeat **10** to **255**, for the corresponding PWM code block outputs 0~255. In this way, the LED light will slowly turn from dark to bright.

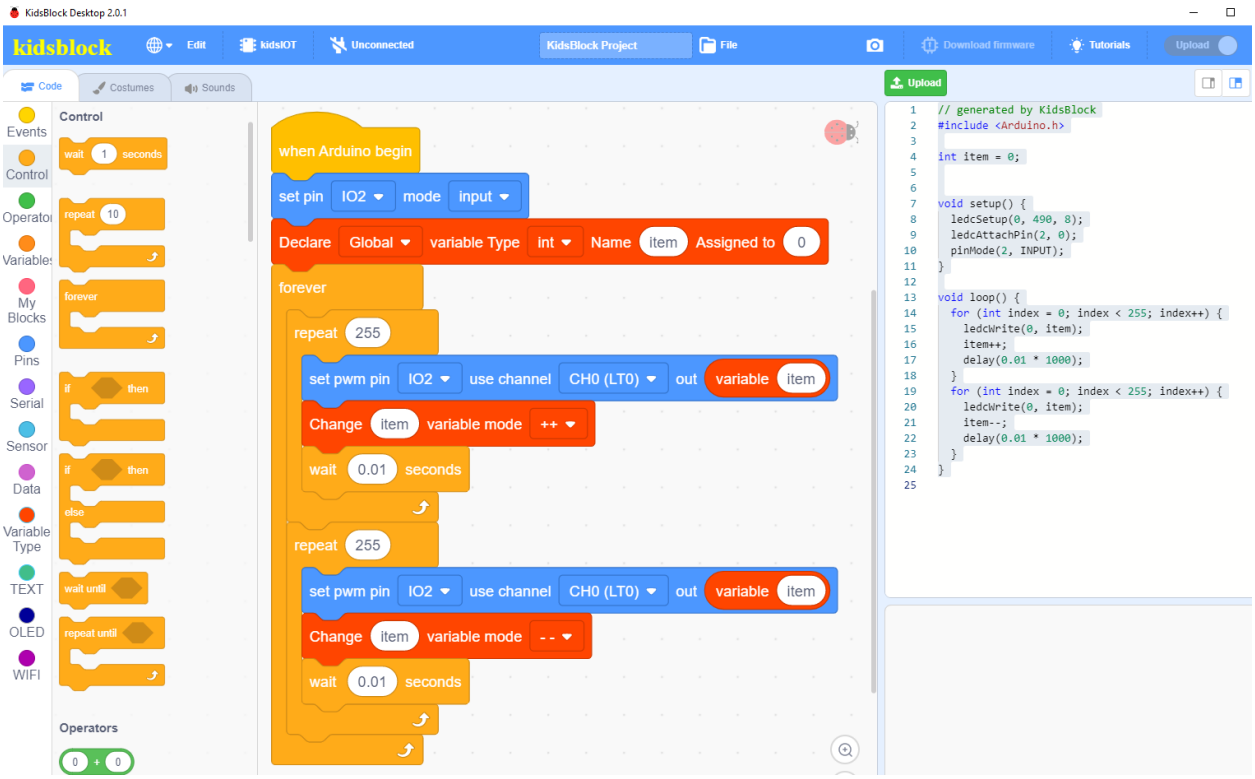


Copy the code block 

change "++" to "–", then LED will slowly turn from bright to dark.



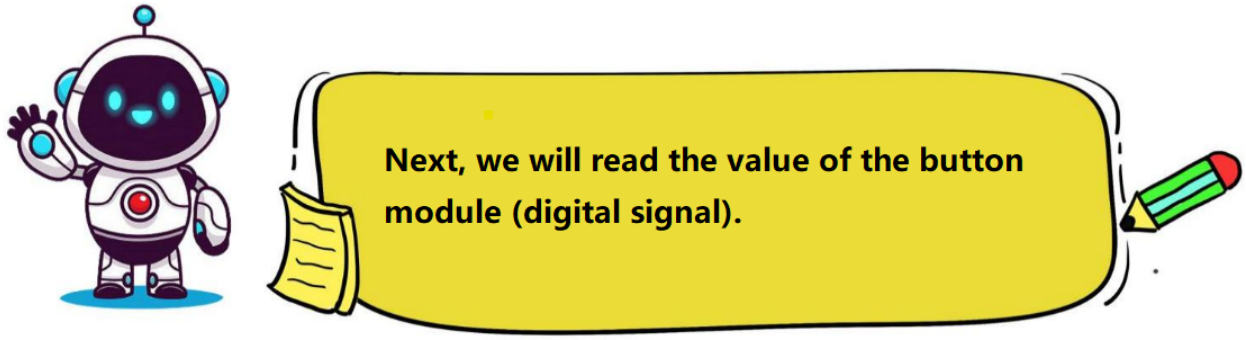
Complete Program



(3). Test Result

Click  to upload the above complete code to the kidsIOT motherboard. After powering up via the USB cable, the LED will gradually brighten and then dim, like breathing.

7. Read the value of the button module



(1). Knowledge

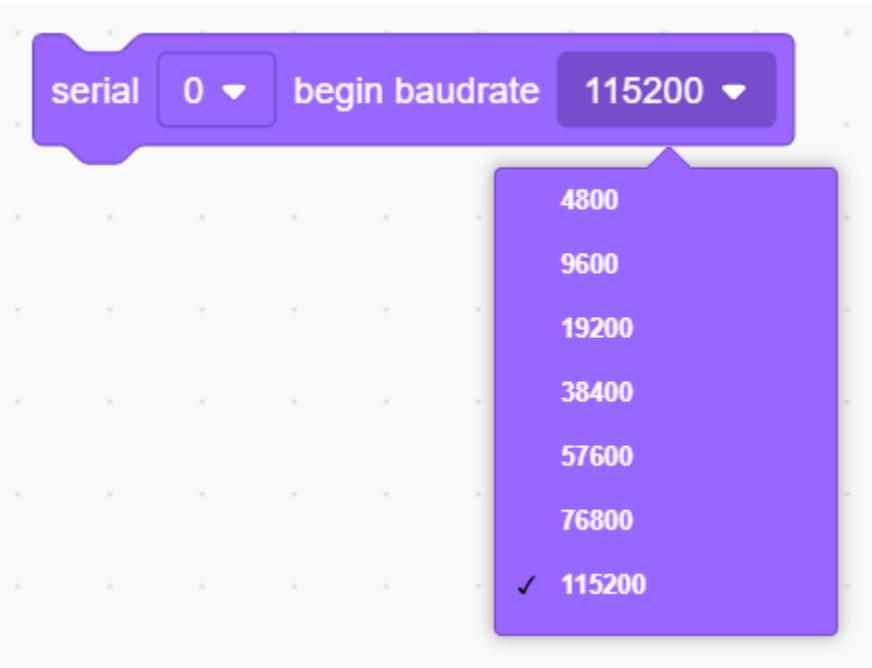
The principle of the button module is based on the switch circuit.

When the button is pressed, the switch closes, allowing current to pass through the button to GND, then the digital input pin of the kidsIOT motherboard detects a low level signal.

When not pressed, the switch is in the off state, the pin is pulled high by the pull-up resistor, and the digital input pin detects a high-level signal.

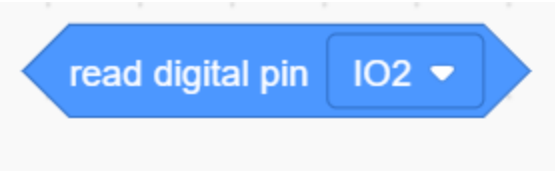
(2). Programming Steps

Step 1Description of the Building Blocks



The block is used to set serial baud rate(generally, the baud rate 9600 is taken as an example)

This block is used to set print mode for the serial port. **warp** means line feed printing, **no-warp** means no line feed printing, **HEX** means hexadecimal printing.

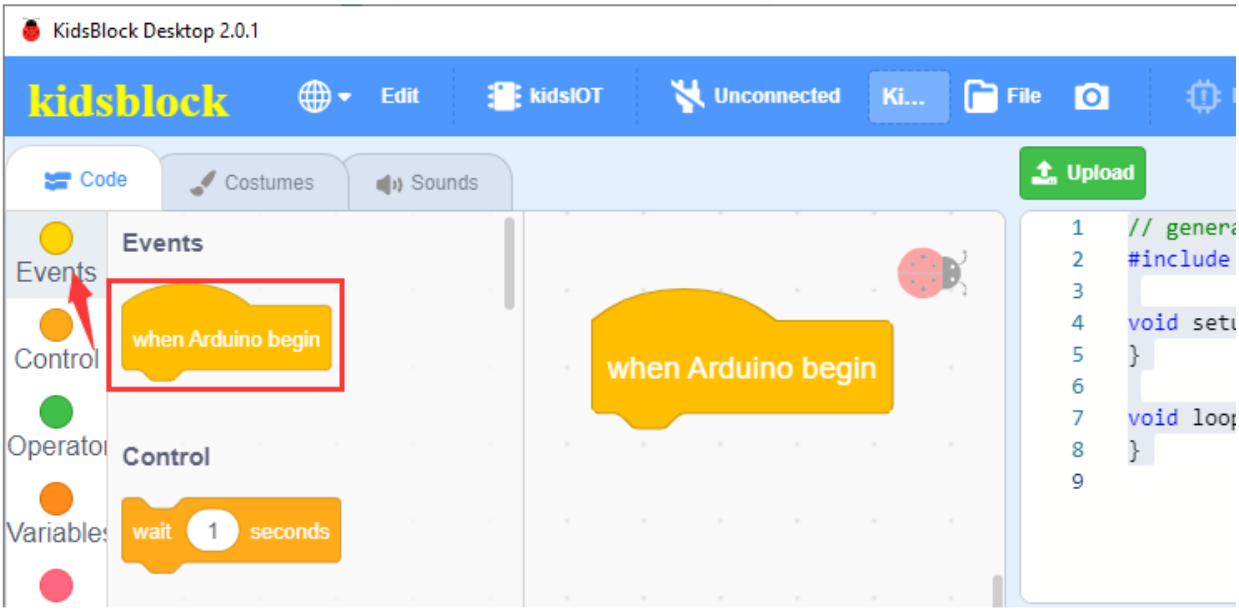


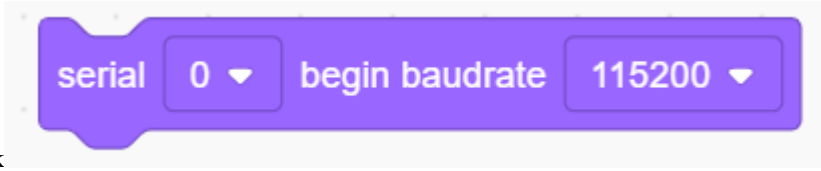
It is used to read the digital signal value of the specified pin0 or 1).

Step 2Write the Program

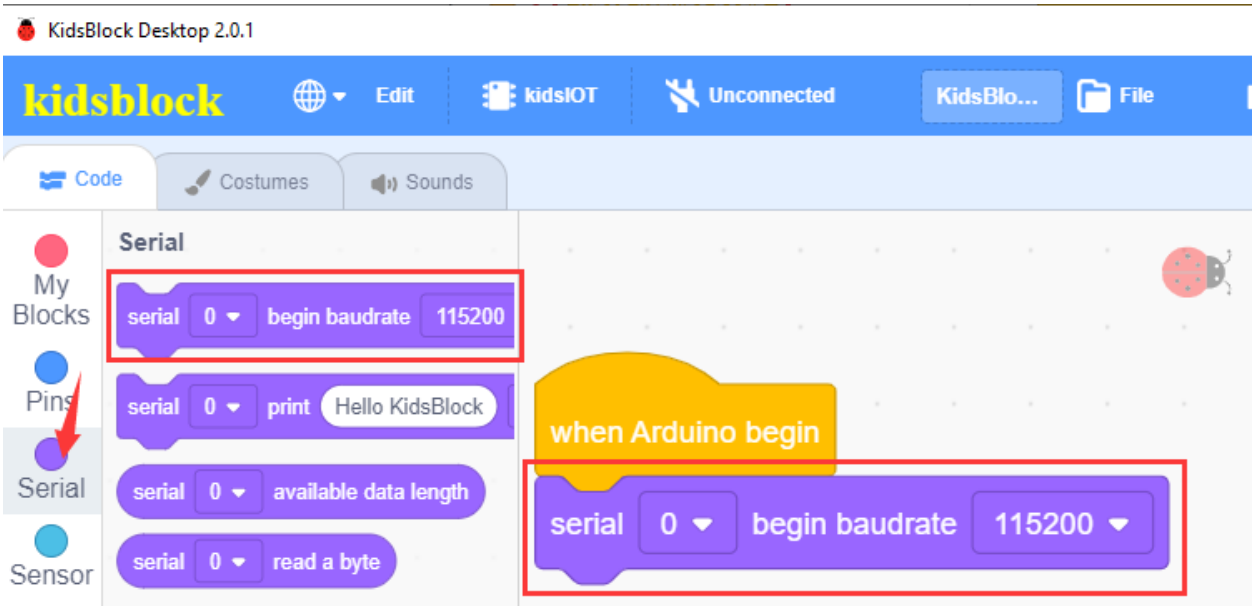
Drag the instruction block 

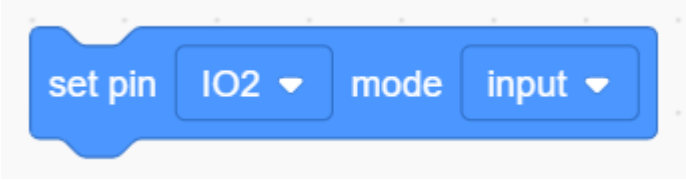
in the **Events** module to the script area.



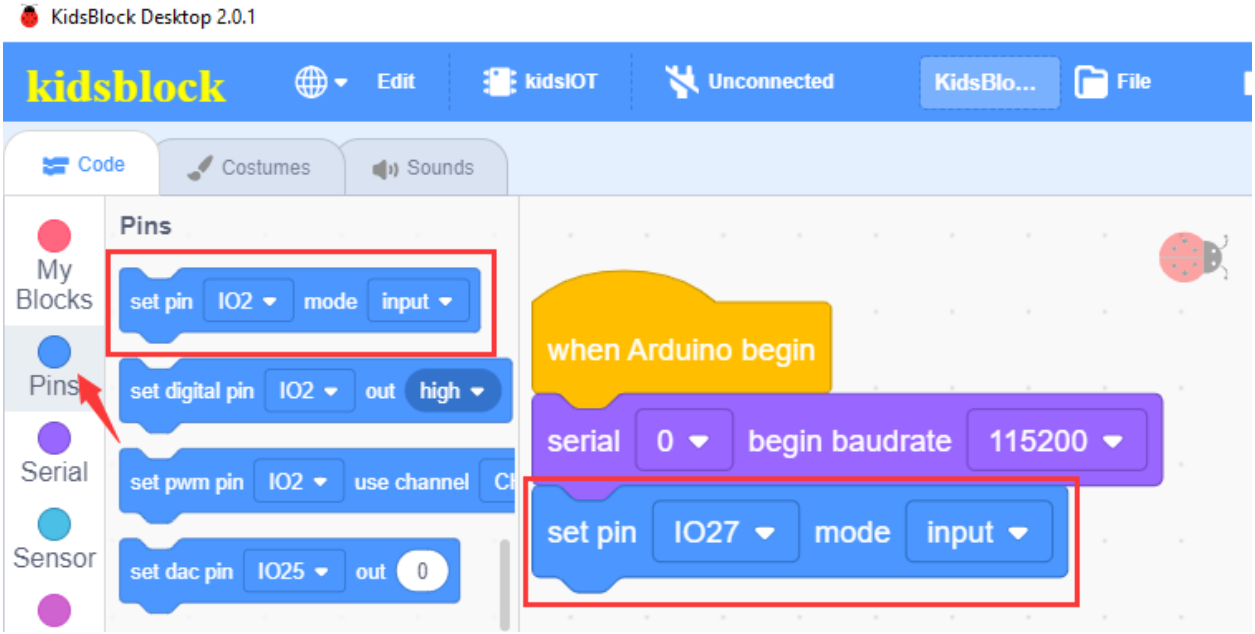
Drag the instruction block 

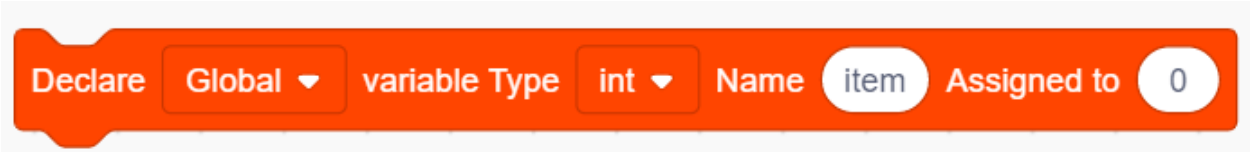
in the **Serial** module to the script area and take the baud rate 15200 as an example.



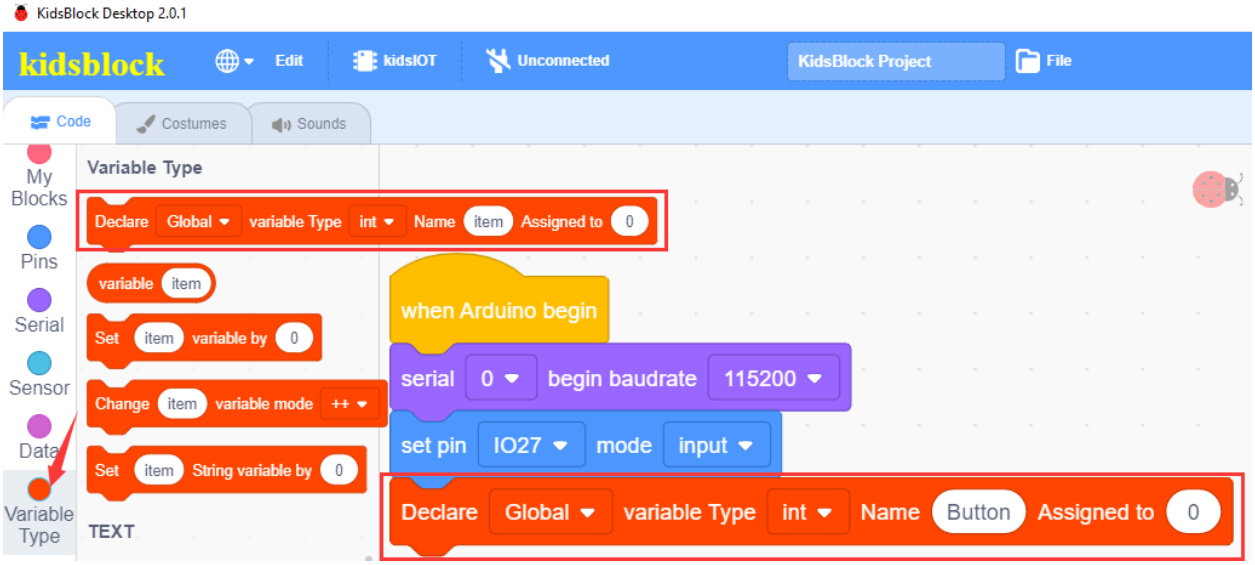
Drag the block 

in the "**Pin**" module to the script area. Since the button module is connected to No. 4 port (the control pin is io27) , so change pin IO2 to IO27.



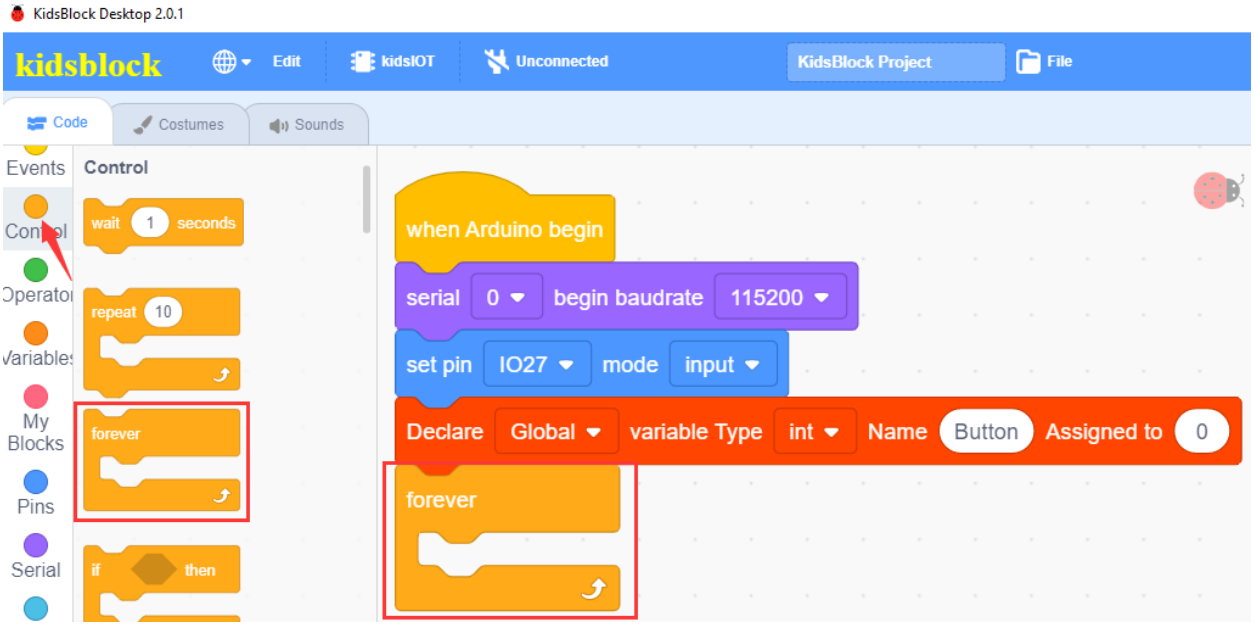
Drag the block 

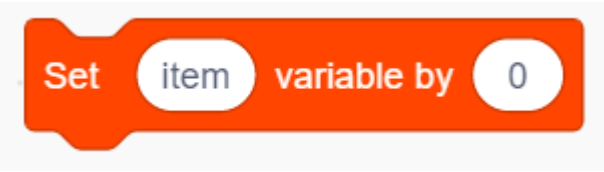
in the "**Variable Type**" module to the script area, then change item to "Button".



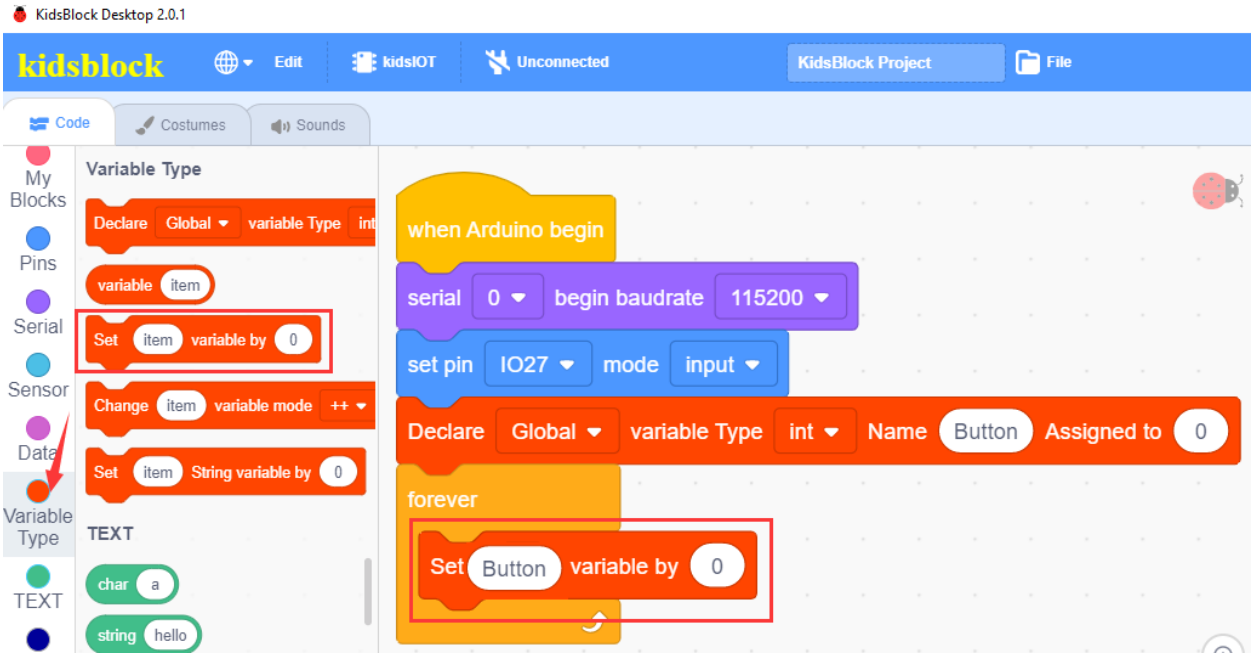
Drag the block 

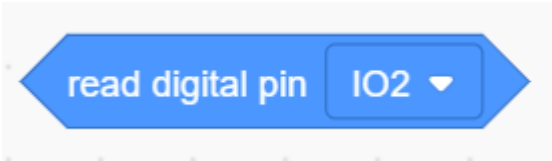
in the "**Control**" module to the script area.



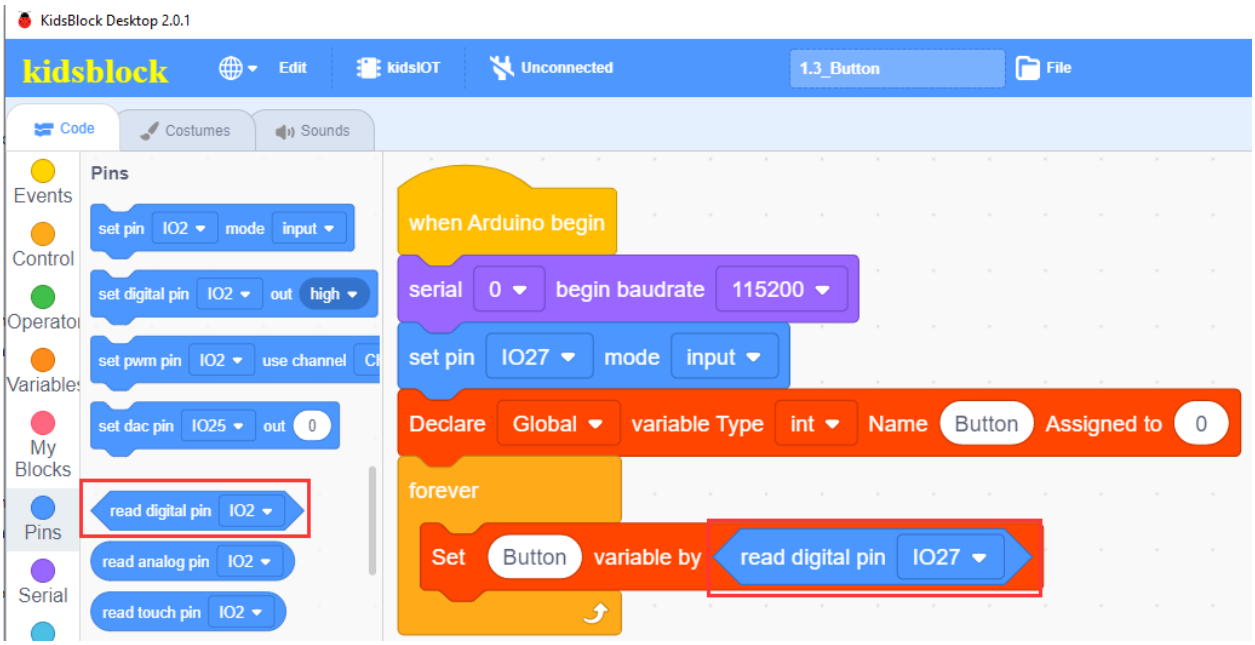
Drag the block 

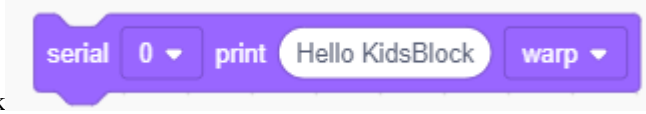
in the "**Variable Type**" module to the script area, then change item to "Button".



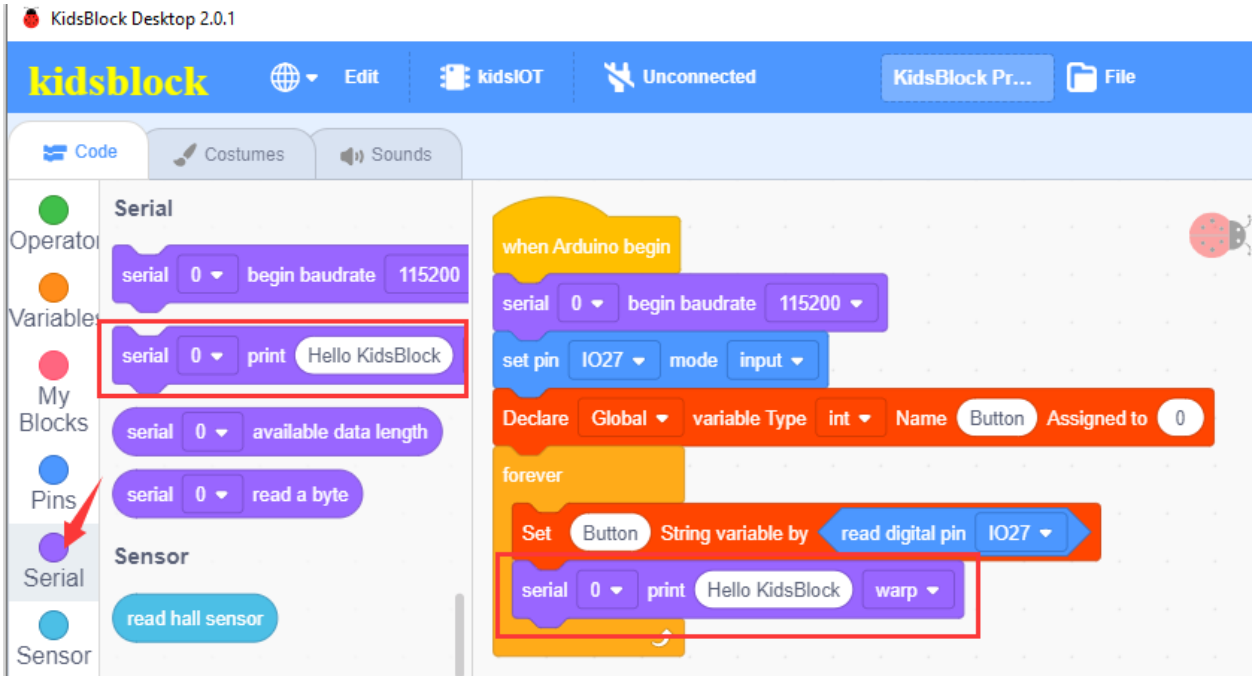
Drag the block 

in the "**Pin**" module to the script area, then change pin IO2 to IO27.



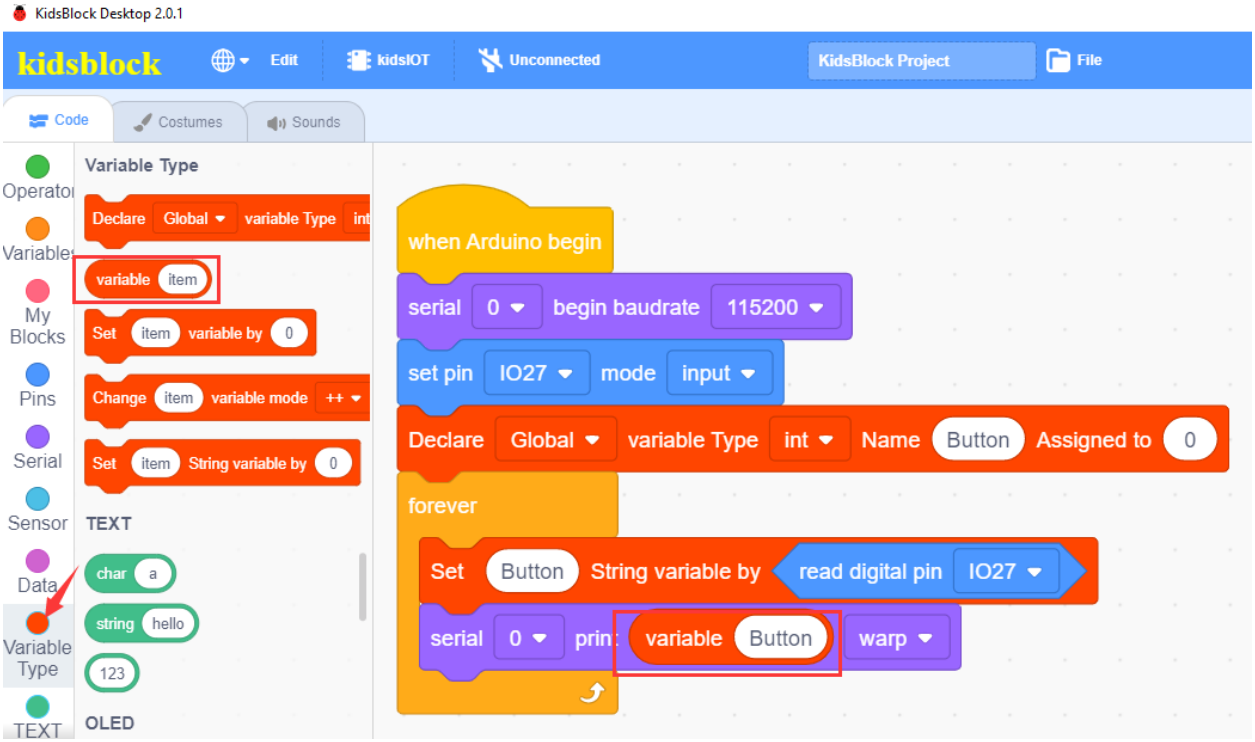
Drag the instruction block 

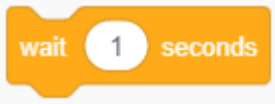
in the **Serial** module to the script area.



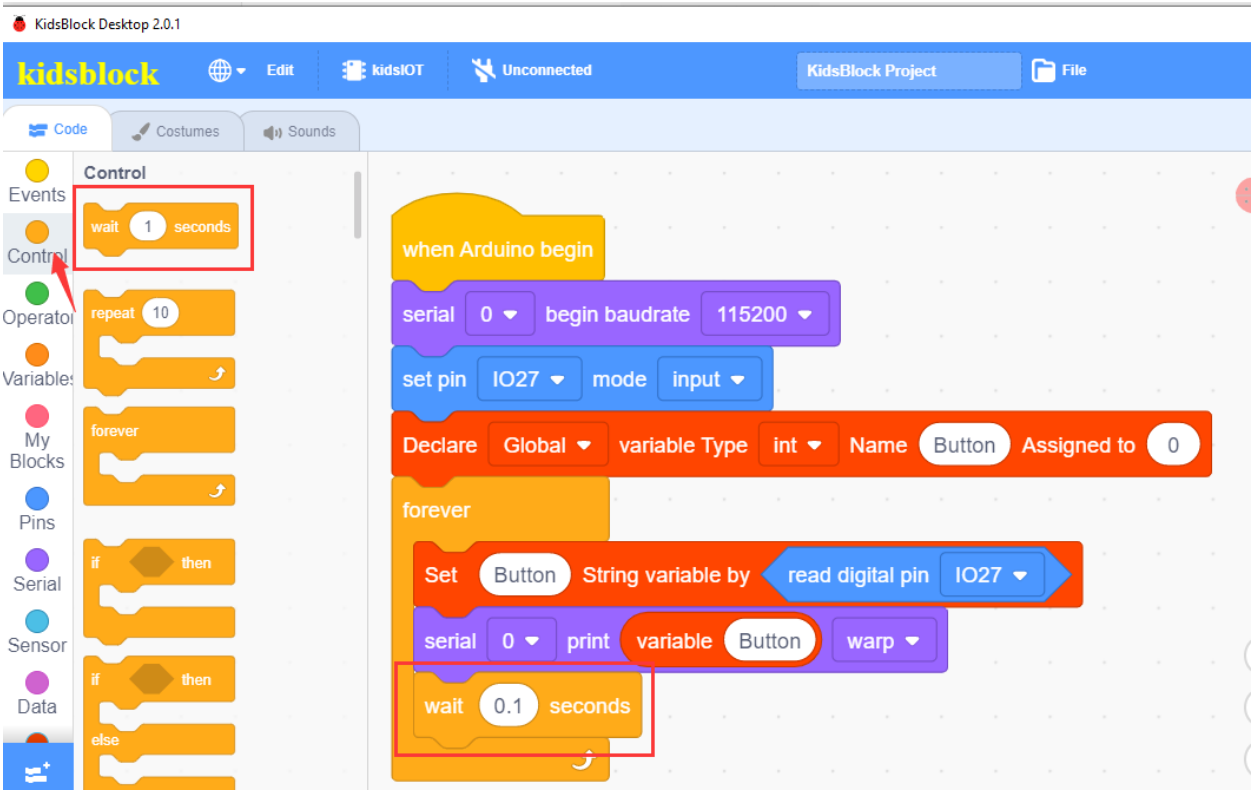
Drag the block 

in the "**Variable Type**" module to the script area, then change item to "Button".

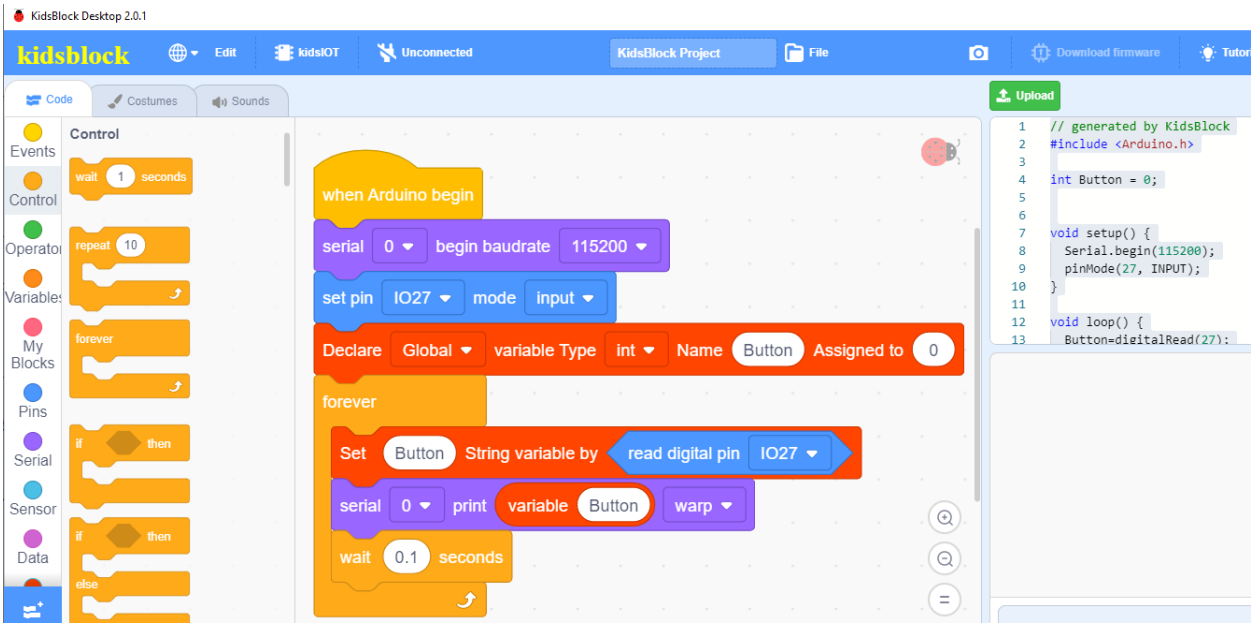


Drag the block 

in the "**Control**" module to the script area and set the delay to **0.1** second.



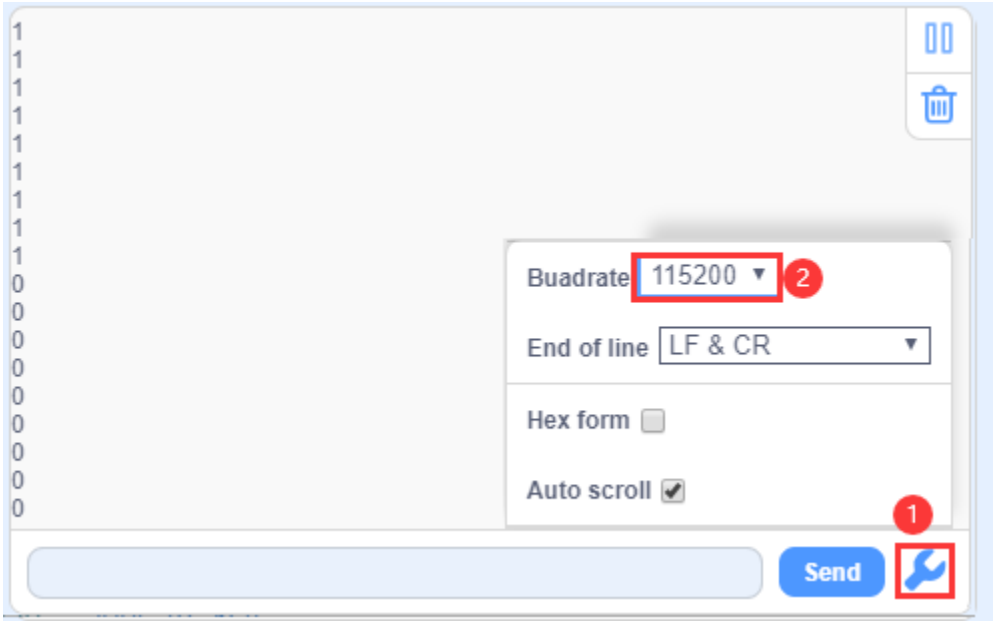
Complete Program



(3). Test Result

Click 

to upload the above complete code to the kidsIOT motherboard. After powering up via the USB cable, click in the serial monitor and set the baud rate to 15200. When the button is not pressed, the serial monitor prints the digital signal 1, when pressed, the digital signal 0 is printed.



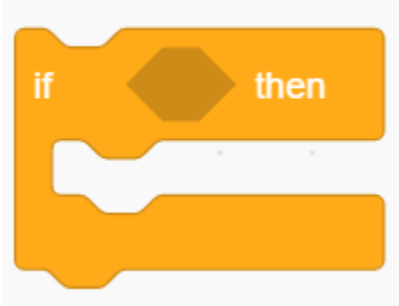
8. Self-locking button function

(1). Knowledge

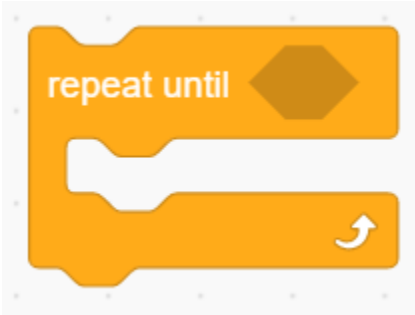
Self-locking button: It locks when pressed and will not automatically pop up. It will pop up only when pressed again, which is very similar to a switch. The switch can be turned on and off using self-locking buttons. If it is controlled by the kidsIOT mainboard, this operation can also be achieved via software.

(2). Programming Steps

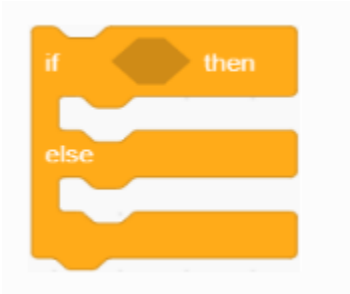
Step 1Description of the Building Blocks



It is a conditional statement code executing if-then function: If the logical judgment statement in  is satisfied, the code statement below **then** is executed.



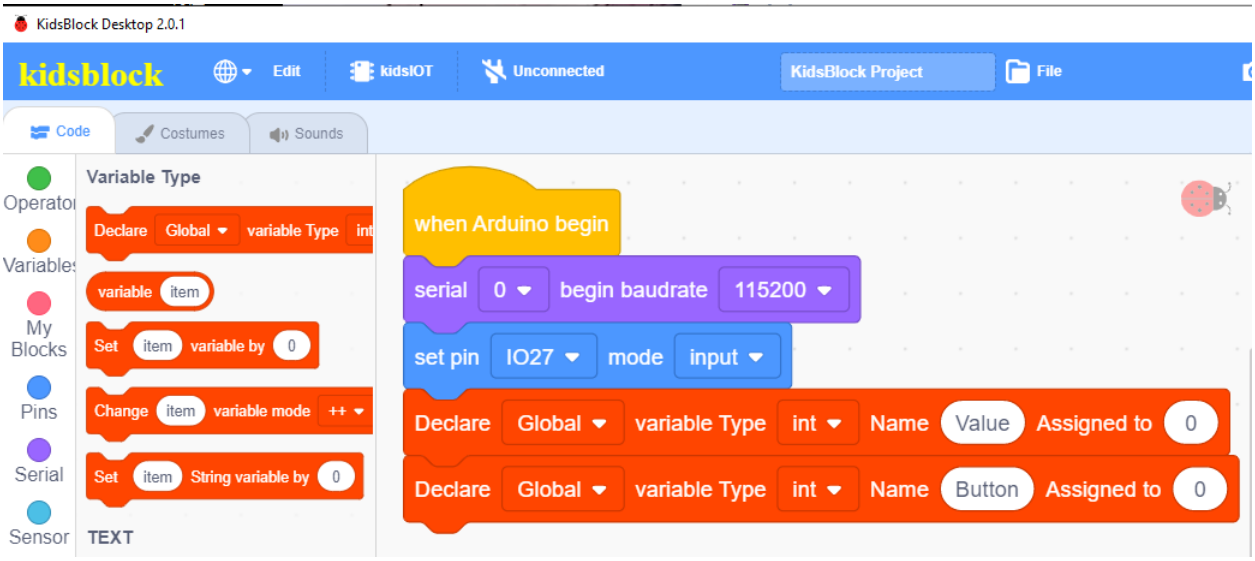
It is a conditional statement: If the logical judgment statement in  is satisfied, the loop will be executed continuously. If not, the loop will be terminated.



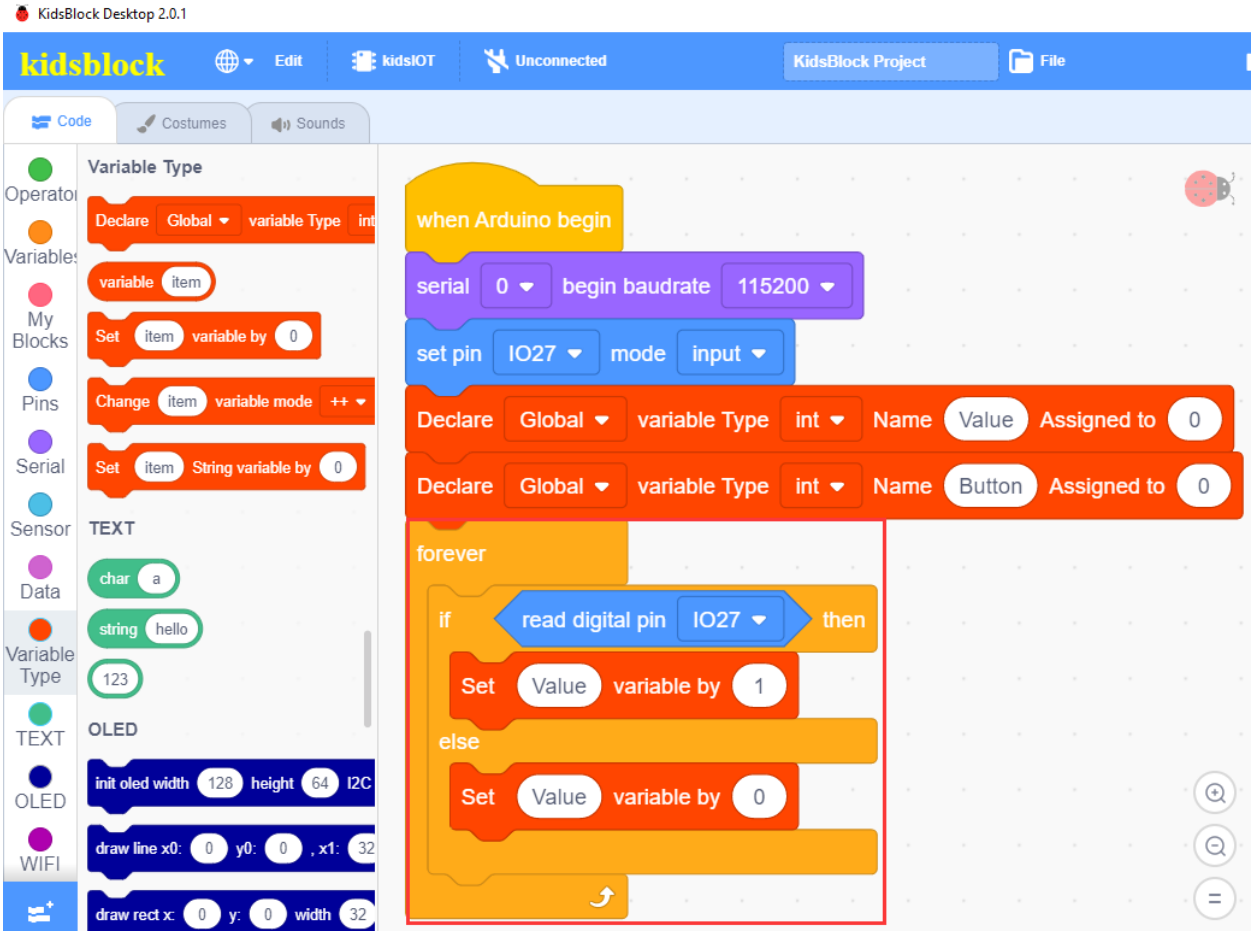
It is a conditional statement code executing if-then-else function: If the logical judgment statement in  is satisfied, the code statement below **then** is executed, otherwise, the code below **else** is executed.

Step 2Write the Program

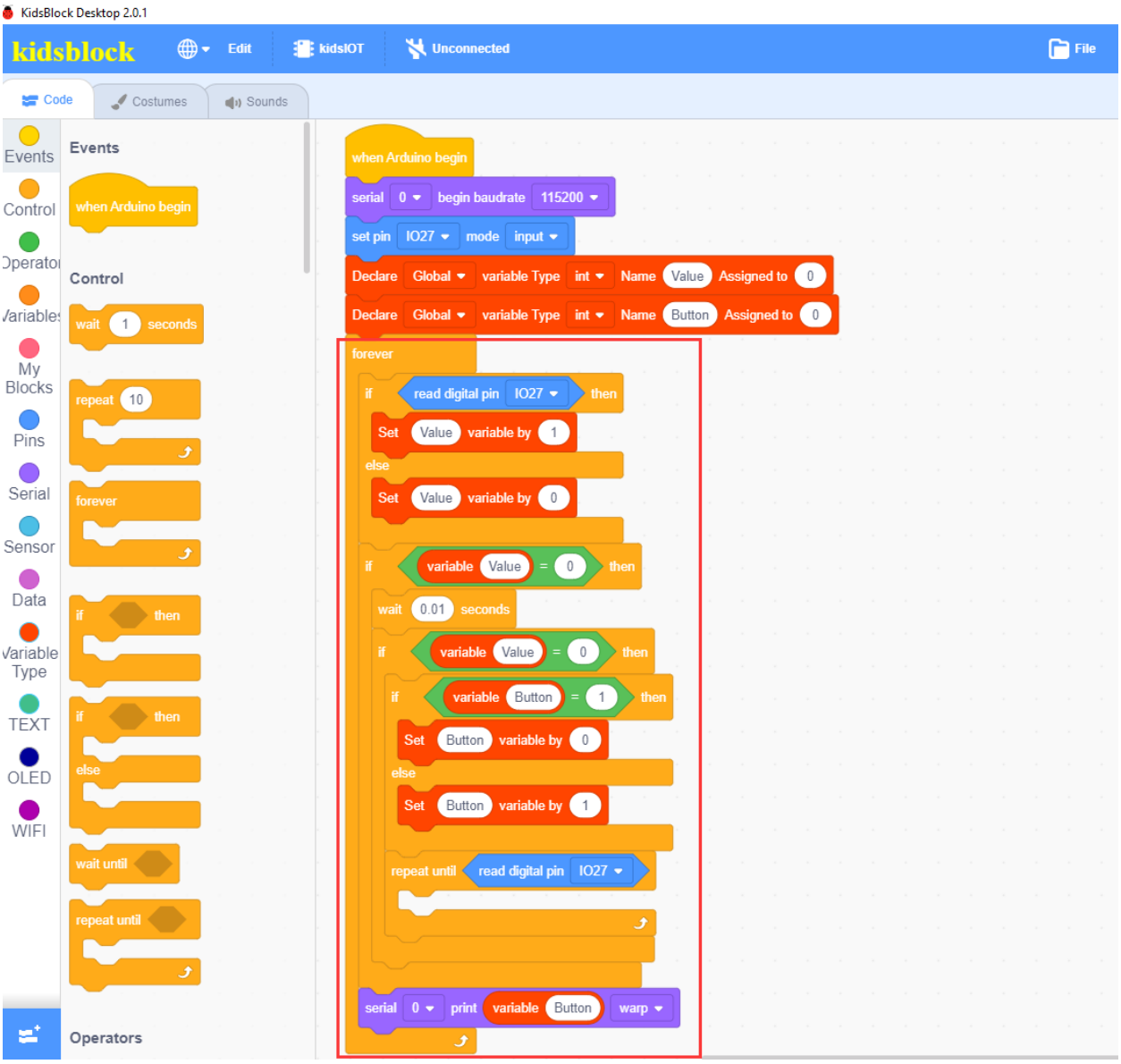
First define a Value to get the button status, and then define a Button. At the same time, select the serial baud rate to 15200 and the control button pin IO27 to "**input**" mode.



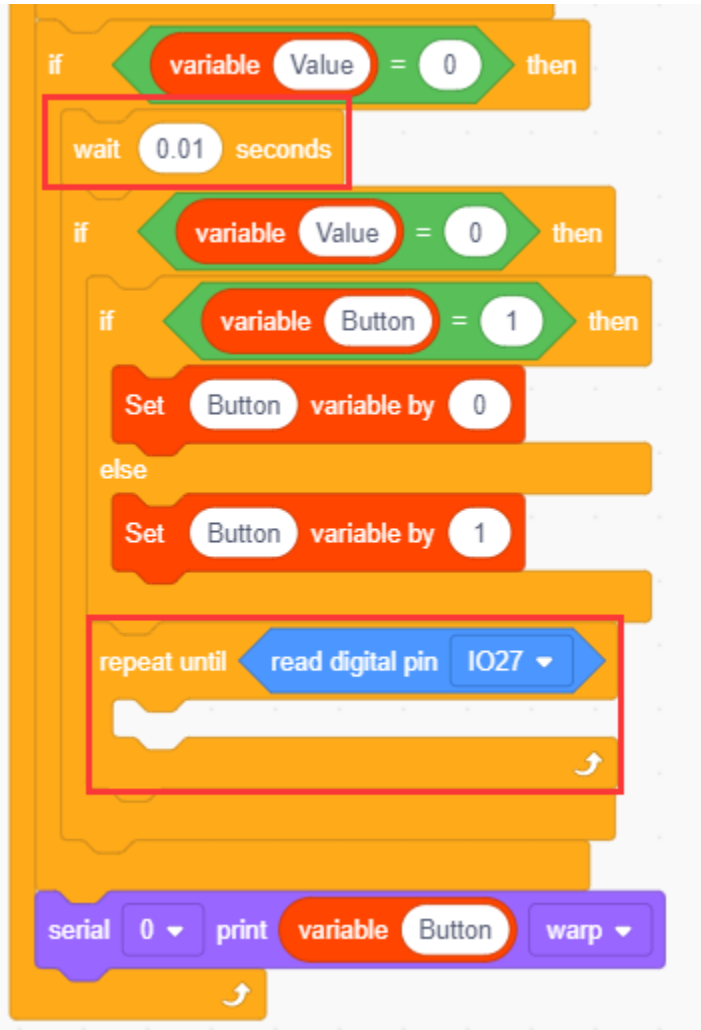
Assign the read button value to "Value".



Determine whether the button is pressed. When pressed, change the value of “Button” and print it.



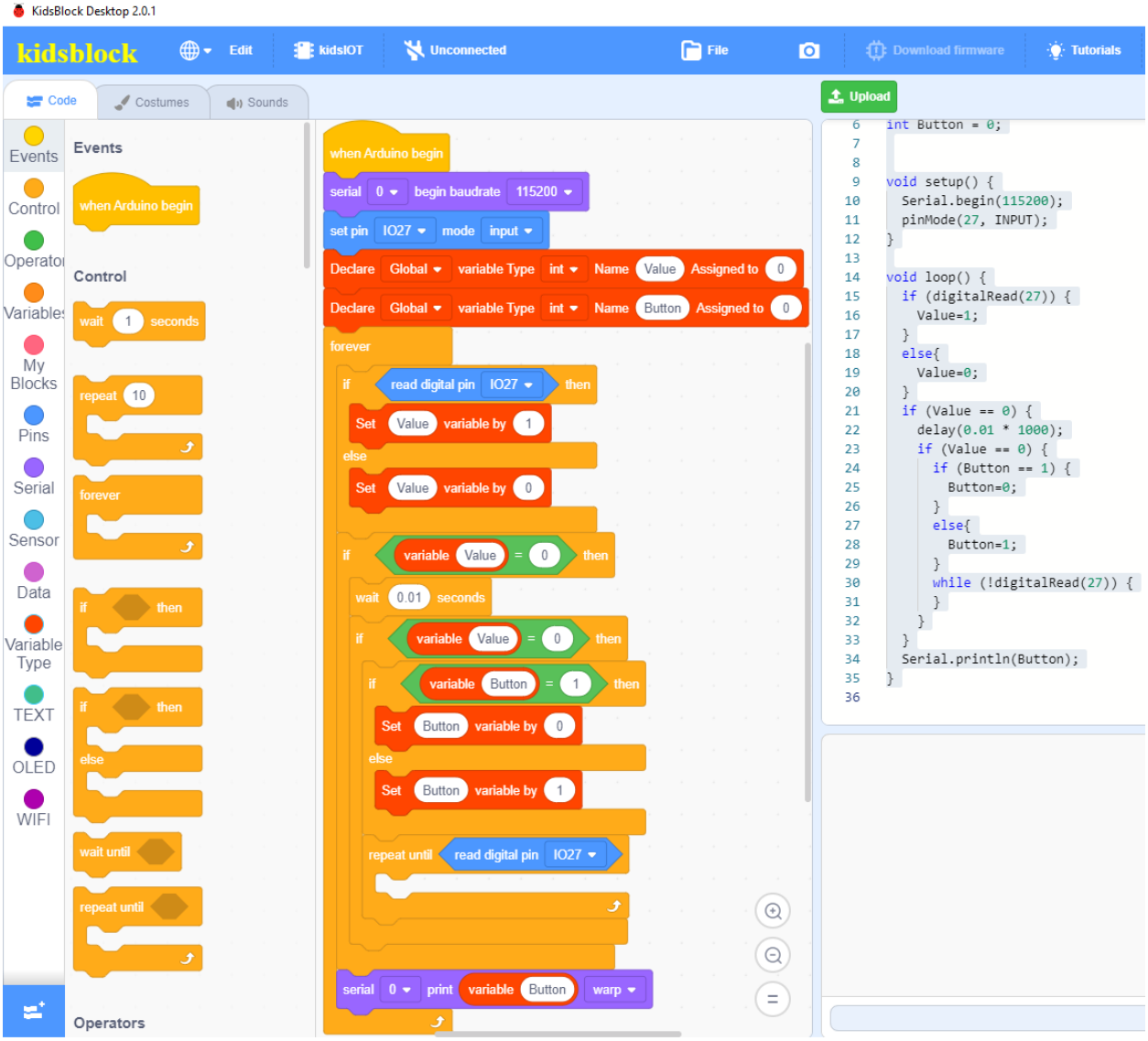
Description



The delay of 0.01 seconds is the button anti-shake function. After detecting that the button is closed, a delay program is executed, with a delay of 5ms 10ms (depending on the mechanical characteristics). After the jitter disappears, the button status is detected again. If the closed state level is still maintained, then a button is pressed.

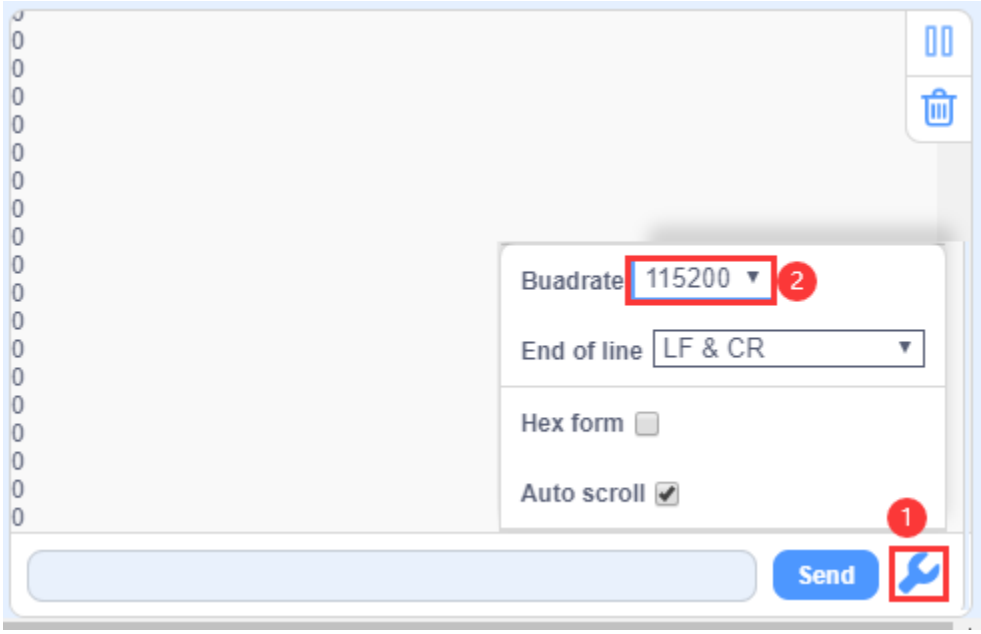
When it is released, a delay of 5ms 10ms is also required. Only after the jitter disappears can the button processing program be transferred. When the button is pressed once, the button becomes 1, when pressed again, it becomes 0.

Complete Program

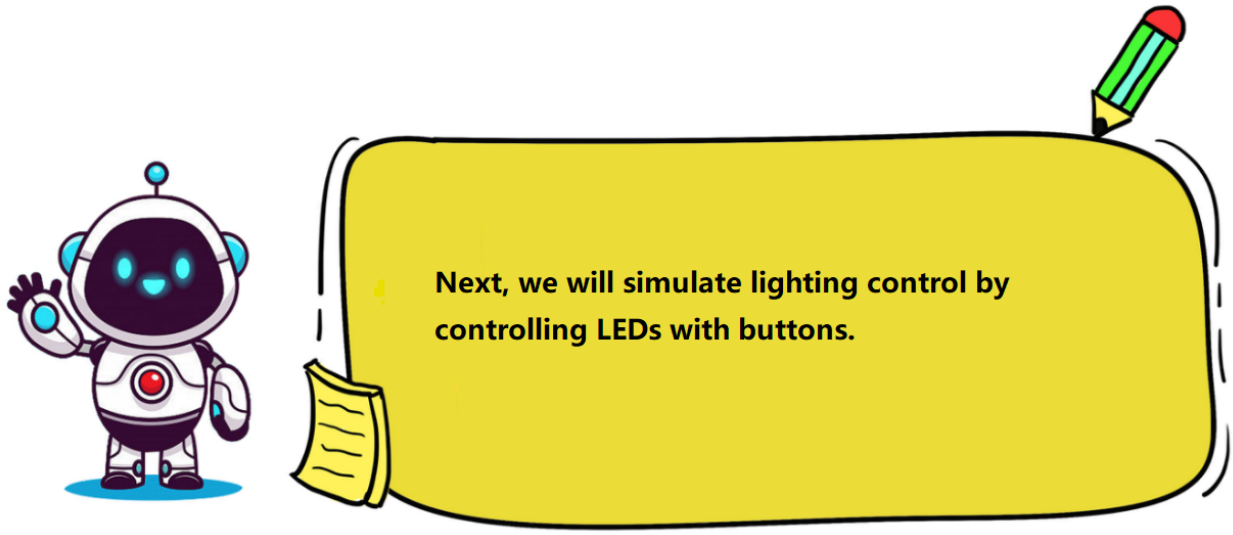


(3). Test Result

Click  to upload the above complete code to the kidsIOT motherboard. After powering up via the USB cable, click  in the serial monitor and set the baud rate to 15200. When the button is pressed, the serial monitor prints the number 1, and when pressed again, the monitor prints the number 0, so as to achieve the self-locking function of buttons.

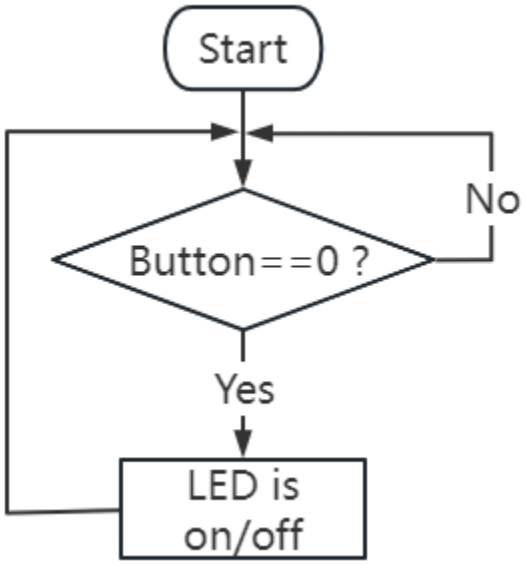


9. Lighting control system



(1). Programming Steps

Step 1Flow Chart

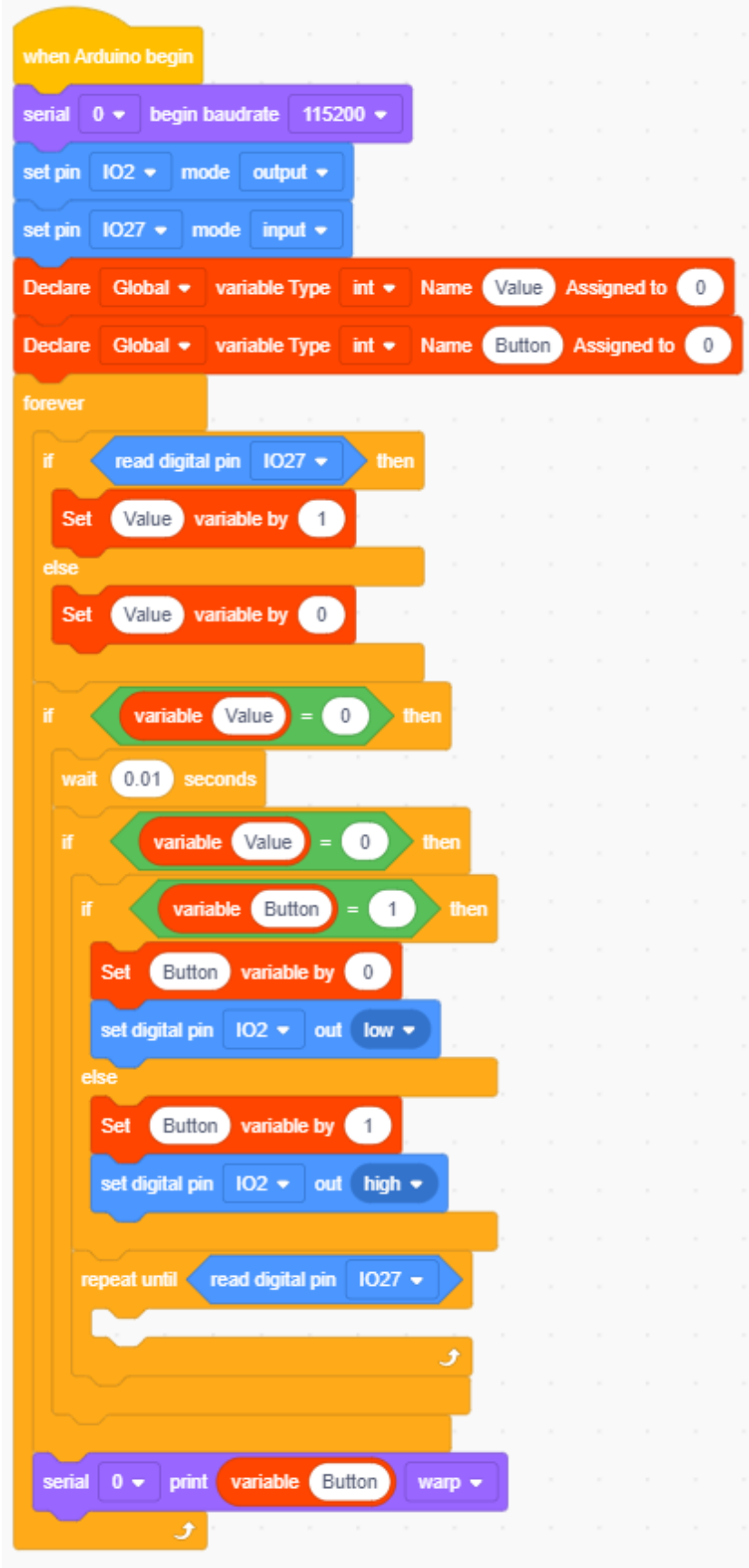


Step 2Write the Program

Based on the code of the self-locking button above, add the relevant blocks for turning the LED on and off.


when Arduino begin
serial 0 begin baudrate 115200
set pin IO2 mode output
set pin IO27 mode input
Declare Global variable Type int Name Value Assigned to 0
Declare Global variable Type int Name Button Assigned to 0
forever
if read digital pin IO27 then
Set Value variable by 1
else
Set Value variable by 0
if variable Value = 0 then
wait 0.01 seconds
if variable Value = 0 then
if variable Button = 1 then
Set Button variable by 0
set digital pin IO2 out low
Set Button variable by 1
set digital pin IO2 out high
repeat until read digital pin IO27
serial 0 print variable Button warp

Complete Program


when Arduino begin
serial 0 begin baudrate 115200
set pin IO2 mode output
set pin IO27 mode input
Declare Global variable Type int Name Value Assigned to 0
Declare Global variable Type int Name Button Assigned to 0
forever
if read digital pin IO27 then
Set Value variable by 1
else
Set Value variable by 0
if variable Value = 0 then
wait 0.01 seconds
if variable Value = 0 then
if variable Button = 1 then
Set Button variable by 0
set digital pin IO2 out low
else
Set Button variable by 1
set digital pin IO2 out high
repeat until read digital pin IO27
serial 0 print variable Button warp

(2). Test Result

Click 

to upload the above complete code to the kidsIOT motherboard, then power up via the USB cable. When the button is pressed for the first time, the LED is turned on. When pressed for the second time, the LED is turned off. When pressed for the third time, the LED is turned on again…, which is consistent with the LED switch in real life.

10. Common Problems

Q1: There are errors in kidsIOT board programming program

A: Please check whether the board type is correct.

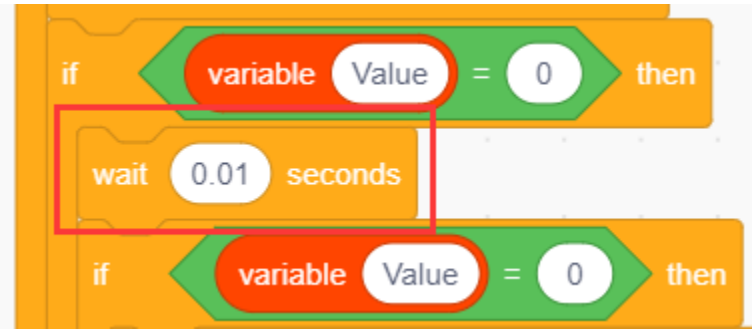
Please check whether the USB port number is selected correctly.

Q2: The LED does not light up after burning the code

A: Please confirm whether the pins in the code are consistent with the actual wiring. If there is an error, please wire correctly according to the pins in the code.

Q3: Are the buttons insensitive? Sometimes it can be detected, sometimes not?

A: Modify the button delay time and set it to an appropriate delay.



4.3.2 Project 02: Light Controlled System

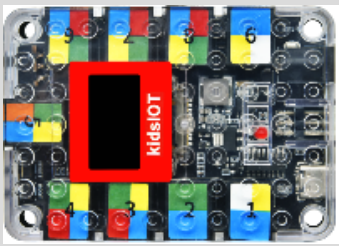
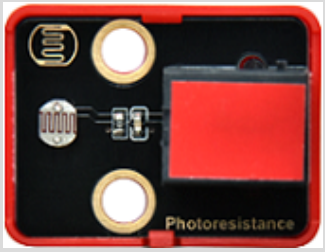
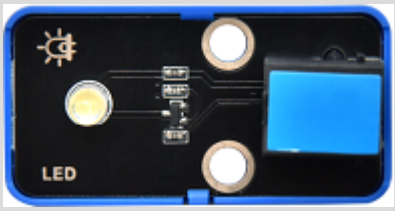
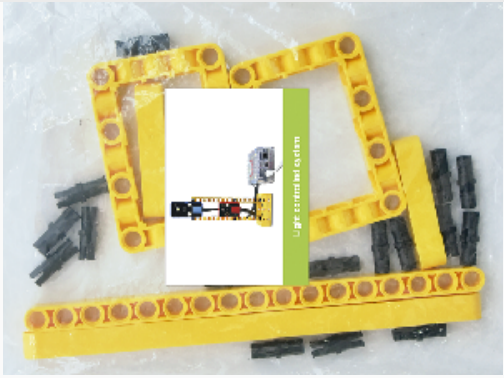
1. Description

This light controlled system is composed of a photoresisitor, a LED and a kidsIOT mainboard, which can realize intelligent lighting control, saving energy and improving usage efficiency.

It empowers to automatically detect day and night as well as light intensity, making the entire system more intelligent and energy-saving.

When the ambient brightness is lower than the set value, the photoresisitor will detect a signal and automatically turn on the LED; when higher than the set value, it will send another signal to turn off the LED.

2. Components

		
kidsIOT Mainboard×1	Photoresistor×1	White LED Module×1
		
Wire×2	USB Cable×1	Light Controlled System LEGO Pieces×1

About Photoresistor

It is a sensor that uses a photoresistor to convert optical signals into electrical signals(the stronger the light, the greater the light intensity value).

Parameters:

Working voltage: DC 3.3V-5V

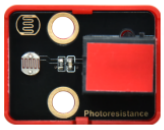
Working current: (Max)0.2mA@5V

Maximum power: 1mW

Spectral peak: 540nm

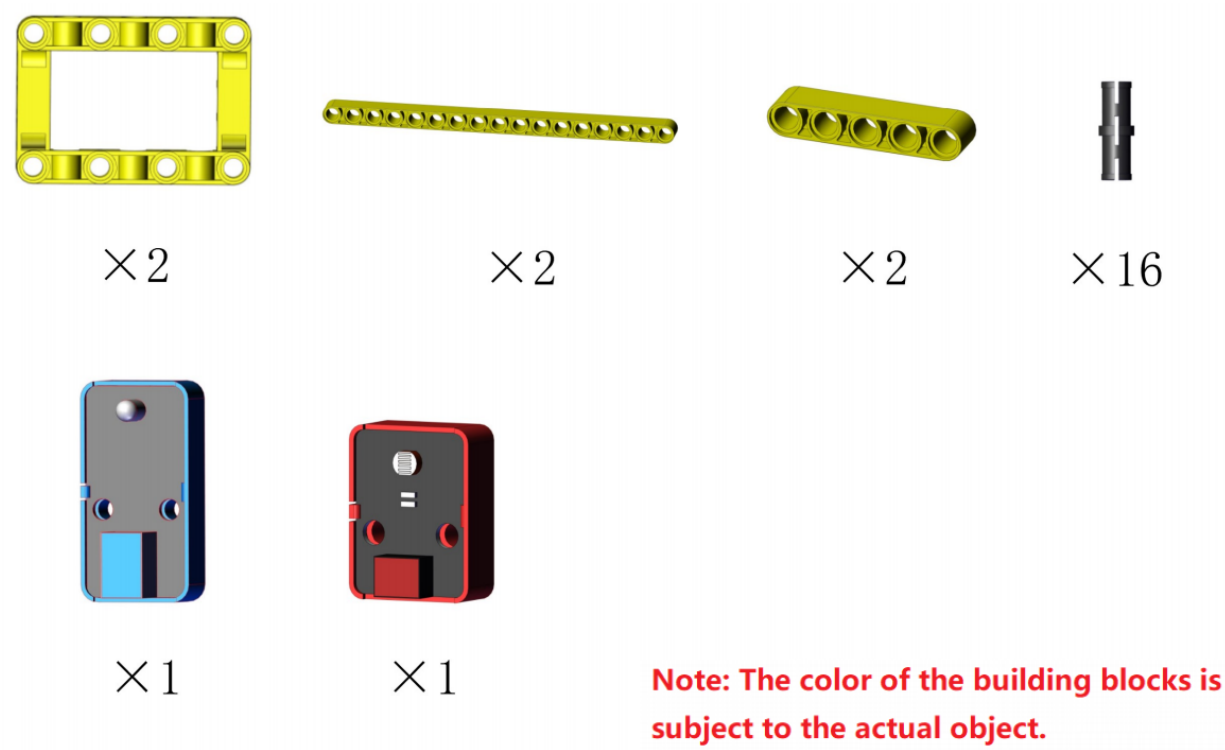
Bright resistance: (10lux): 5-10KR

Dark resistance: 0.5MR



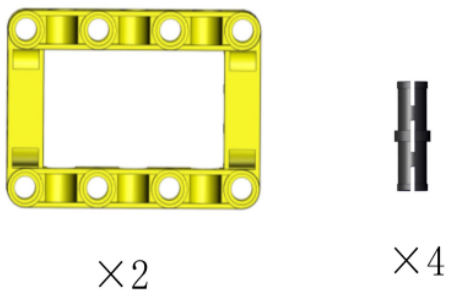
3. Assembly Steps

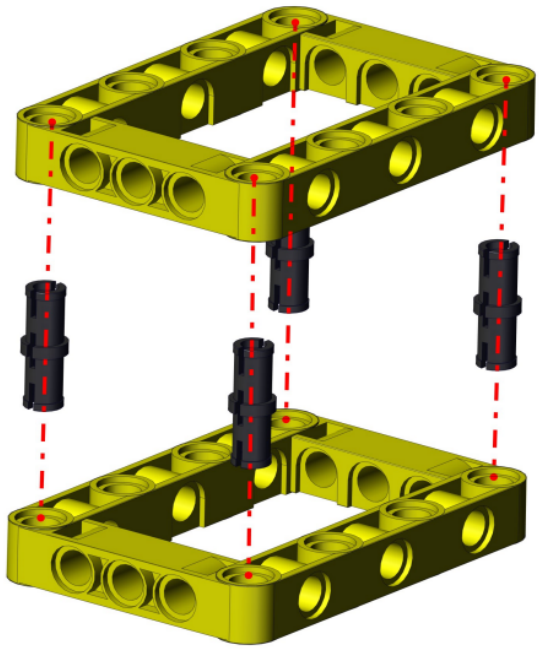
Step 1Components Needed

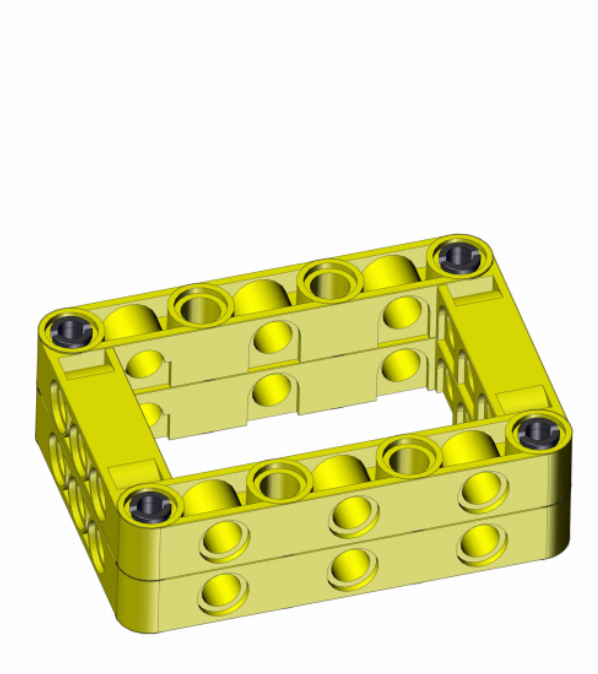


Step 2Process

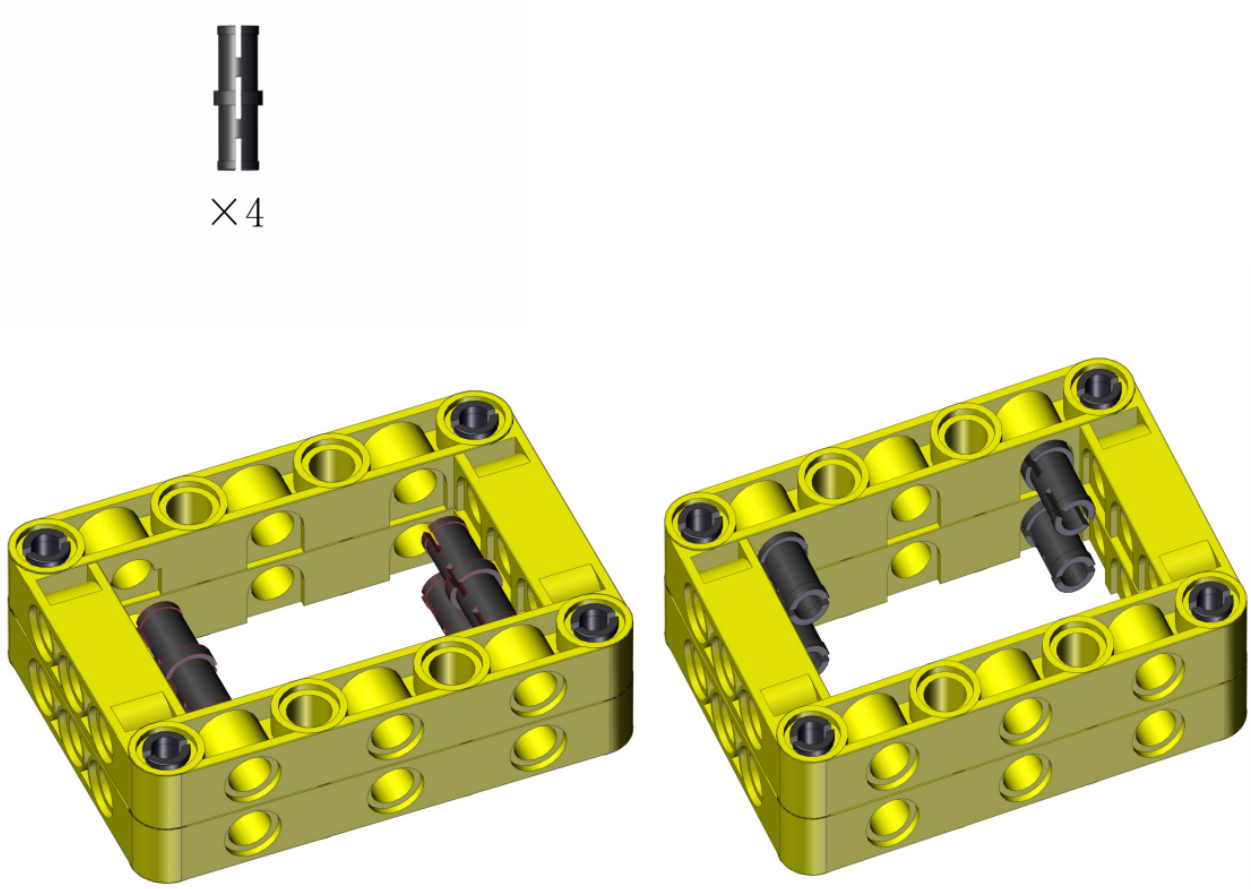
Process 1



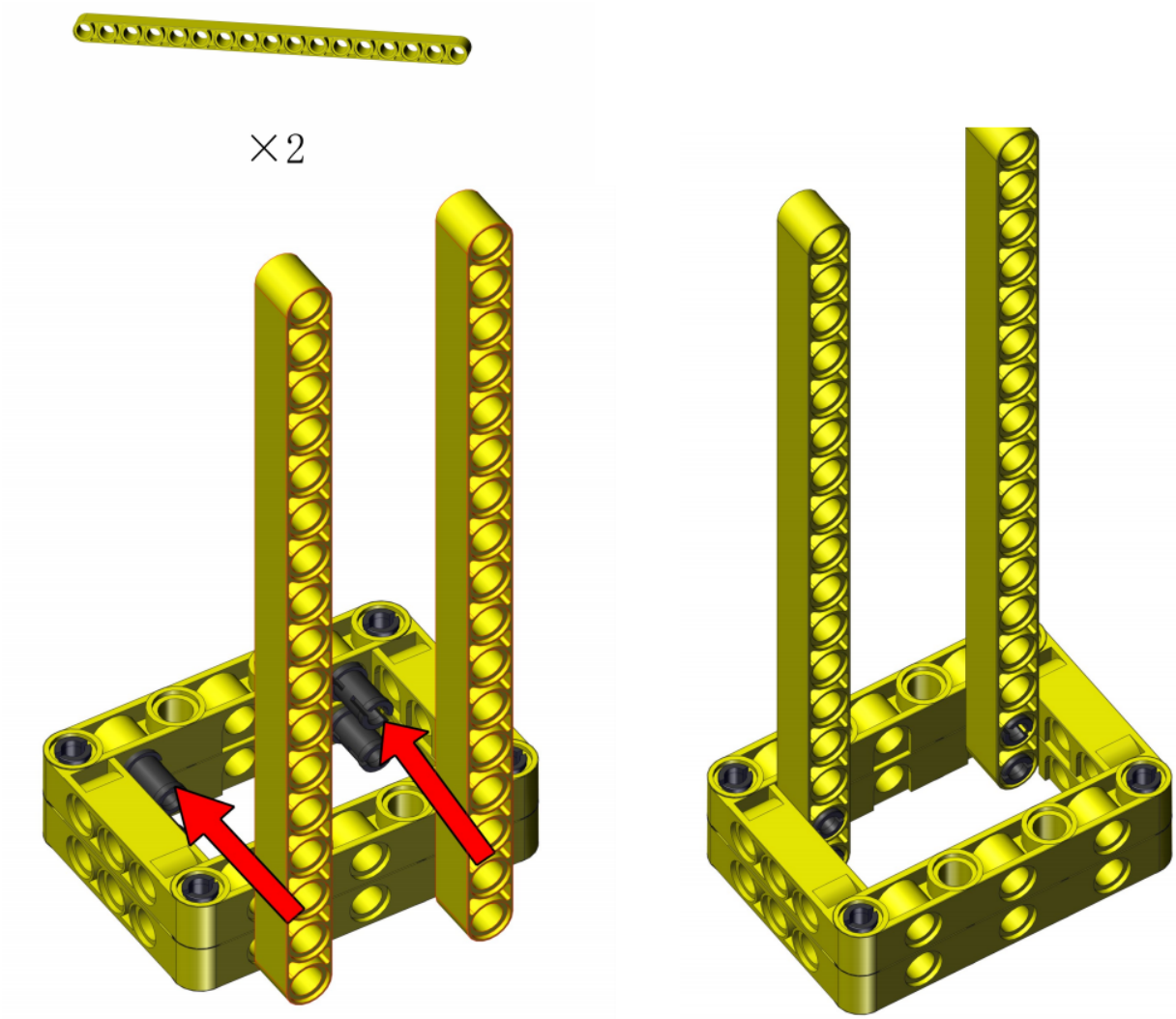




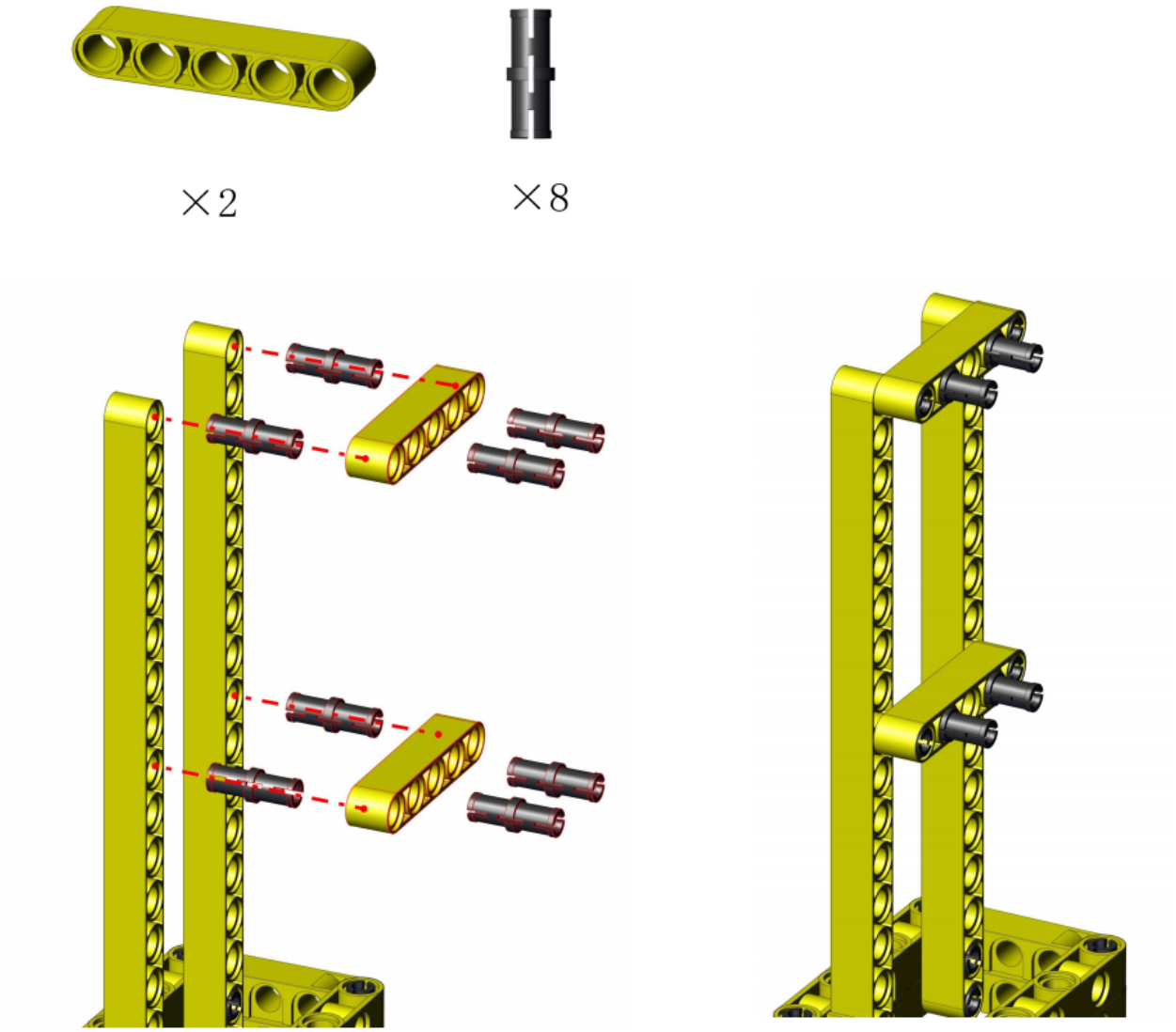
Process 2



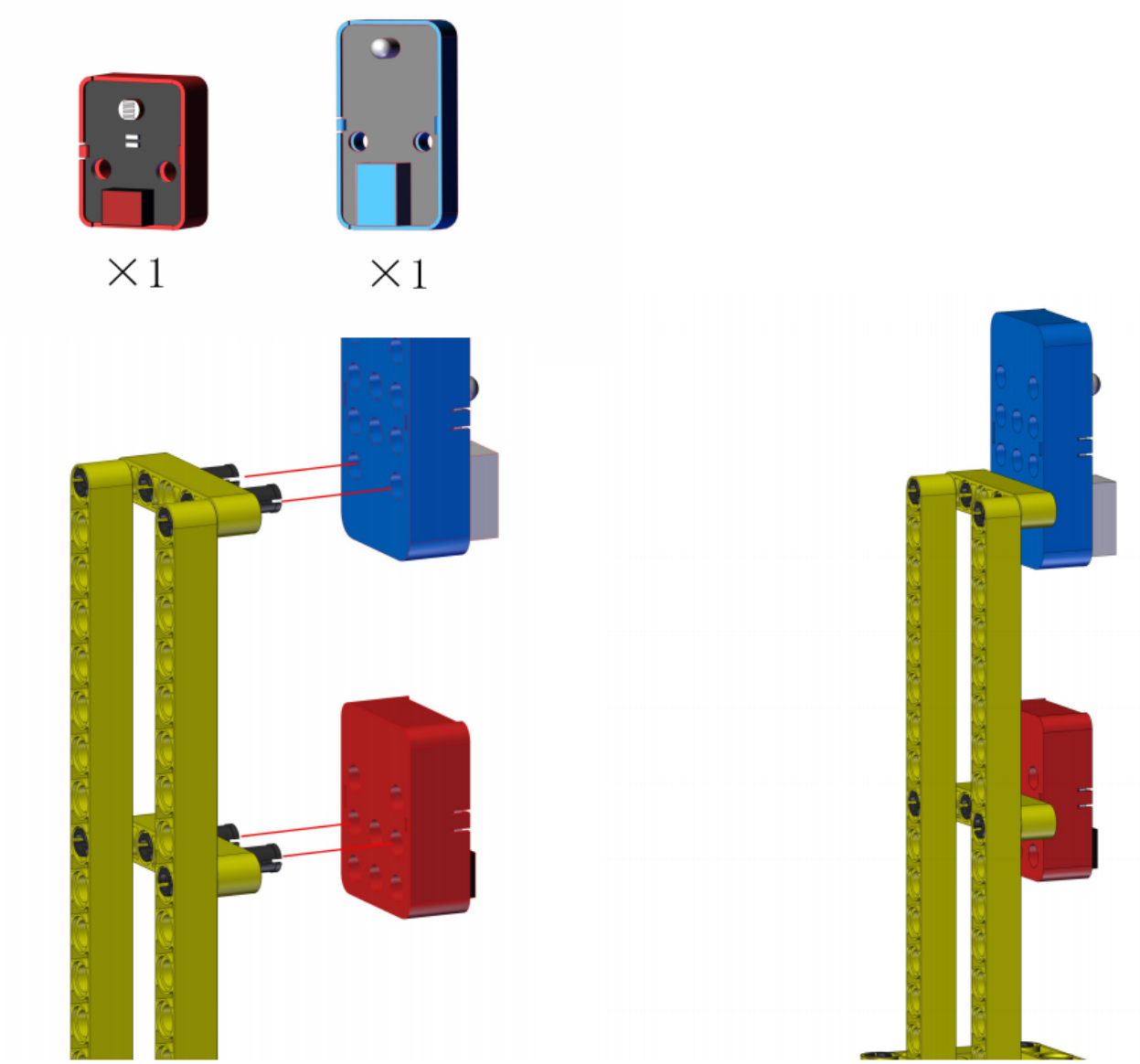
Process 3



Process 4



Process 5



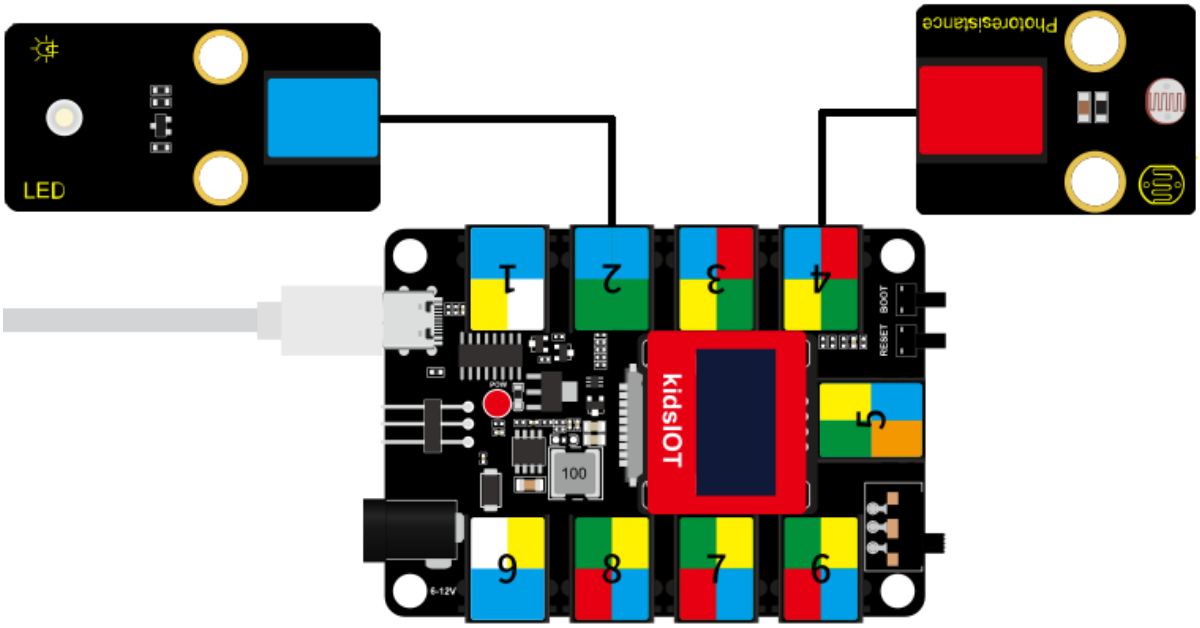
Complete

(Note: Do not dismantle it, it will still be used in project 10.)

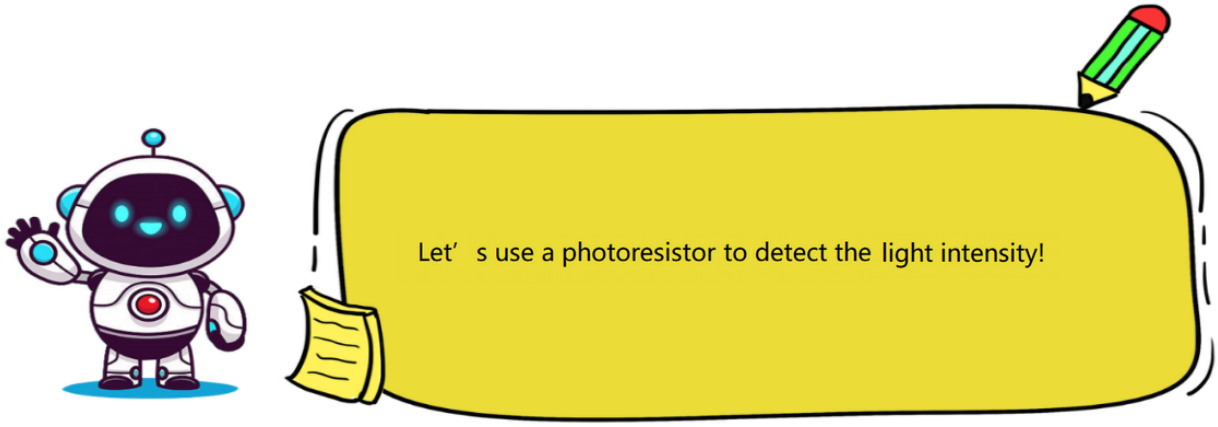
4. Wiring Diagram

Module	kidsIOT Mainboard
White LED Module	No.2 portcontrol pin is io2
Photoresistor	No.4 portcontrol pin is io39

Connect the kidsIOT mainboard to your computer via USB cable.

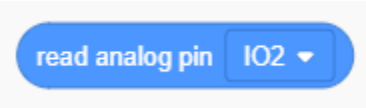


5. Read the value of the photoresistor



(1). Programming Steps

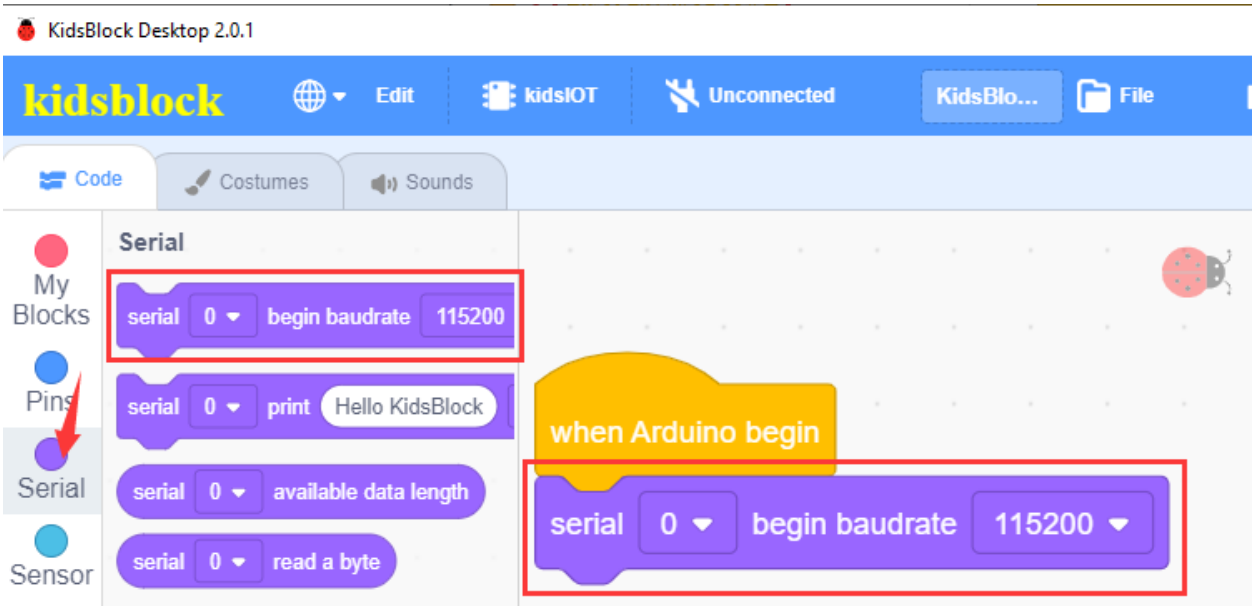
Step 1Description of the Building Blocks



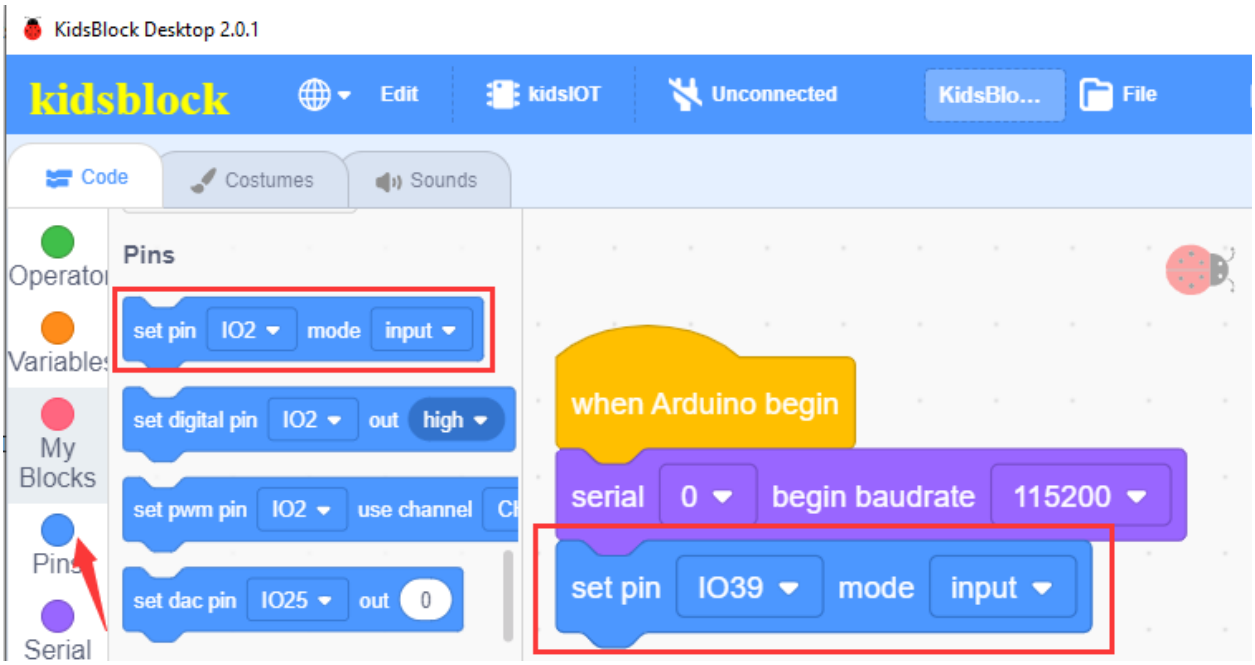
It is used to read the analog signal value of the specified pin.

Step 2Write the Program

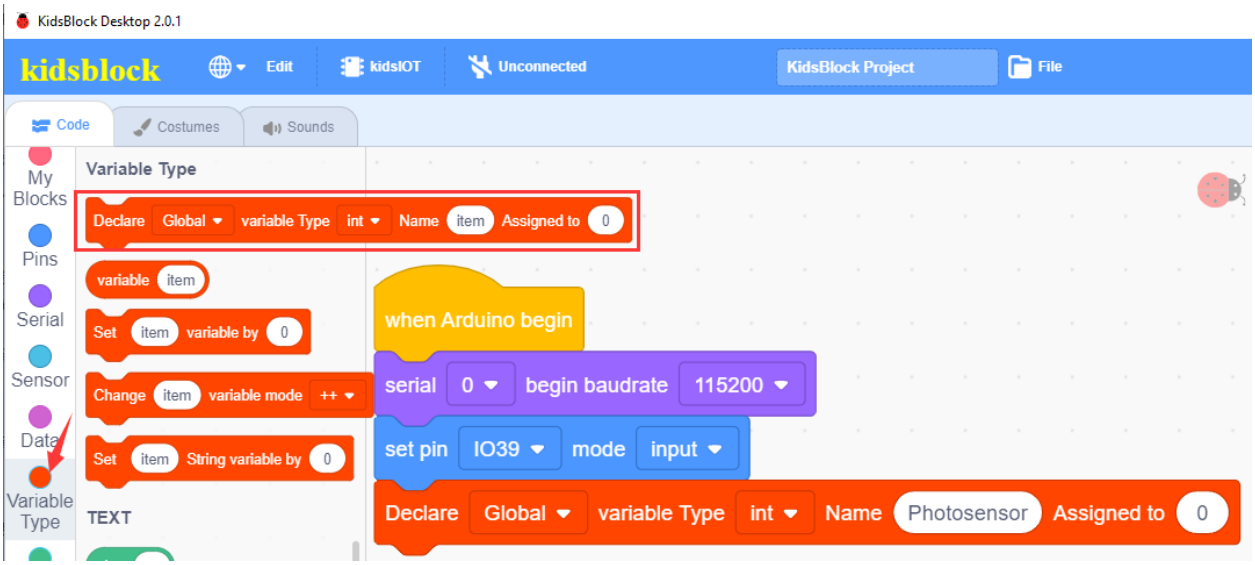
Set the baud rate to 15200.



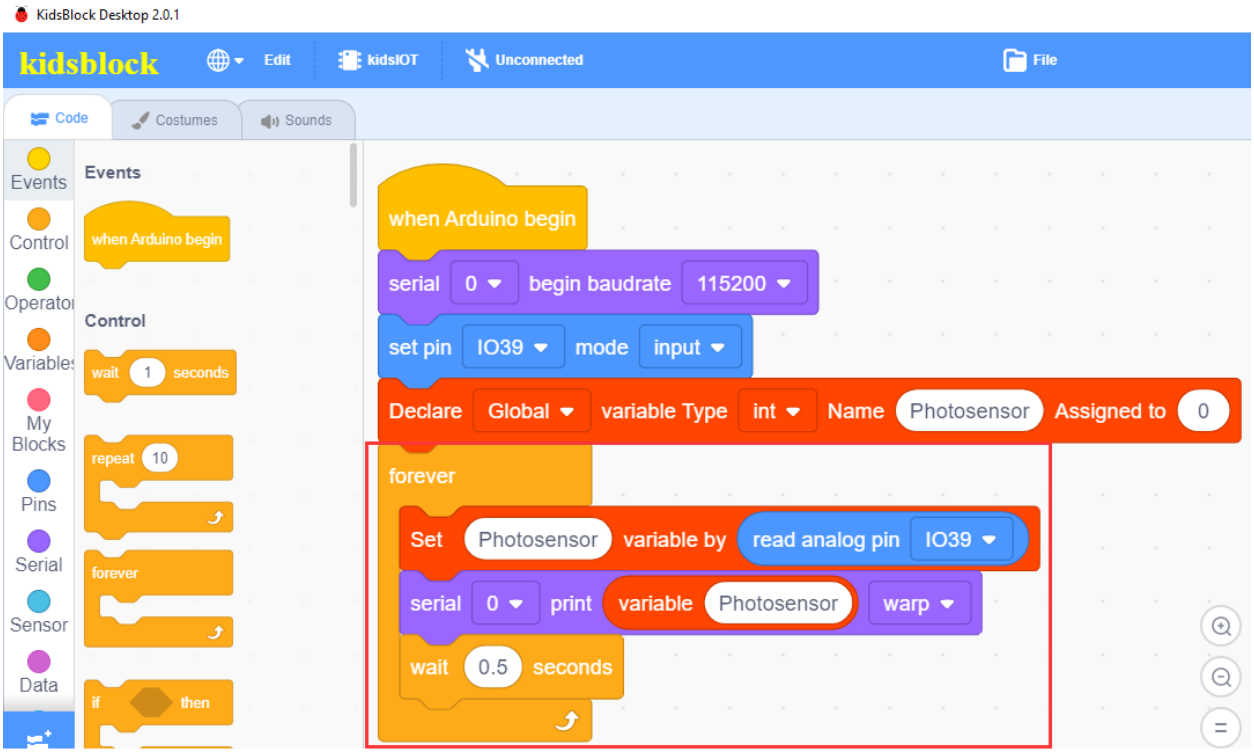
Set the pin IO39 connected to the photoresistor (control pin io39) to the "**input**" mode.



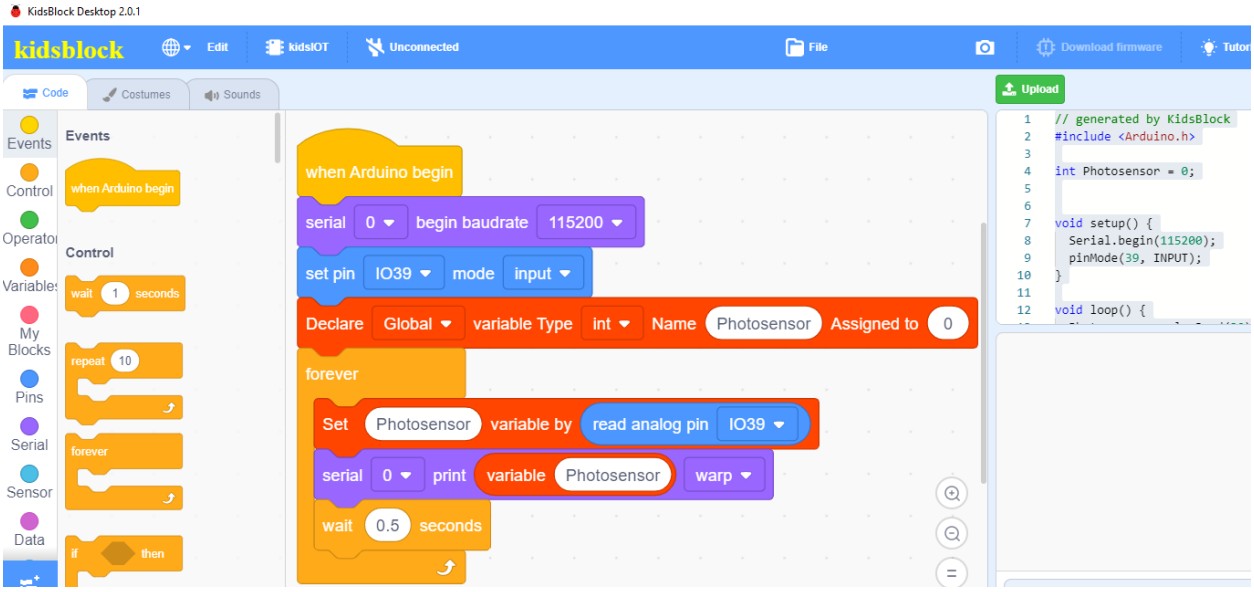
Define a "Photosensor" variable to store the value of the photoresistor.



Store the read value of the photoresistor in the "Photosensor" variable.



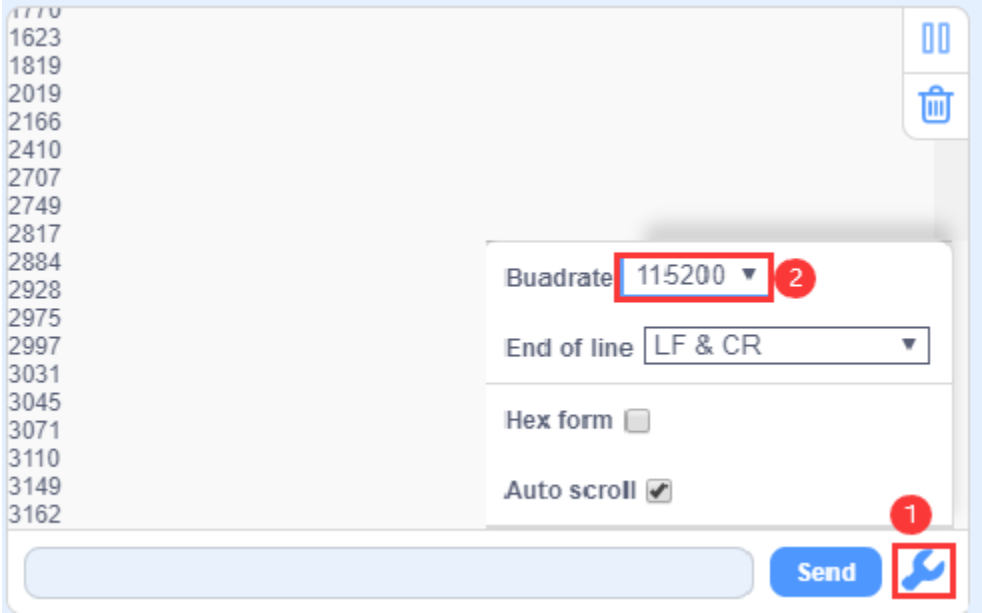
Complete Program



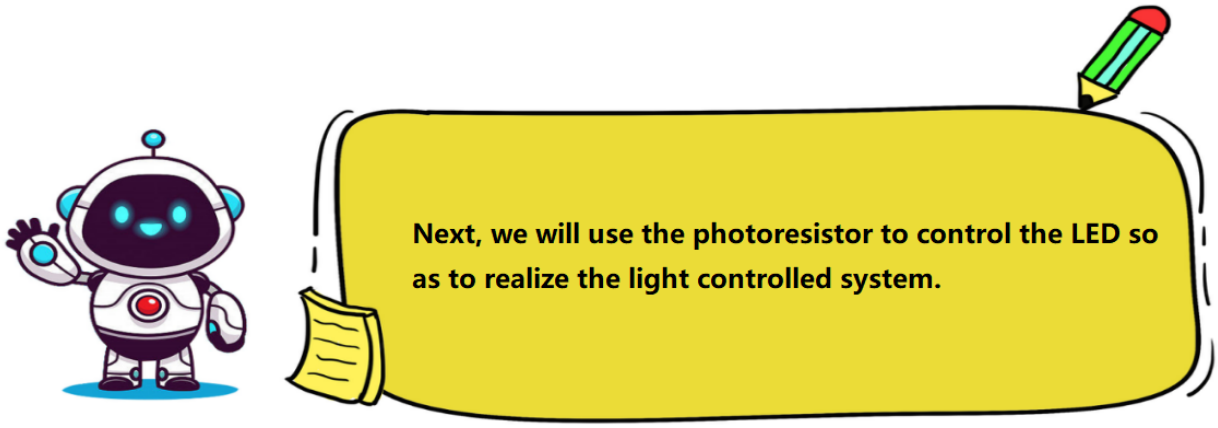
(2). Test Result

Click 

to upload the above complete code to the kidsIOT motherboard. After powering up via the USB cable, click  in the serial monitor and set the baud rate to 15200. Then the serial monitor will print the value read by the photoresistor. When the light detected by the photoresistor is brighter, the monitor prints a larger value, otherwise, the monitor prints a smaller value.

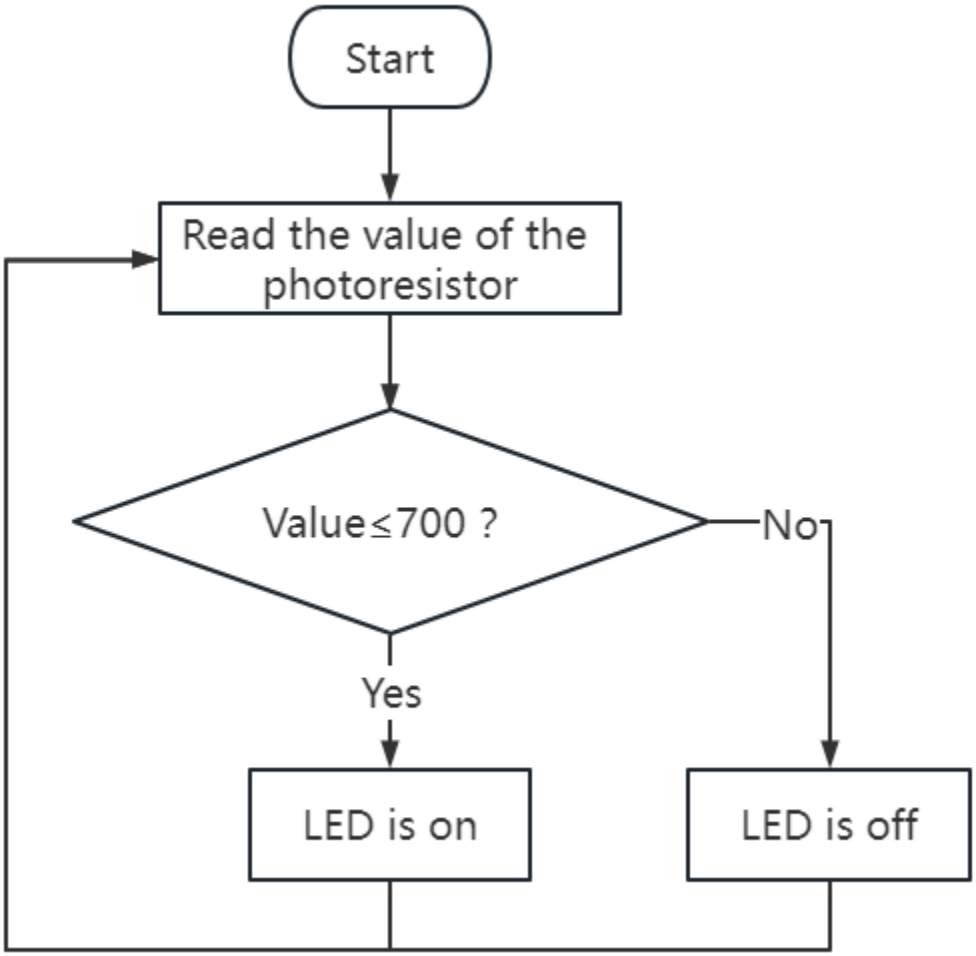


6. Light Controlled System



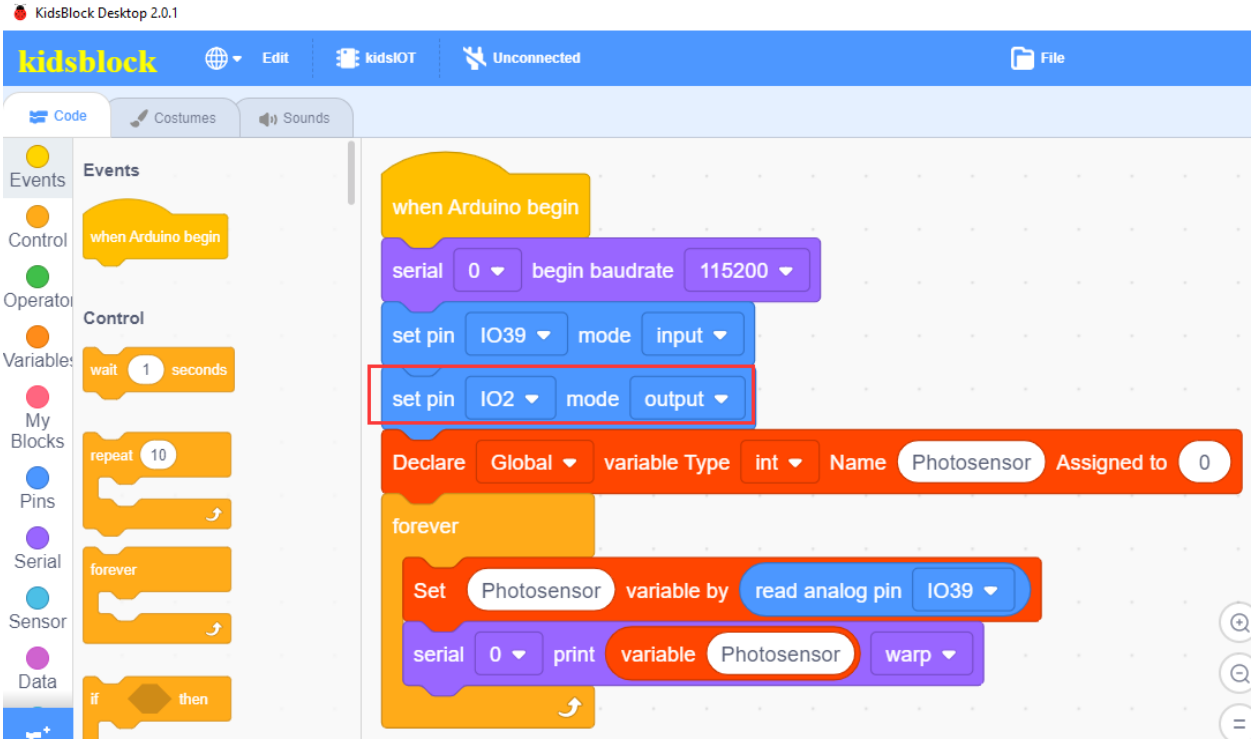
(1). Programming Steps

Step 1Flow Chart

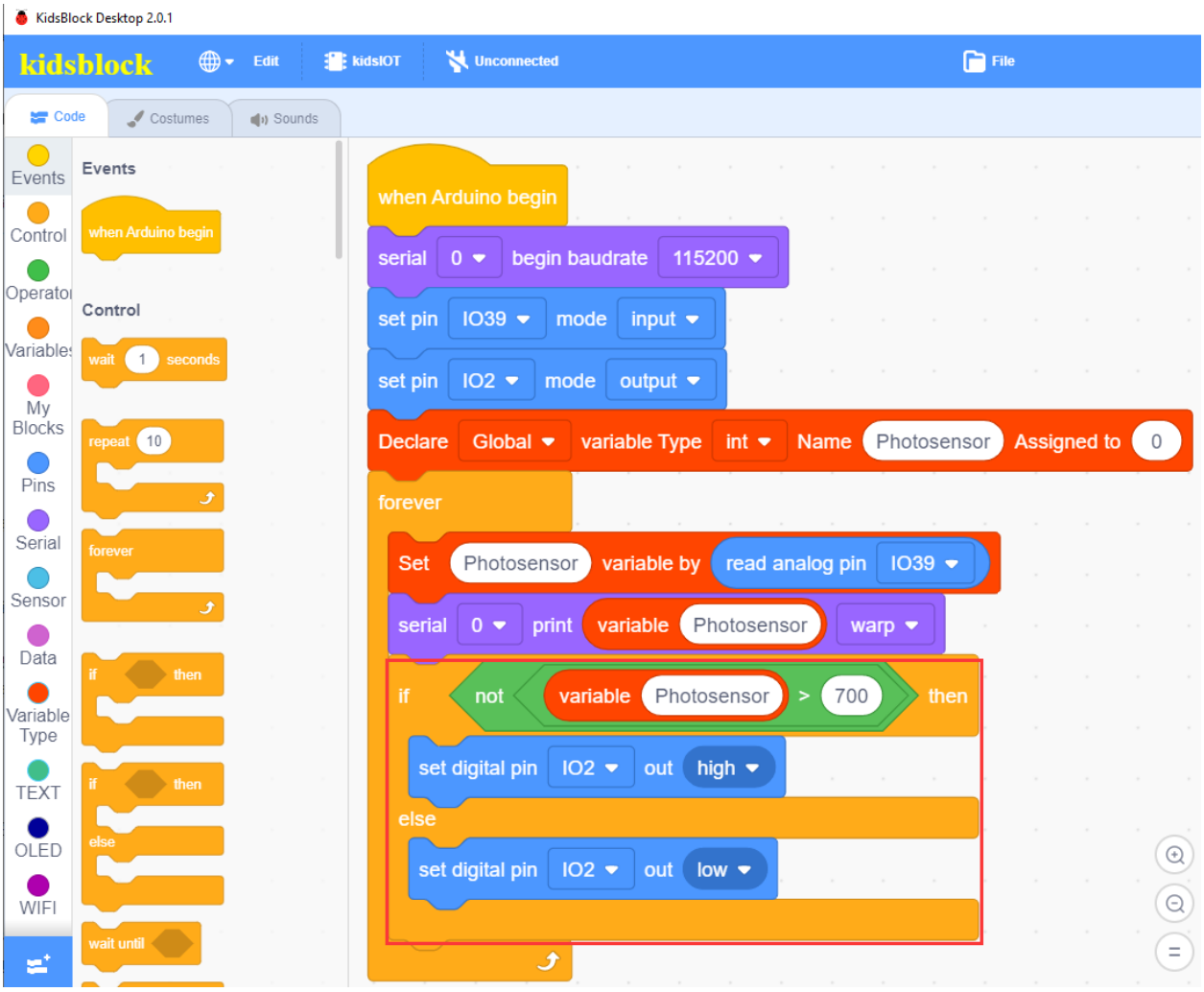


Step 2Write the Program

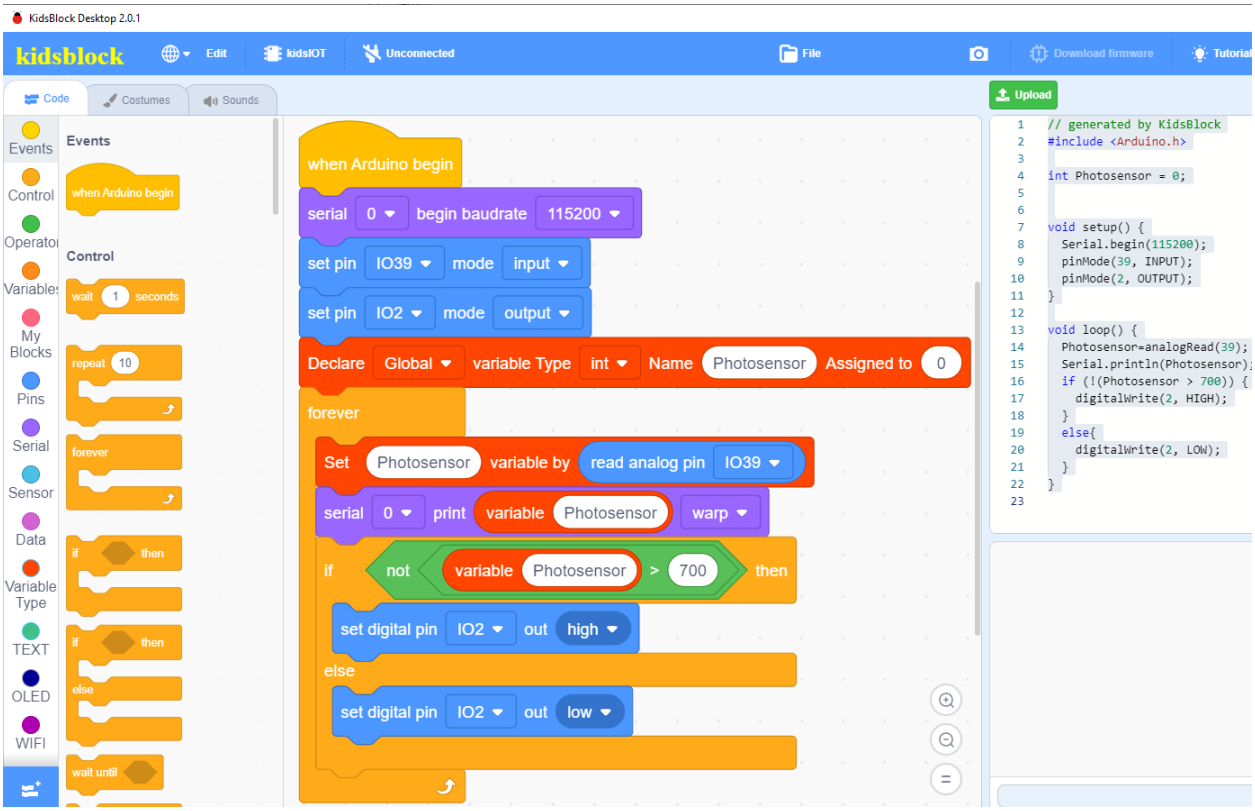
Delete the "Wait 0.5 seconds" block in the complete program above, and then drag "**Set pin IO2 mode input**" block from the "**Pin**" module to the script area. Change "**input**" to "**output**".



Determine the value read by the photoresistor. When the value is less than or equal to 700, the LED will turn on, otherwise, the LED will turn off.



Complete Program



(2). Test Result

Click  to upload the above complete code to the kidsIOT motherboard, then power up via the USB cable. When the light is strong during the day and the value of the photoresistor is greater than 700, the LED will turn off. After dark, when the value is less than or equal to 700, the LED will automatically turn on.

7. Common Problems

Q1: The value detected by the photoresistor cannot be 0?

A: Because in actual operation, even if all the lights are turned off and the room is very dark, the value is only close to 0.

Q2: After burning the sample code, why can't the LED light up even when the lights are turned off in the room?

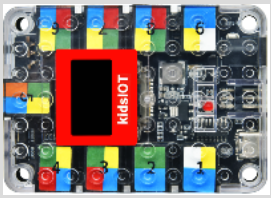
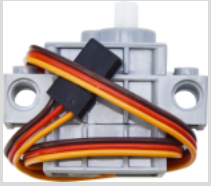
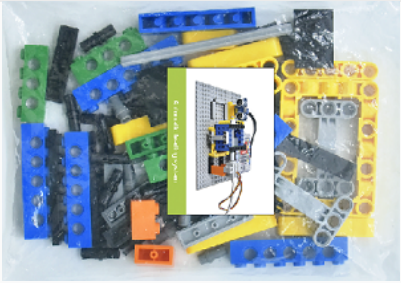
A: You can set the value read by the photoresistor to be larger. The value in the sample code is 700, which can be adjusted to 1000 or even larger.

4.3.3 Project 03: Automatic Feeding System

1. Description

The automatic feeding system is composed of a kidsIOT main board, an ultrasonic sensor and a servo. The ultrasonic sensor is used to detect the distance of pets in the feeding area. When the pet approaches the food bowl, the sensor detects that the distance is getting closer. After triggering the signal, it controls the servo to open the feed box and automatically feed the animals.

2. Components

			
kidsIOT Mainboard×1	Ultrasonic Adapter Board×1	Ultrasonic Sensor×1	Servo×1
			
Wire×1	USB Cable×1	Automatic Feeding System LEGO Pieces×1	

About Ultrasonic Sensor, Ultrasonic Adapter and OLED

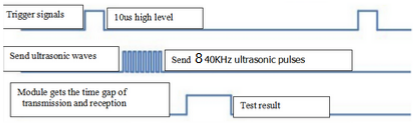
Ultrasonic Sensor: It uses sonar to determine distance to an object like bats do and it comes with an ultrasonic transmitter and a receiver module.

Ultrasonic wave is also a sound wave whose speed V is related to temperature. Generally, the transmission speed of the wave in the air is 340m/s. According to the time t recorded by the timer, the distance s from the sensor to the obstacle can be calculated, that is, S =340t/2:

(1) Pull down TRIG then trigger high level signals with least 10us;

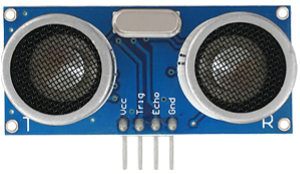
(2) After triggering, the module will automatically send eight 40KHz ultrasonic pulses and detect whether there is a signal return;

(3) If there is a signal return, ECHO outputs a high level.



Parameters:

Working voltage: 3.3V-5V
Quiescent current: <2mA
Working current: 15mA
Sensing angle: <15°
Distance range: 2cm – 400 cm
Precision: 0.3 cm
Measuring angle: 30 degree
Input trigger pulse: 10us



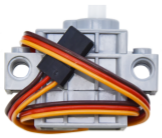
Ultrasonic Adapter: It is mainly convenient to connect and use the ultrasonic sensor. sensor.

Parameters:

Working voltage: 3.3V-5V

Servo: It is a component featuring slower speed and stronger strength than a motor, which is able to make robotic arms do more flexible, more difficult and more precise movements. What's more, it can rotate to different angles according to different models(0°-180°, 0°-270°), here we will use a 270° servo.

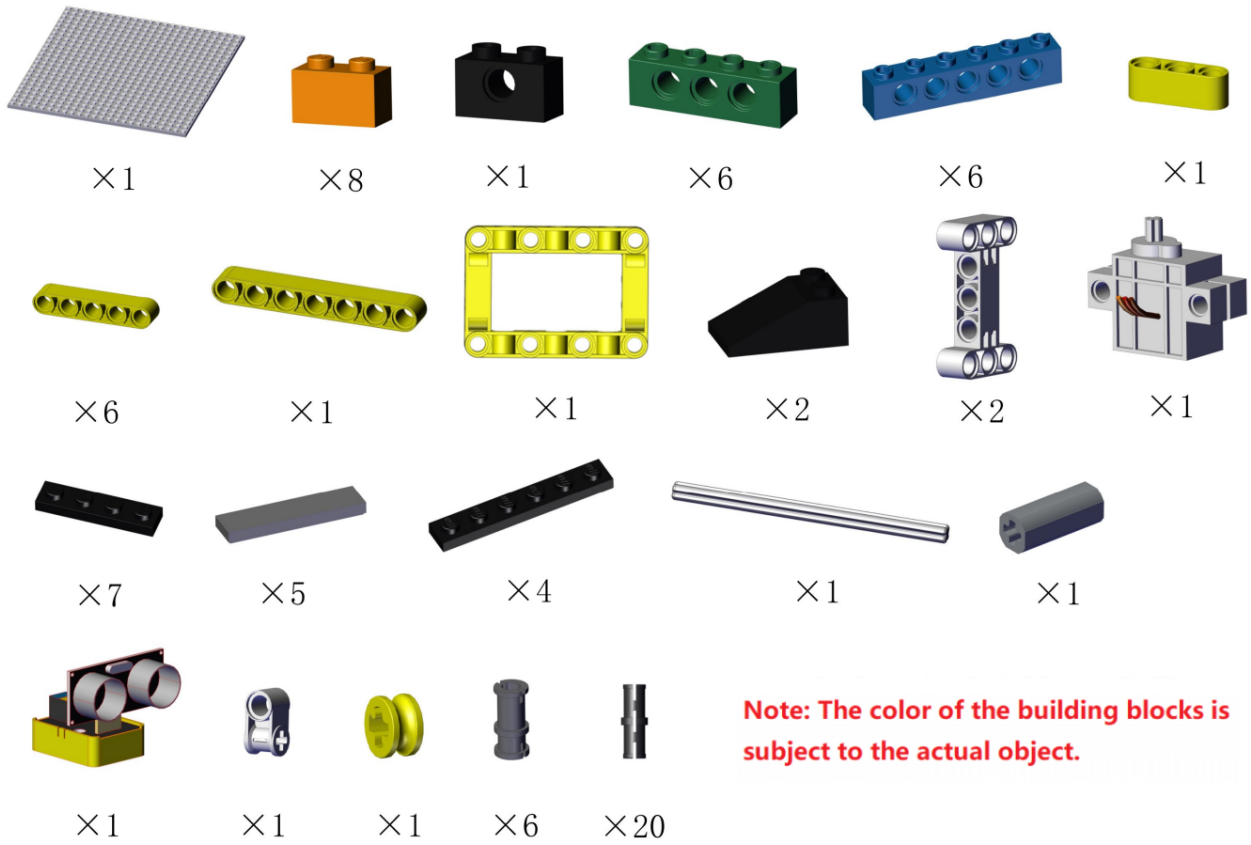
Parameters:



Working voltage: DC 3.3V-6V
Rated voltage: DC 4.8V
Deceleration ratio: 1/266
Neutral position: 1500usec
Control system: change pulse width
Range of pulse width: 600-2400usec
No-load speed: 0.12±0.01sec/60° (4.8V test); 0.10±0.01sec/60° (6V test)
No-load current: 50±20mA (4.8V test);70±20mA (6V test)
Stop current: 0.8±0.1A (4.8V test):0.9±0.1A (6V test)
Stop torque: 0.9±0.2kg/cm(4.8V test); 1.0±0.2kg/cm(6V test)
Stop current: 0.6±0.1A (4.8V test);0.7±0.1A (6V test)
Standby current: 7±1mA (4.8V test);7±1mA (6V test).
Operating angle(Max) : 270°±10°
Operating angle: 0°- 270° (600-2400usec)
No-load life: more than 50,000 times (4.8V test); more than 40,000 times (6V test).

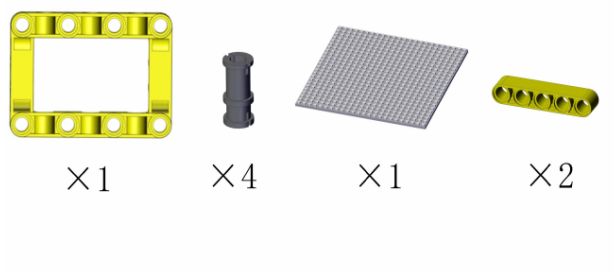
3. Assembly Steps

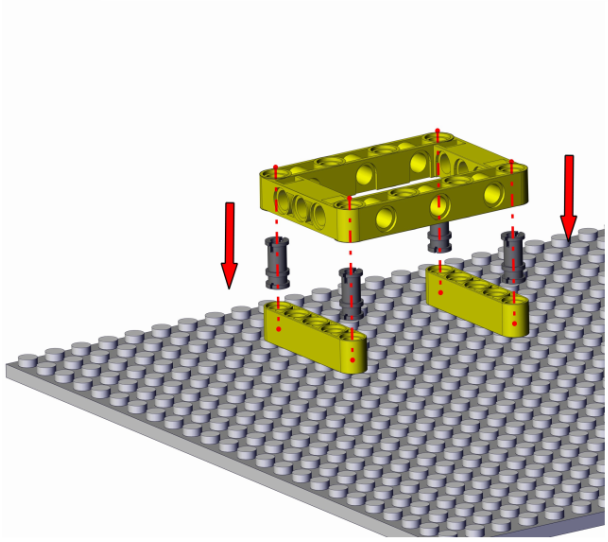
Step 1Components Needed

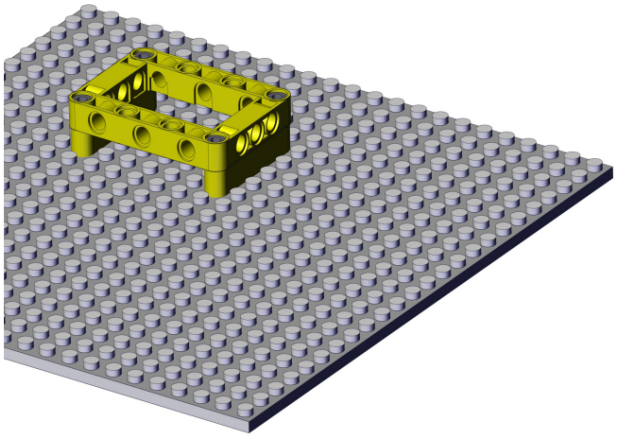


Step 2Process

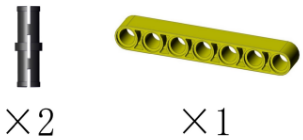
Process 1

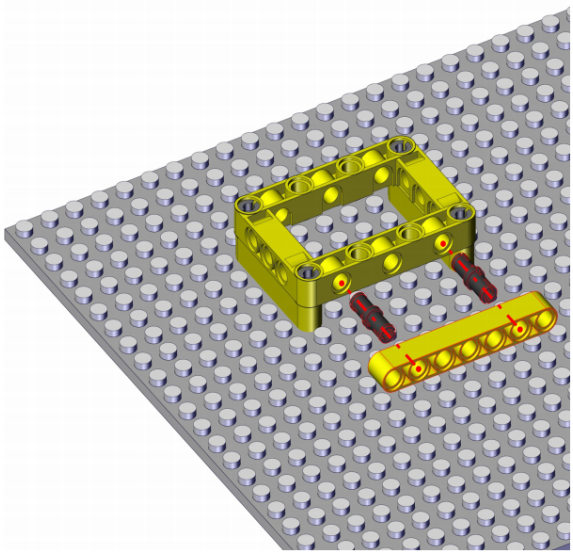


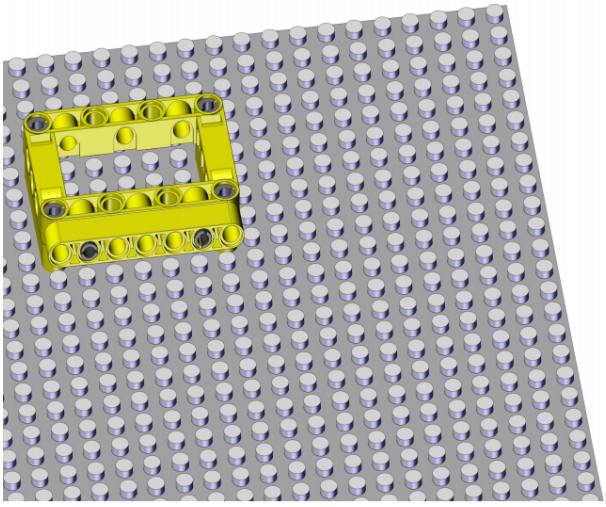




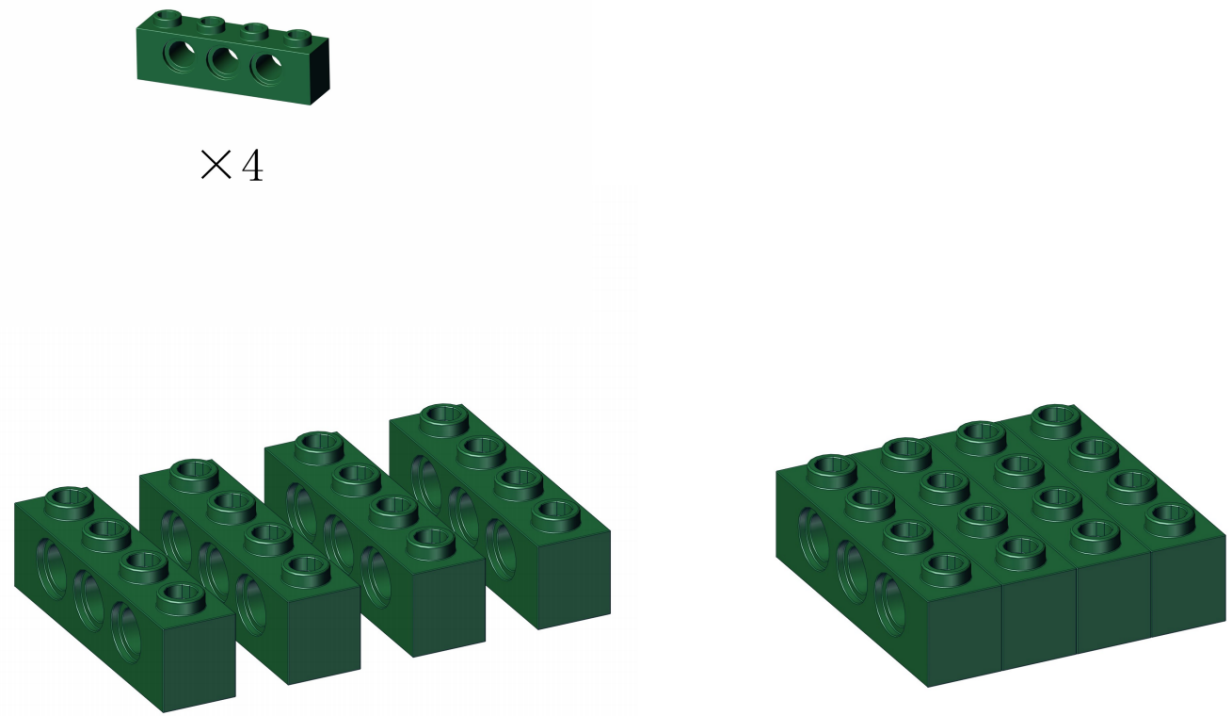
Process 2



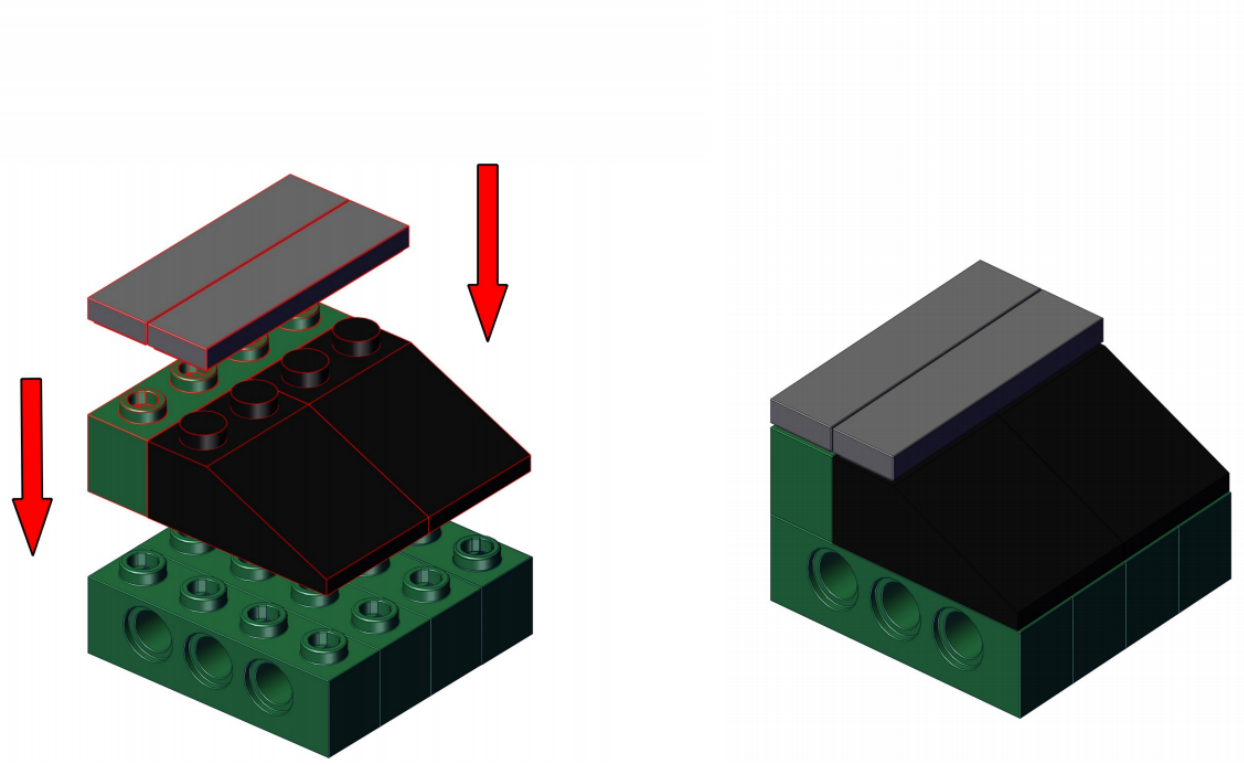




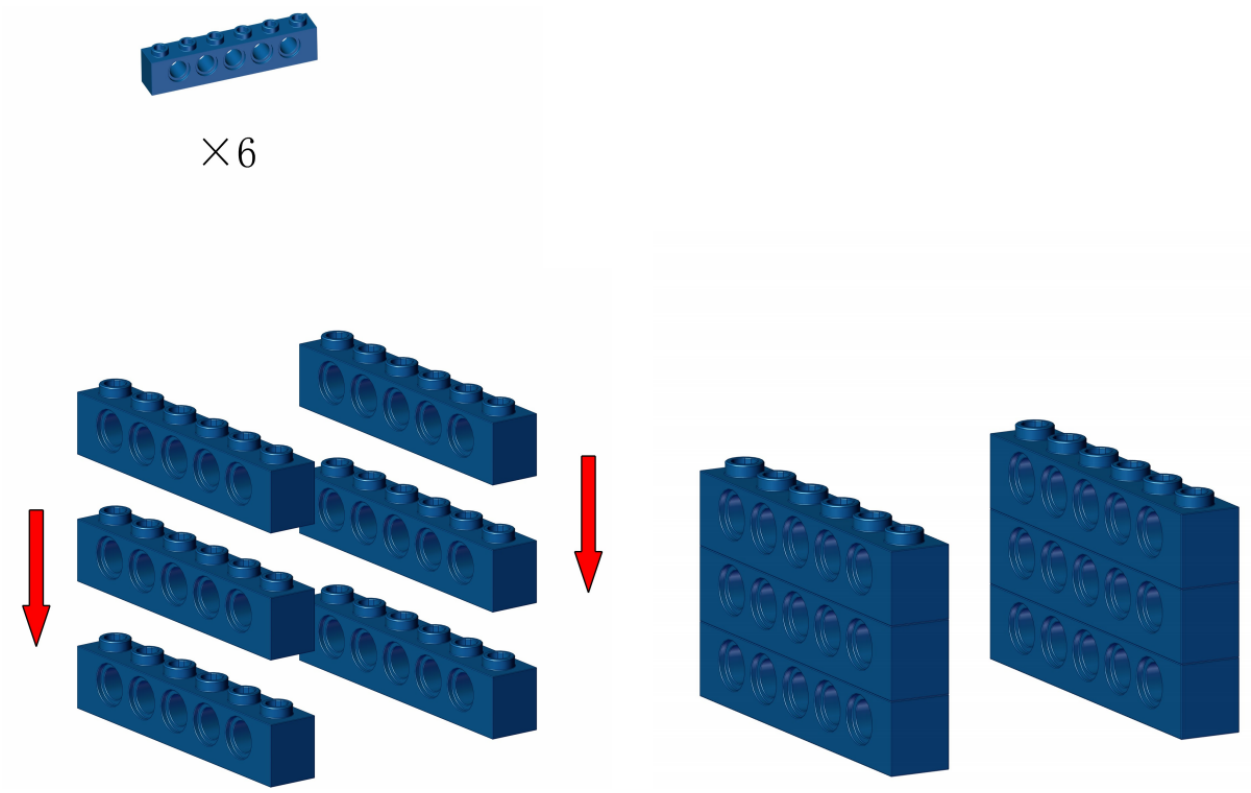
Process 3



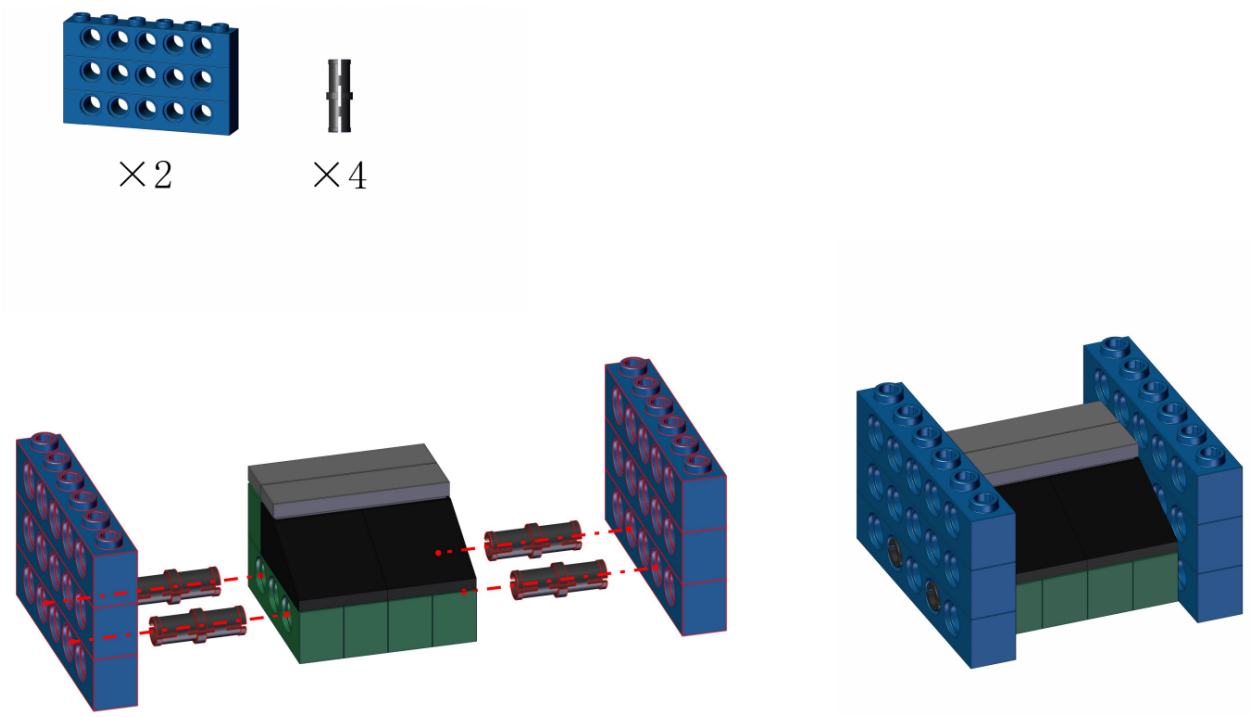
Process 4



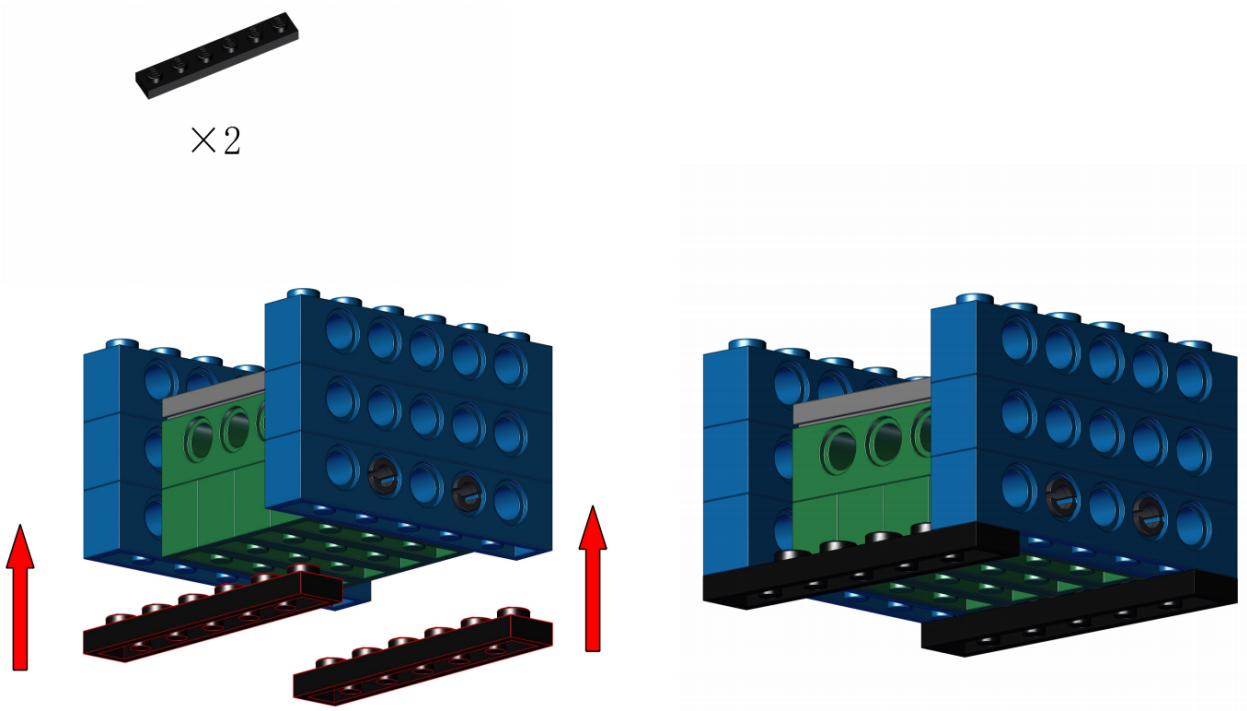
Process 5



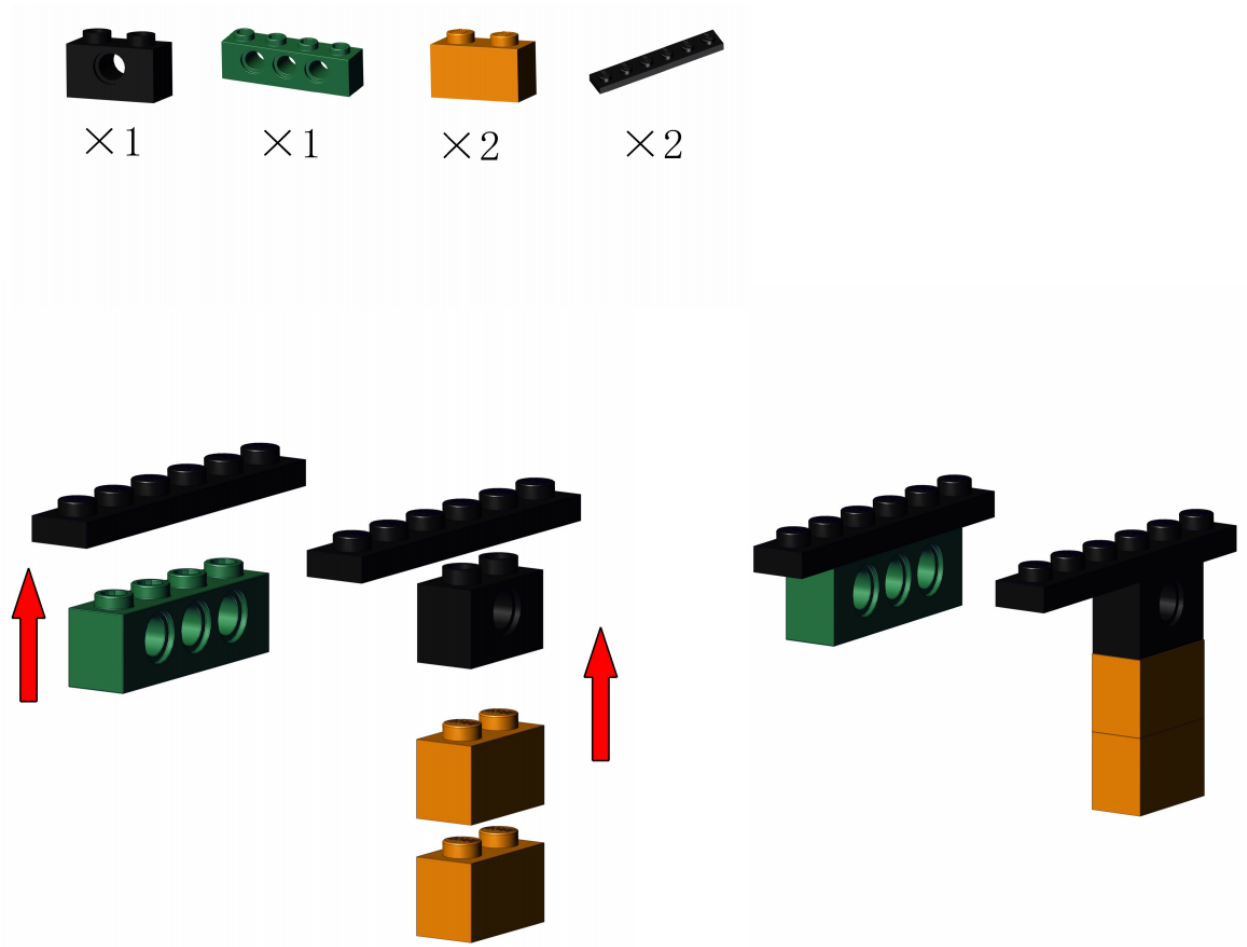
Process 6



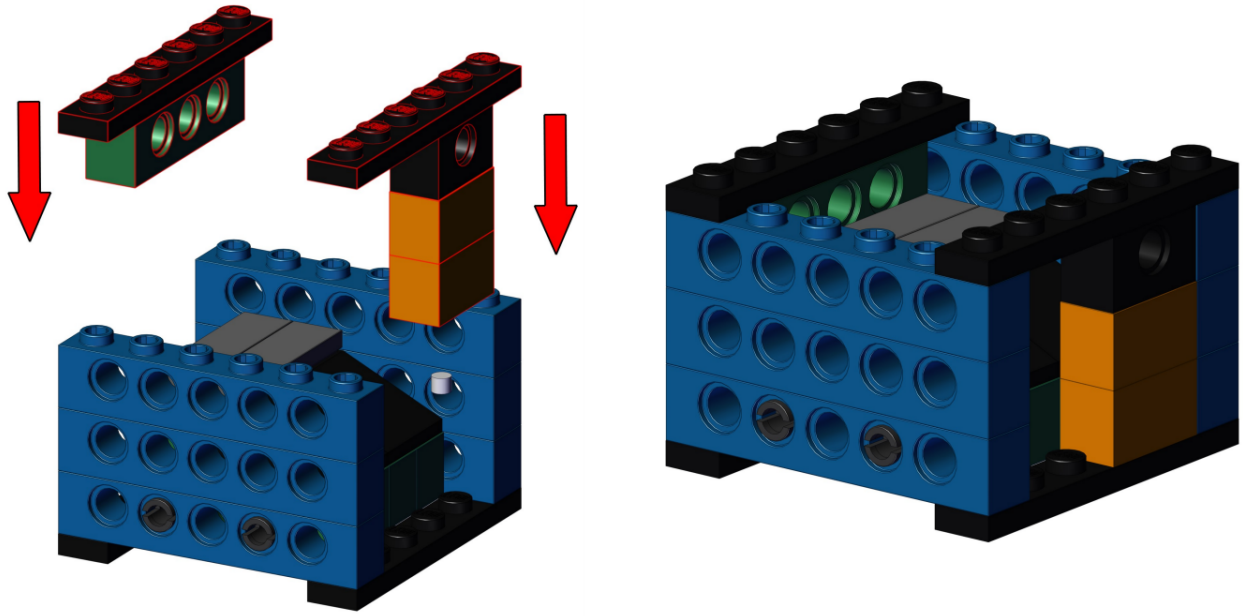
Process 7



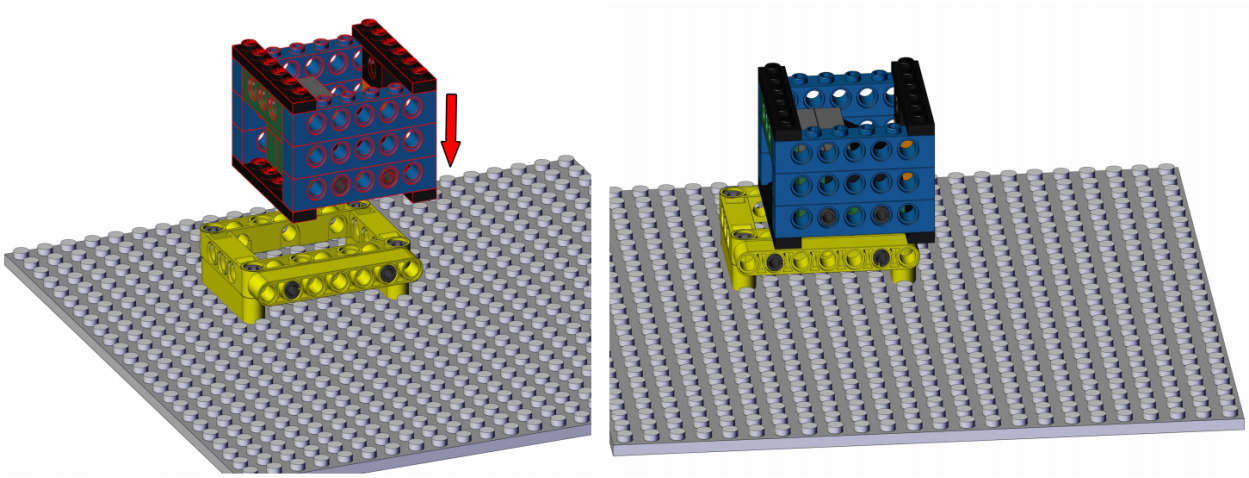
Process 8



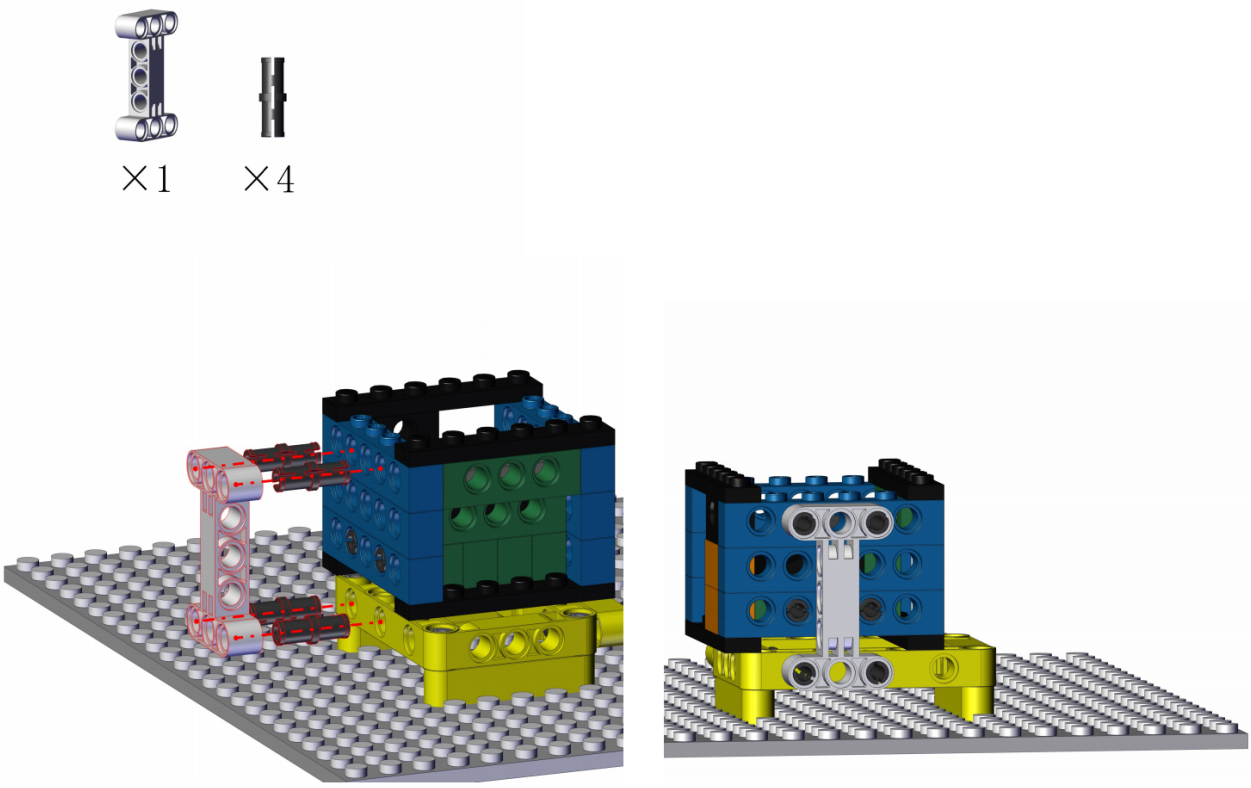
Process 9



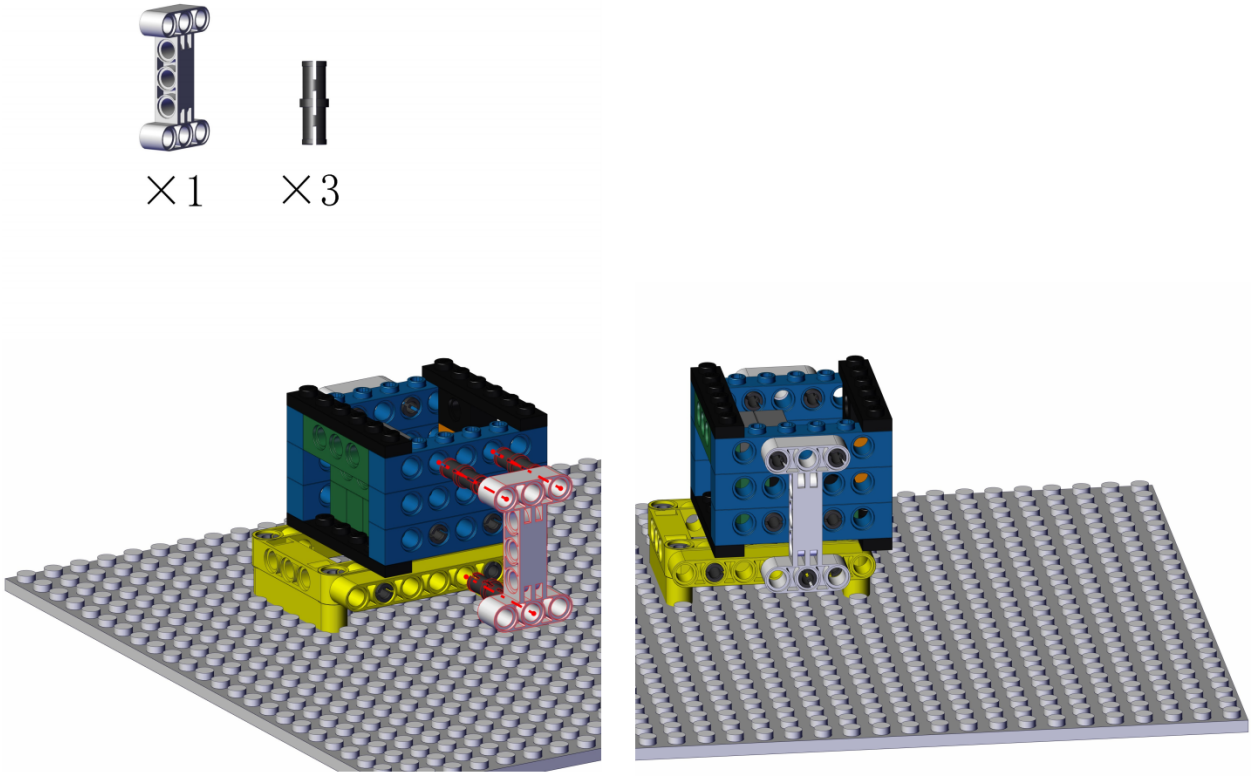
Process 10



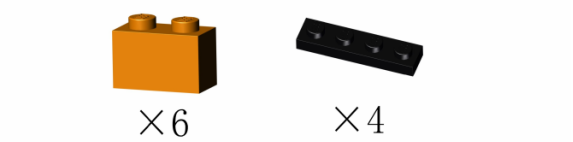
Process 11

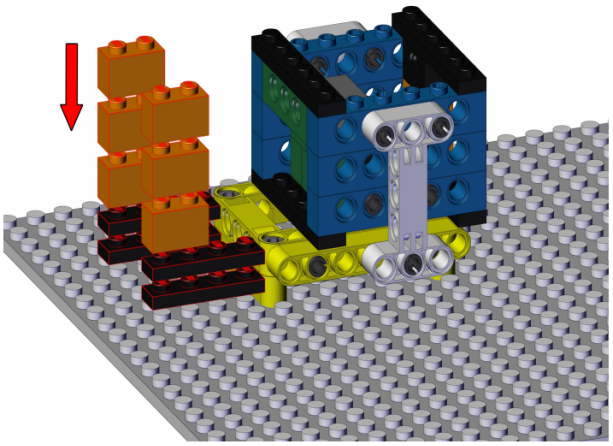


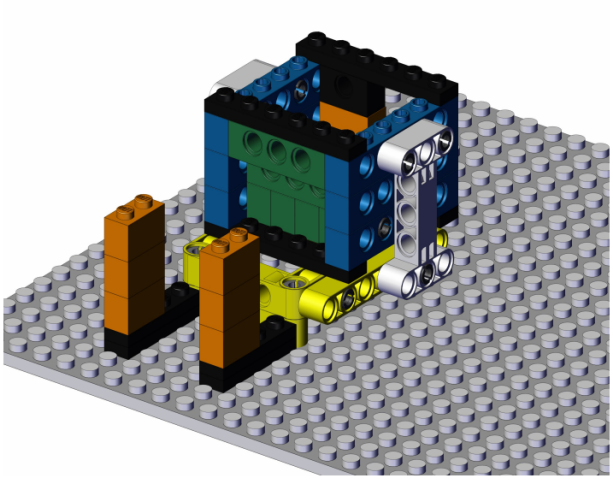
Process 12



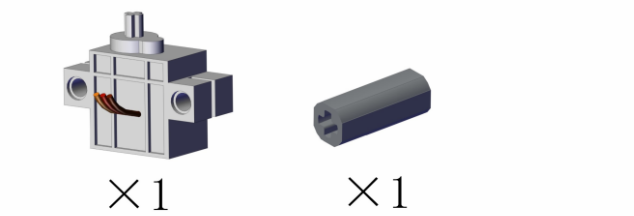
Process 13

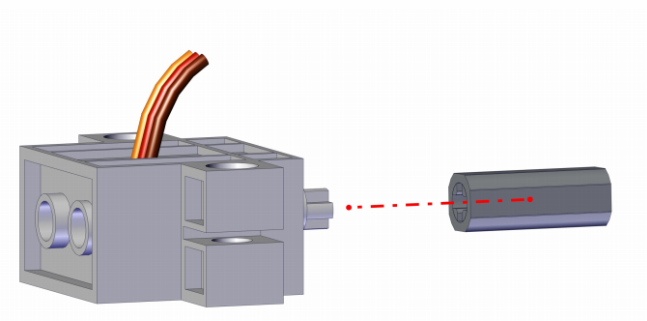


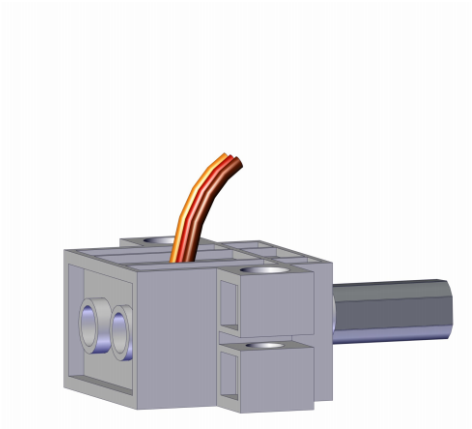




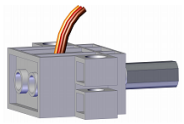
Process 14



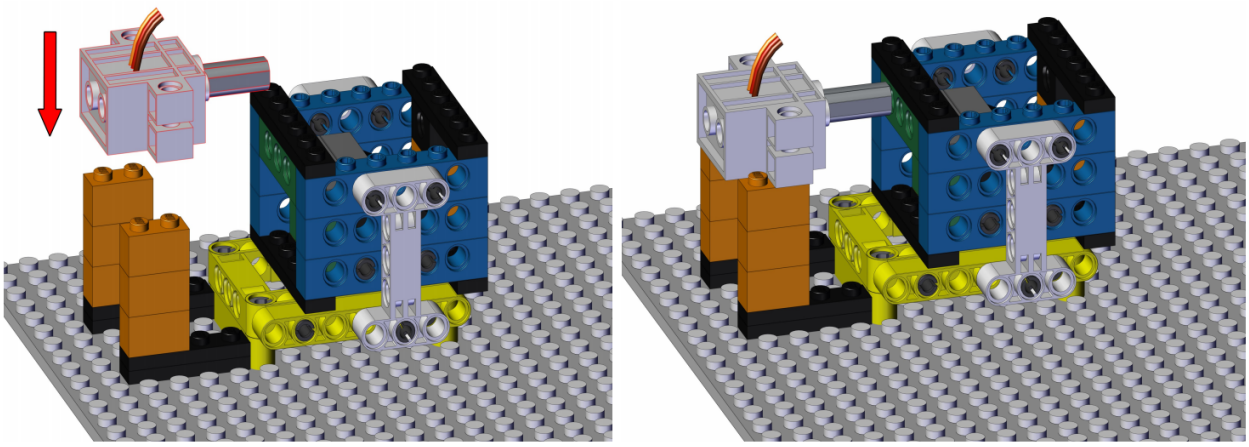




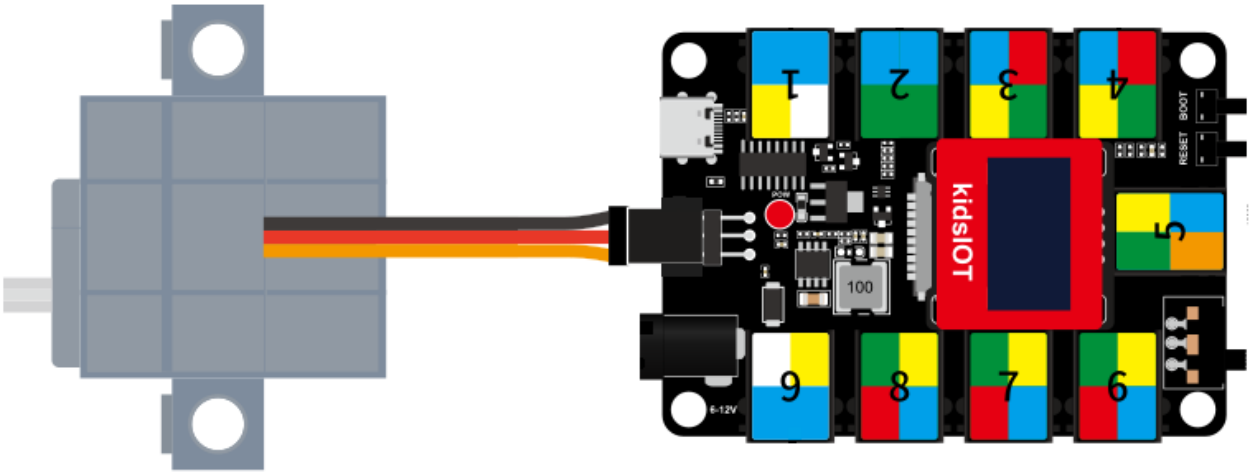
Process 15



×1

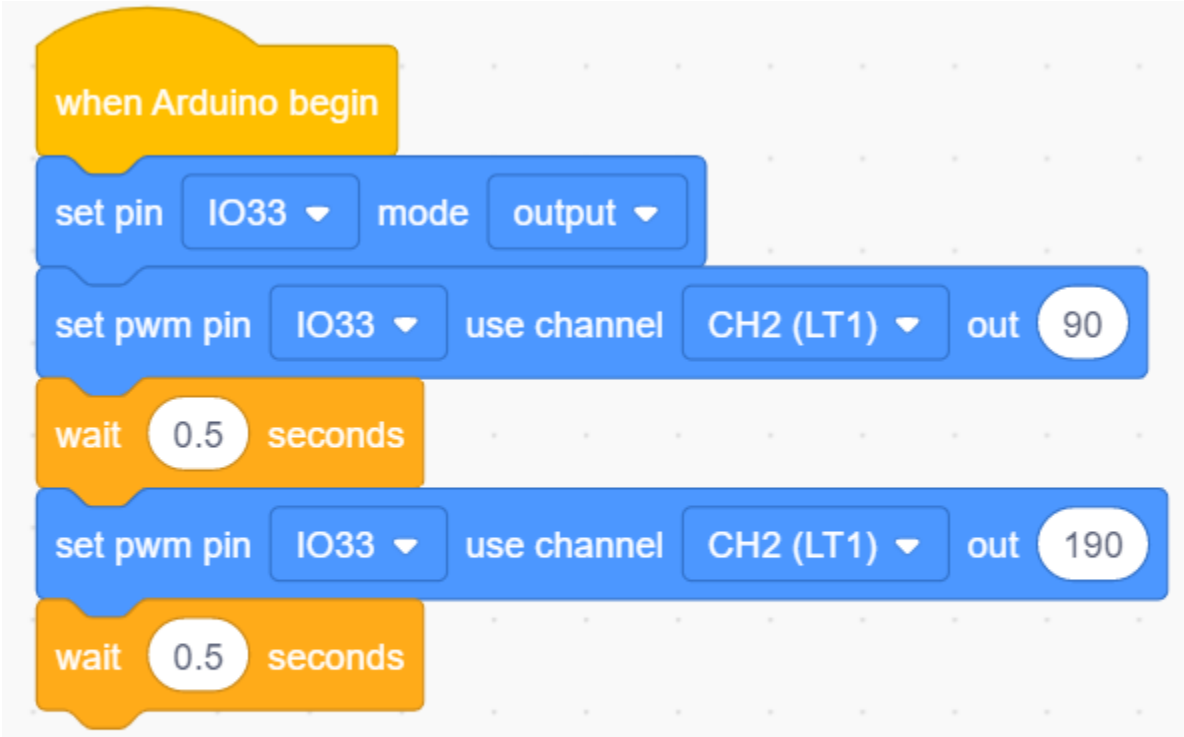


Process 16Initialize the servo angle

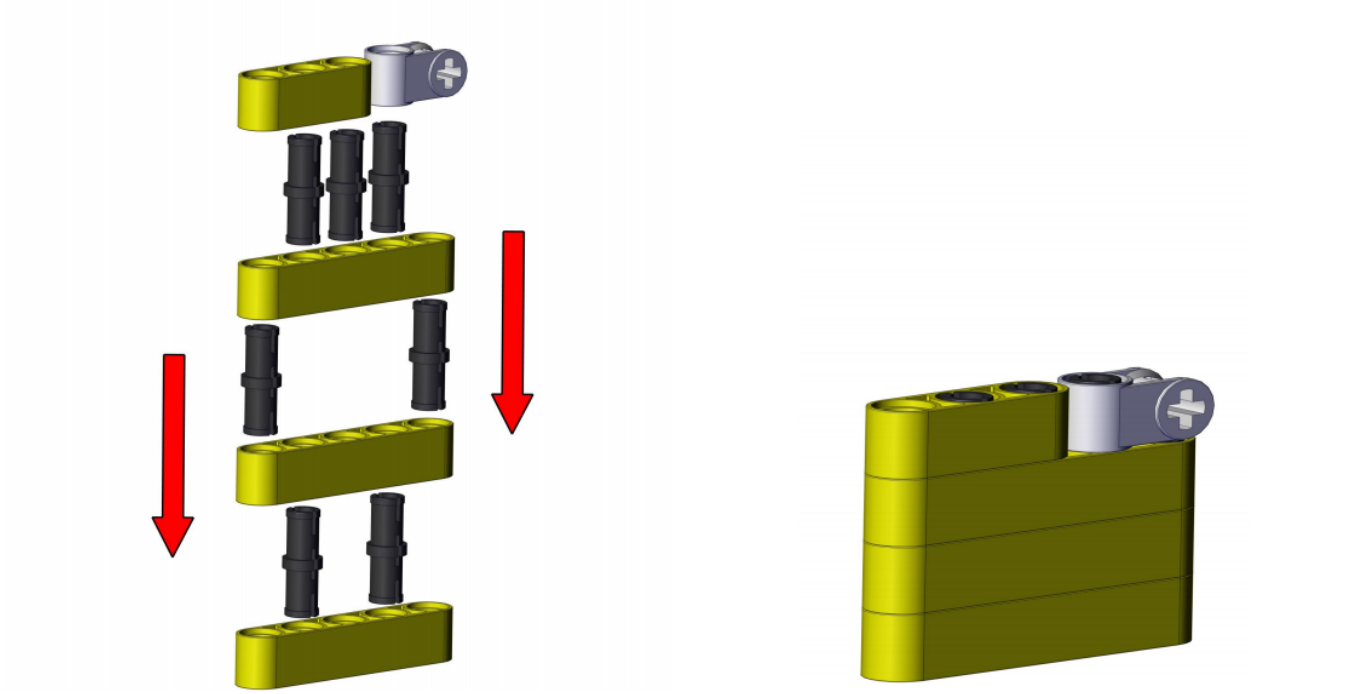


Wiring of servo

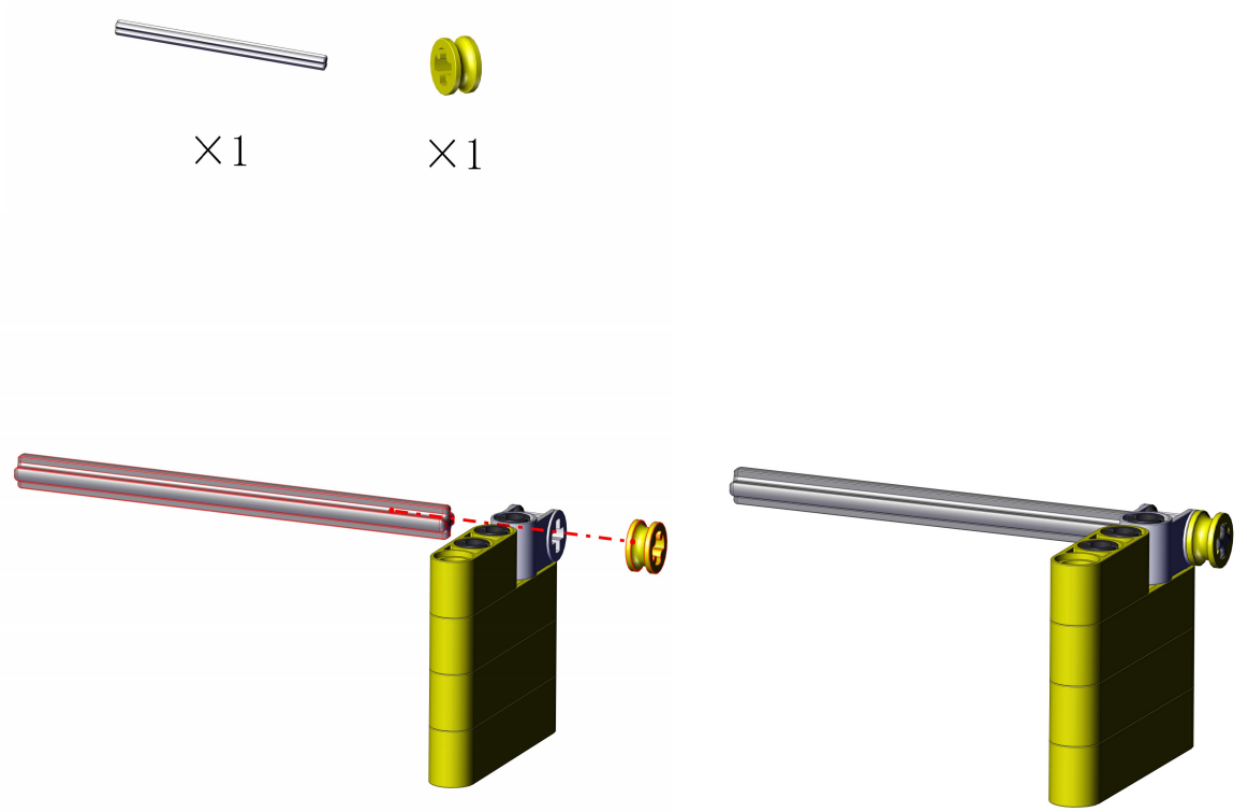
First write the following code in KidsBlock software and upload the code to the kidsIOT mainboard, then the servo will rotate 190° . (Note: If the servo can not rotate, you can press the RESET button on the kidsIOT board.)


when Arduino begin
set pin IO33 mode output
set pwm pin IO33 use channel CH2 (LT1) out 90
wait 0.5 seconds
set pwm pin IO33 use channel CH2 (LT1) out 190
wait 0.5 seconds

Process 17

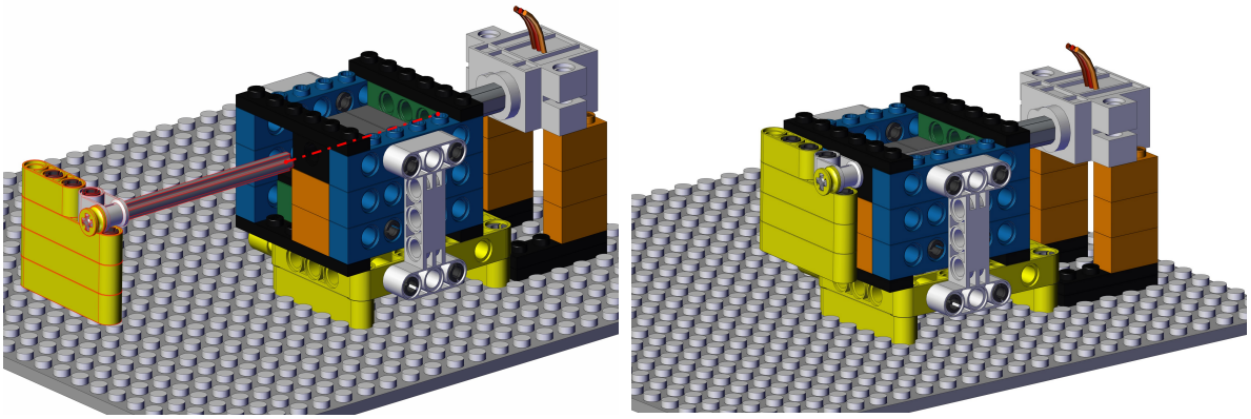


Process 18

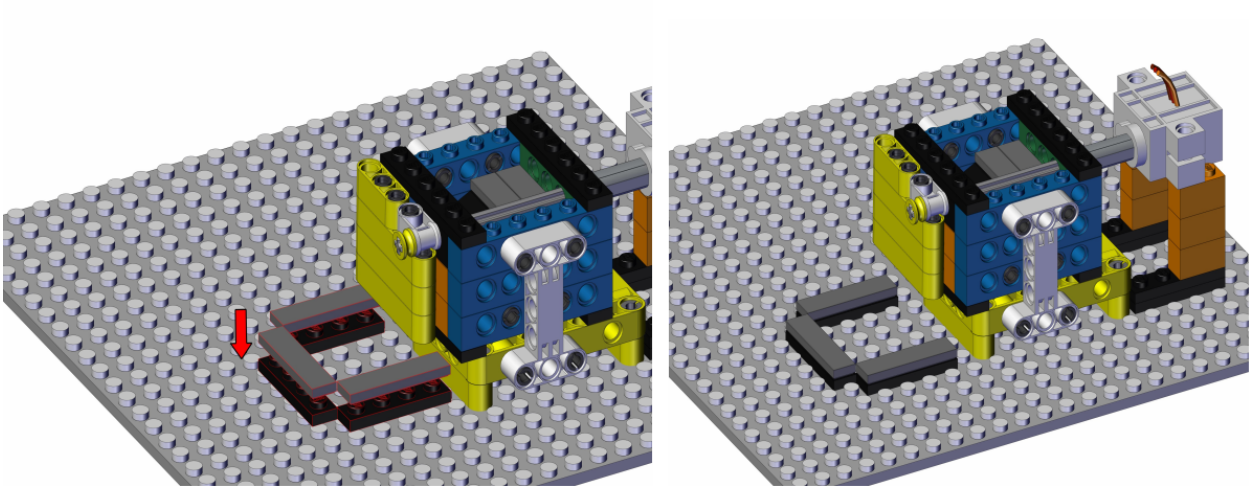


Process 19

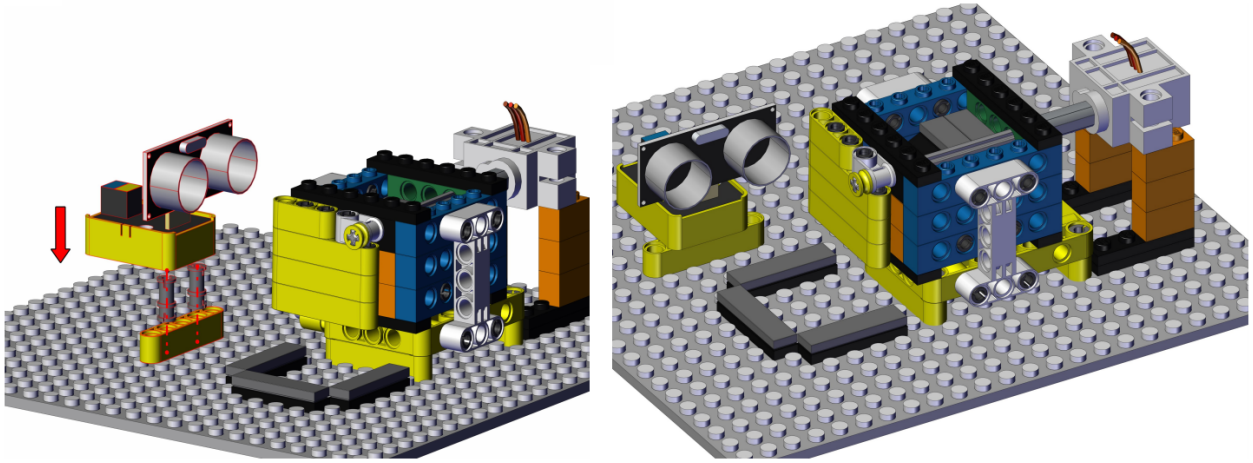
(Note: Do not twist the servo)



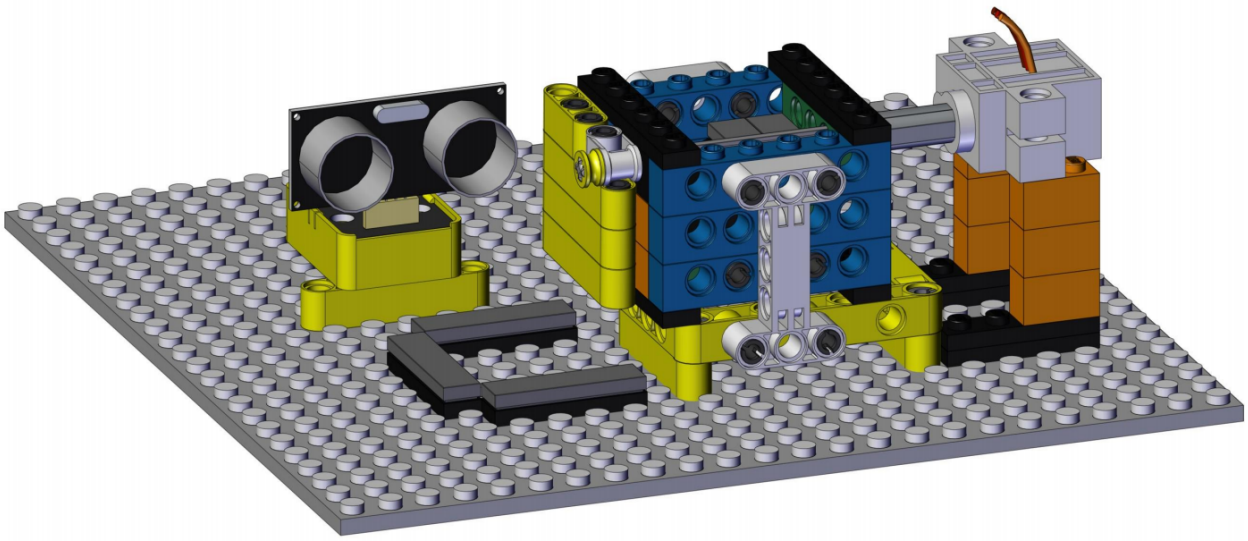
Process 20



Process 21



Complete

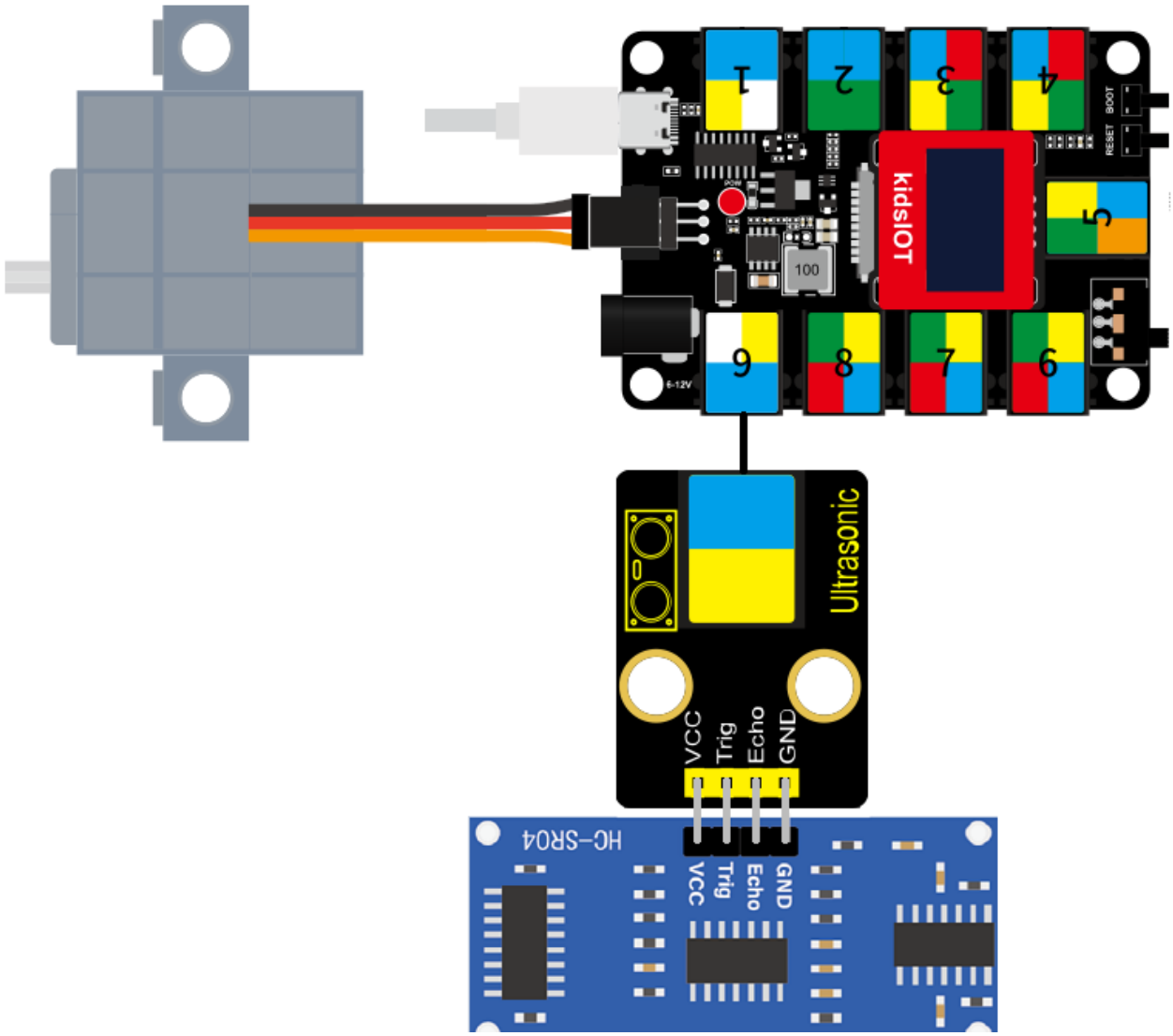


4. Wiring Diagram

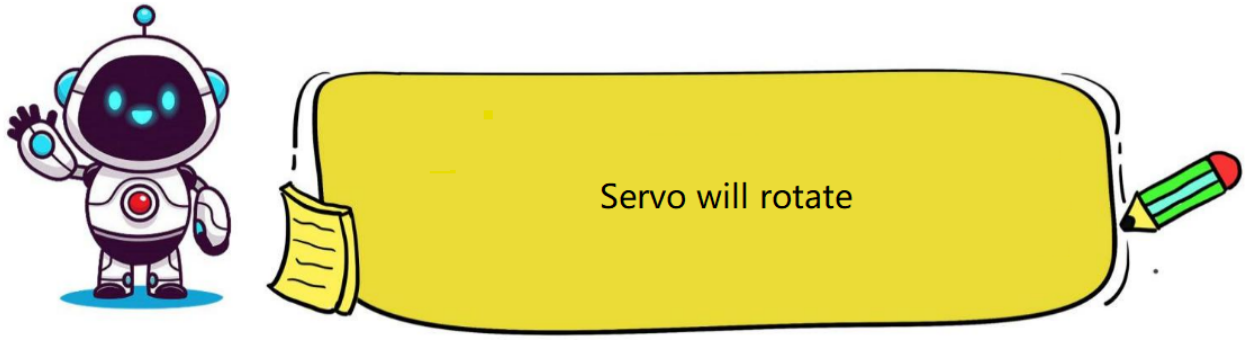
Module	kidsIOT Mainboard
Ultrasonic Adapter Board	No.9 portTrig–io18Echo–io19
Servo	G/V/io33 portBrown→GRed→VOrange→io33

Ultrasonic Sensor	Ultrasonic Adapter Board
Vcc	VCC
Trig	Trig
Echo	Echo
Gnd	GND

Connect the kidsIOT mainboard to your computer via USB cable.

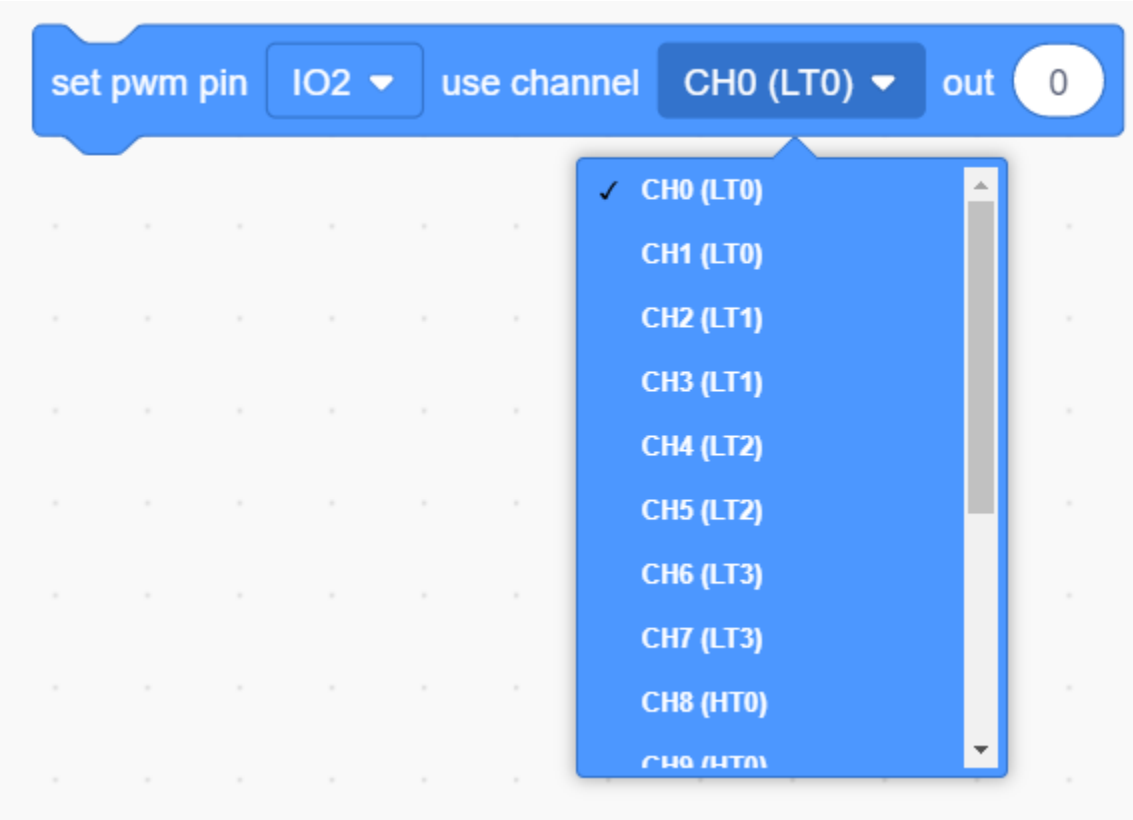


5. Servo rotation



(1). Programming Steps

Step 1Description of the Building Block

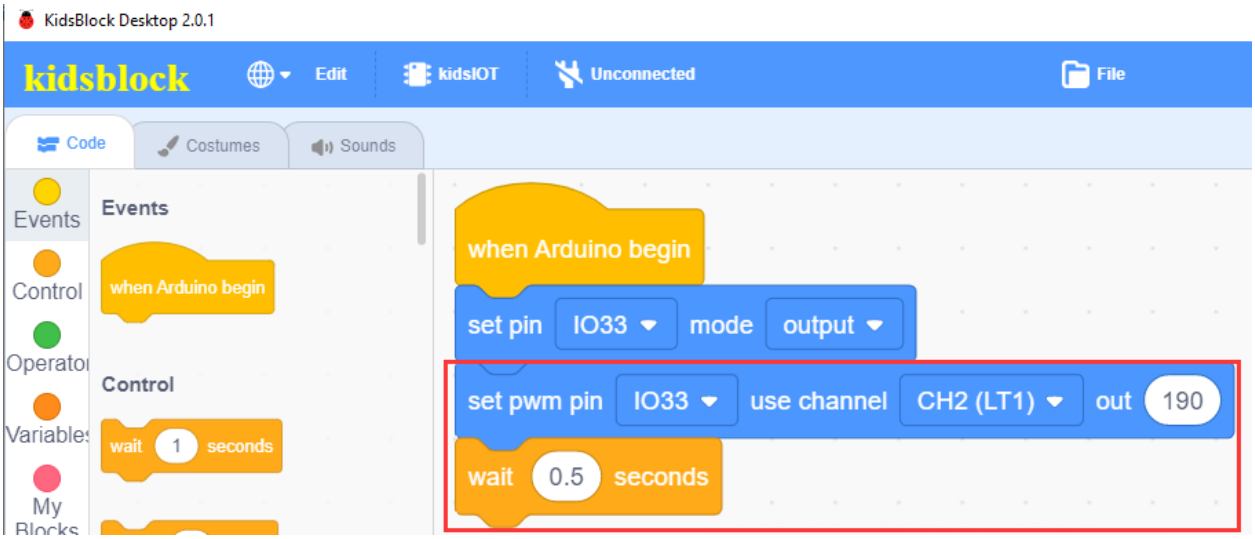


Set the servo's channel and output (rotation) angle to the specified PWM pin.

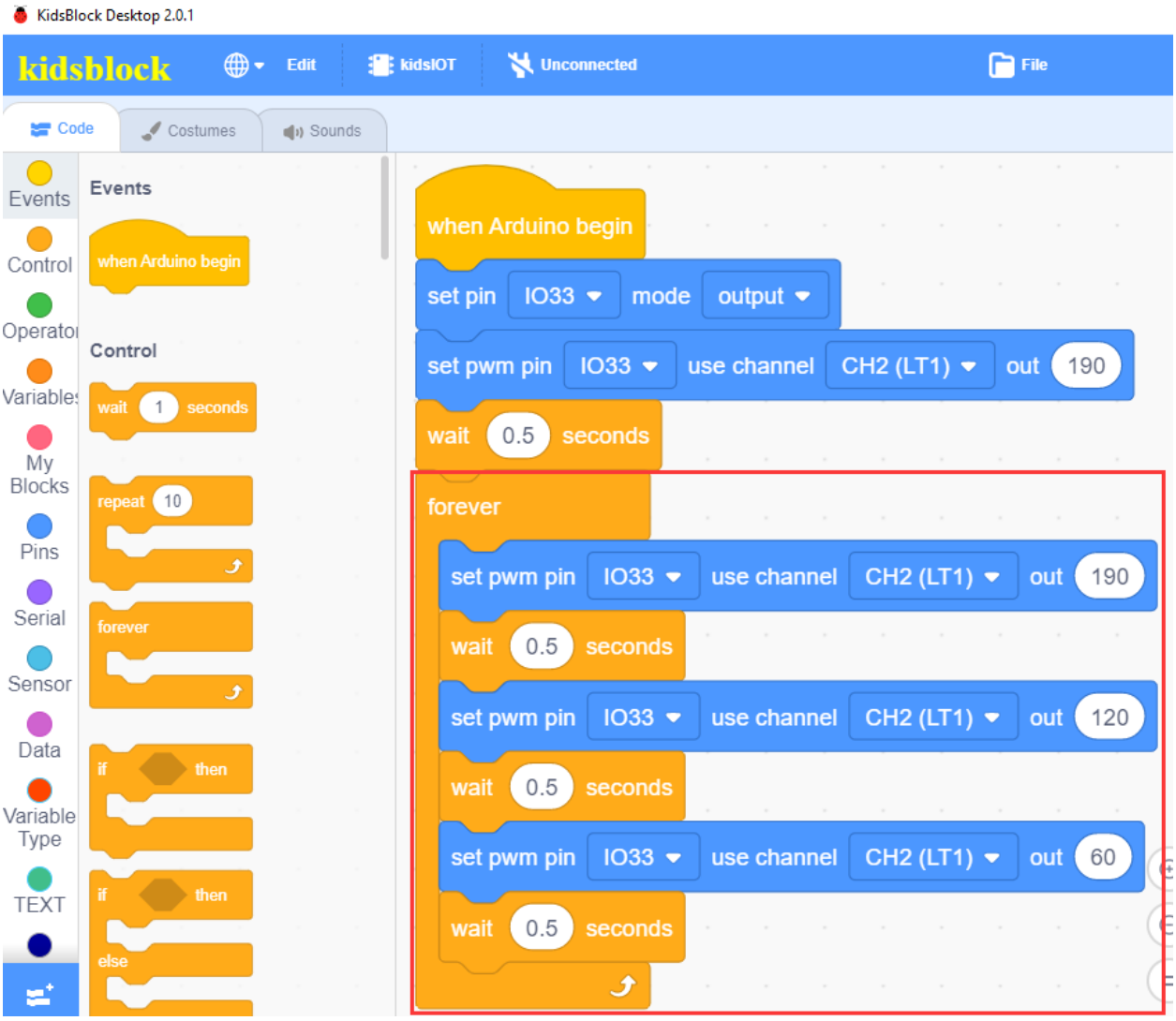
Step 2Write the Program

Set the pin IO33 (control pinio33) connected to the servo to "**Output**" mode.

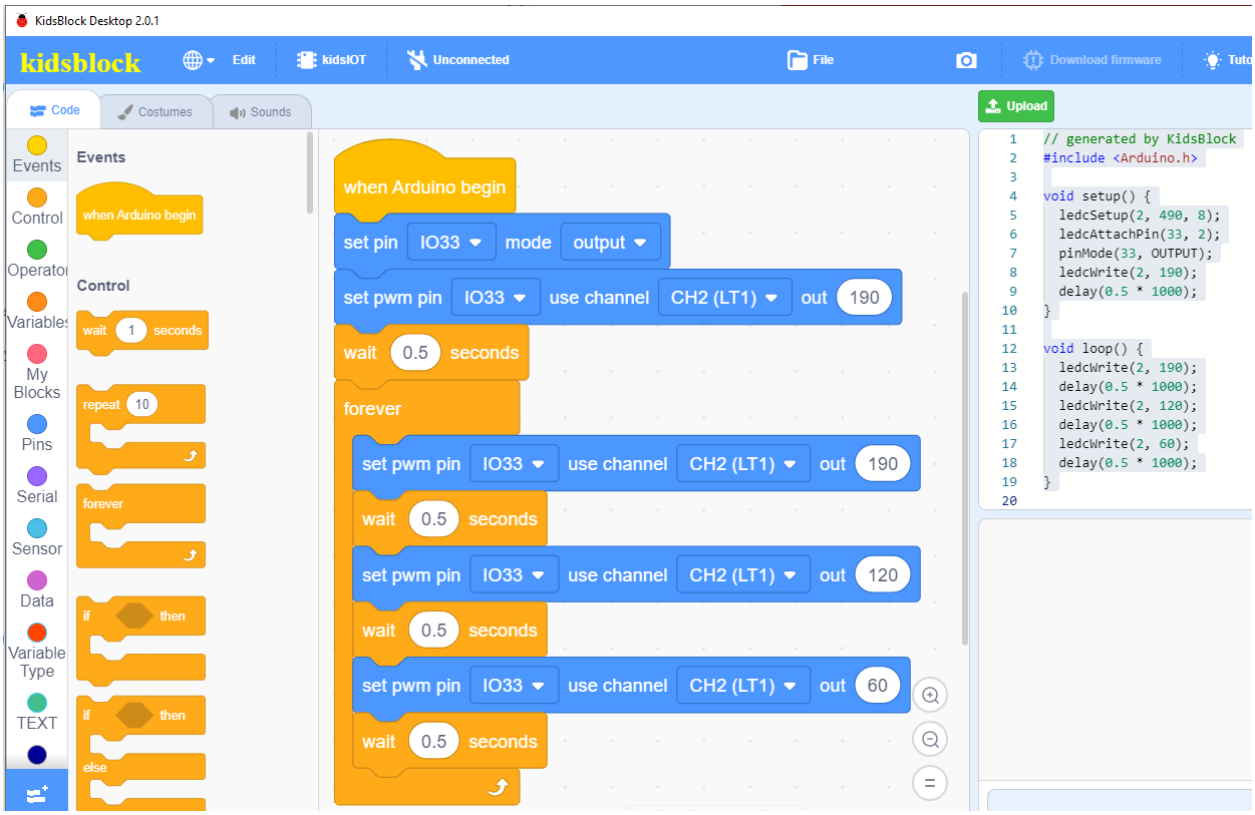
Initialize the control channel of the servo to CH2 (LT1) and the initial angle to 190°, with a delay of 0.5 seconds.



The servo rotates from 190° to 120° and then to 60° every 0.5 seconds.



Complete Program

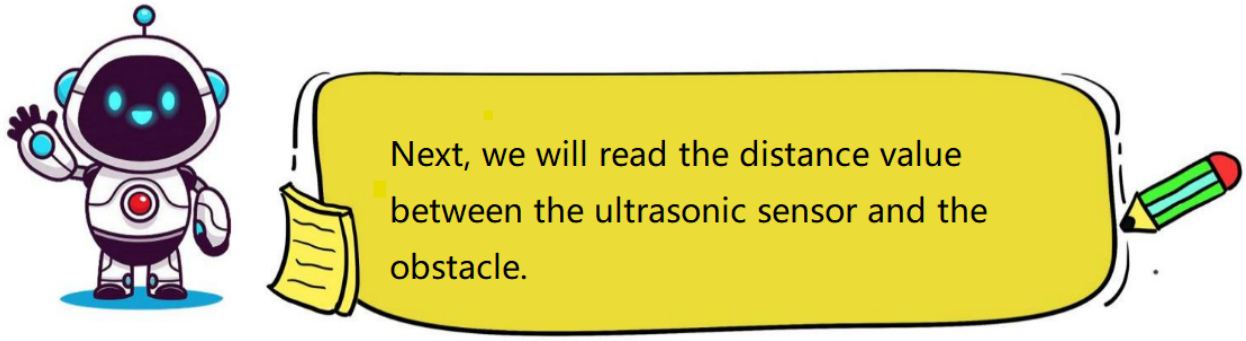


(2). Test Result

Click 

to upload the above complete code to the kidsIOT motherboard, then power up via the USB cable, then servo will rotate.

6. Read the Value of Ultrasonic Sensor

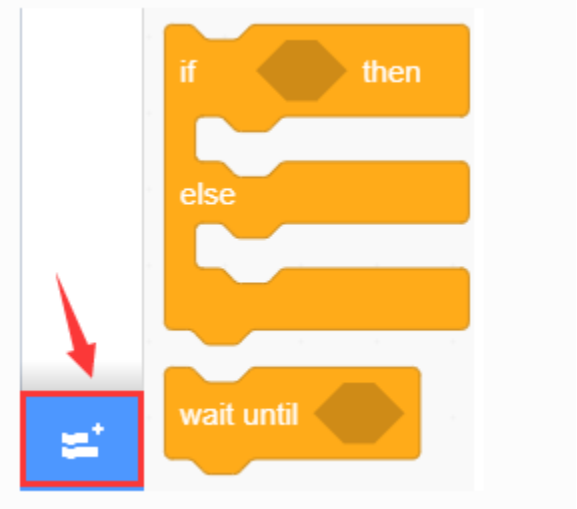


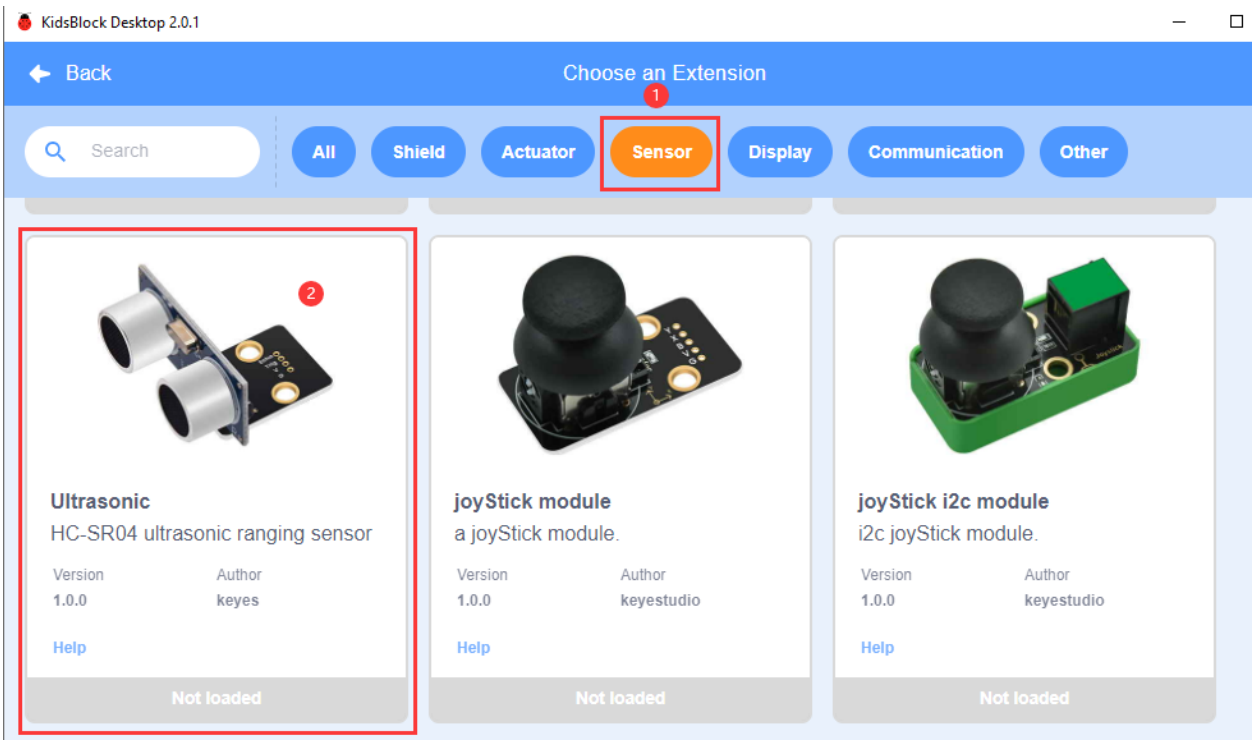
(1). Programming Steps

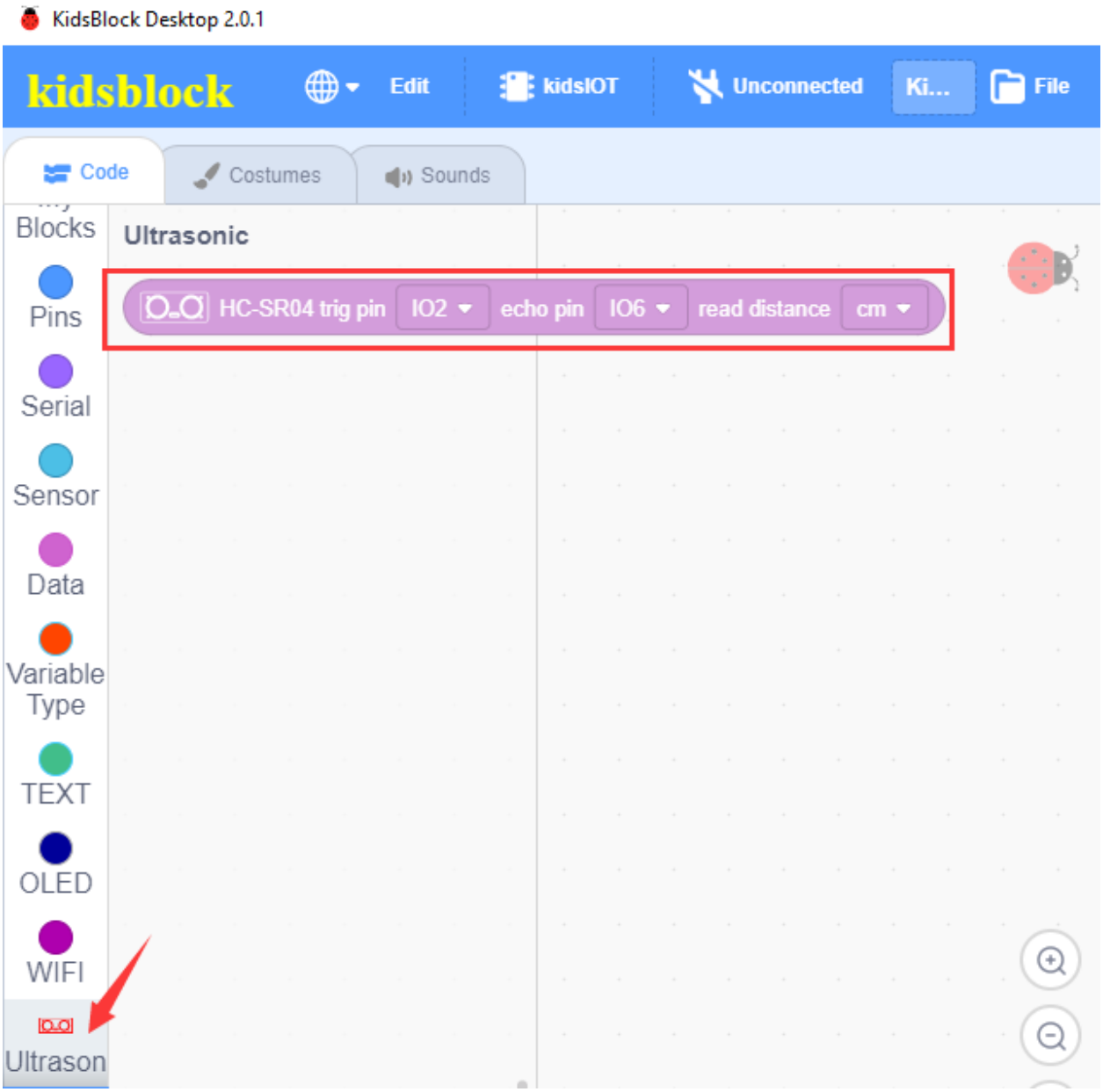
Step 1Add the Ultrasonic Sensor

Tap the "Sensor" module in the "Extension" , then select "**Ultrasonic Sensor**" and click 

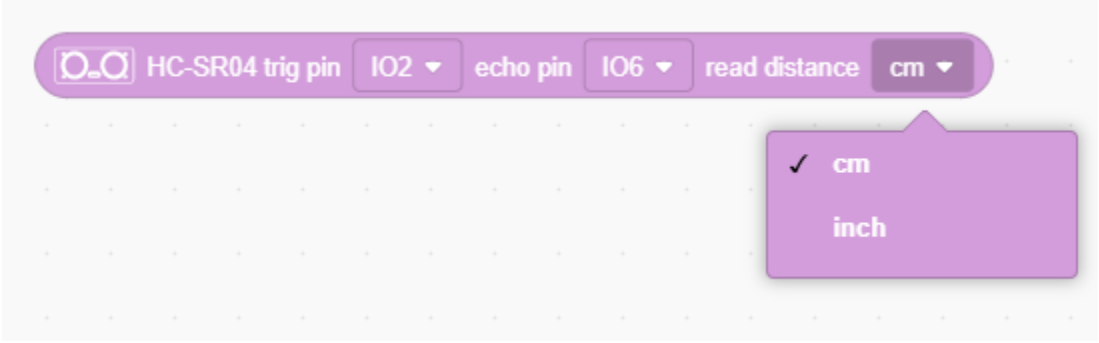
to return to the programming interface.







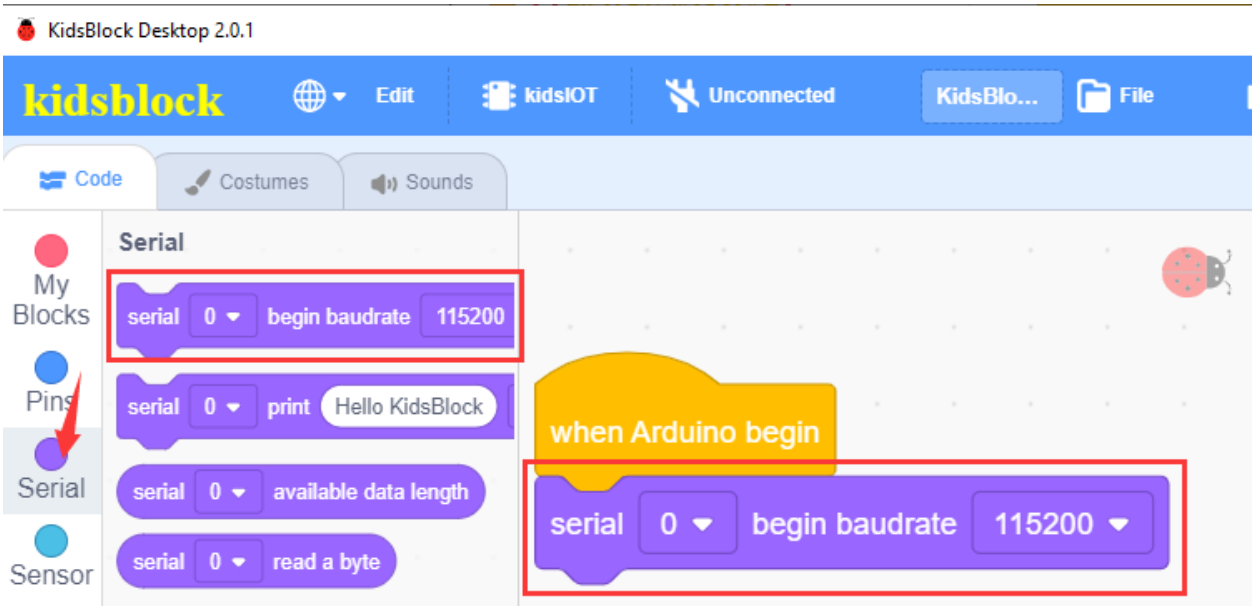
Step 2Description of the Building Block



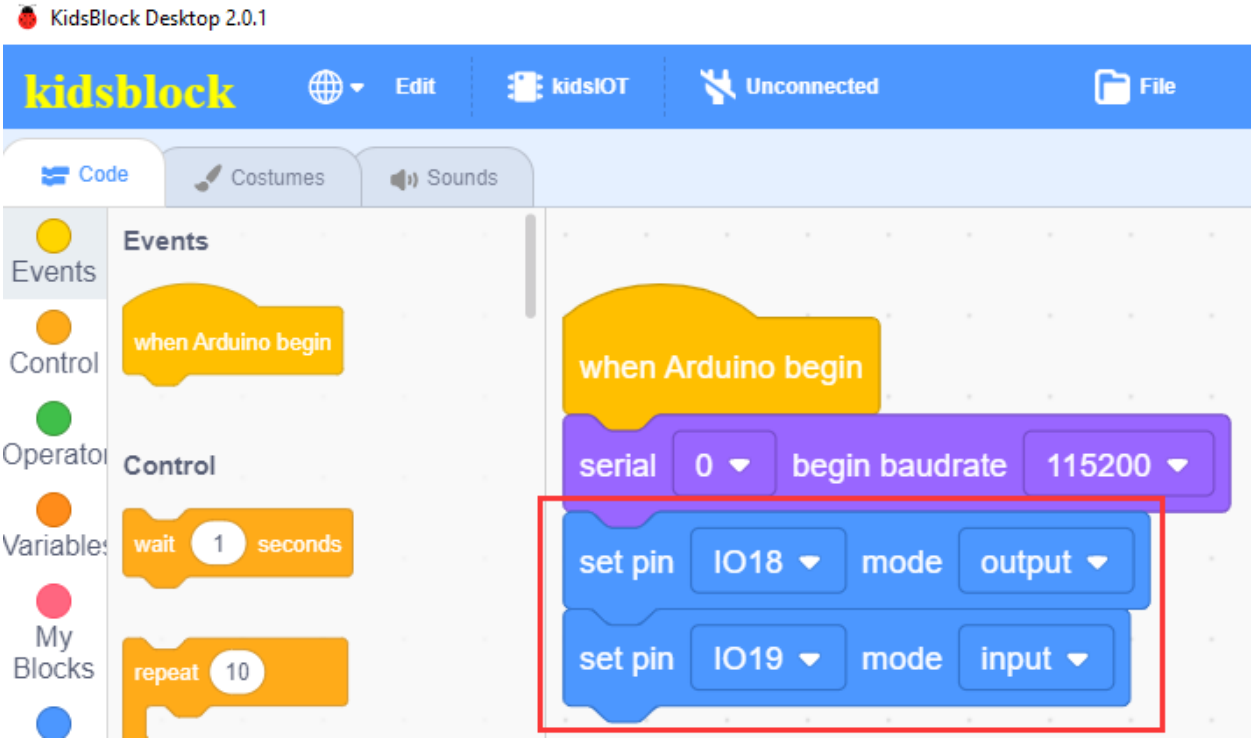
This block is used to measure distance to the specified pin, and the distance unit can be cm or inch.

Step 3Write the Program

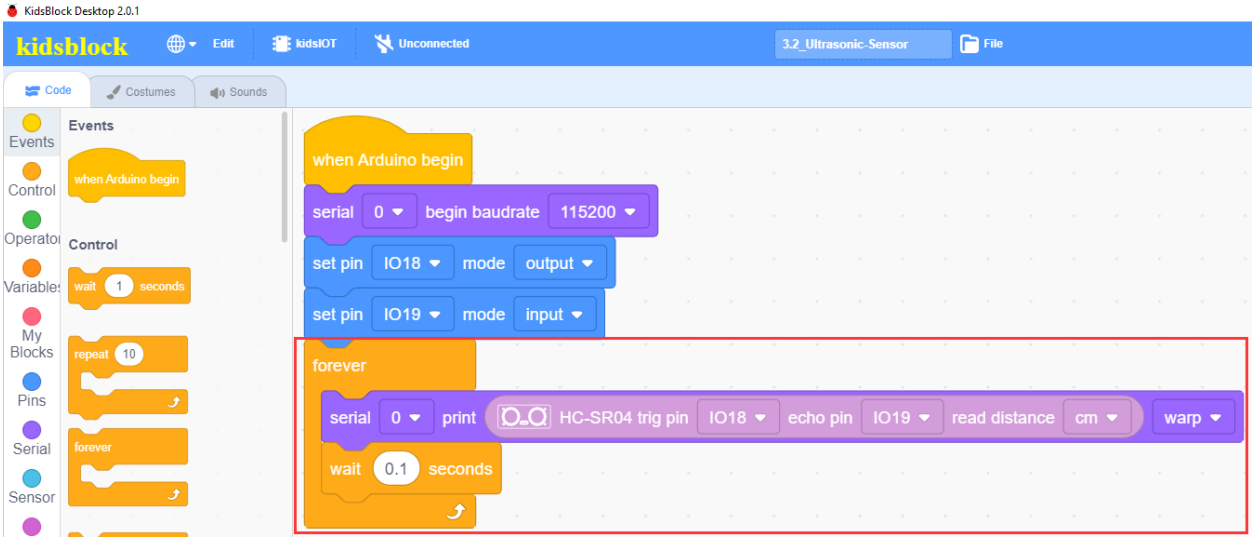
Set the baud rate to 15200.



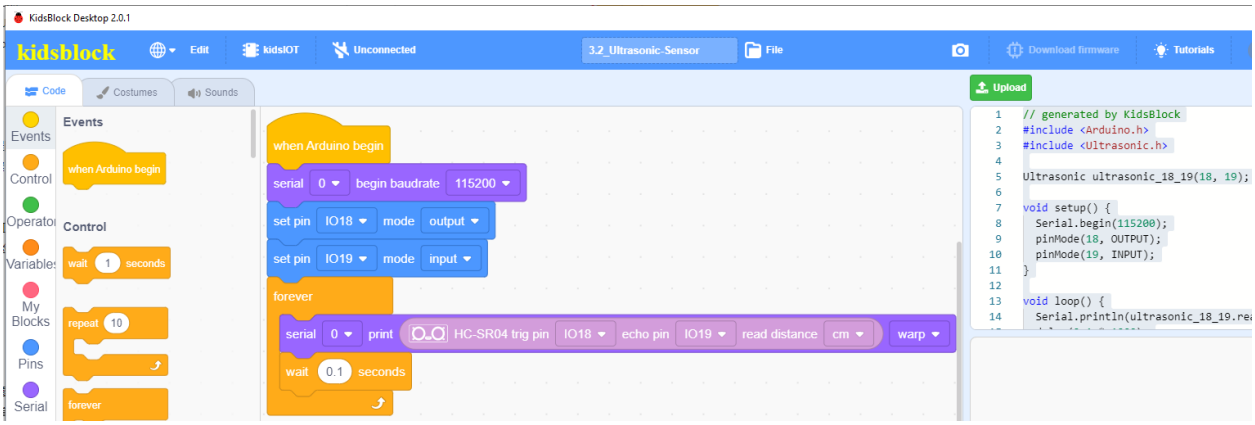
Set the Trig pin of the ultrasonic sensor IO18 to "**output**" mode, and the Echo pin IO19 to "**input**" mode.



Set Trig to IO18 and Echo to IO19, and the serial port prints the distance value detected by the ultrasonic sensor at 0.1 second intervals.



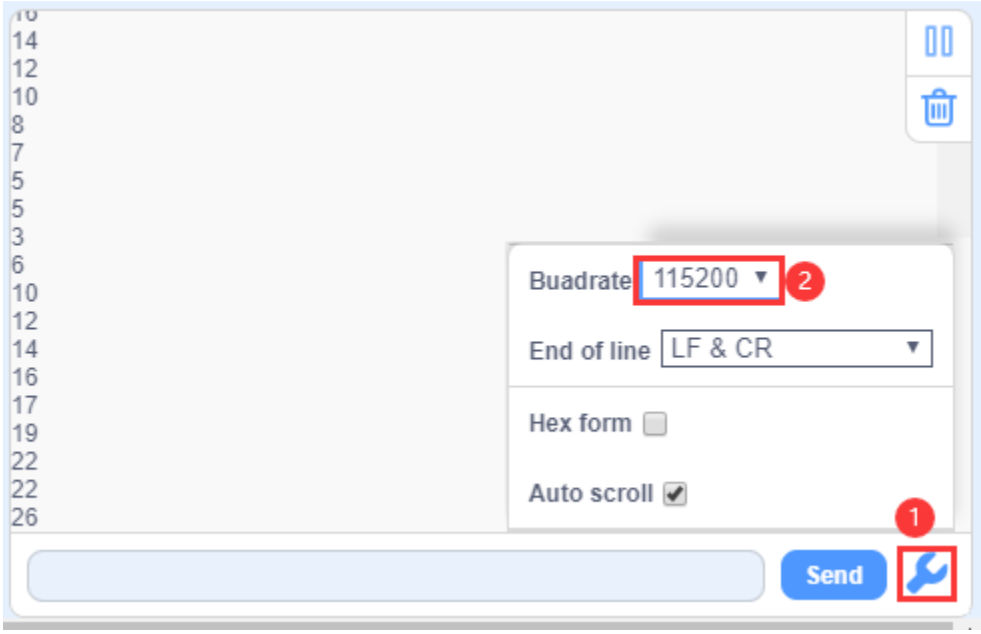
Complete Program



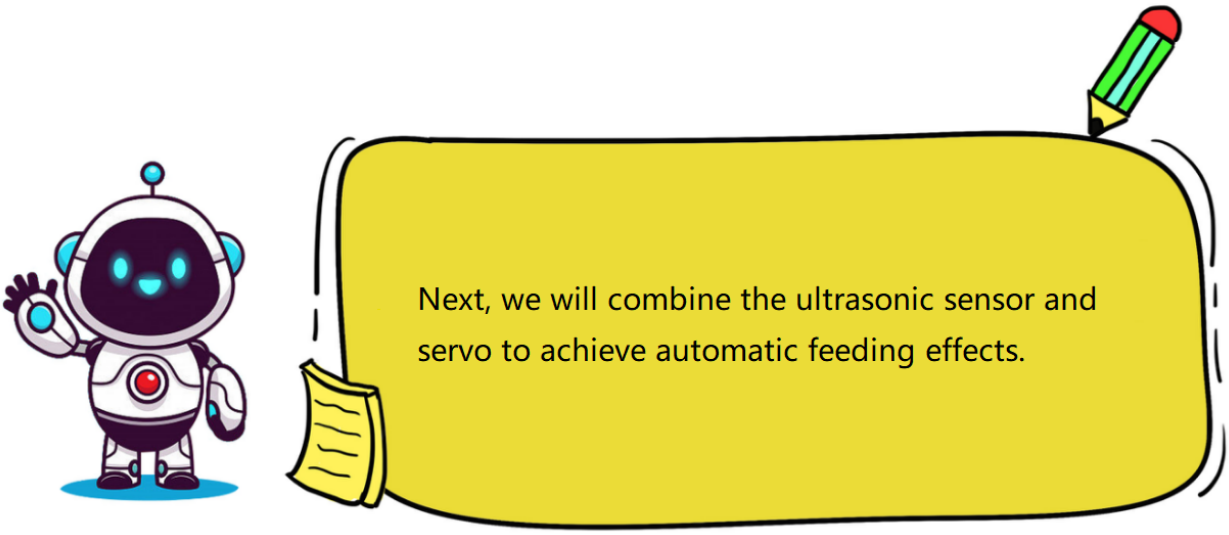
(2). Test Result

Click 

to upload the above complete code to the kidsIOT motherboard. After powering up via the USB cable, click  in the serial monitor and set the baud rate to 15200. Move your hand in front of the ultrasonic sensor. When you are close to it, the displayed distance value becomes smaller. When you move away from it, the value becomes larger.

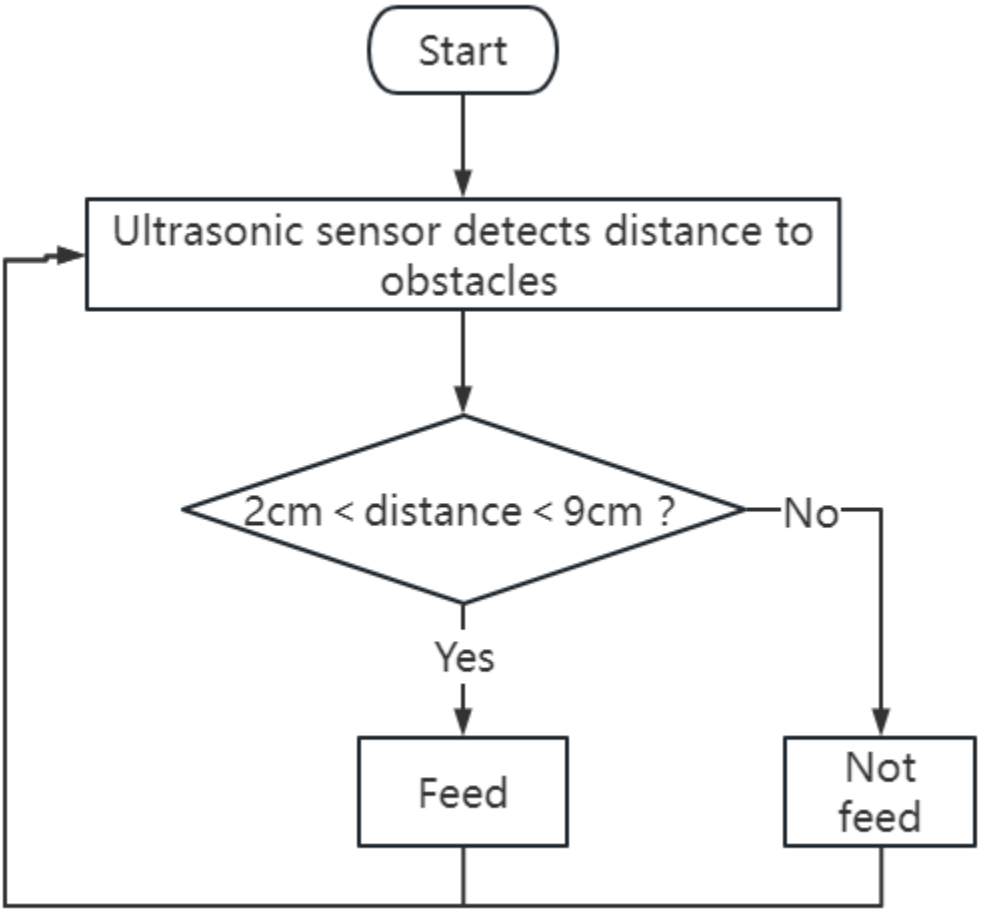


7. Automatic Feeding System



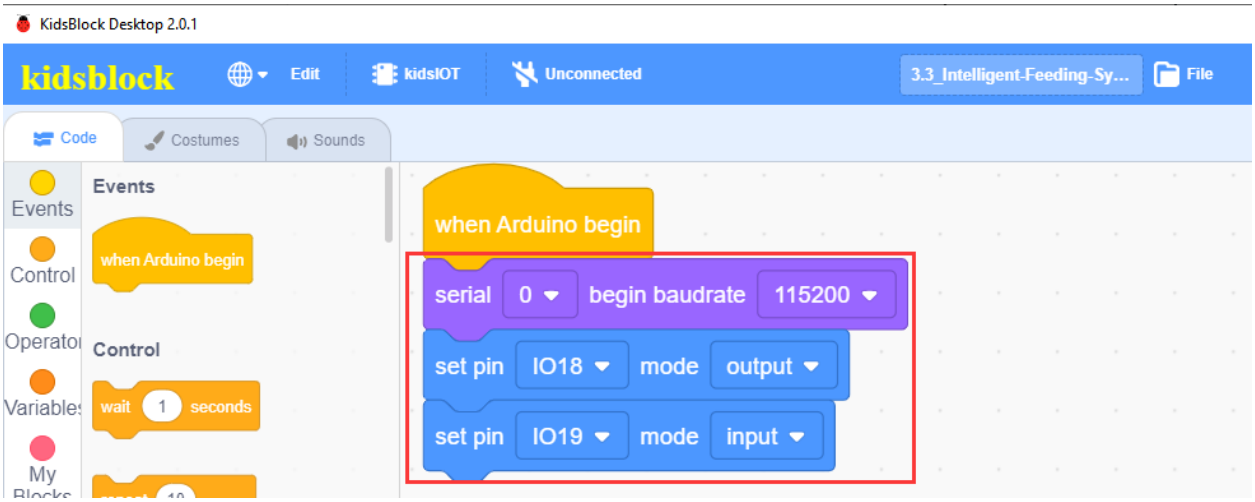
(1). Programming Steps

Step 1Flow Chart

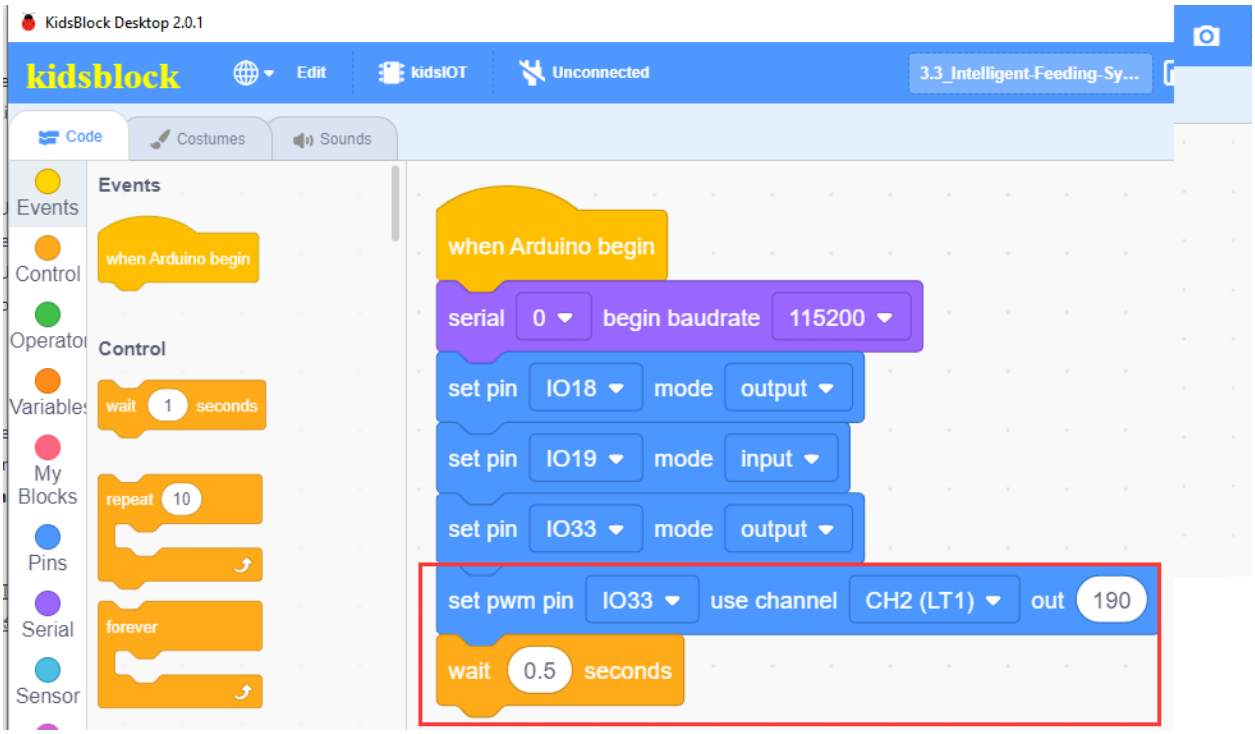


Step 2Programming Steps

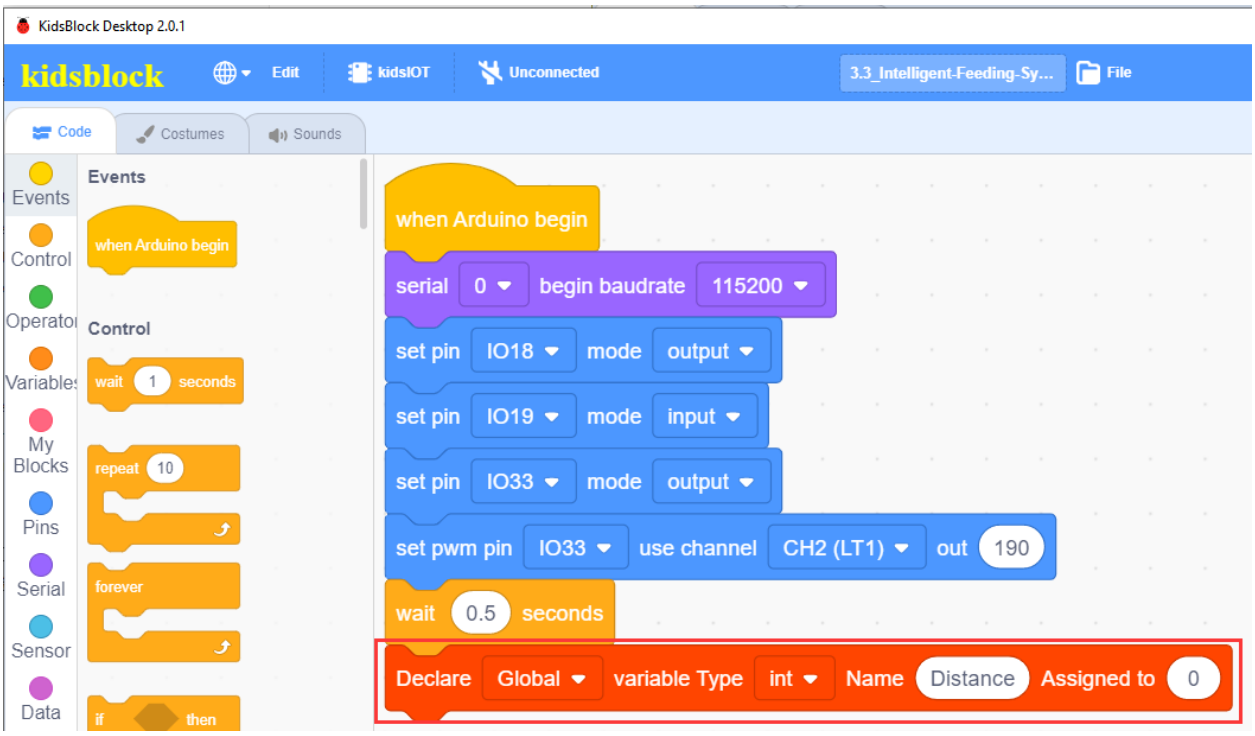
Set the baud rate to 15200, the Trig pin of the ultrasonic sensor IO18 to "**output**" mode, and the Echo pin IO19 to "**input**" mode.



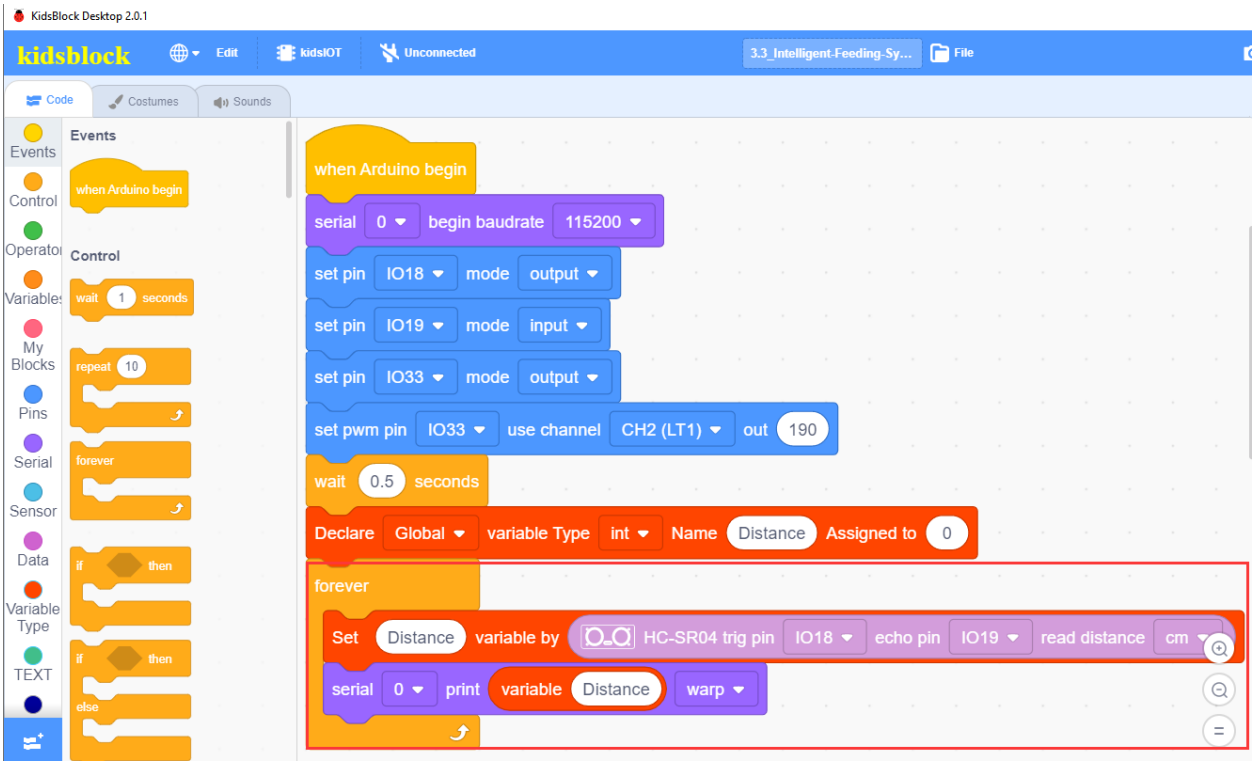
Set the pin IO33 connected to the servo to "**Output**" mode, initialize the control channel of the servo to CH2 (LT1) and the initial angle to 190°, delay 0.5 seconds.



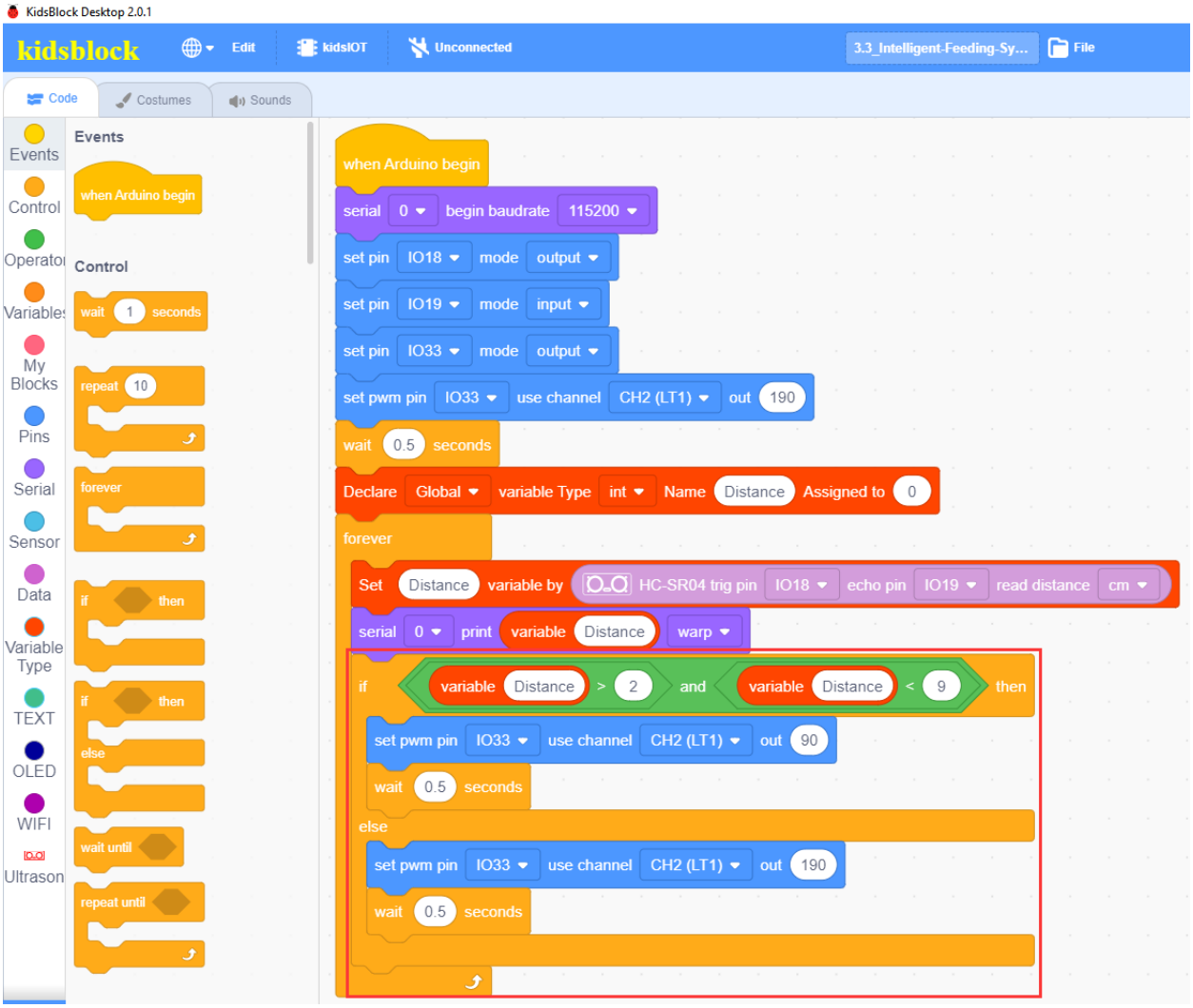
Define a "Distance" global variable to store the distance value detected by the ultrasonic sensor.



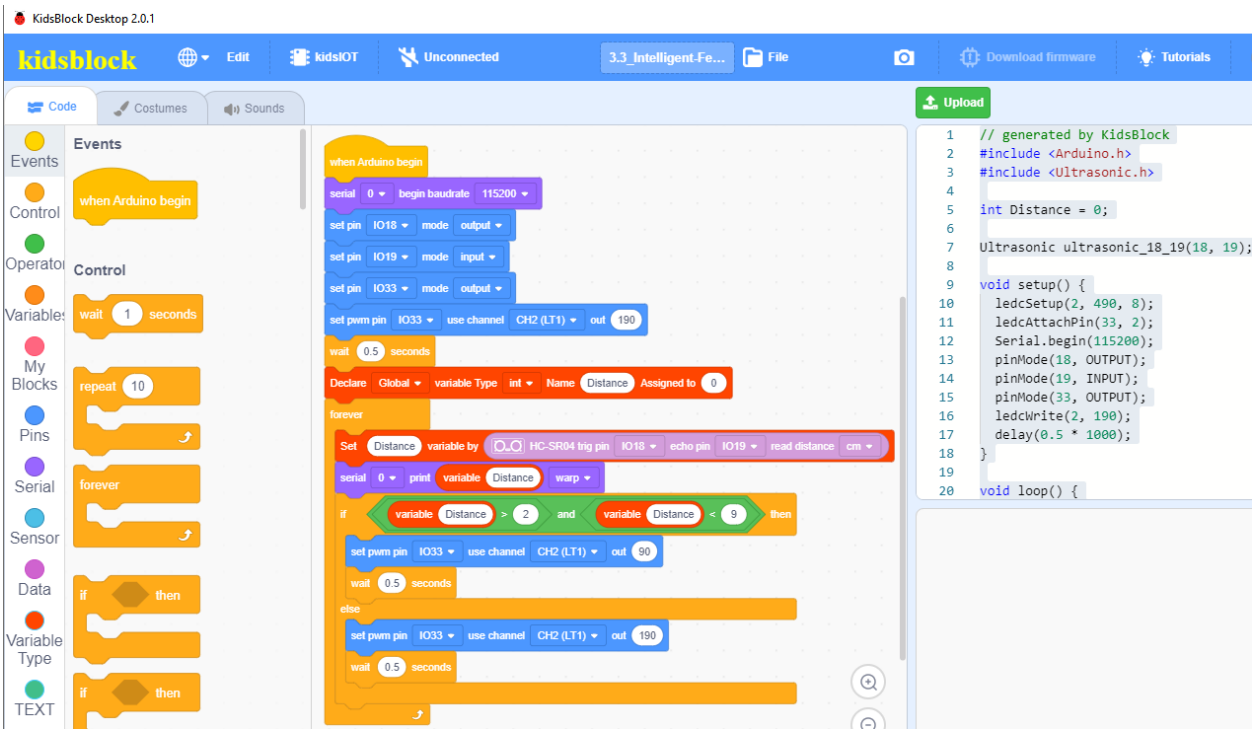
Set the Trig pin and Echo pin of the ultrasonic sensor, and print the read distance value of the ultrasonic sensor on the serial port.



Determine the distance detected by the ultrasonic sensor. If 2cm < distance value < 7cm, the feed box will be opened; otherwise, it will be closed.

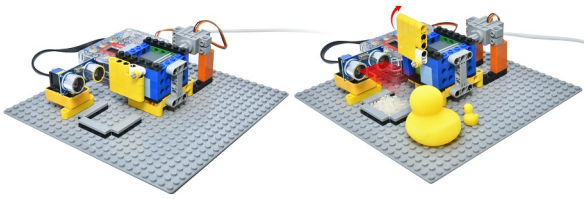


Complete Program



(2). Test Result

Click  to upload the above complete code to the kidsIOT motherboard. After powering up via the USB cable, click  in the serial monitor and set the baud rate to 15200. If an animal is detected within 2cm-9cm, the feed box will be opened to feed the animal.



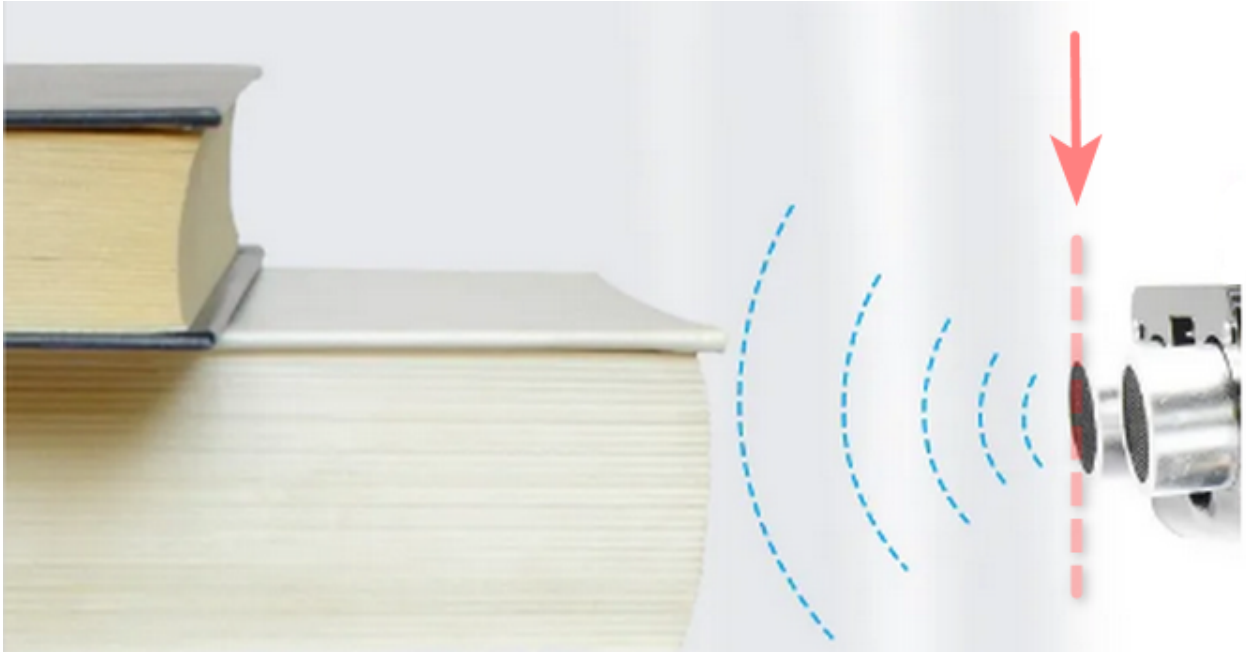
8. Common Problems

Q1: Why doesn't the servo work?

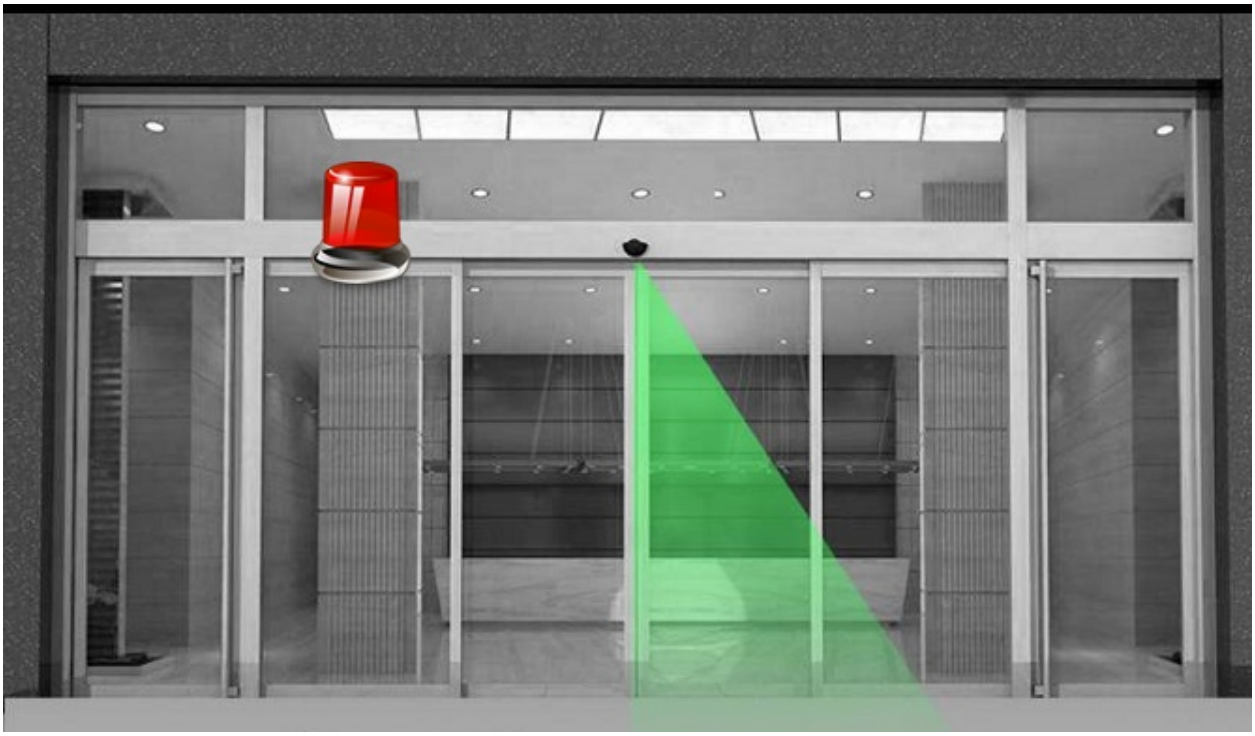
A: It may be stuck. Before assembling the servo, use the code to adjust it to 80°.

Q2: Why is the detection distance inaccurate when using the ultrasonic sensor?

A: Measurement should be started from the transmitting head of the ultrasonic sensor. This module is not a high-precision ultrasonic distance detection module and may exist errors.



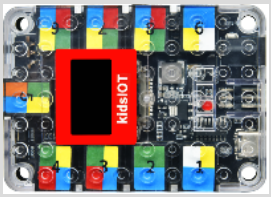
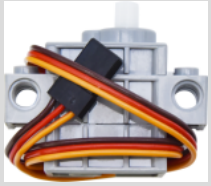
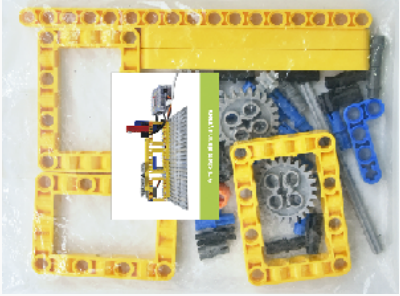
4.3.4 Project 04: Anti-theft Alarm System



1. Description

The anti-theft alarm system is composed of a PIR motion sensor, a buzzer, a LED and a kidsIOT mainboard. When programming via the KidsBlock, you are able to judge whether someone is moving by reading the digital signal detected by the PIR motion sensor. If someone is moving, the buzzer will sound an alarm and the LED will flash to alert the user that someone has entered the area. Thus, a low-cost anti-theft alarm system can be realized, which is suitable for homes or offices.

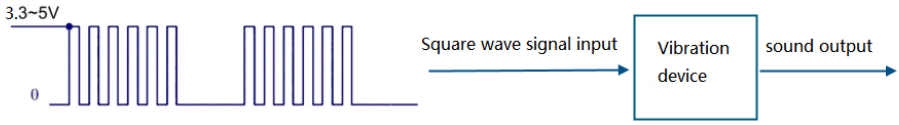
2. Components

			
kidsIOT Mainboard×1	PIR Motion Sensor×1	Passive Buzzer×1	Servo×1
			
Wire×2	USB Cable×1	Anti-theft Alarm System LEGO Pieces×1	

About Passive Buzzer and PIR Motion Sensor

Buzzer: It is able to emit sounds of different frequencies and durations and is powered by DC voltage. What's more, it can be widely used in computers, alarms, electronic toys, automotive electronic equipment and so on to issue reminders or alarm sounds. There are active buzzers and passive buzzers.

Passive buzzer: Only by giving it "square wave signals" of a certain frequency can it make sound. Different square wave frequencies will produce different sounds (it can simulate tunes to achieve musical effects).



Note: The resonant frequency of the passive buzzer we provide is 2048Hz, and the sound is the loudest at this time.

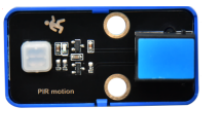
Parameters:

Working voltage: DC 3.3V-5V
Working current: (Max)30mA@5V
Maximum power: 0.15mW
Resonant frequency: 2048Hz
Output sound pressure: (Min)80dB/10cm
Control signal: square wave

Active buzzer: When connected to DC power supply, it will automatically emit sound (the frequency of the sound is fixed), and the sound is single.

PIR Motion Sensor: It is a sensor that detects infrared rays emitted by people or animals so as to output digital signals(1 or 0).

Parameters:

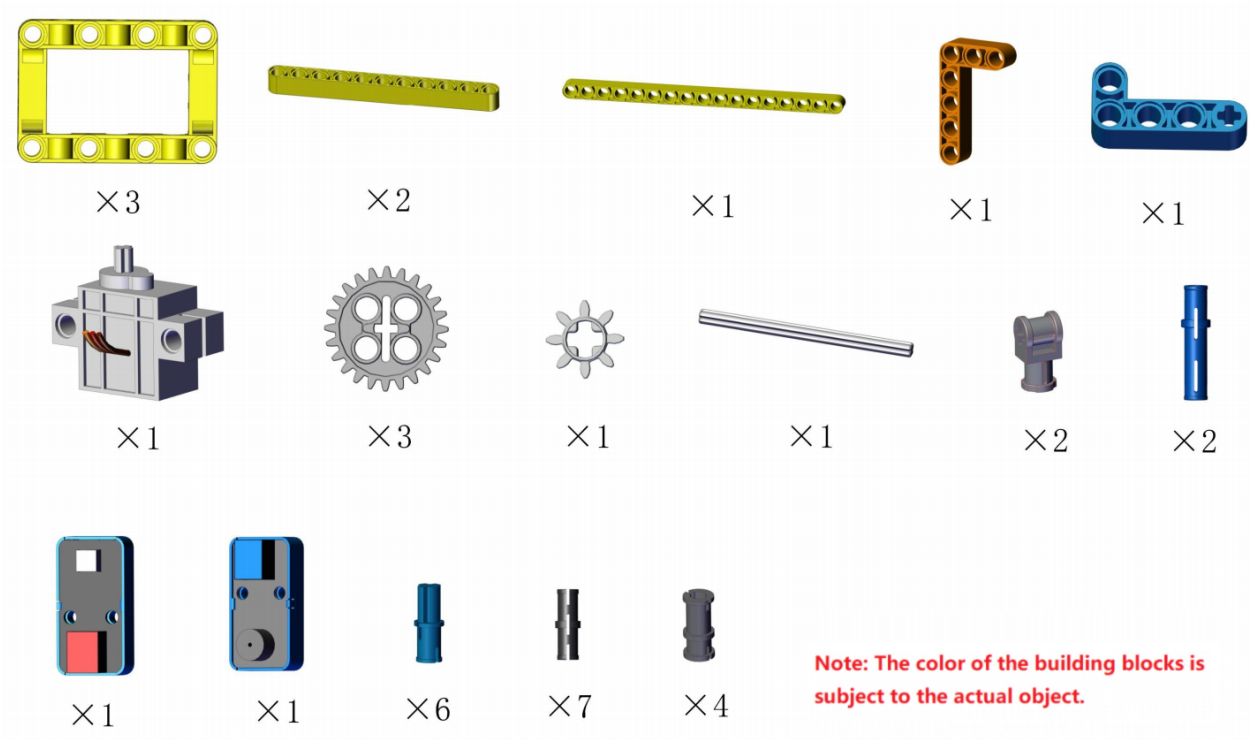


Working voltage: DC 3.3V~5V
Working current: (Max)3.6mA@5V
Maximum power: 18mW
Viewing angle: Y = 90°, X = 110°
Detection distance: ≤5m
Control signal: digital signals(1 or 0)

Note: It should be away from the direct irradiation of sunlight, car headlights and incandescent lamps, or heaters and air conditioners so as to avoid the wrong alarms caused by the environmental temperature changes.

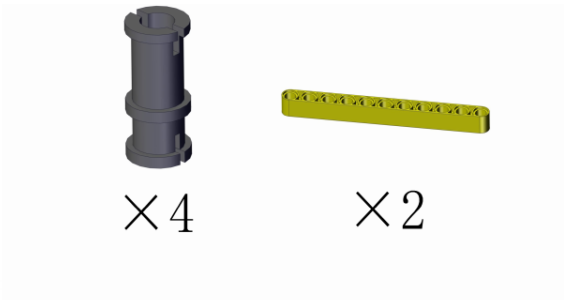
3. Assembly Steps

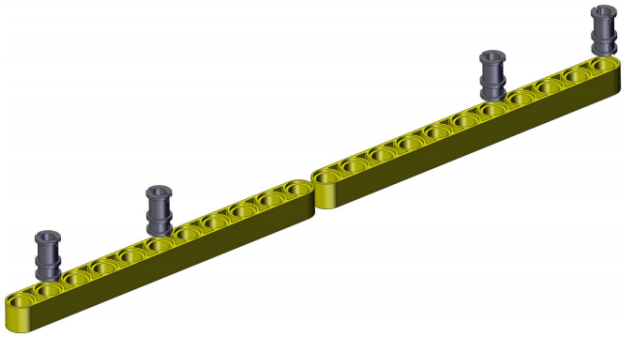
Step 1Components Needed

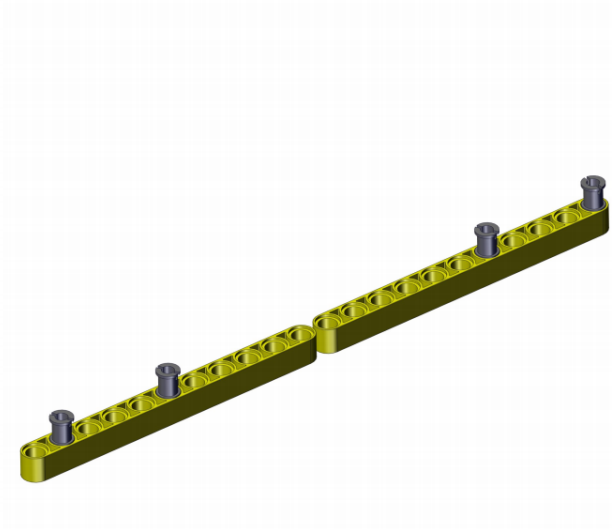


Step 2Process

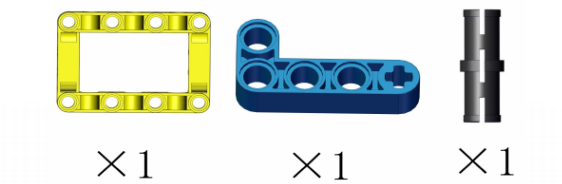
Process 1

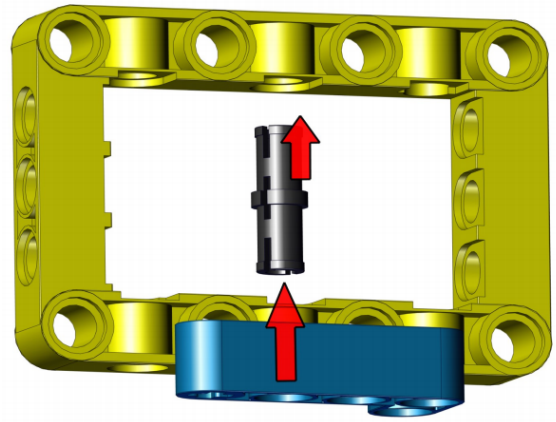


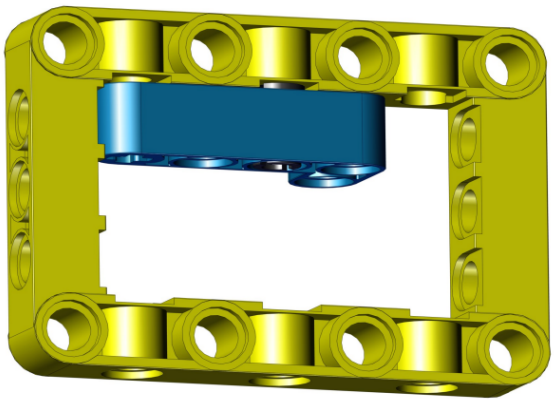




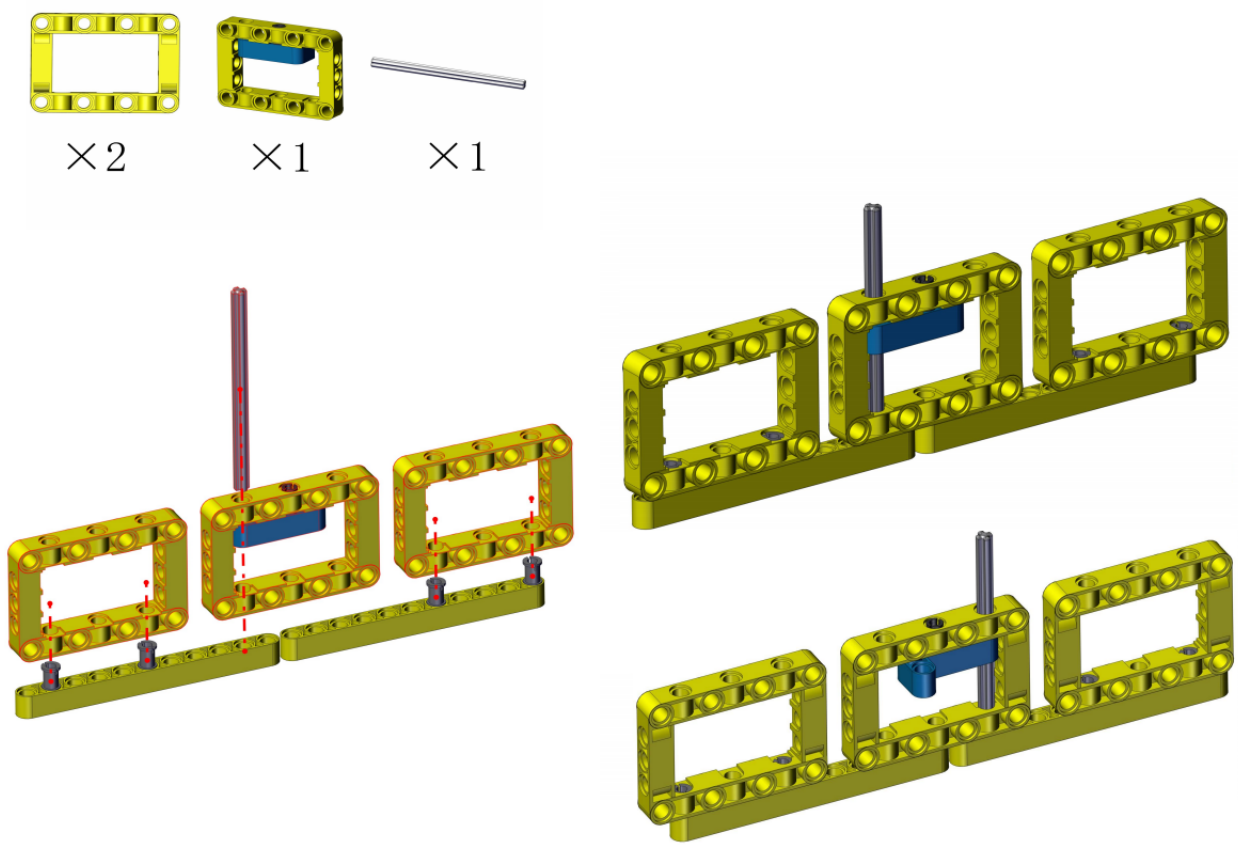
Process 2



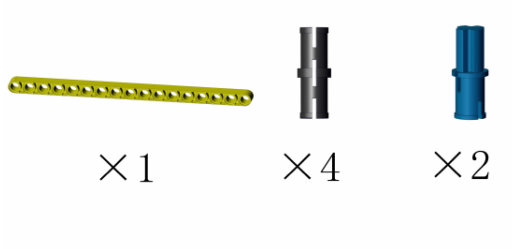


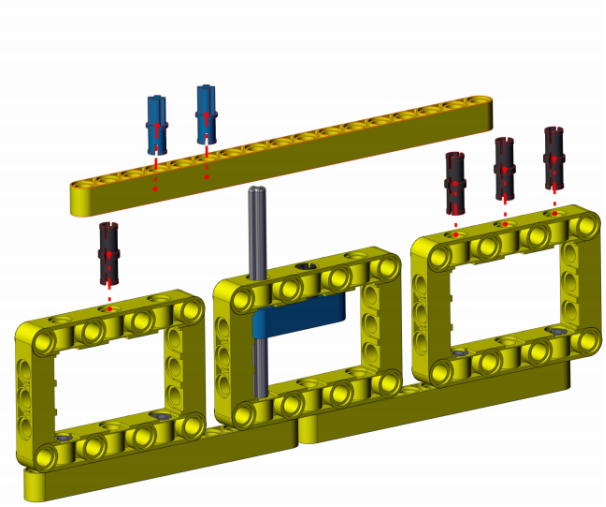


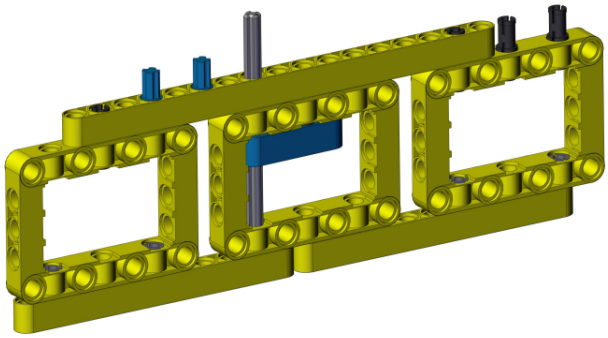
Process 3



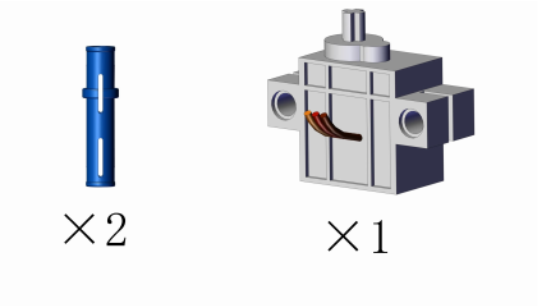
Process 4

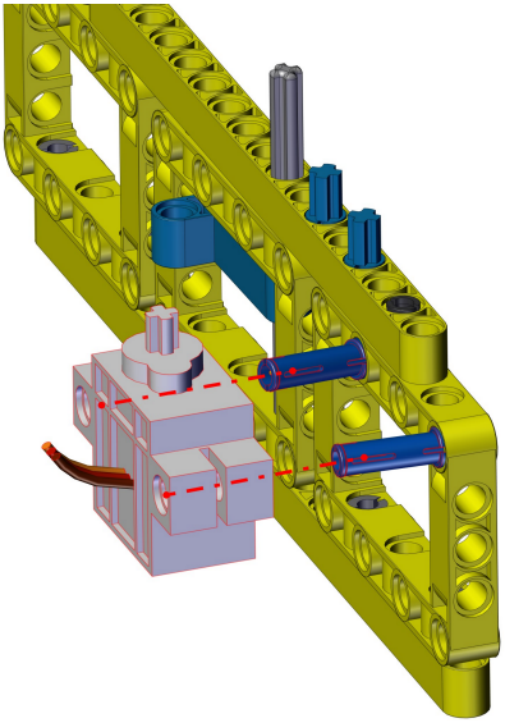


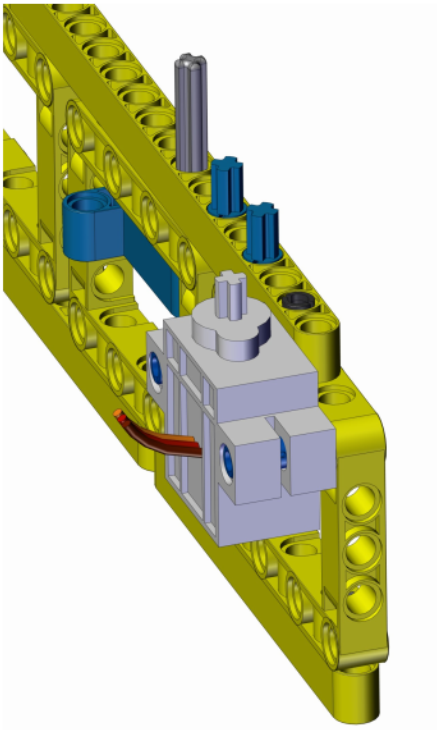




Process 5

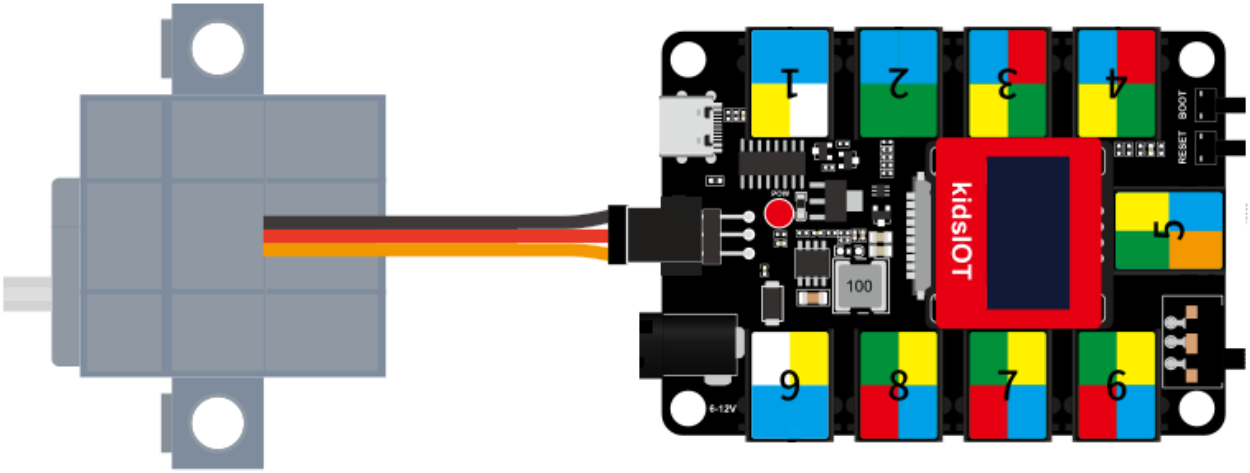




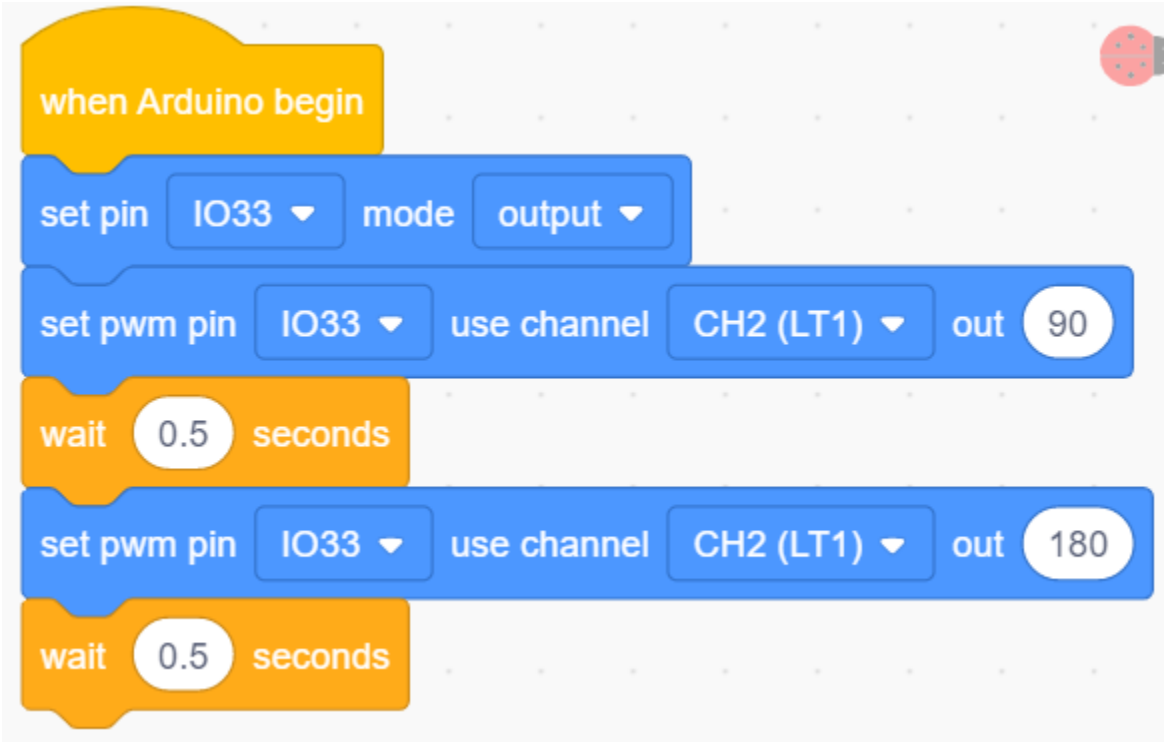


Process 6Initialize the servo angle

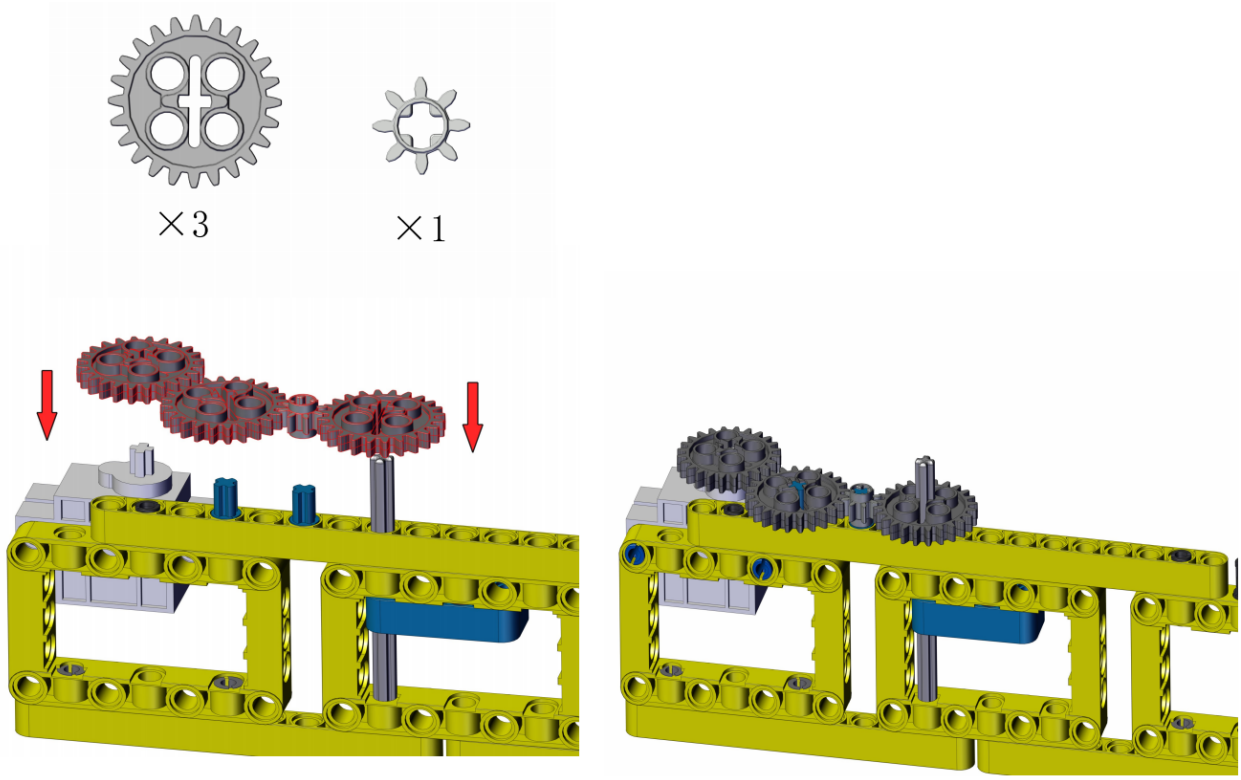
Wiring of servo(it is the same as project 03)



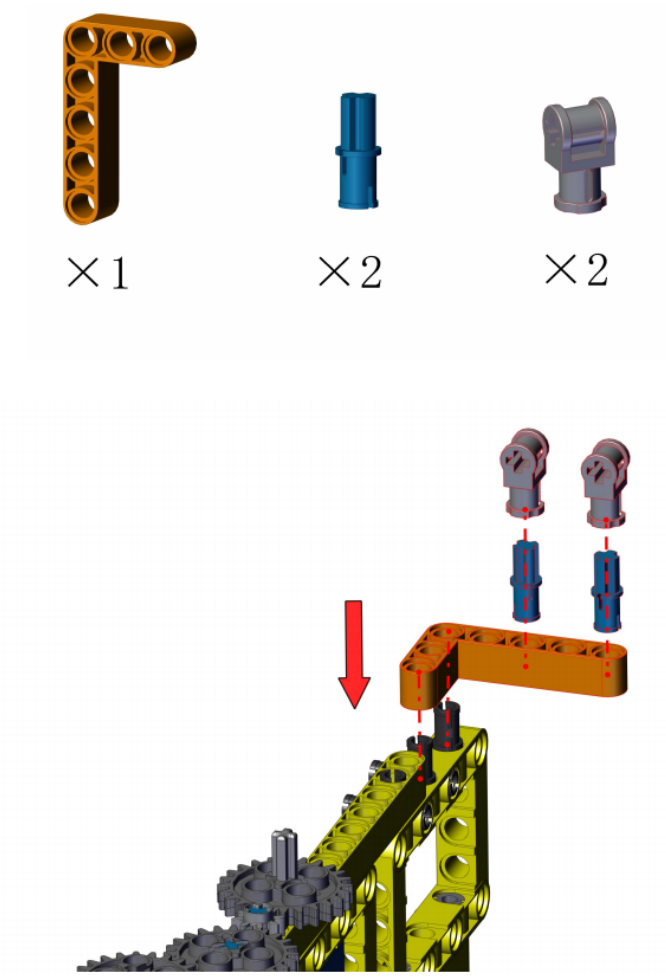
First write the following code in KidsBlock software and upload the code to the kidsIOT mainboard, then the servo will rotate 180° . (Note: If the servo can not rotate, you can press the RESET button on the kidsIOT board.)

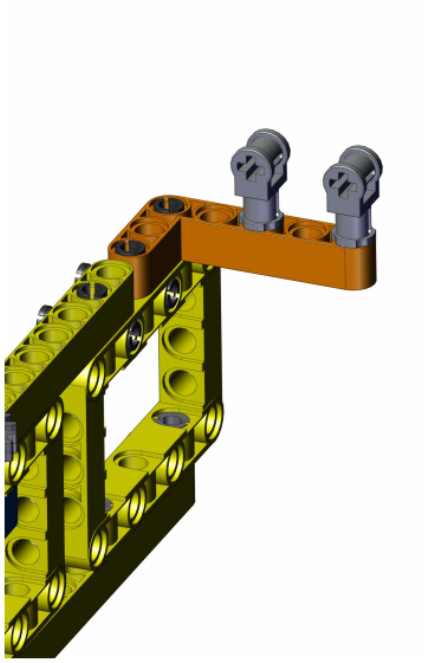


Process 7(Place the three Lego boxes on the same side, then assemble the four gears.)

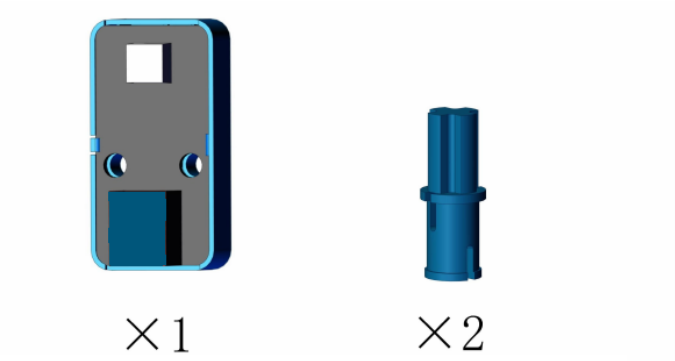


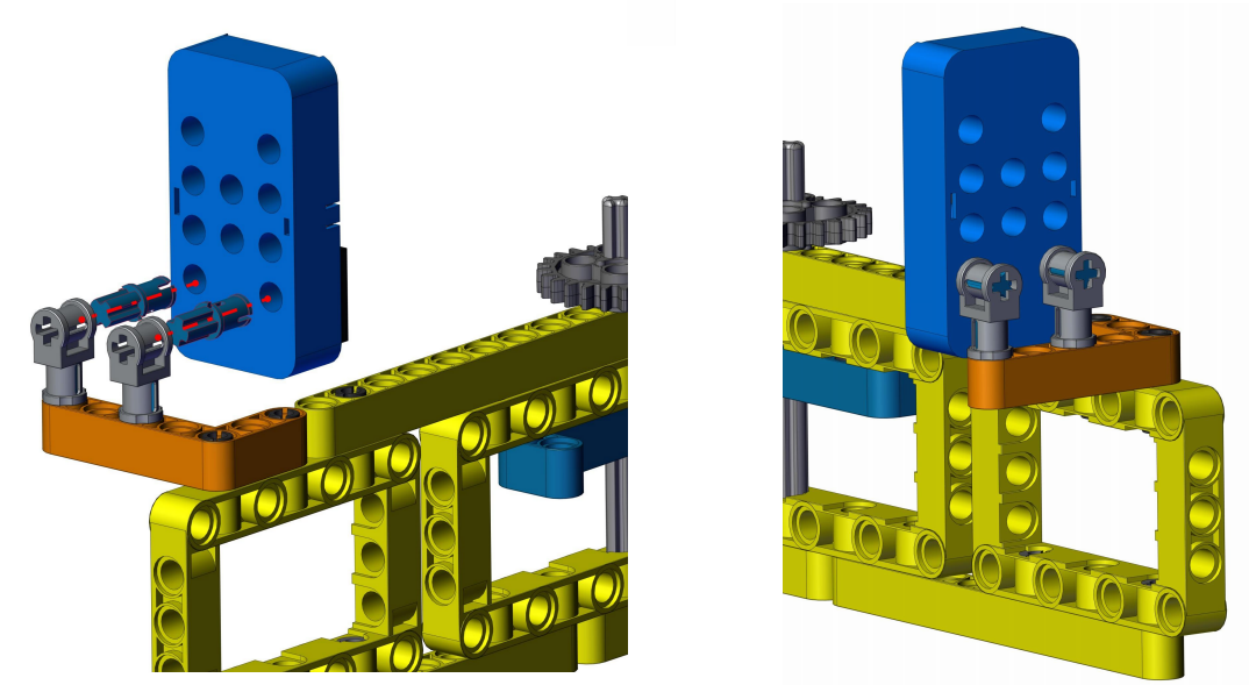
Process 8



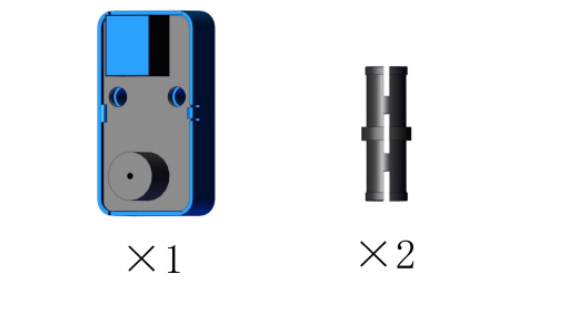


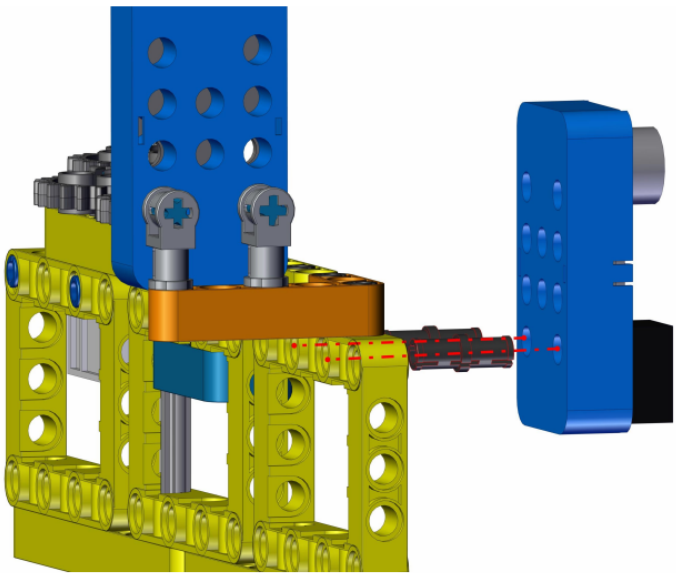
Process 9

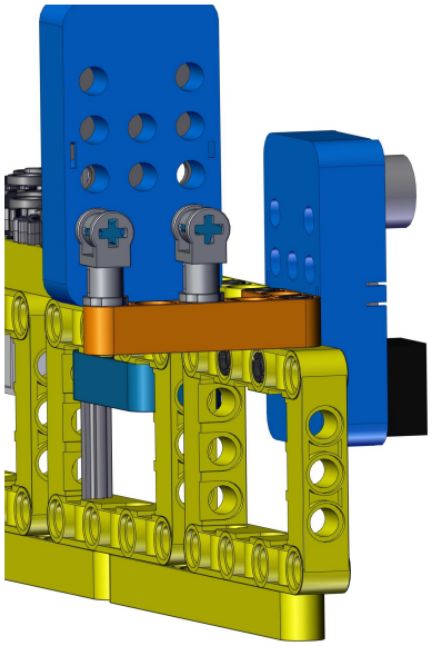




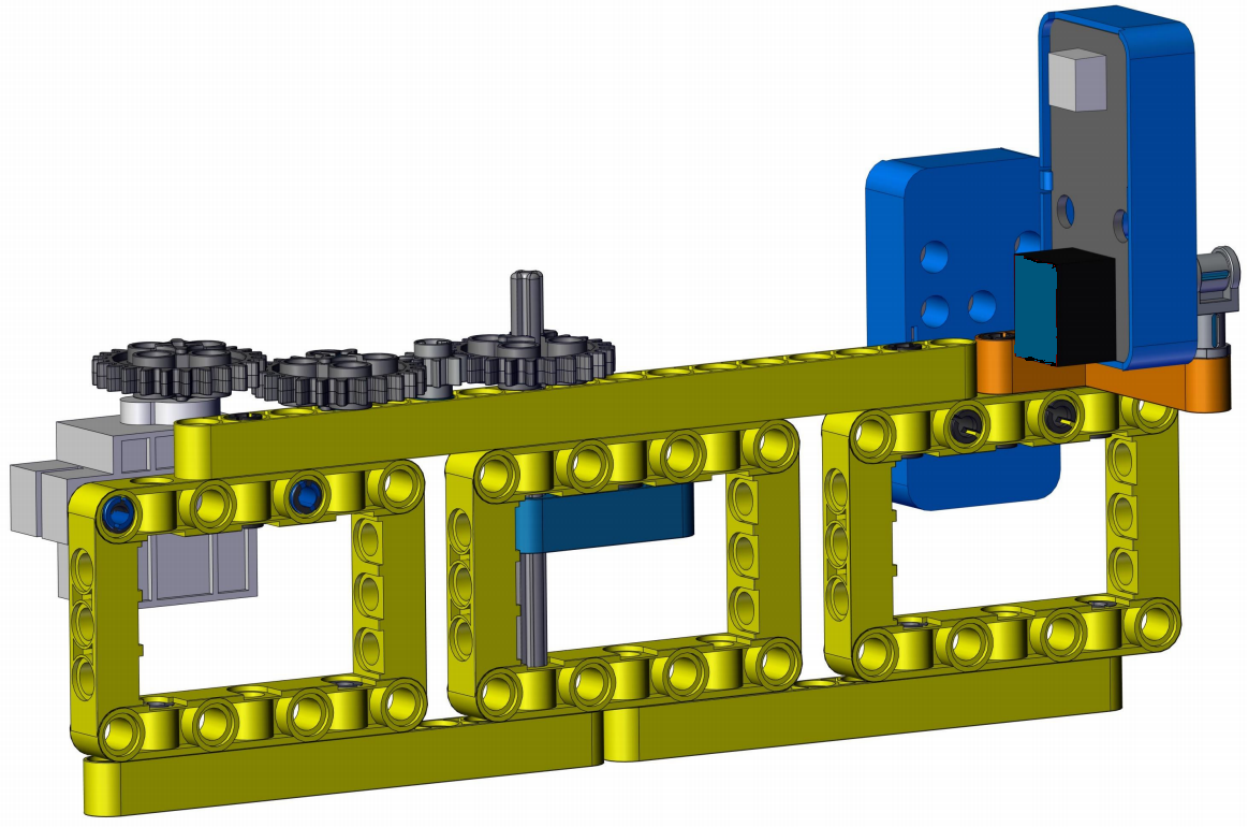
Process 10



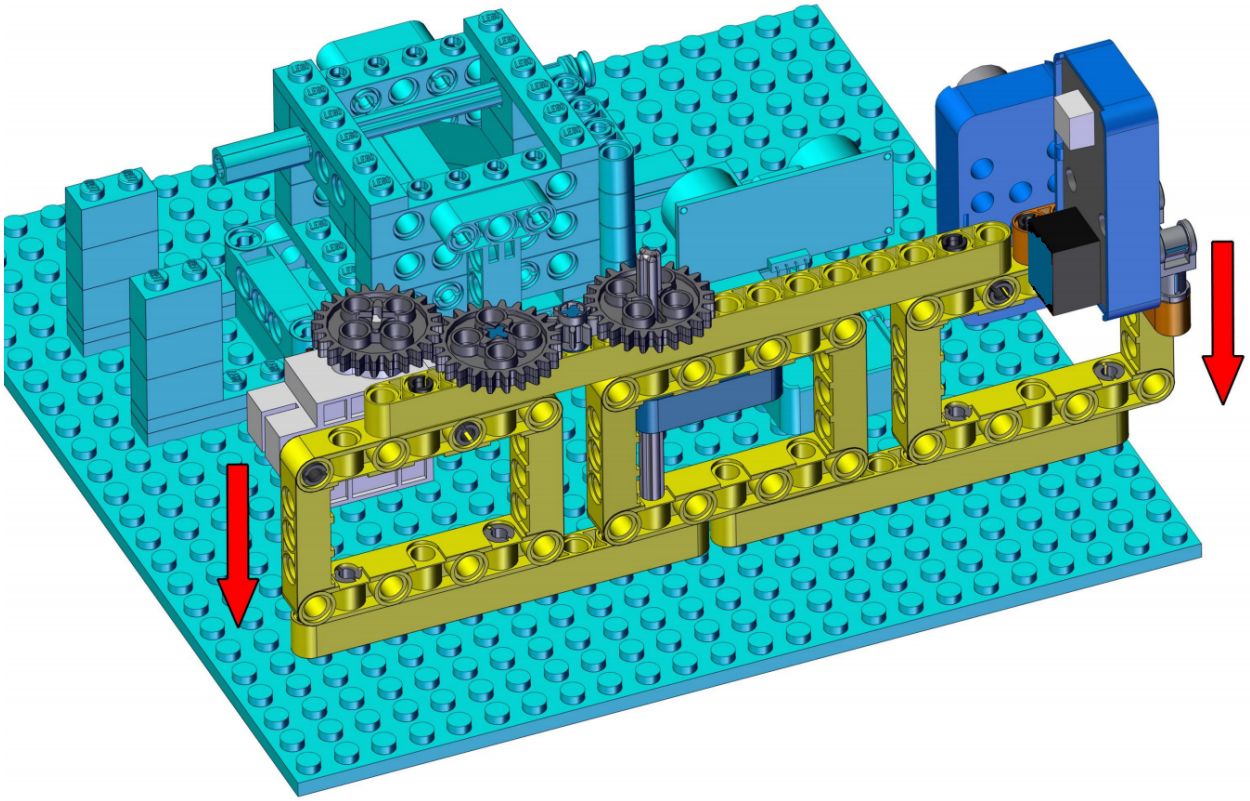




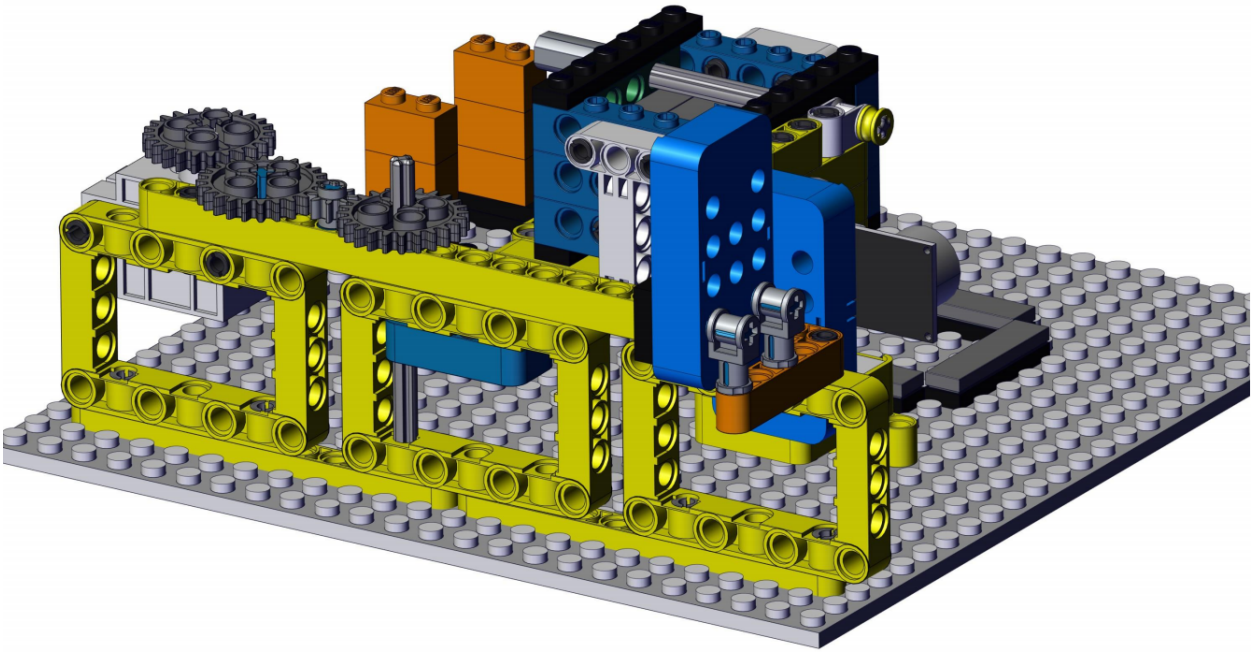
Complete 1



Process 11Share the LEGO board with project 03



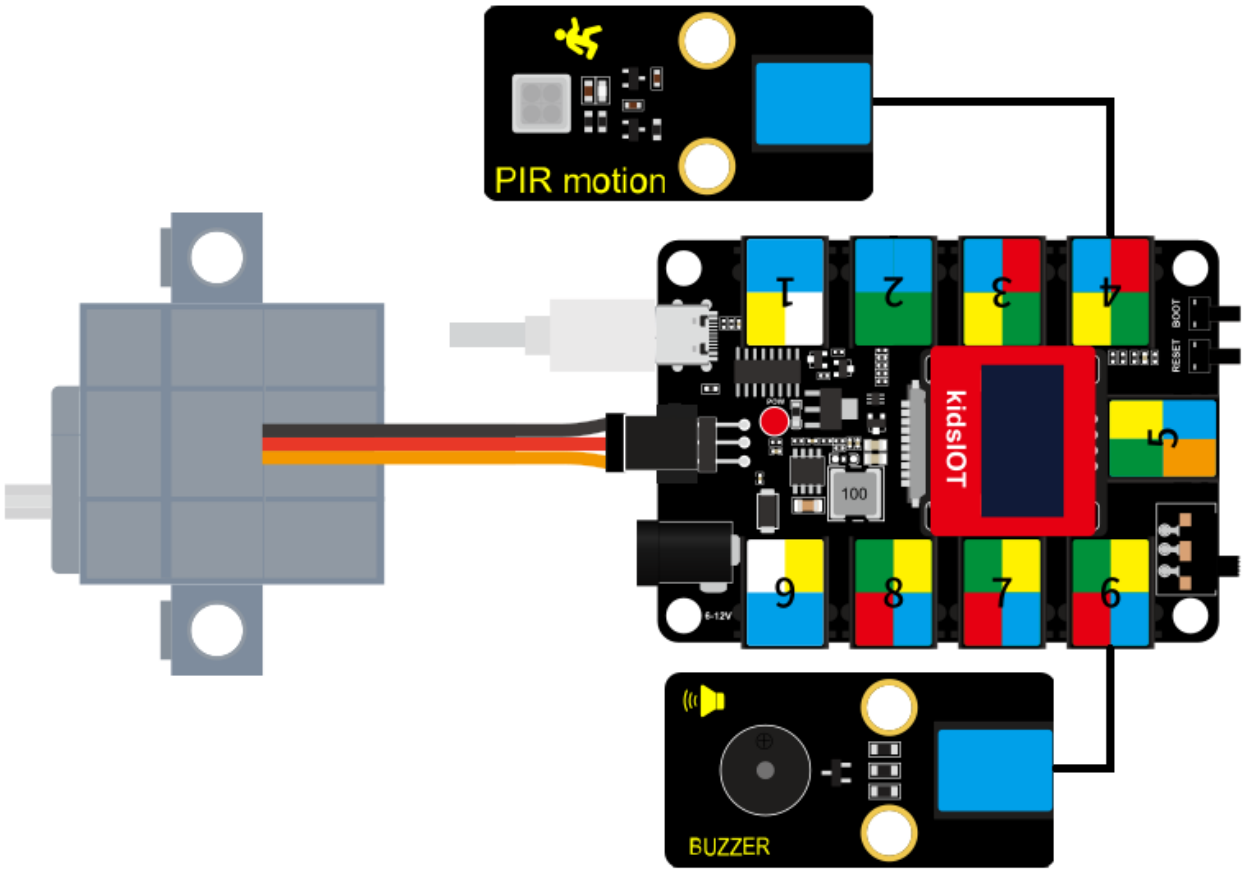
Complete 2



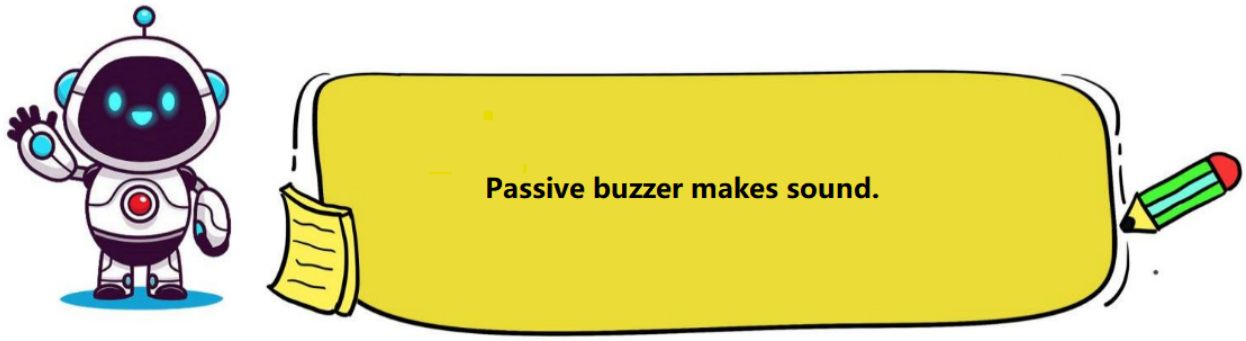
4. Wiring Diagram

Module	kidsIOT Mainboard
PIR Motion Sensor	No.4 portcontrol pin is io27
Passive Buzzer	No.6 portcontrol pin is io23
Servo	G/V/io33 portBrown→GRed→VOrange→io33

Connect the kidsIOT mainboard to your computer via USB cable.



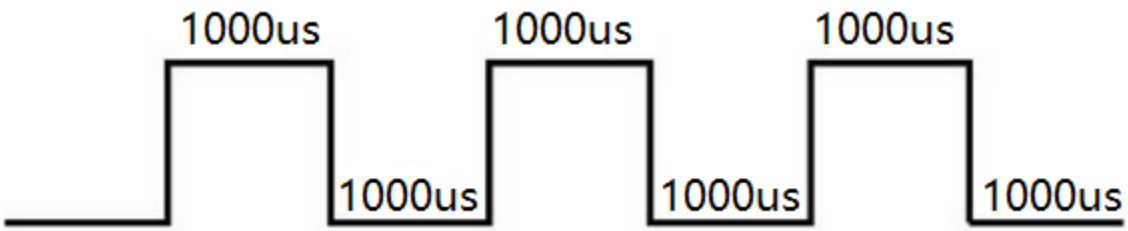
5. Passive buzzer makes sound



Method 1

(1). Knowledge

The passive buzzer is driven by square waves. Let's simulate the square waves below. The high and low levels of the pin can simulate a square wave: keeping the high level for 1000us and the low level for 1000us can make the buzzer sound.



Changing the time of high and low level can change the sound volume of the buzzer. You can try changing it to 1500us, 2000us, 3000us…

(2). Write the Program

Initialize the buzzer's pin **IO23** and "**Output**" mode.

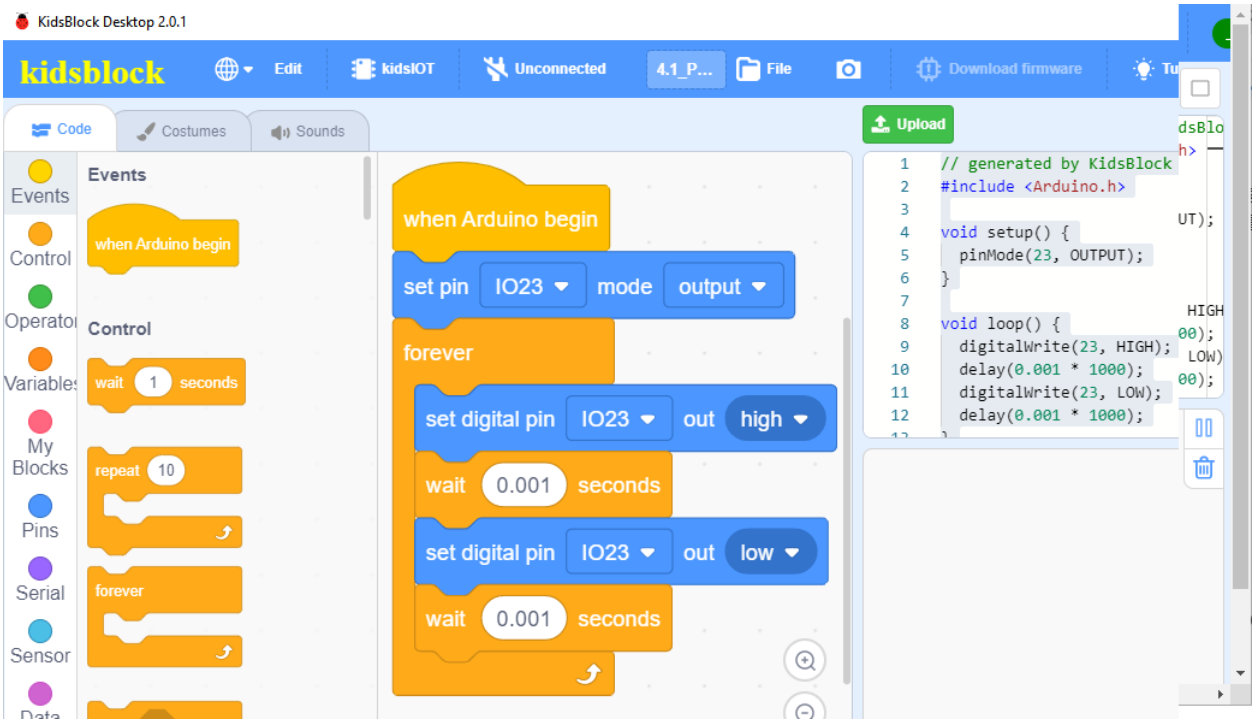
Set the buzzer pin **IO23** to "**High**" and "**Low**". Here we take the delay of 0.001 second (1000 microseconds) as an example to make the buzzer emit sound.

The delay function is a microsecond delay, which means the time delay is 1000 microseconds.

Note: The conversion relationship between seconds, milliseconds and microseconds is: 1 second = 1000 milliseconds = 1000000 microseconds.

By f=1/T, changing high and low levels in 1000us, we can know that the frequency of such a square wave is 1000Hz (that is, the number of high and low level changes per second is 1000 times).

Complete Program



(3). Test Result

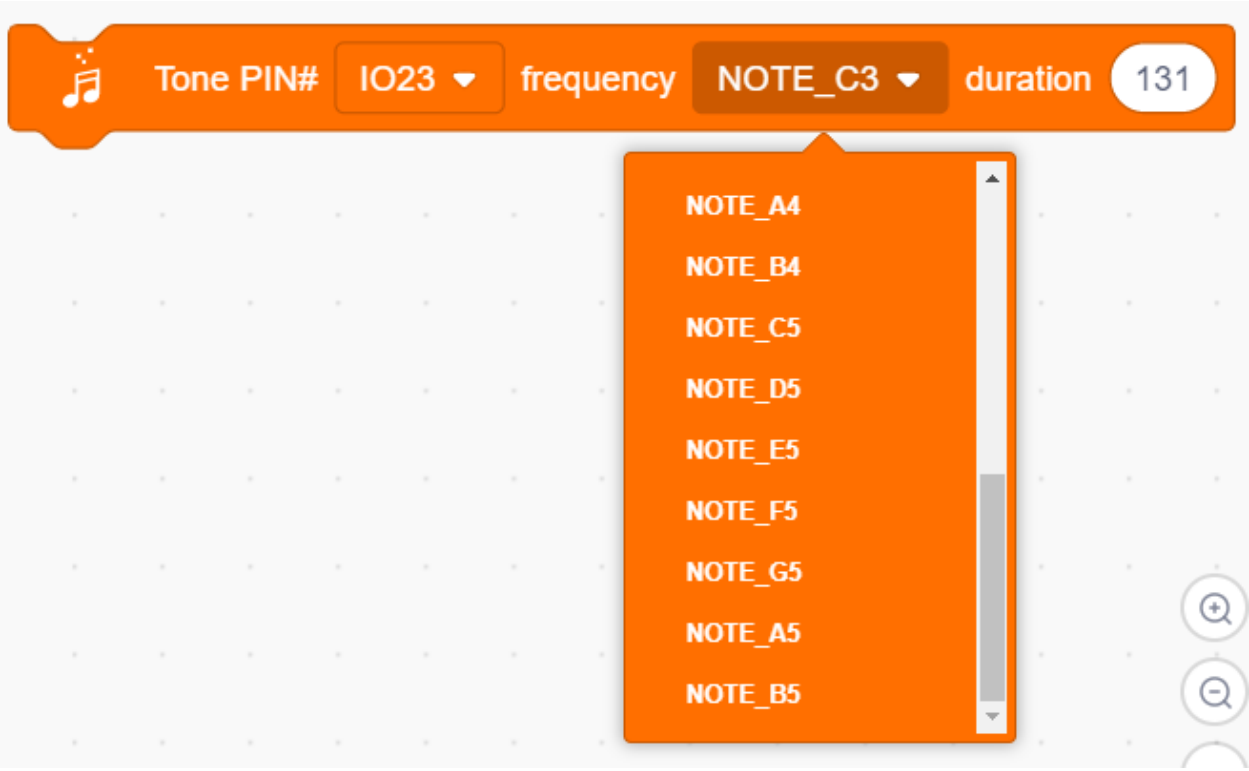
Click 

to upload the above complete code to the kidsIOT motherboard. After powering up via the USB cable, the passive buzzer will make sound.

Method 2

(1). Knowledge

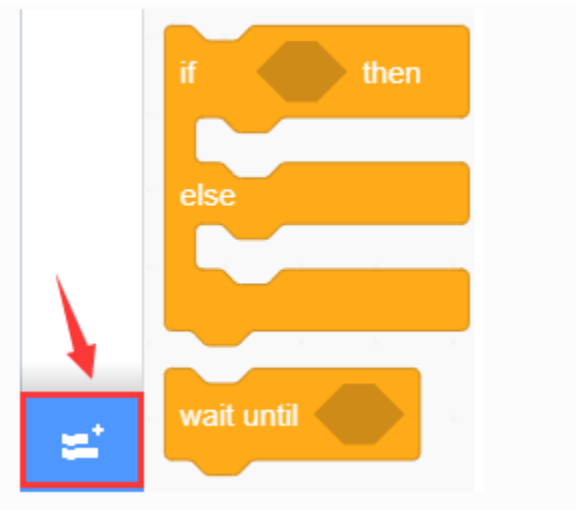
Use the "passive buzzer" code block to drive. The "passive buzzer" code block can generate a fixed-frequency PWM signal to drive the passive buzzer to sound. The sounding time length (beat) and sounding frequency can be controlled via parameters.

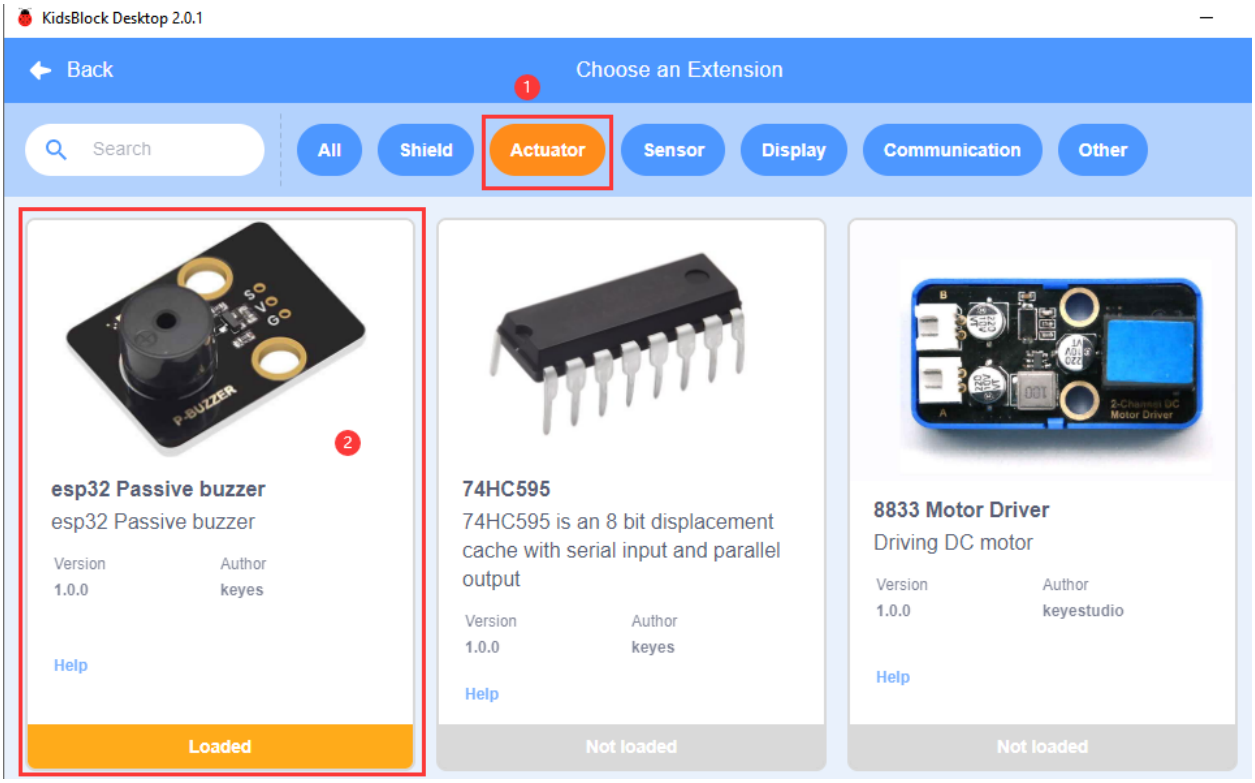


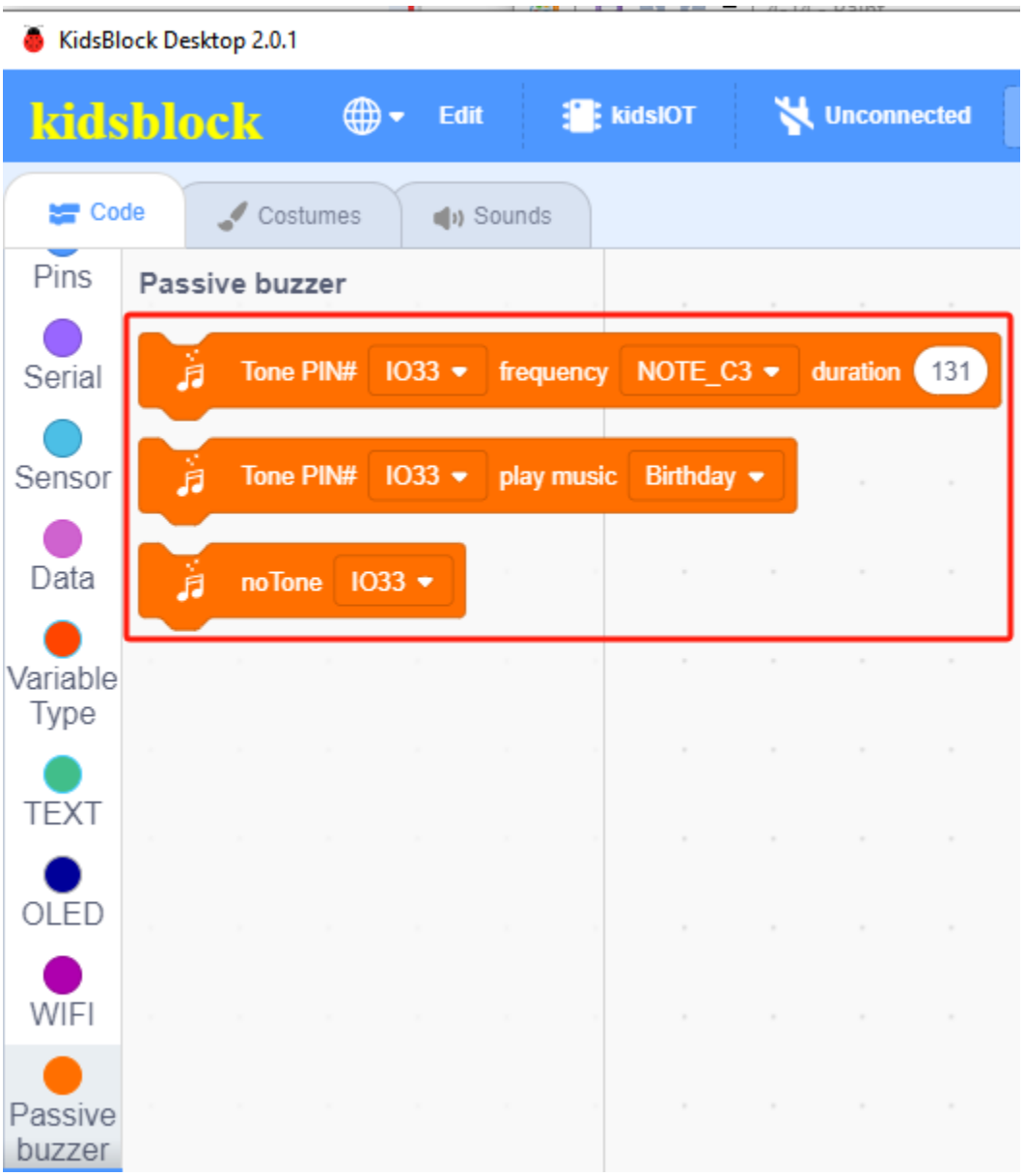
(2). Add "passive buzzer"

Tap the "Actuator" module in the "Extension" , then select "**esp32 Passive buzzer**" and click 

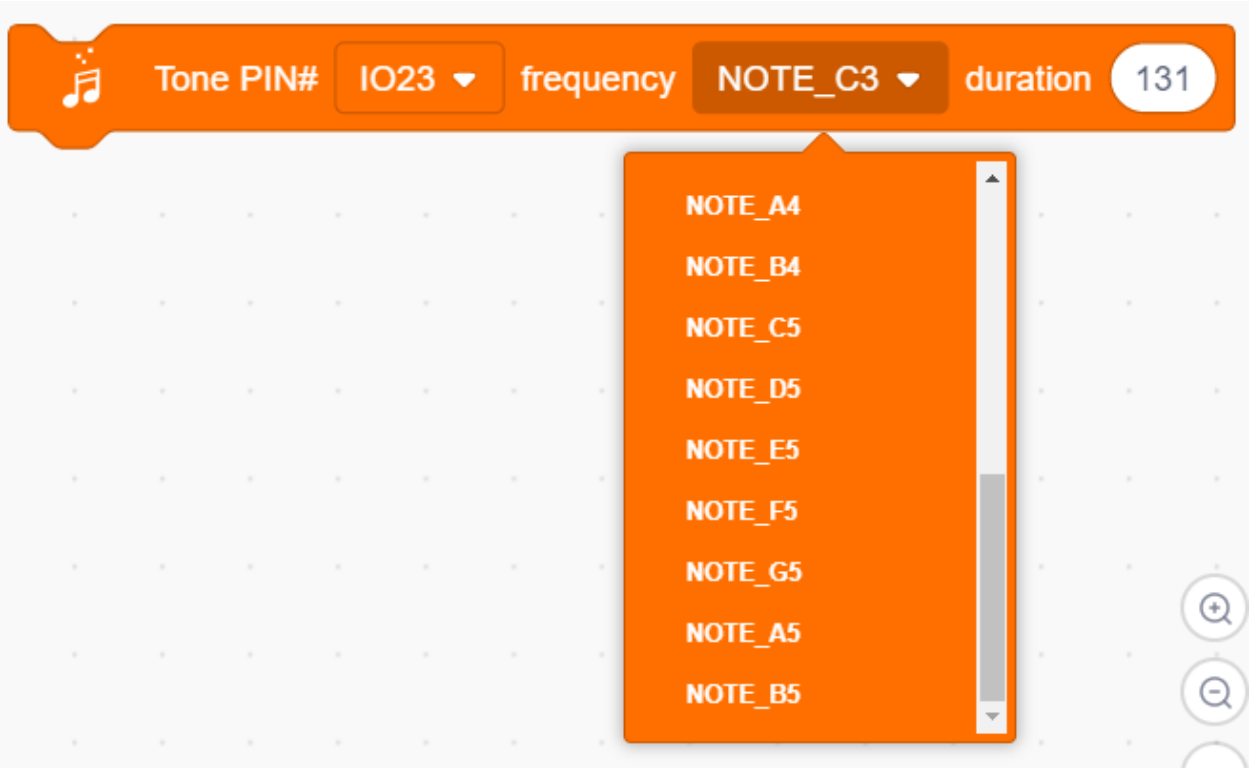
to return to the programming interface.



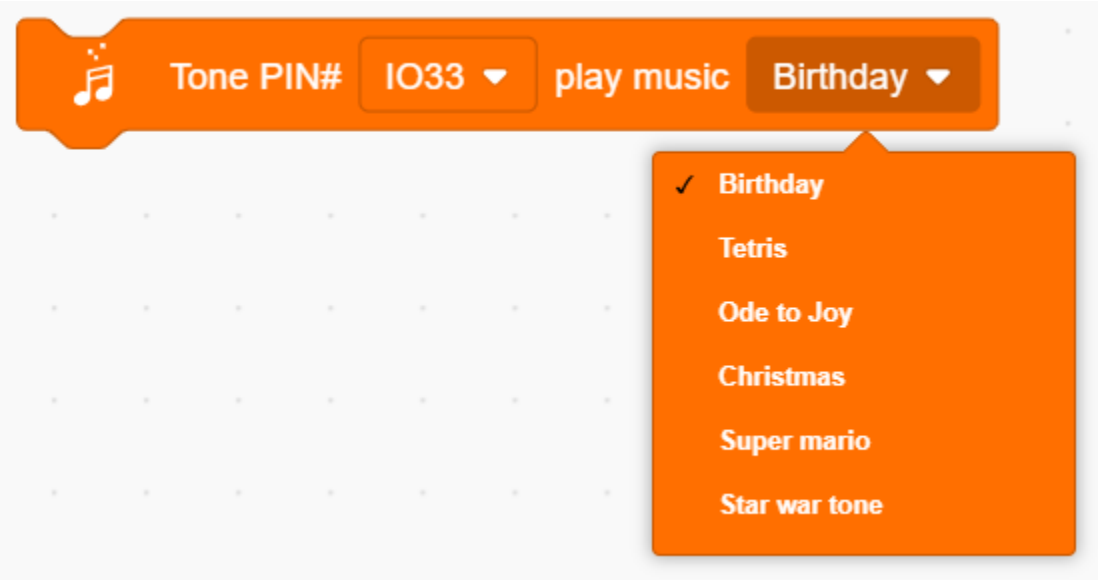

KidsBlock Desktop 2.0.1
Back
Choose an Extension
1
Search
All
Shield
Actuator
Sensor
Display
Communication
Other
2
esp32 Passive buzzer
esp32 Passive buzzer
Version
1.0.0
Author
keyes
Help
Loaded
74HC595
74HC595 is an 8 bit displacement cache with serial input and parallel output
Version
1.0.0
Author
keyes
Help
Not loaded
8833 Motor Driver
Driving DC motor
Version
1.0.0
Author
keyestudio
Help
Not loaded


KidsBlock Desktop 2.0.1
kidsblock
Edit
kidsIOT
Unconnected
Code
Costumes
Sounds
Pins
Serial
Sensor
Data
Variable Type
TEXT
OLED
WIFI
Passive buzzer
Passive buzzer
Tone PIN# IO33 frequency NOTE_C3 duration 131
Tone PIN# IO33 play music Birthday
noTone IO33

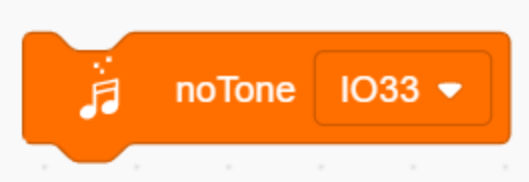
(3). Description of the Building Block



Set the frequency and beat of the passive buzzer to the specified pin.



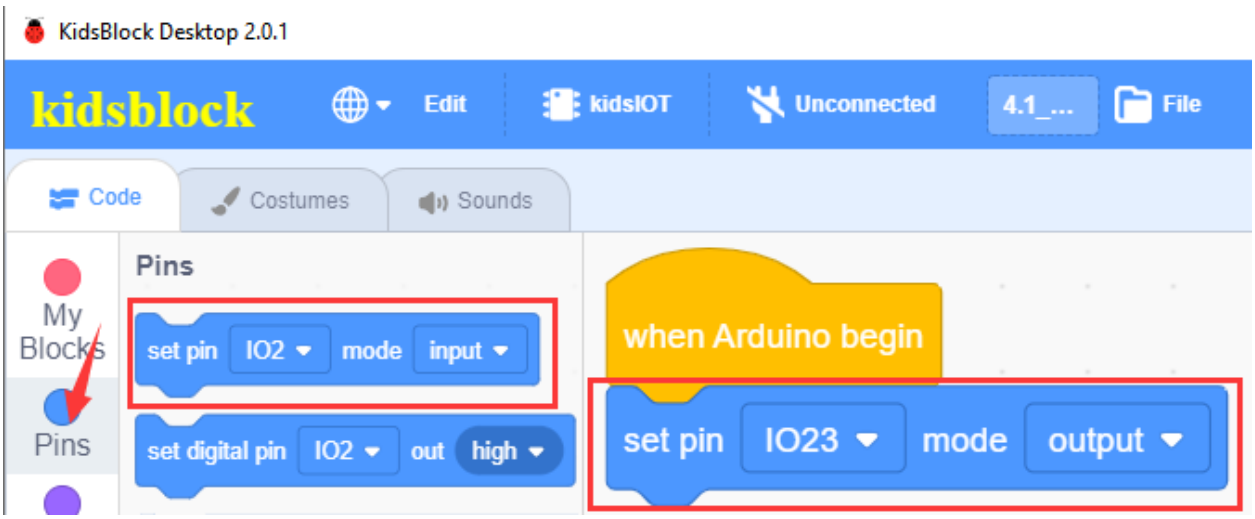
Set the passive buzzer to play specific music to the specified pin.



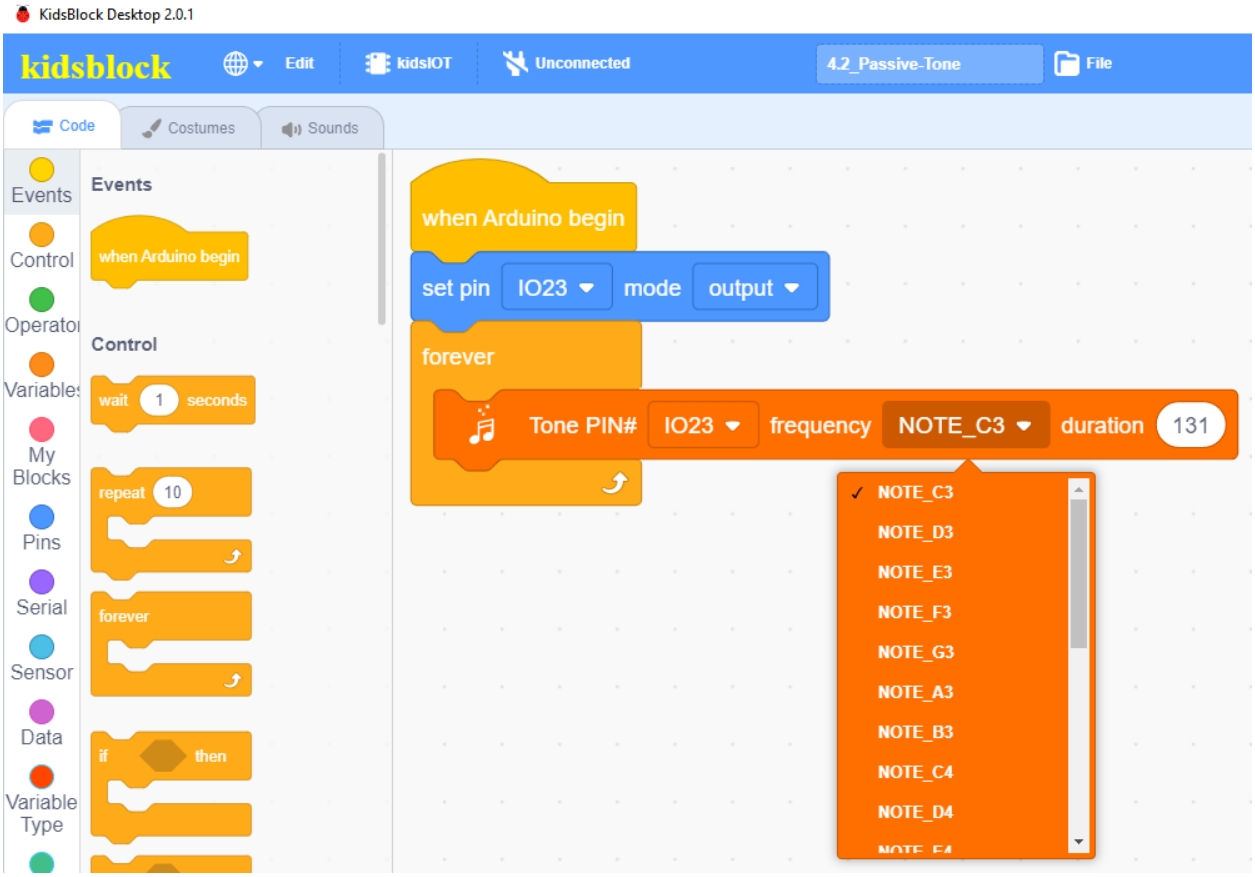
Set the passive buzzer to make no sound to the specified pin.

(4). Write the Program

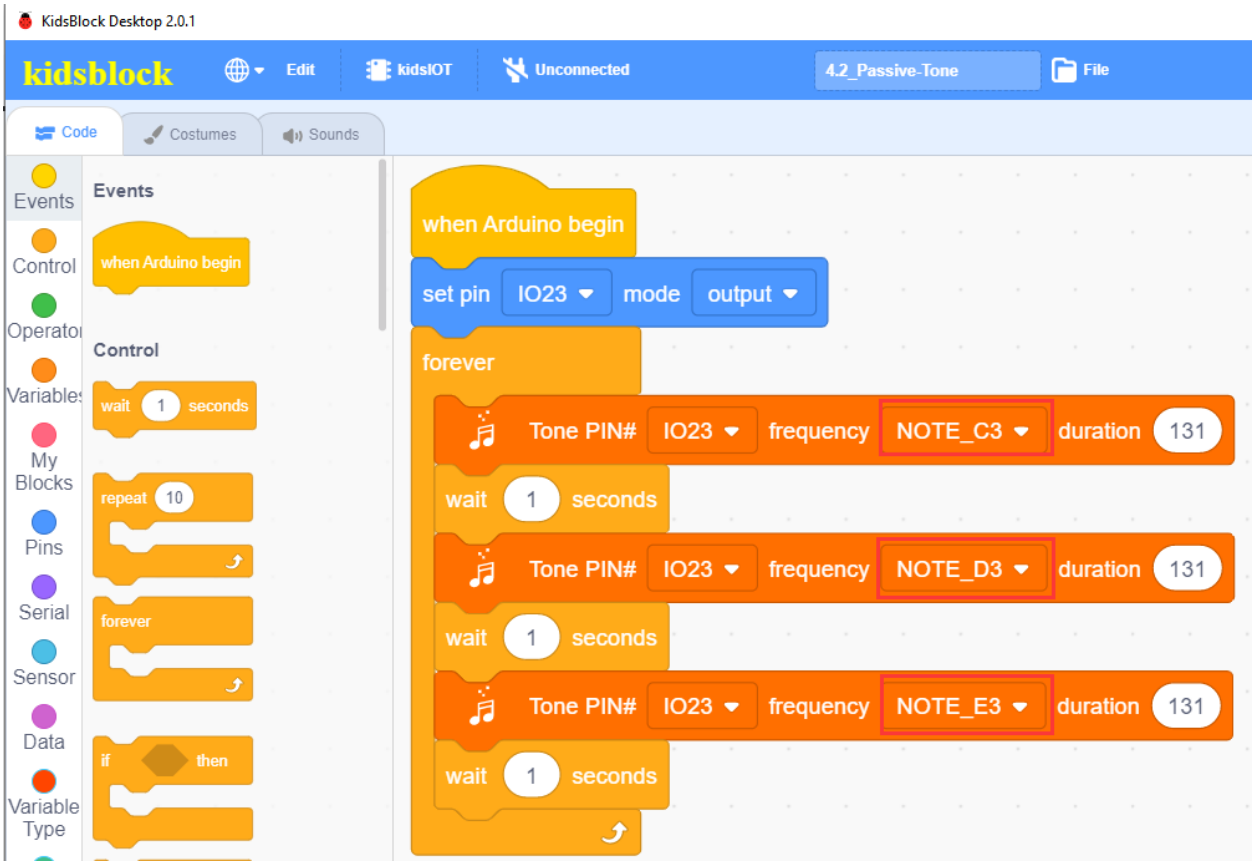
Initialize the buzzer's pin **IO23** and "**Output**" mode.



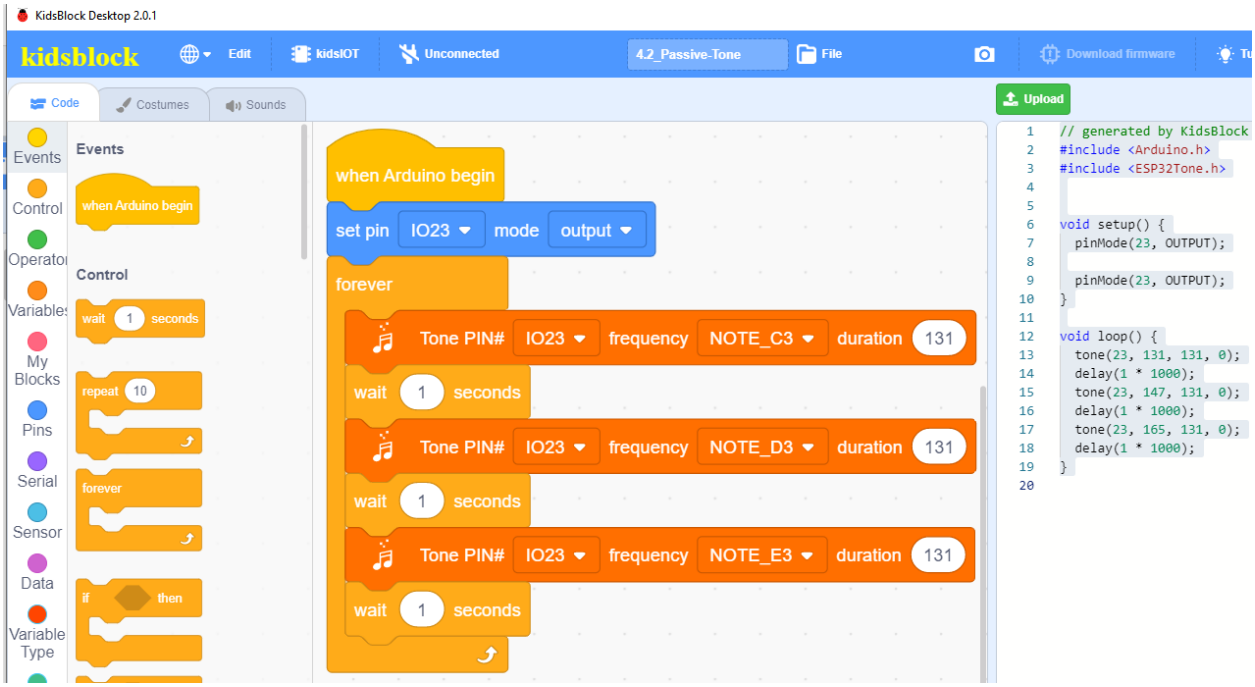
Set the sound pin, frequency and beat can be selected by yourself.



Produce different tones.



Complete Program

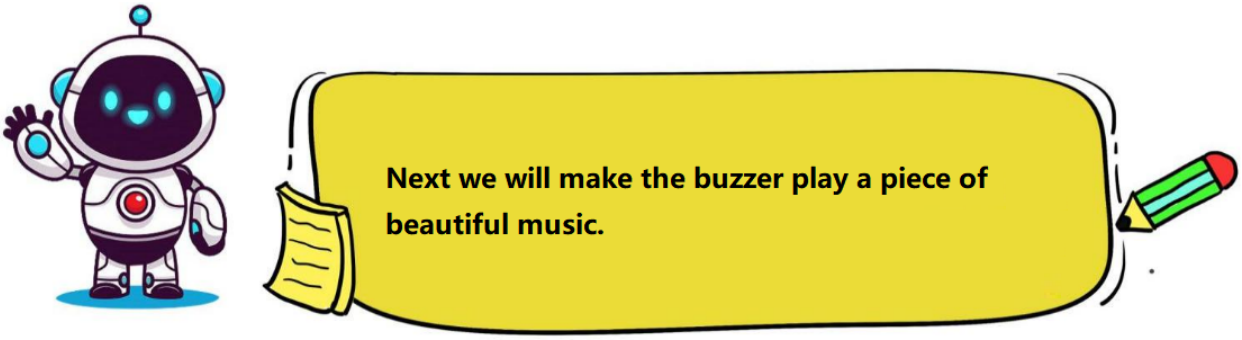


(5). Test Result

Click 

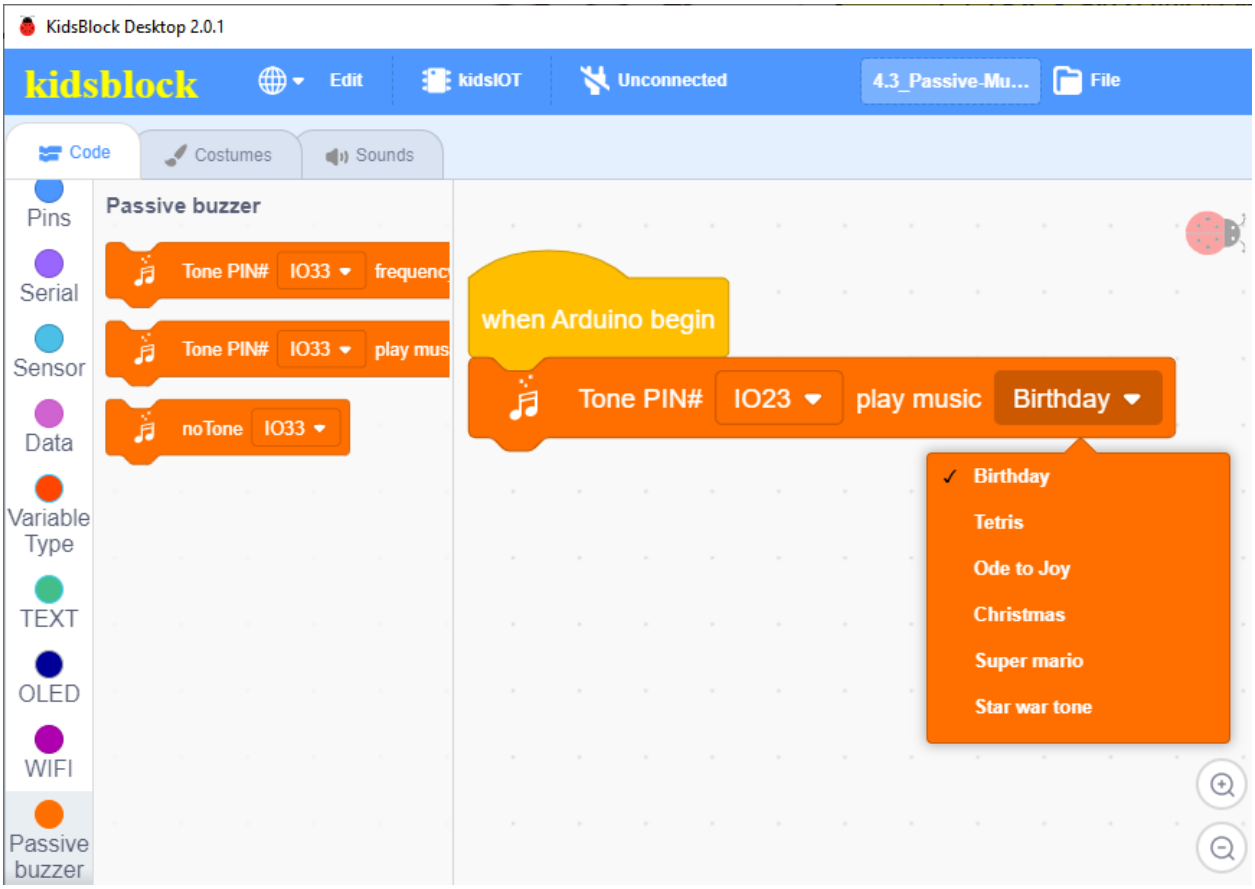
to upload the above complete code to the kidsIOT motherboard. After powering up via the USB cable, the passive buzzer will make sounds with different tones.

6. Passive buzzer plays music

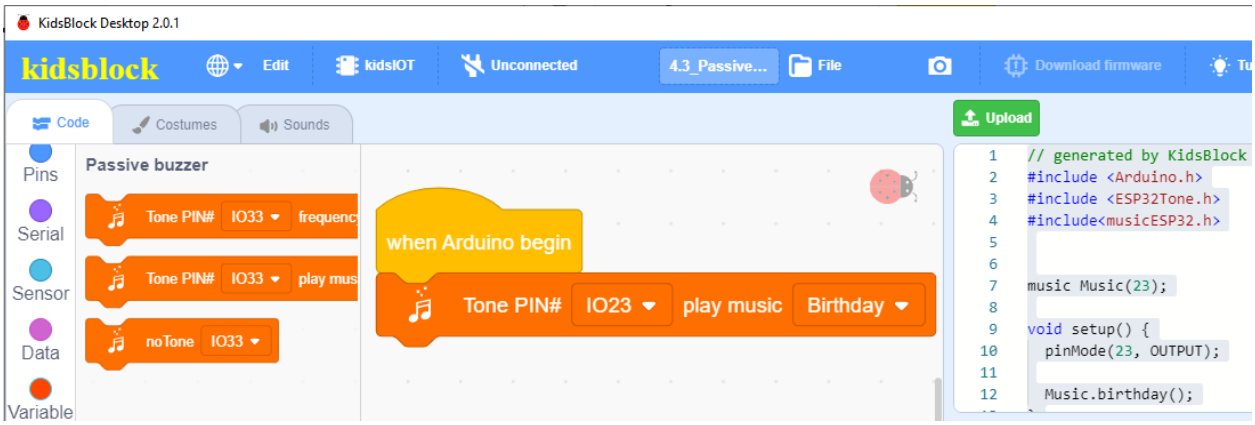


(1). Write the Program

The buzzer pin is **IO23**, and then select a piece of music (we take **Birthday** as an example here) .



Complete Program

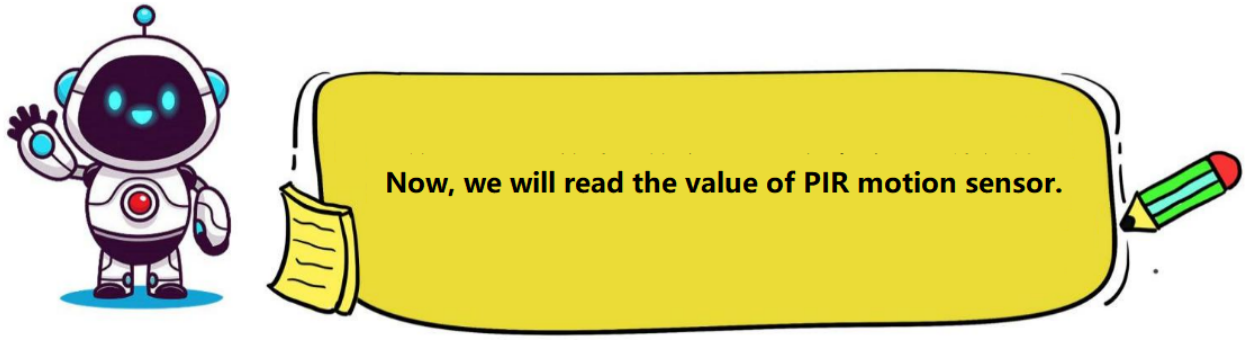


(2). Test Result

Click 

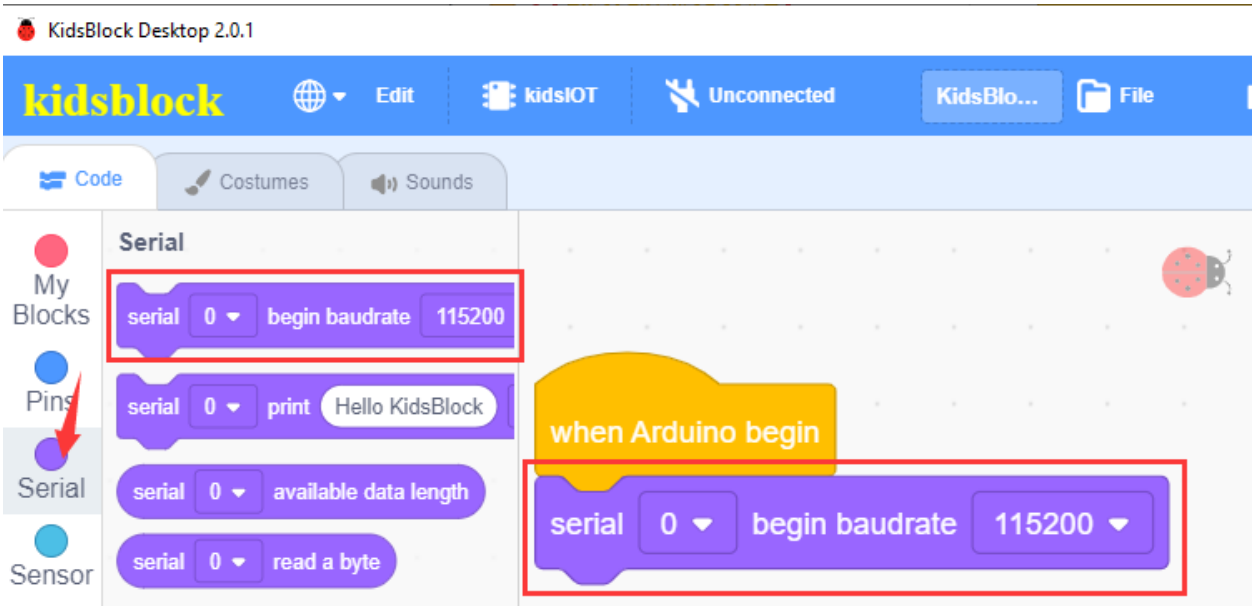
to upload the above complete code to the kidsIOT mainboard. After powering up via the USB cable, the passive buzzer will play a "Happy Birthday" music.

7. Read the value of PIR Motion Sensor

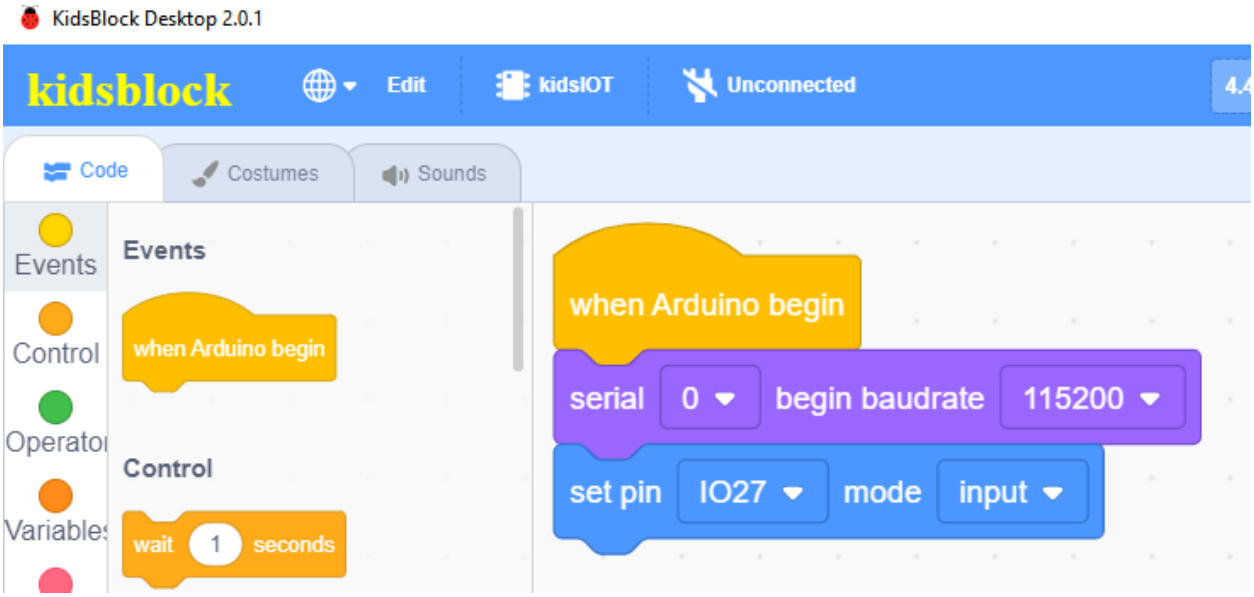


Step 1Write the Program

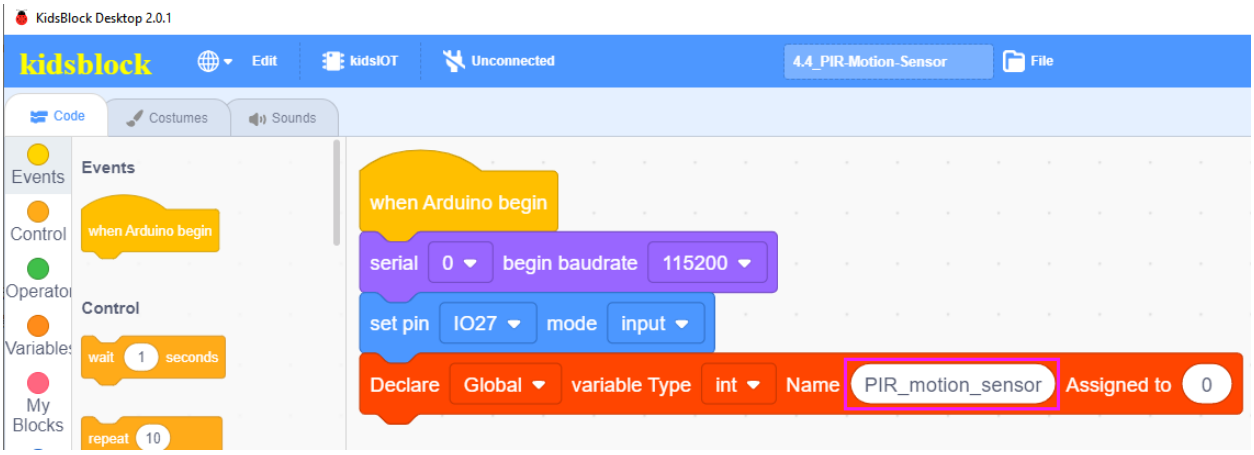
Set the baud rate to 15200.



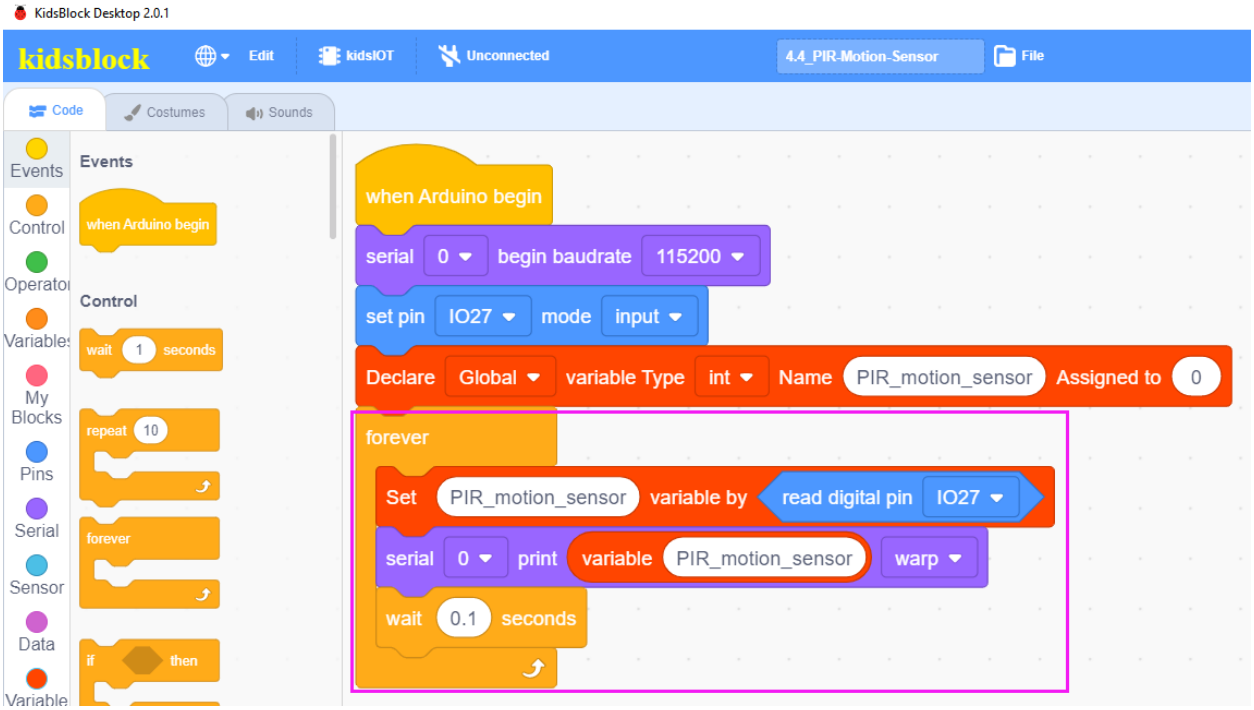
Set the pin IO27 connected to the PIR motion sensor to "**input**" mode.



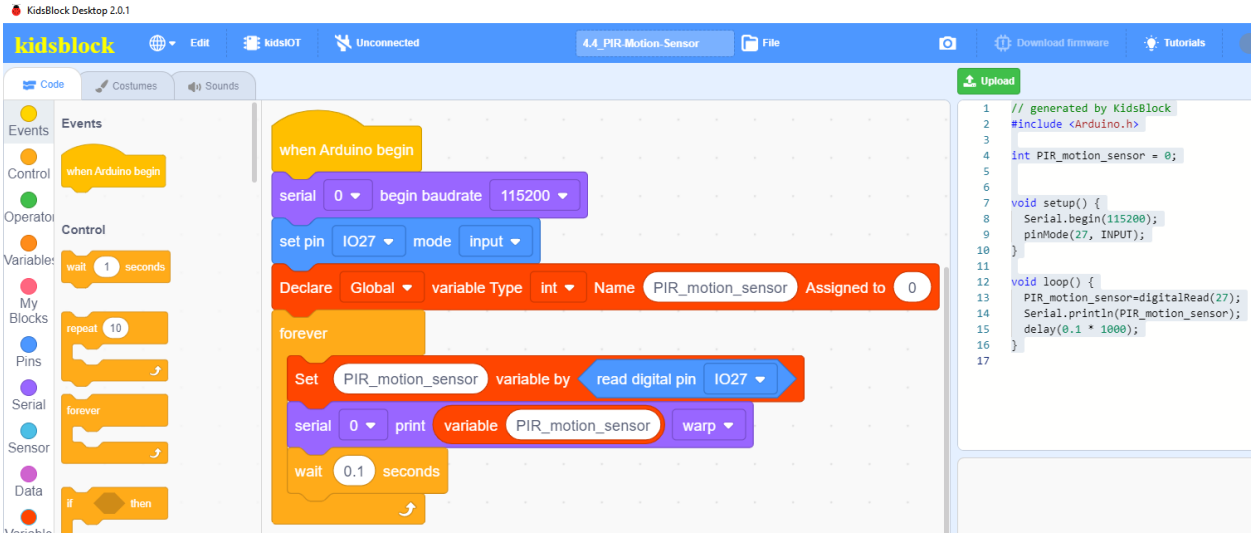
Define a "PIR_motion_sensor" global variable to store the value of the sensor.



Store the read value of the sensor in the "PIR_motion_sensor" variable and print it on the serial port.



Complete Program



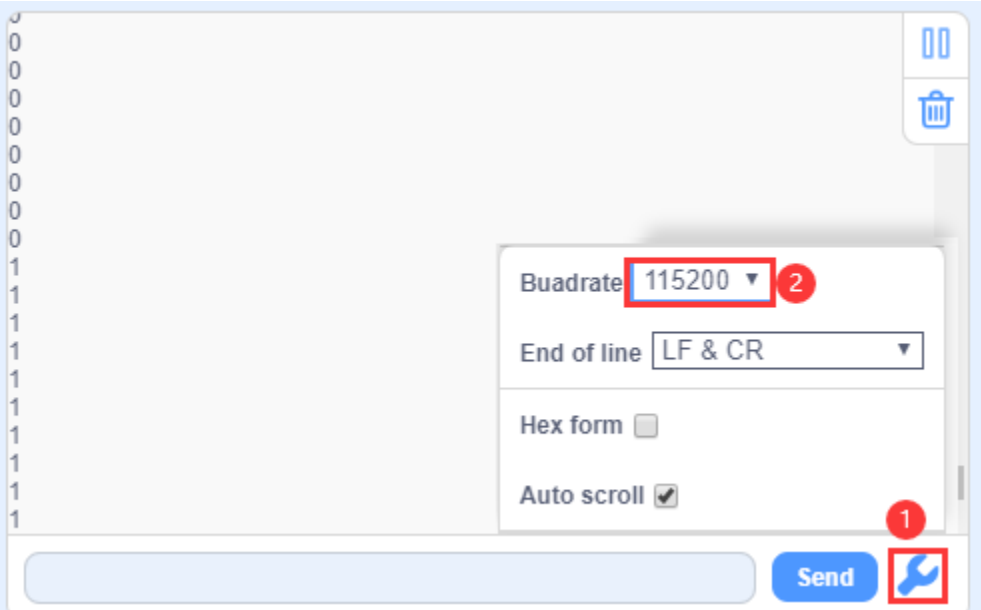
Step 2Test Result

Click 

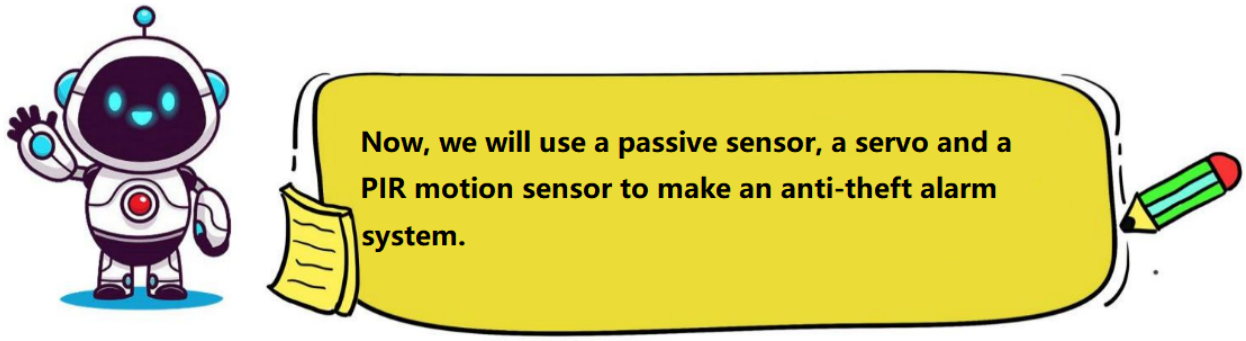
to upload the above complete code to the kidsIOT motherboard. After powering up via the USB cable, click  in the serial monitor and set the baud rate to 15200.

When the sensor detects movement of a person or animal, the serial monitor window prints 1, and the red LED on the sensor will be off; otherwise, the monitor prints 0, and the red LED on the sensor will be on.

Note: The sensor does not have penetrating capabilities. When detecting human movement, please do not block it.

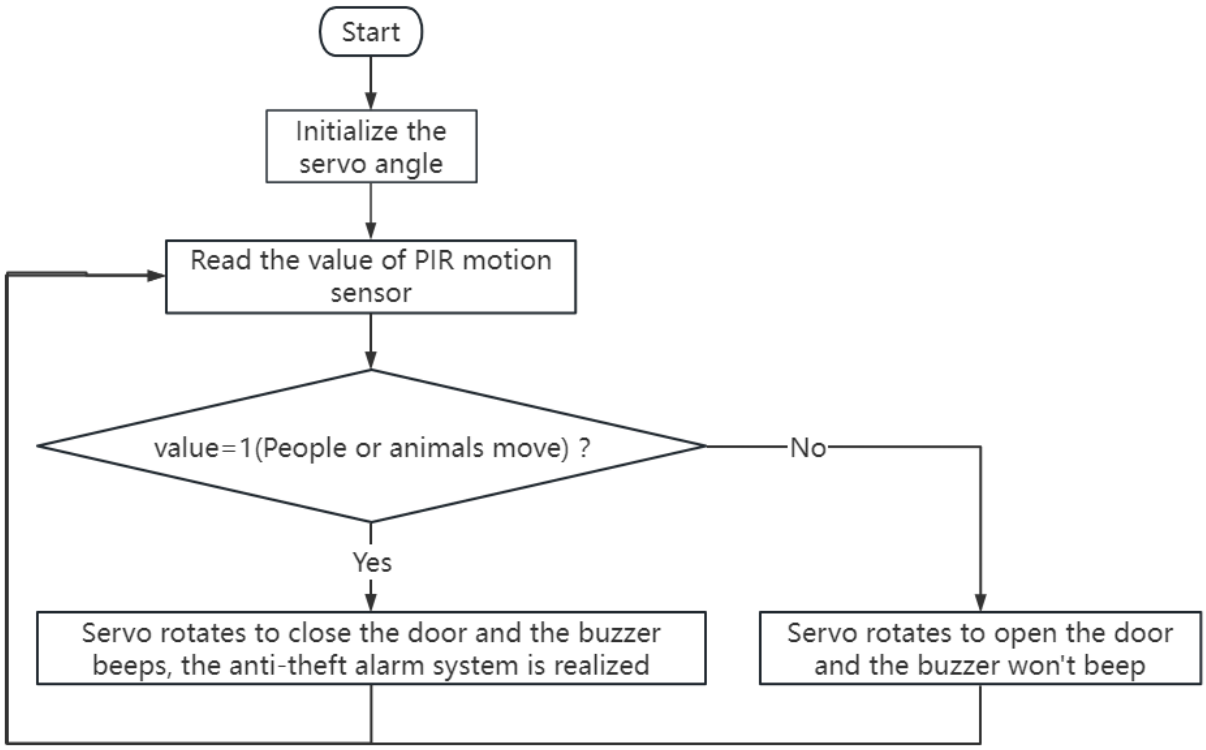


8. Anti-theft Alarm System



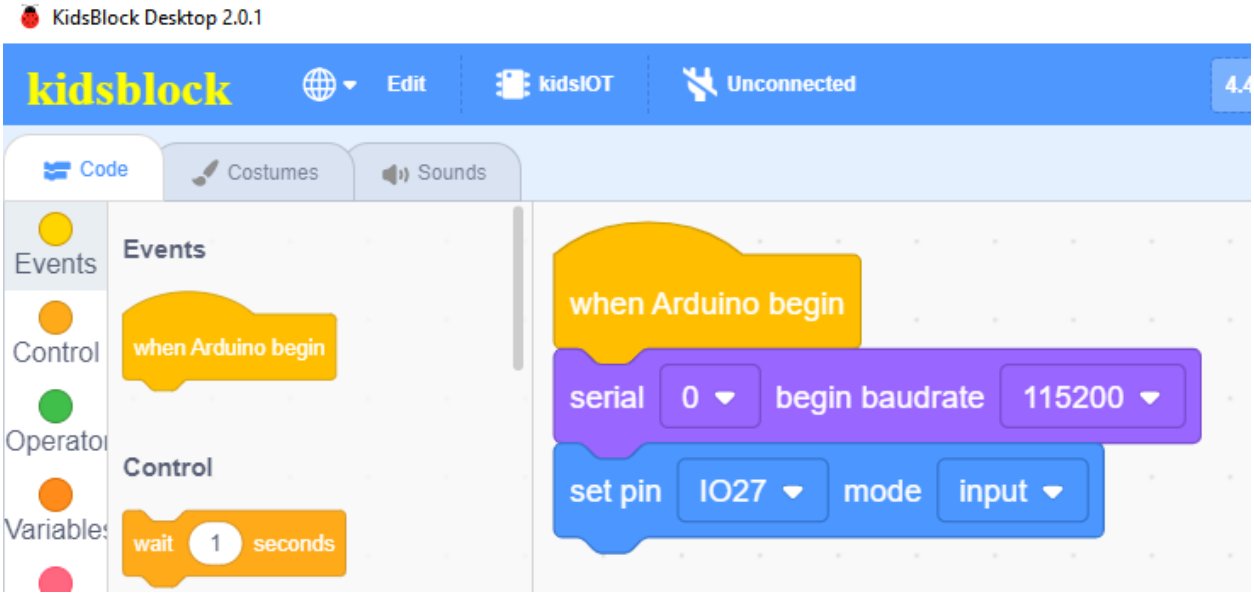
(1). Programming Steps

Step 1Flow Chart



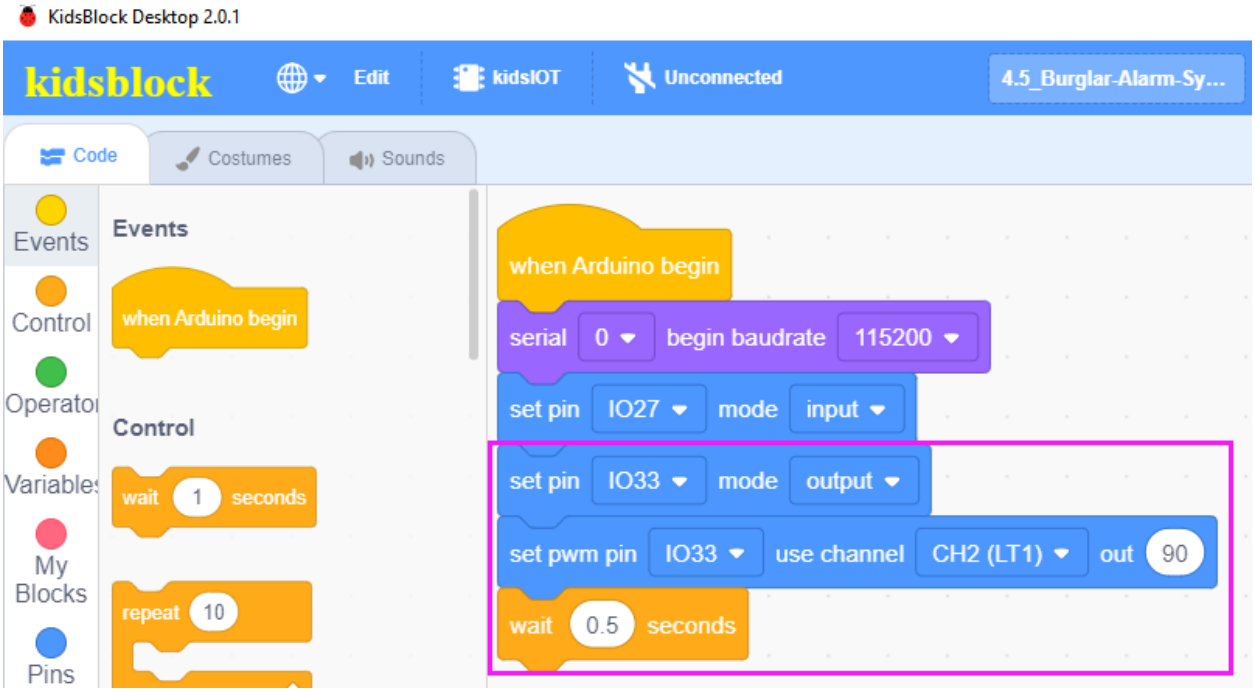
Step 2Write the Program

Set the baud rate to 15200, the IO27 pin of PIR motion sensor to "**input**" mode.

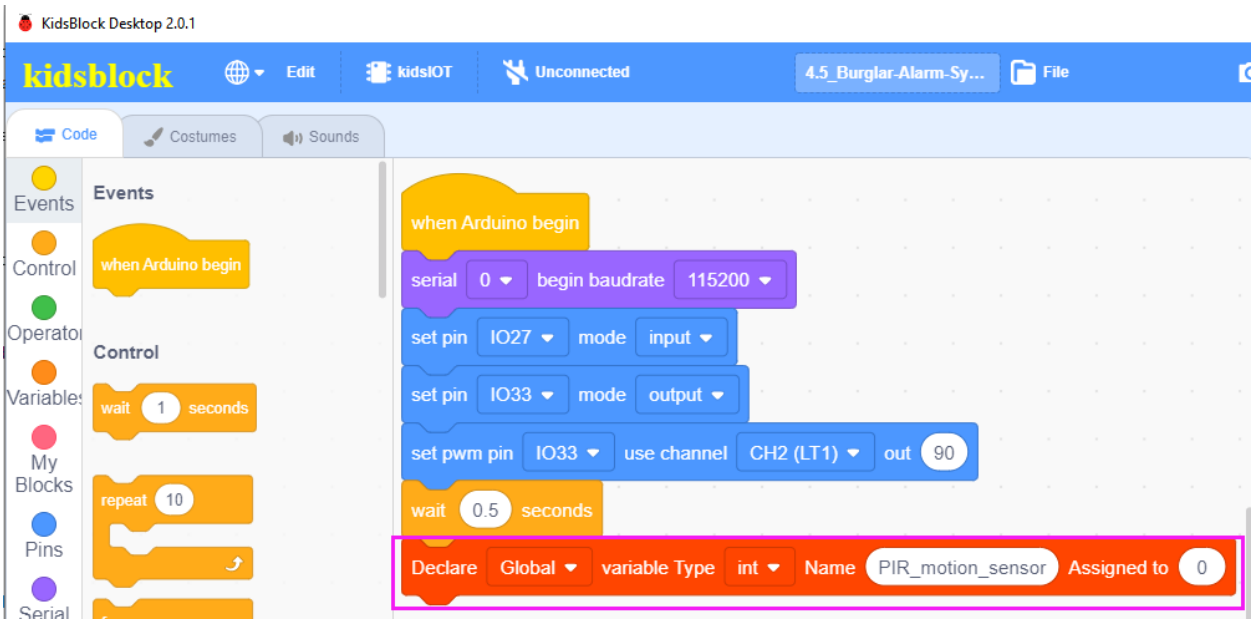


Set the pin IO33 connected to the servo to "**Output**" mode, initialize the control channel of the servo to CH2 (LT1)

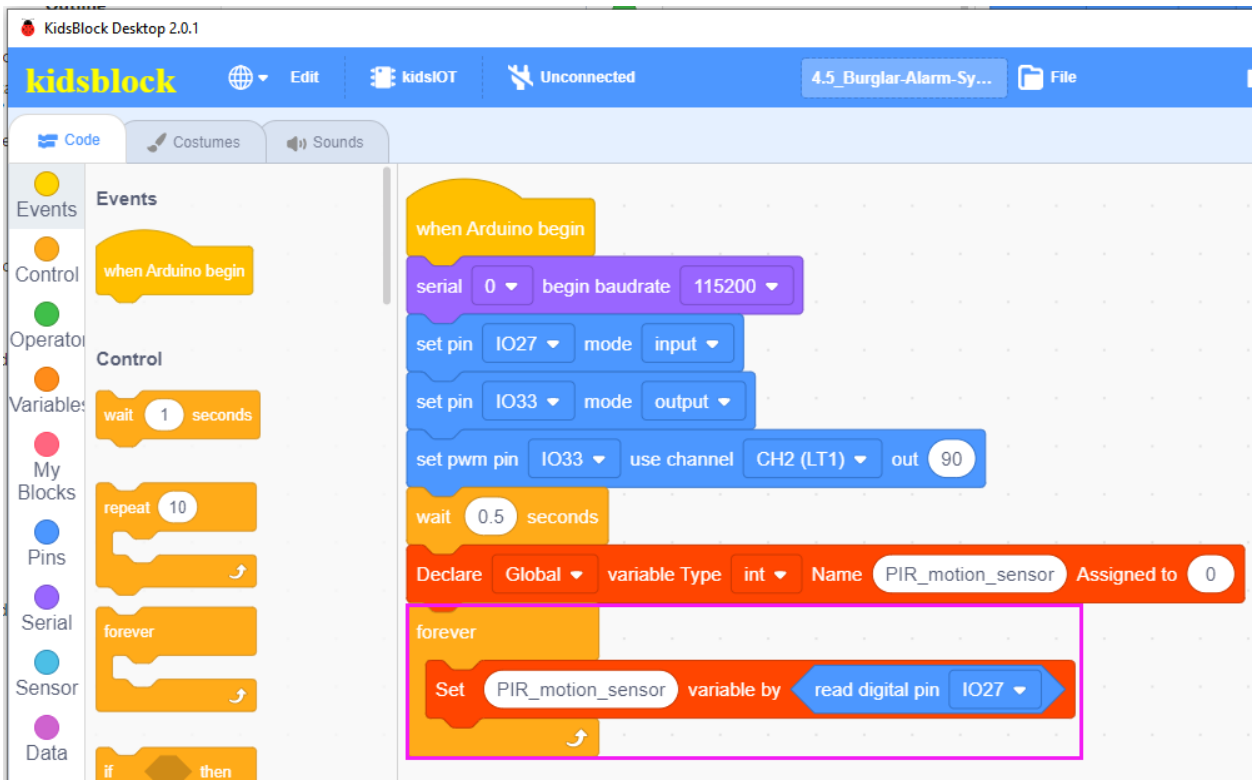
and the initial angle to 90°, delay 0.5 seconds.



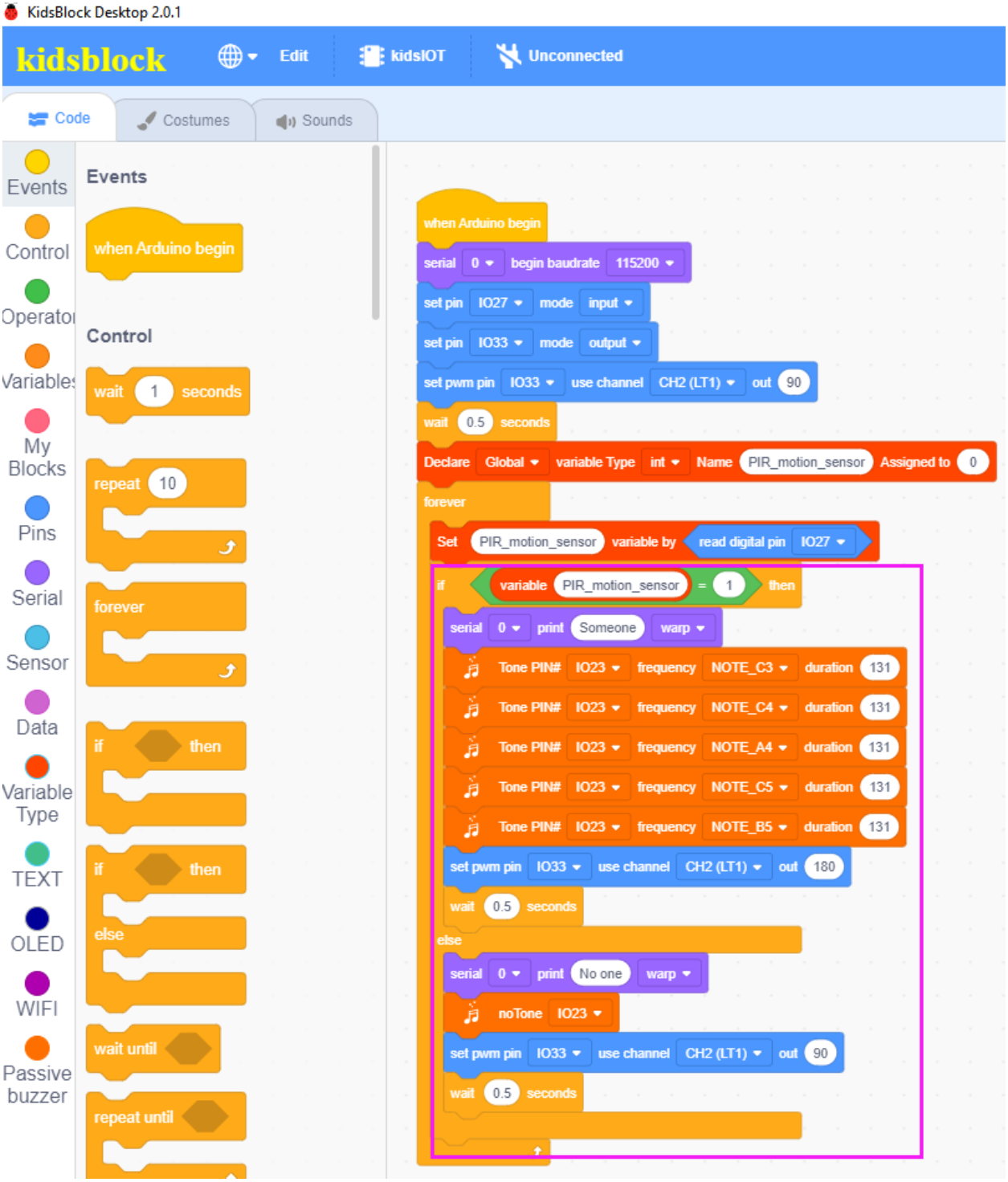
Define a "PIR_motion_sensor" global variable to store the value of the PIR motion sensor.



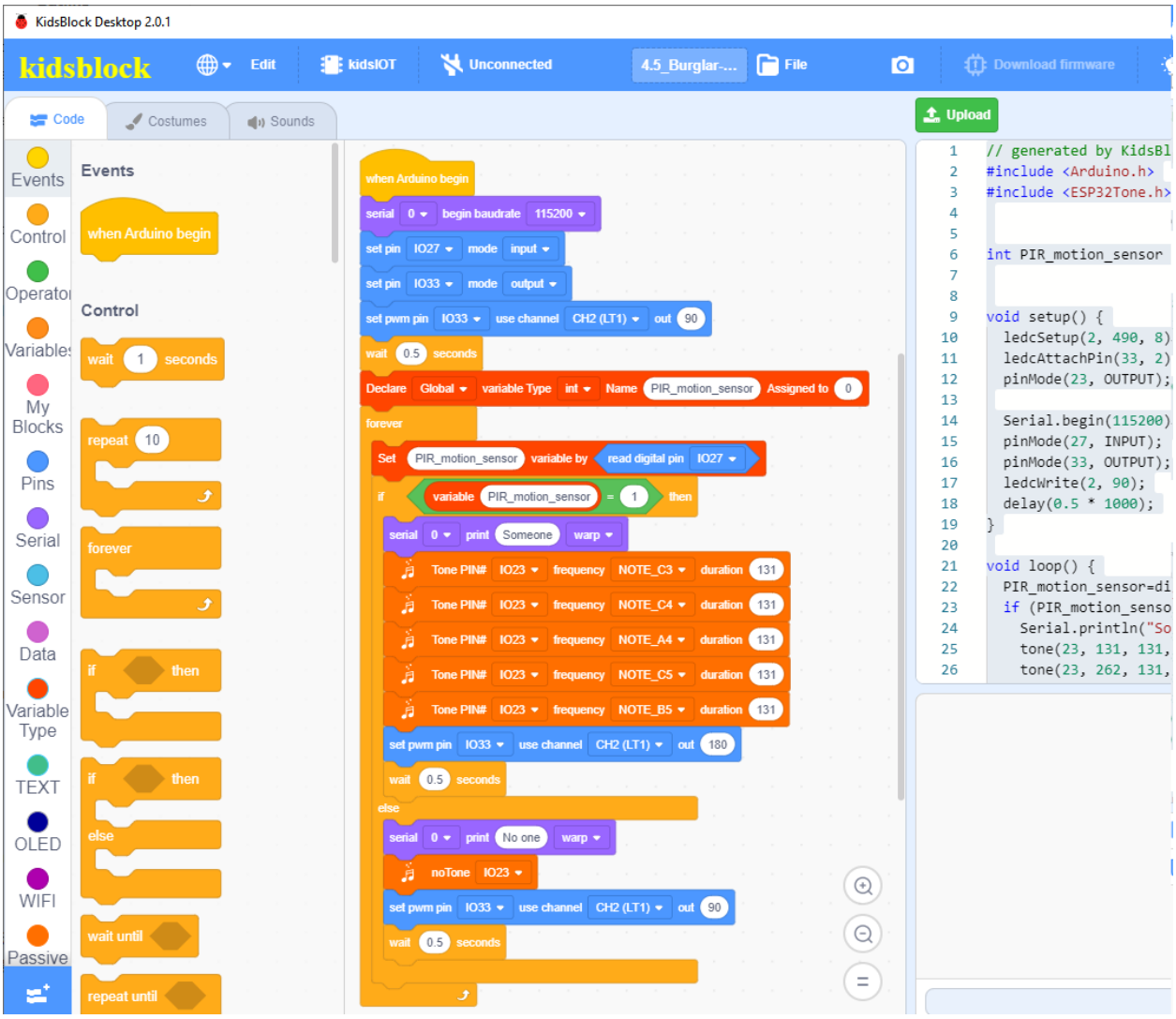
Store the read value of the sensor in the "PIR_motion_sensor" variable.



Judge whether the sensor detects that a person or animal is moving. When someone or an animal is moving, the buzzer sounds, the servo rotates to close the door, and the serial monitor prints “Someone”; otherwise, the buzzer does not sound, the servo rotates to open the door, and the monitor prints ” No one”.



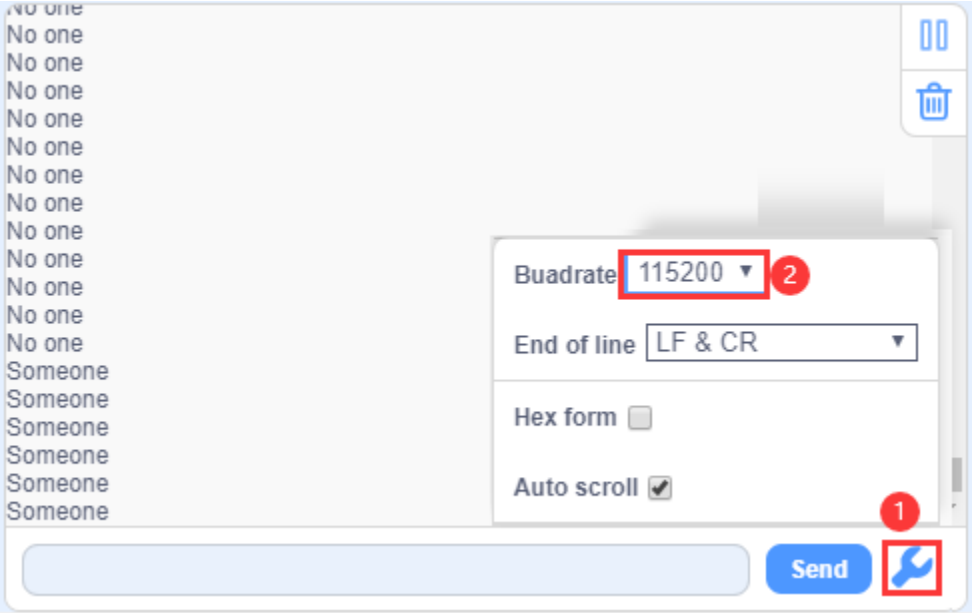
Complete Program

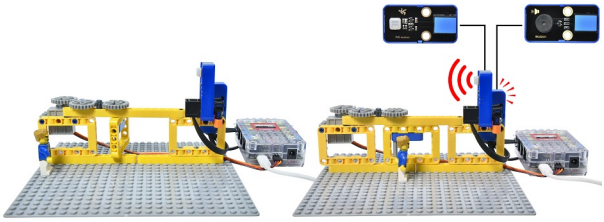


(2). Test Result

Click  to upload the above complete code to the kidsIOT mainboard. After powering up via the USB cable, click  in the serial monitor and set the baud rate to 15200.

When the sensor detects that someone or an animal is moving, the buzzer sounds, the servo rotates to close the door, and the serial monitor prints “Someone”; otherwise, the buzzer does not sound, the servo rotates to open the door, and the monitor prints “No one”.





9. Common Problems

Q1: The tone of the passive buzzer is not accurate to the actual tone?

A: The tones simulated by ordinary passive buzzers cannot meet the requirements of professional tones. If you need accurate tones, you need to use a more professional passive buzzer.

Q2: Does the PIR motion sensor make false alarms?

A: Non-professional PIR motion sensor.

The requirements for the sensor to avoid false alarms are as follows: Within the detection range, avoid objects that are caused by the wind, such as curtains, clothing, flowers, etc. Avoid interference from strong light within the detection range, such as sunlight, car lights, spotlights, lighting and other light sources.

4.3.5 Project 05Rainwater Control System

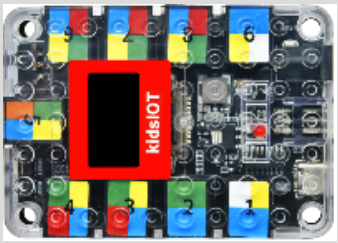
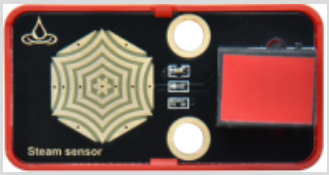
Note: Sprinkling water on other sensors will cause a short circuit and device failure. Sprinkling water on batteries will cause heating and explosion. Please be careful when using the device, especially when used by young children, it must be under the supervision of parents. To ensure safe operation of the device, please follow relevant usage guidelines and safety regulations.

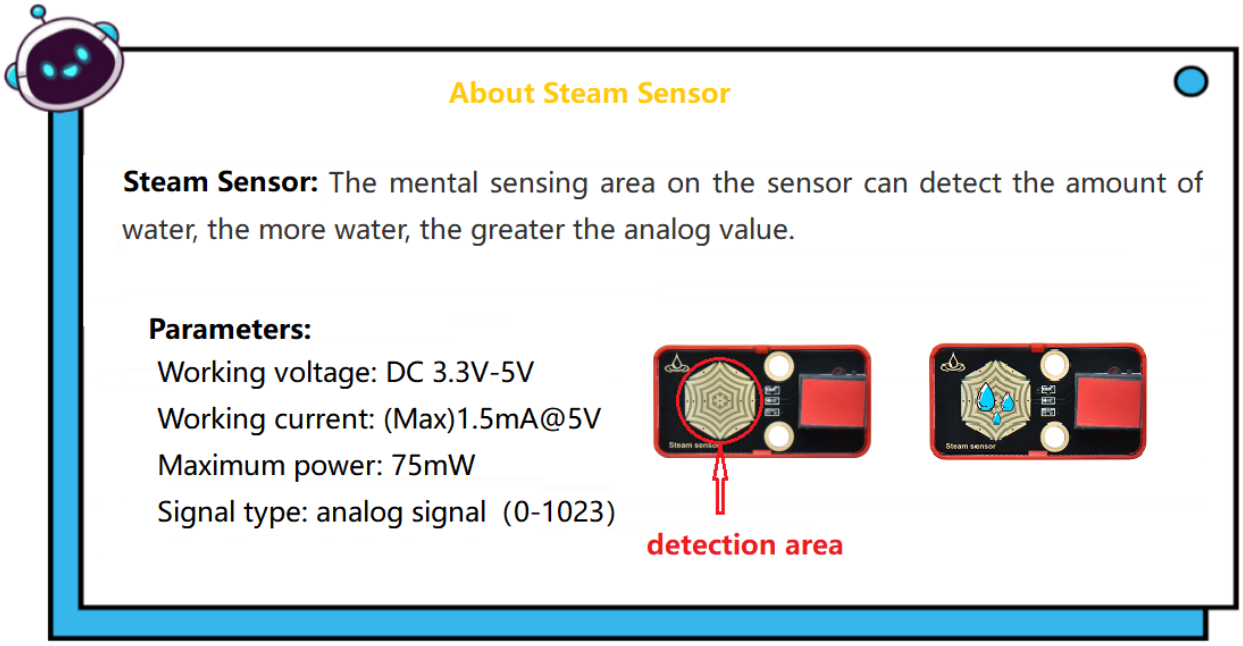
1. Description

This project explains how to use a steam sensor, a passive buzzer and a kidsIOT board to make a rain detection system. When the sensor detects rain, it sends a signal to the kidsIOT mainboard to trigger various actions.

For example, the buzzer can be used to sound an alarm to alert the user that it is raining. This system is able to monitor rainfall in gardening, agriculture, and detect leaks in roofs or buildings. The sensor can be easily connected to the kidsIOT motherboard to form a simple rain detection system.

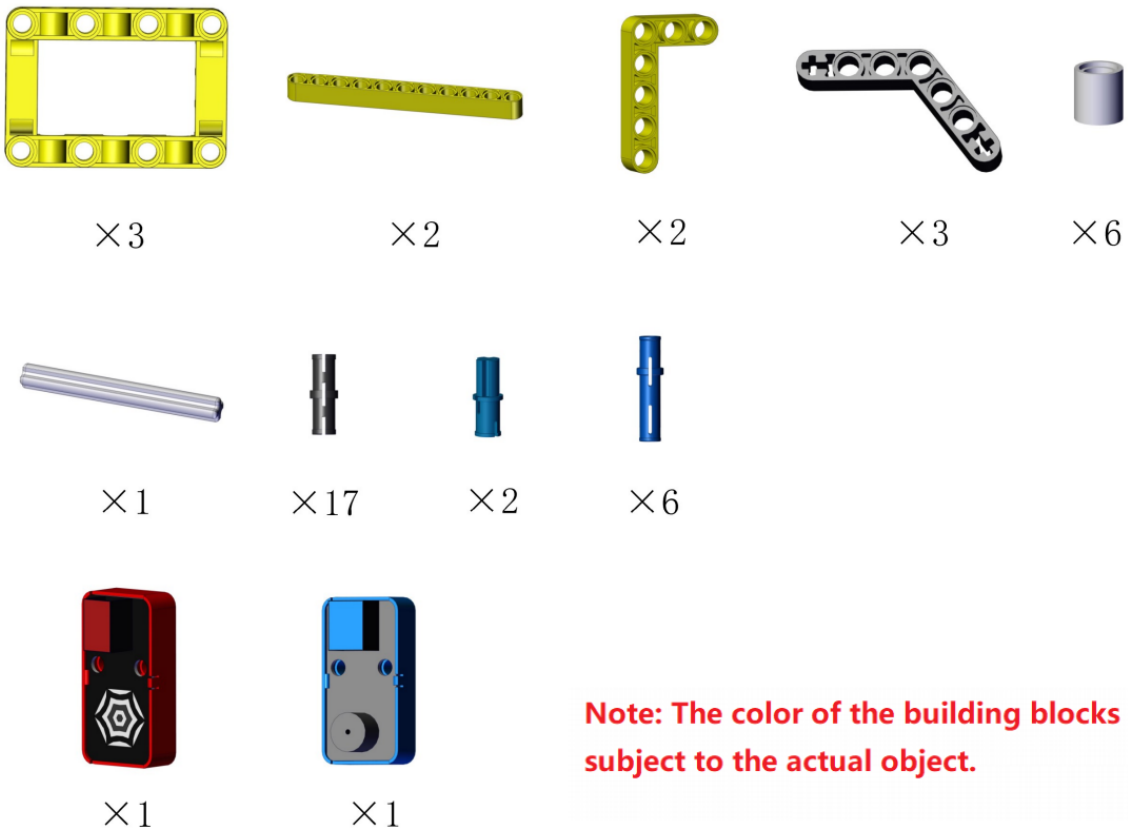
2. Components

		
kidsIOT Mainboard×1	Steam Sensor×1	Passive Buzzer×1
		
Wire×2	USB Cable×1	Rainwater Control System LEGO Pieces×1



3. Assembly Steps

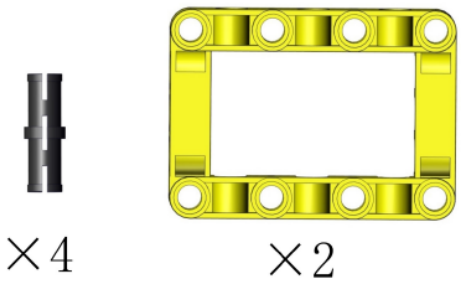
Step 1Components Needed

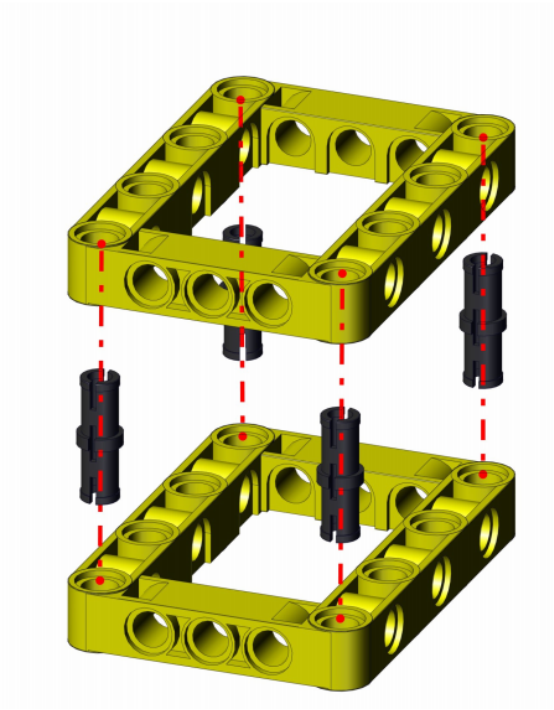


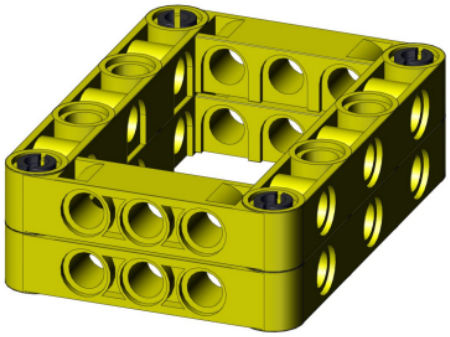
Note: The color of the building blocks is subject to the actual object.

Step 2Process

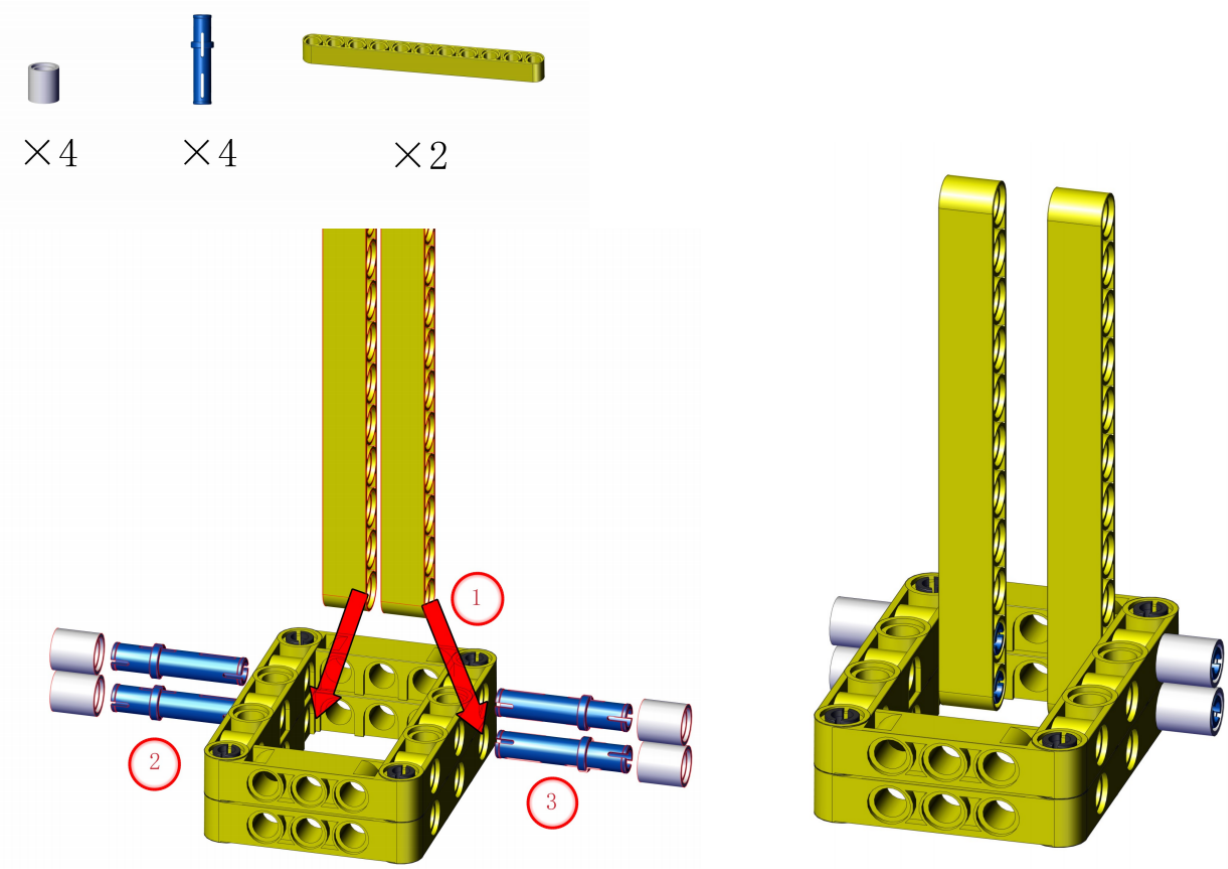
Process 1



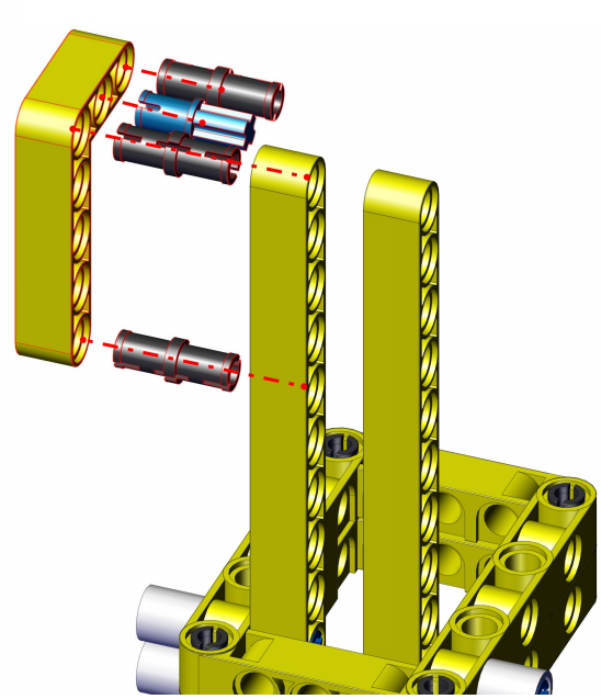


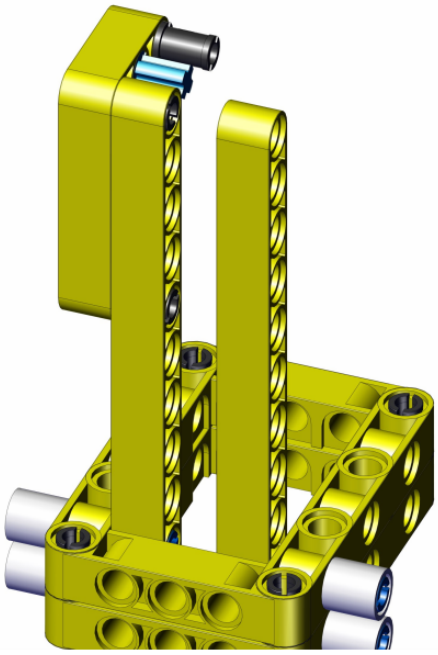


Process 2

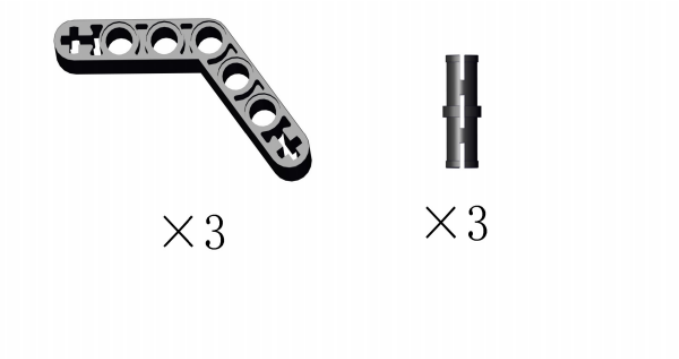


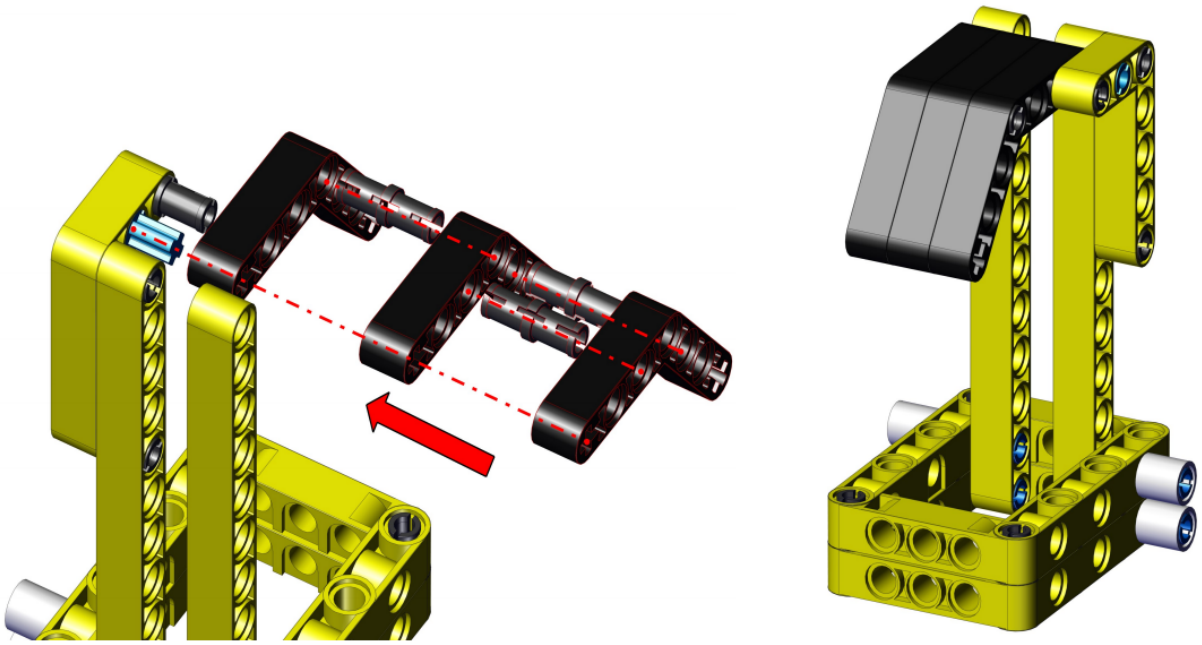
Process 3



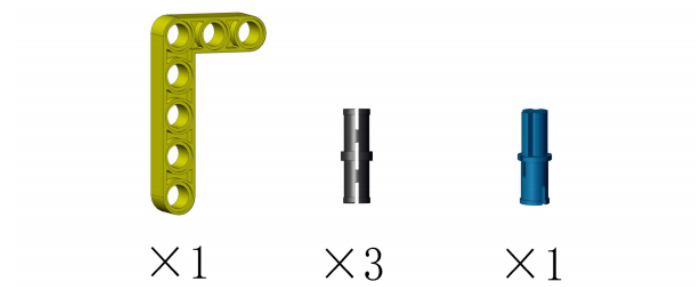


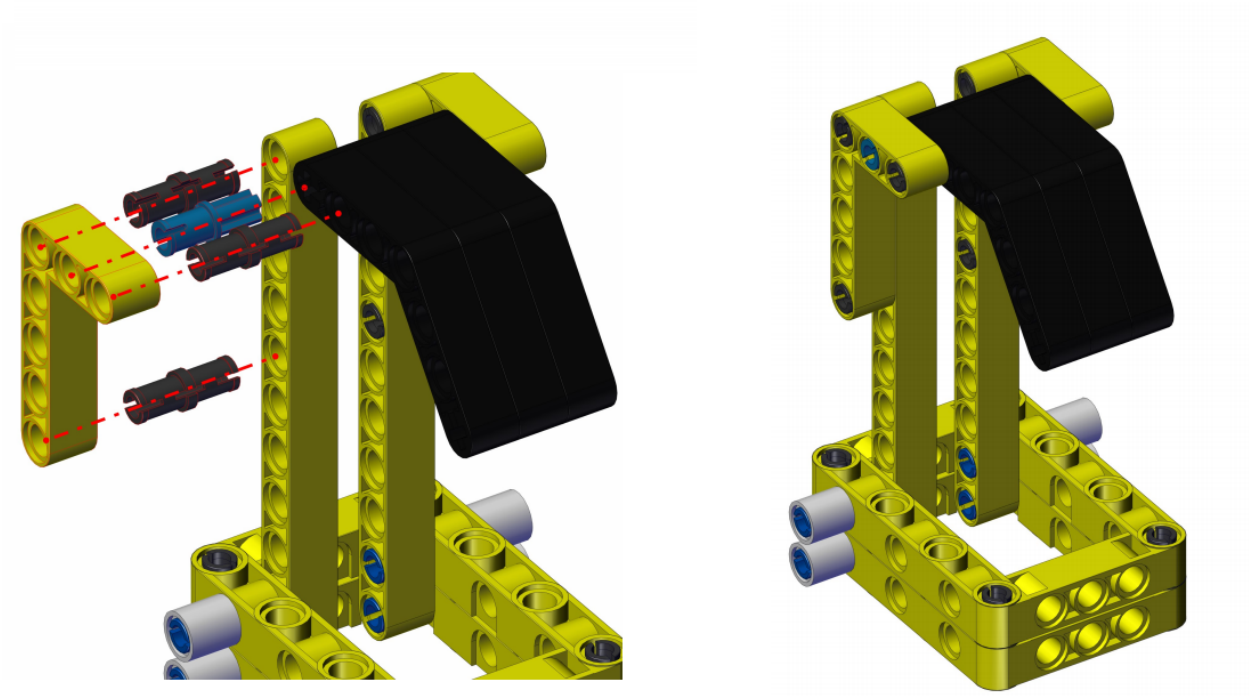
Process 4



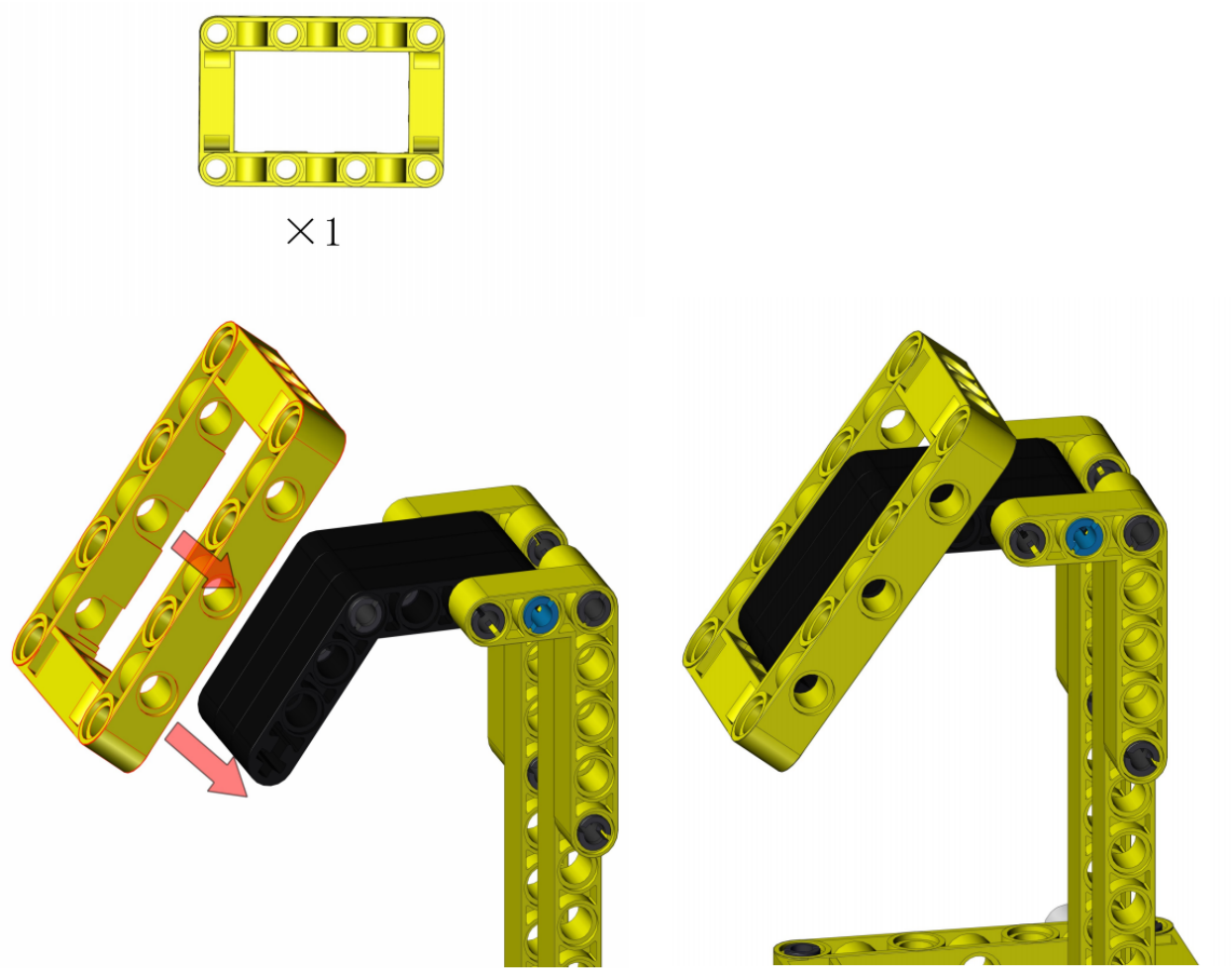


Process 5

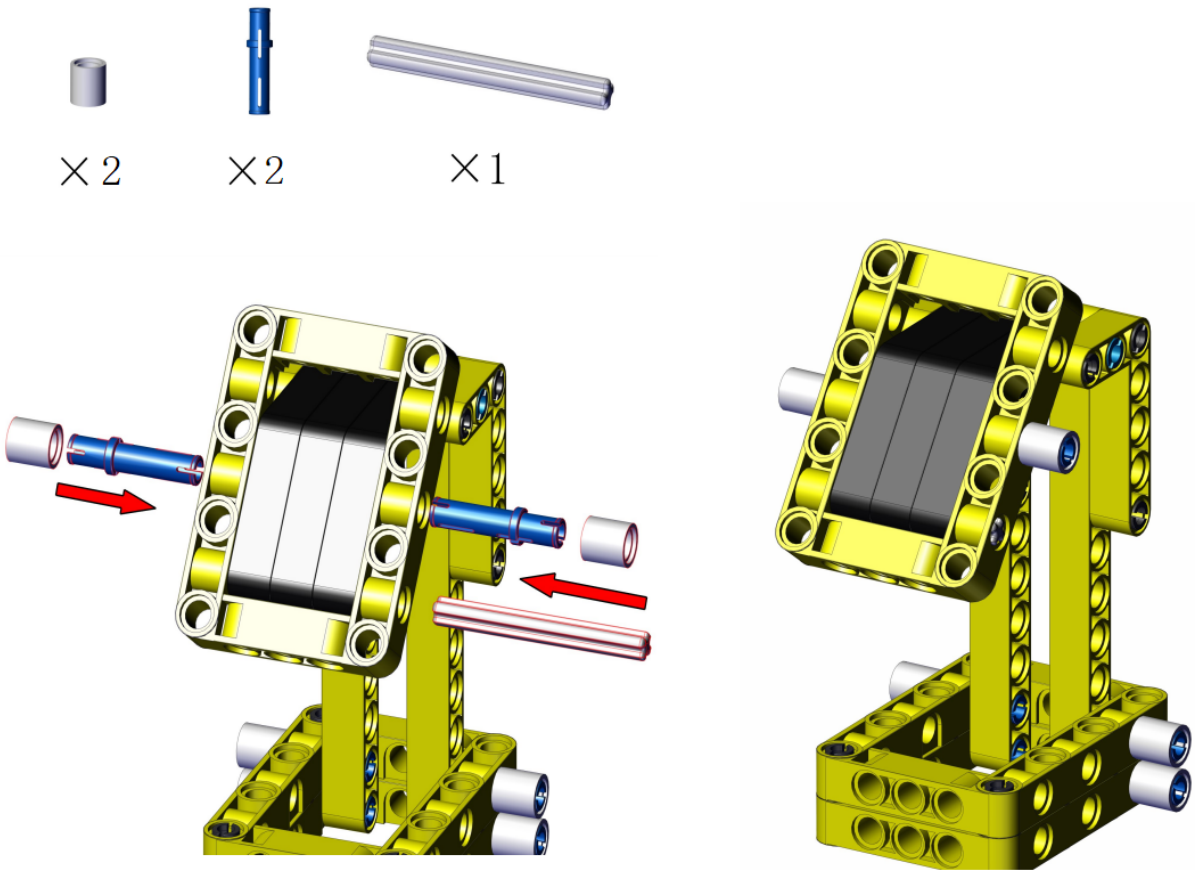




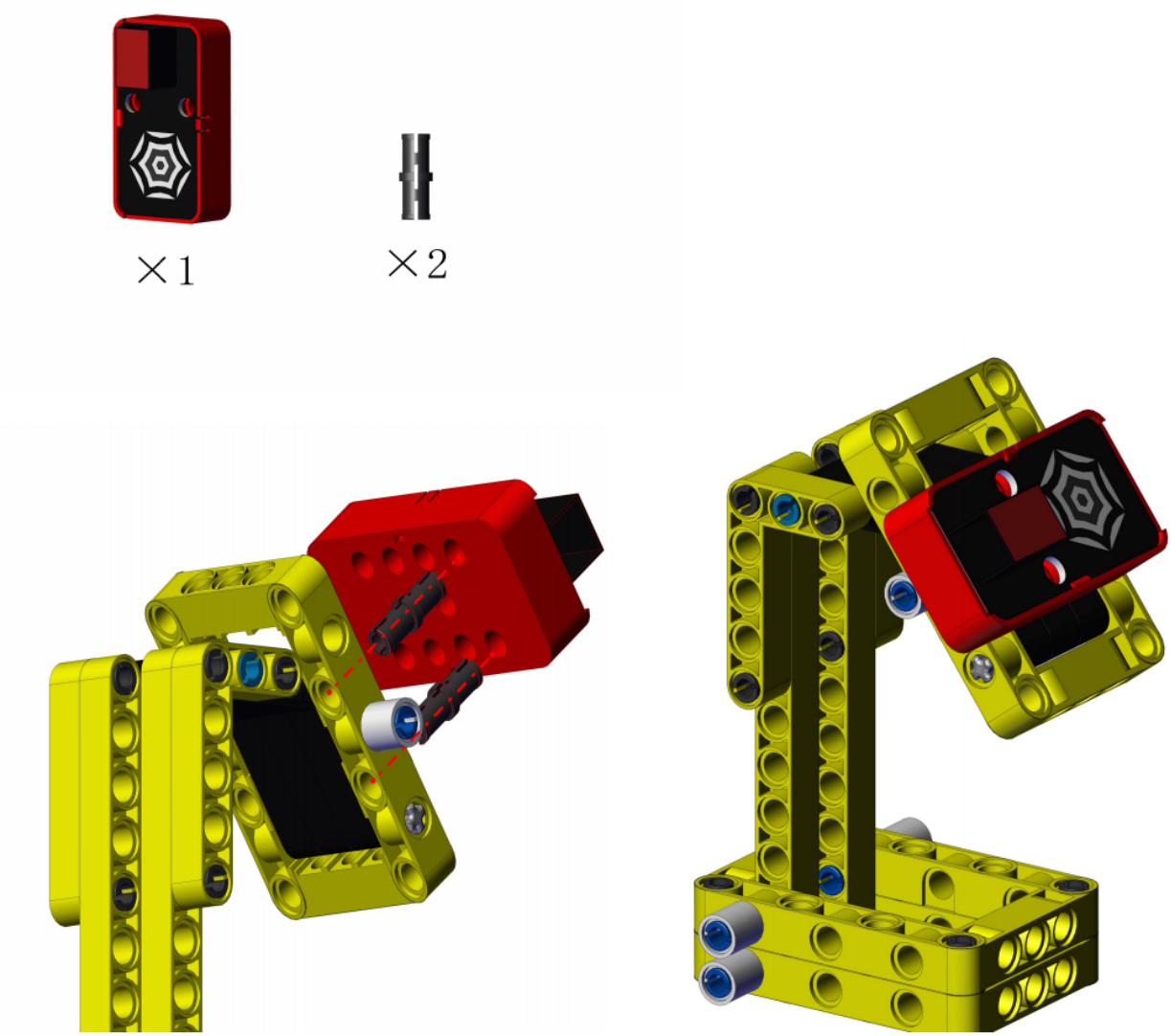
Process 6



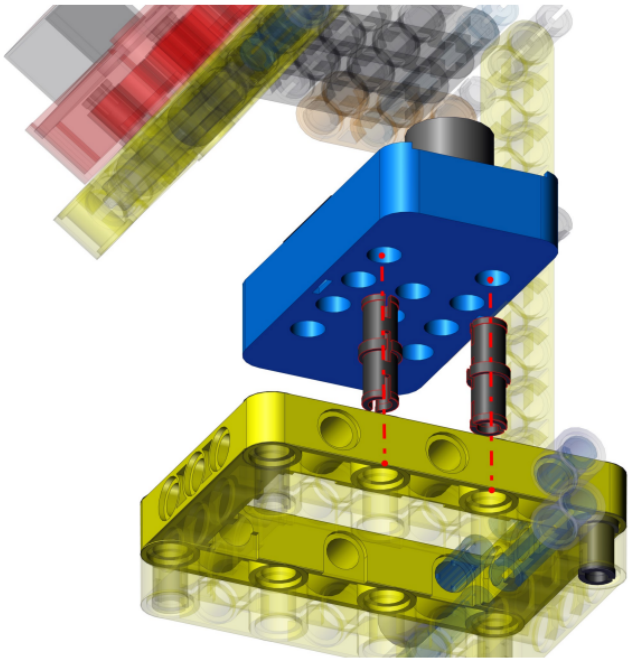
Process 7

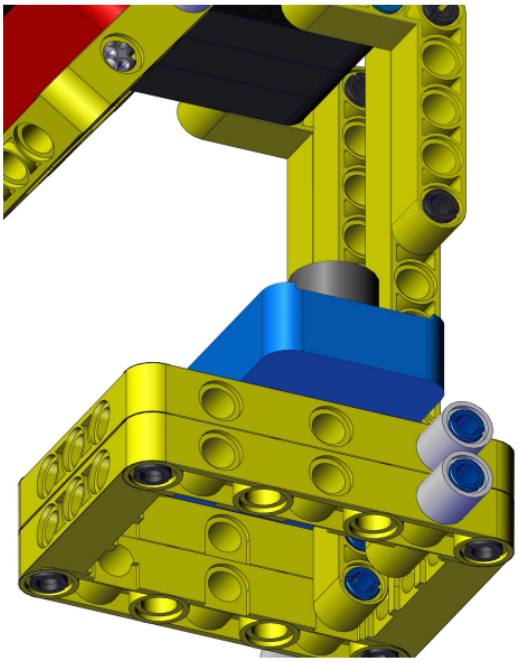


Process 8

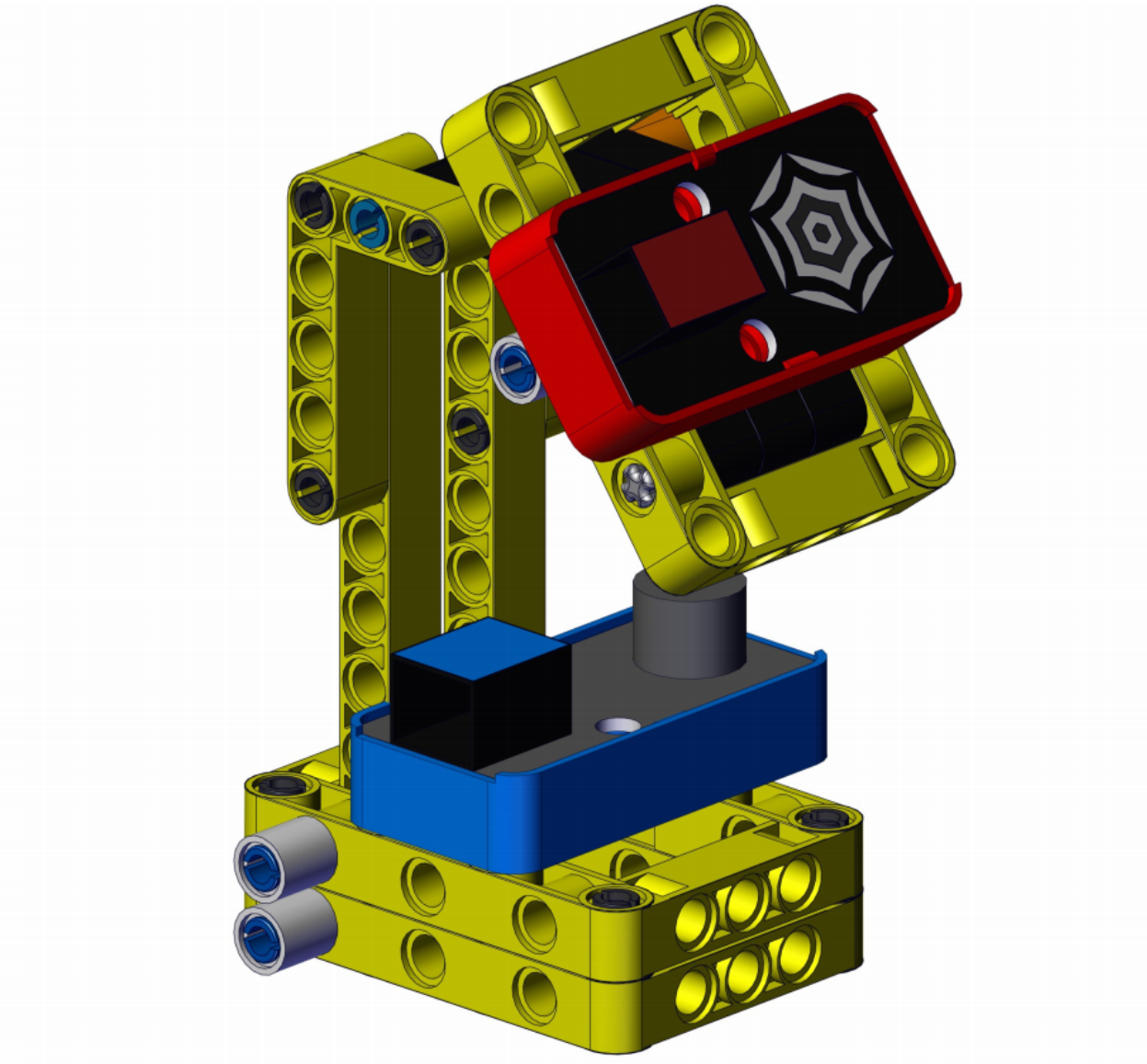


Process 9





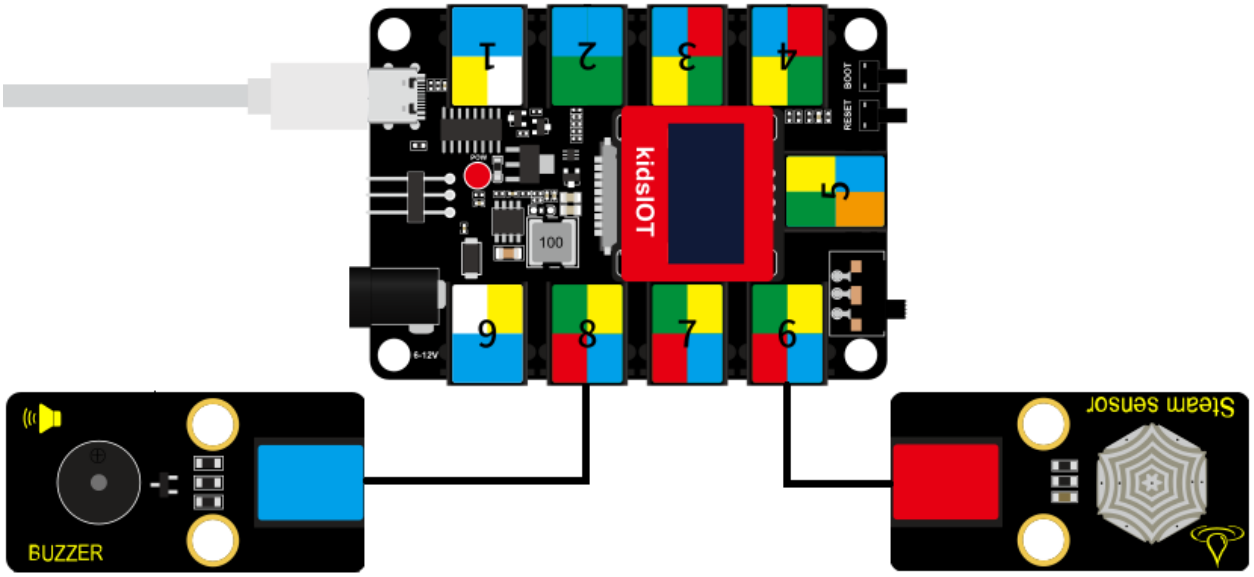
Complete



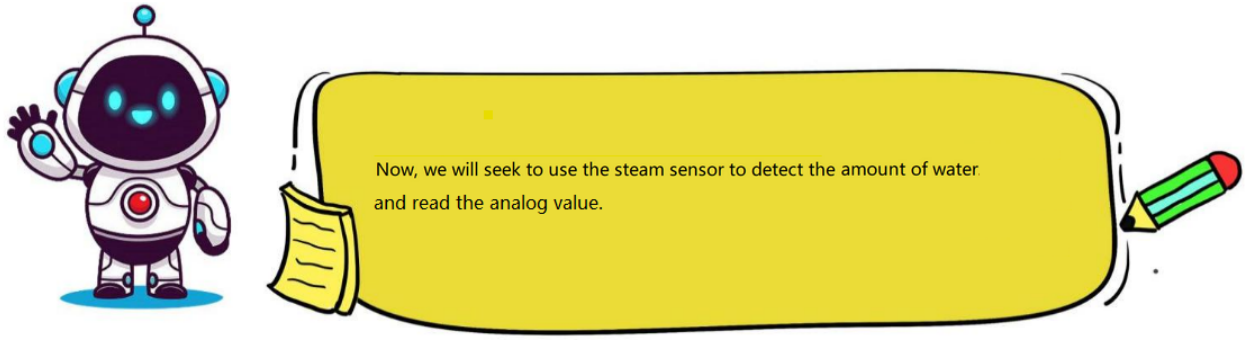
4. Wiring Diagram

Module	kidsIOT Mainboard
Steam Sensor	No.6 portcontrol pin is io36
Passive Buzzer	No.8 portcontrol pin is io5

Connect the kidsIOT mainboard to your computer via USB cable.

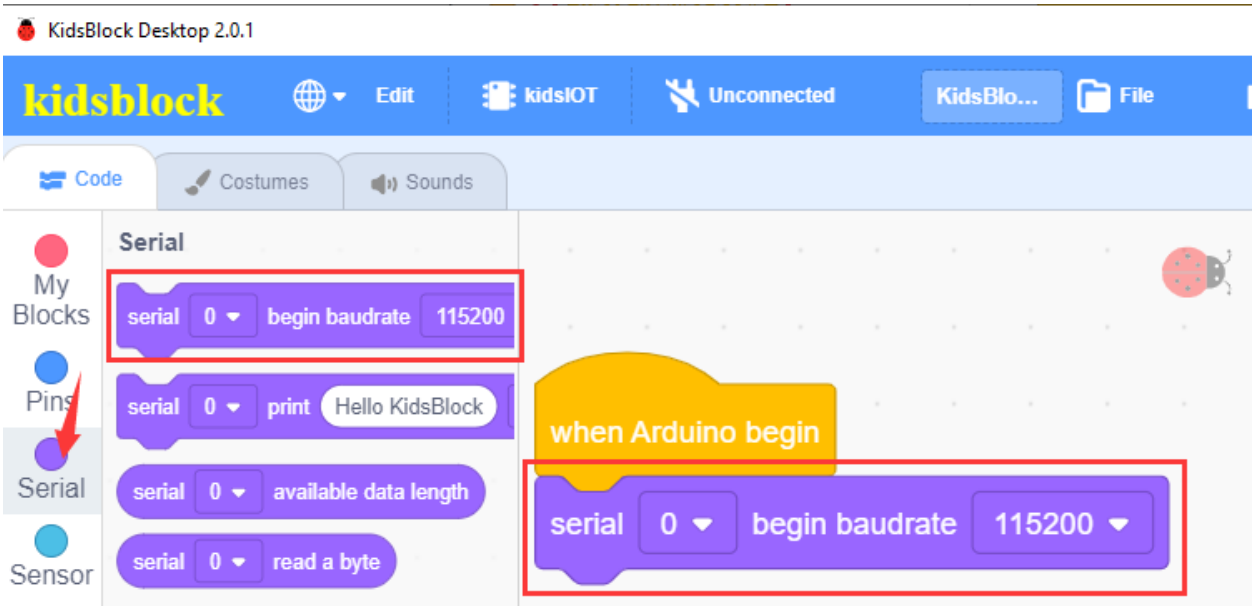


5. Read the analog value of the steam sensor

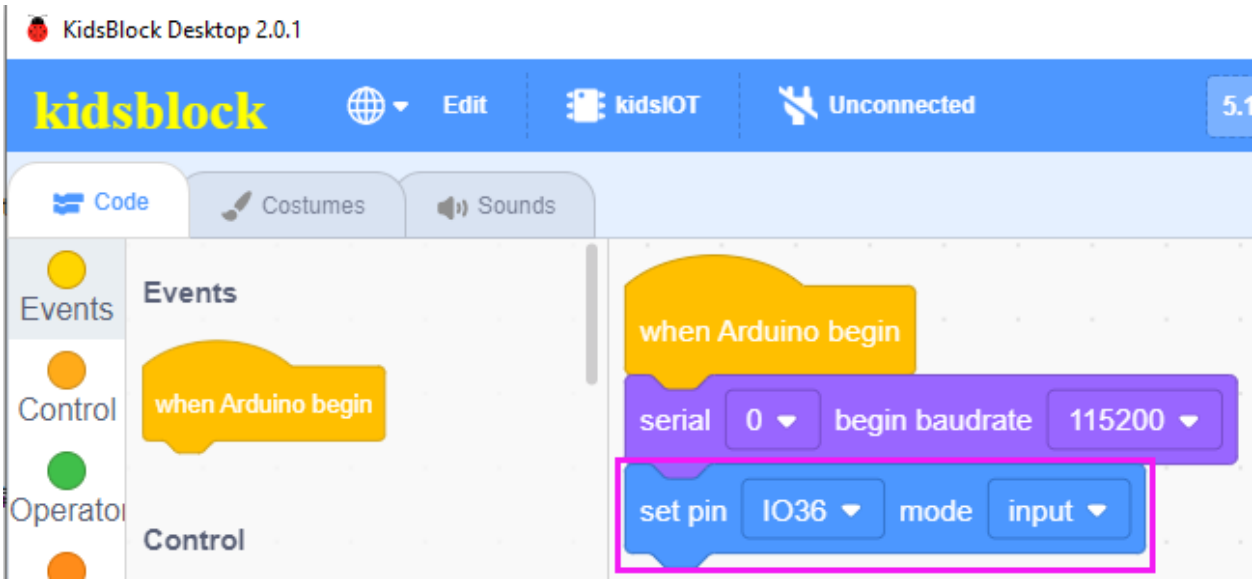


Step 1Write the Program

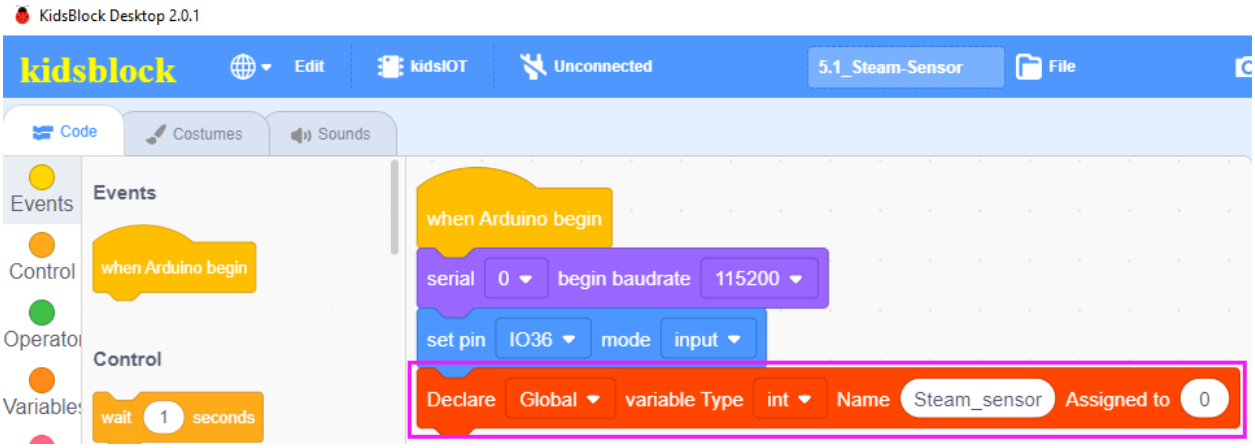
Set the baud rate to 15200.



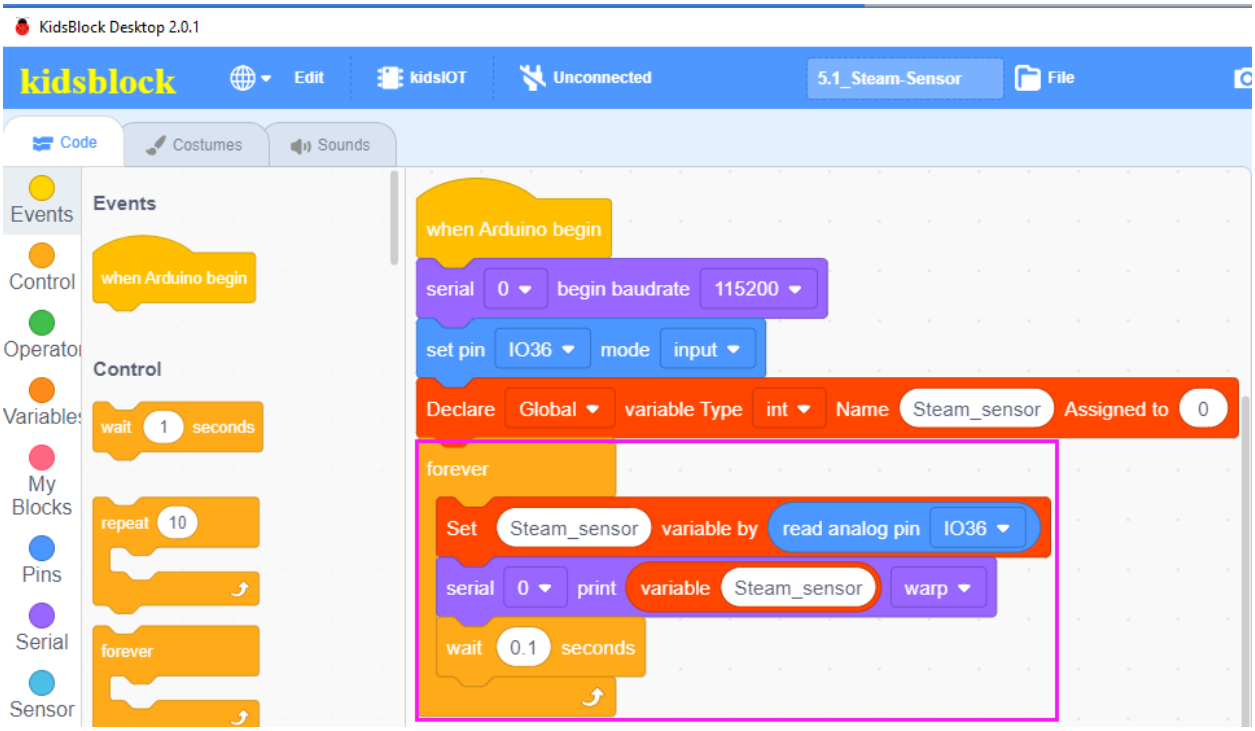
Set the pin IO36 connected to the steam sensor to "**input**" mode.



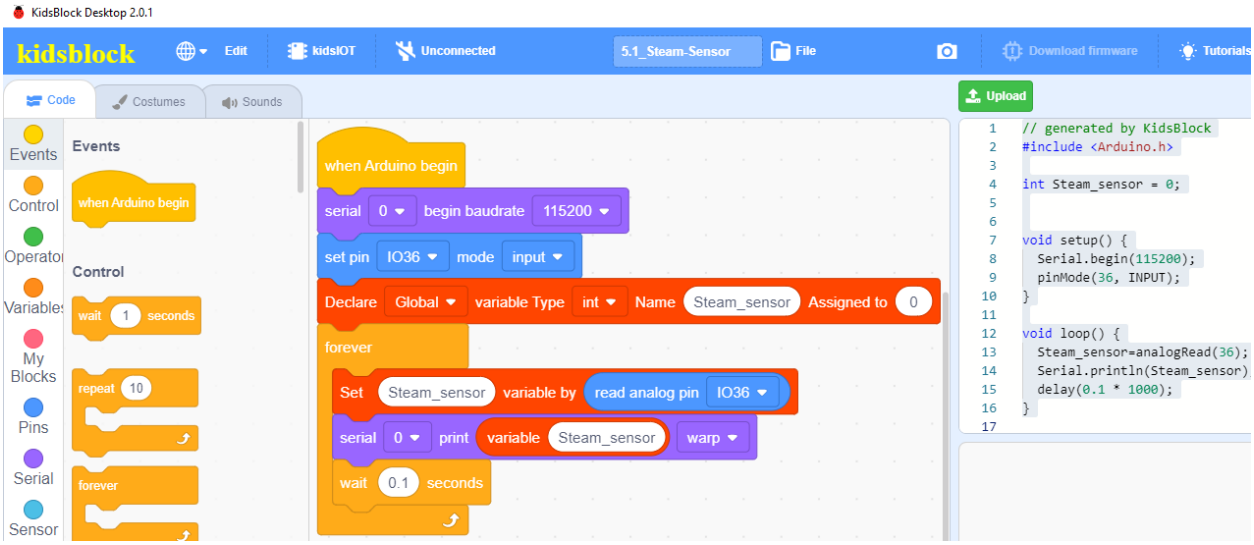
Define a "Steam_sensor" global variable to store the analog value of the steam sensor.



Store the read analog value of the sensor in the "Steam_sensor" variable and print it on the serial port.



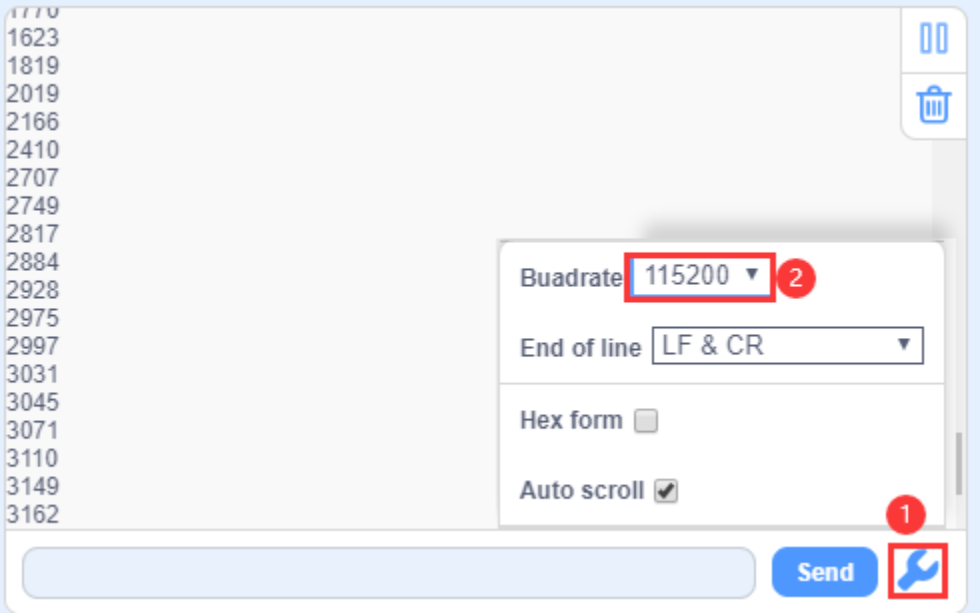
Complete Program



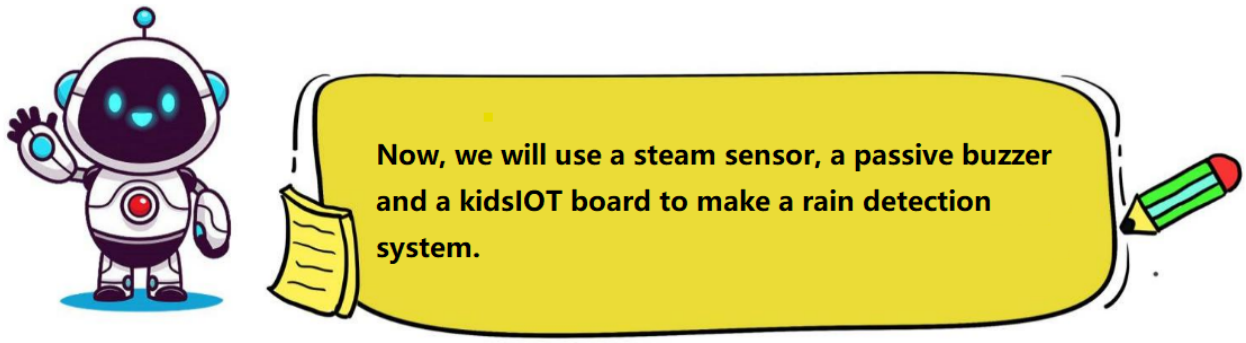
Step 2Test Result

Click  to upload the above complete code to the kidsIOT motherboard. After powering up via the USB cable, click  in the serial monitor and set the baud rate to 15200.

Then the serial monitor will print the value(range0~4095) read by the steam sensor. Touch the detection area on the sensor with a moistened finger, the larger the area, the greater the value!

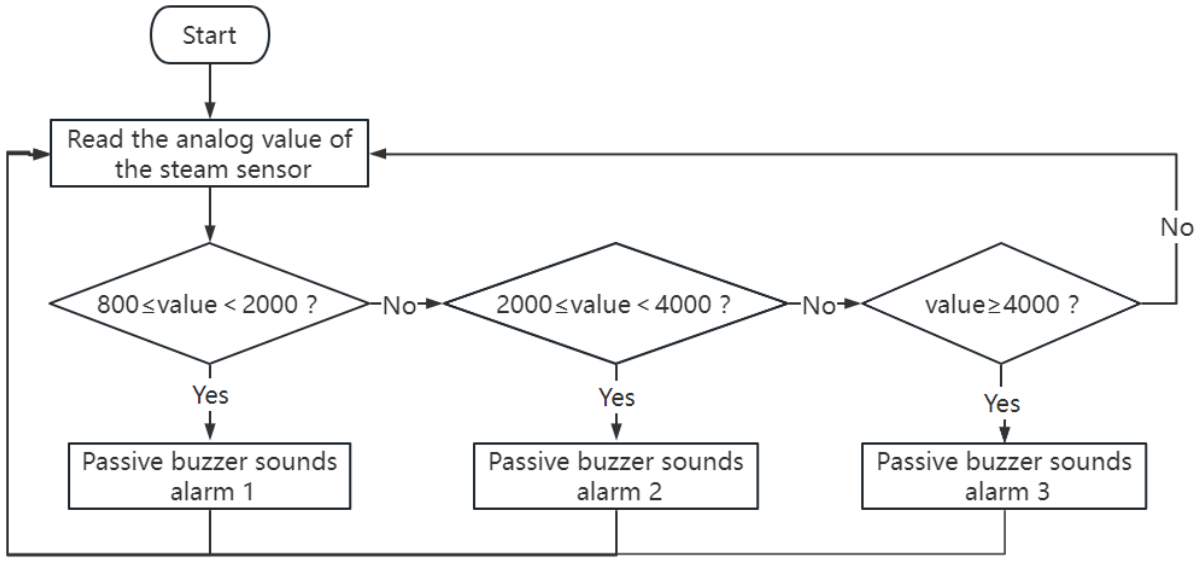


6. Rainwater Control System

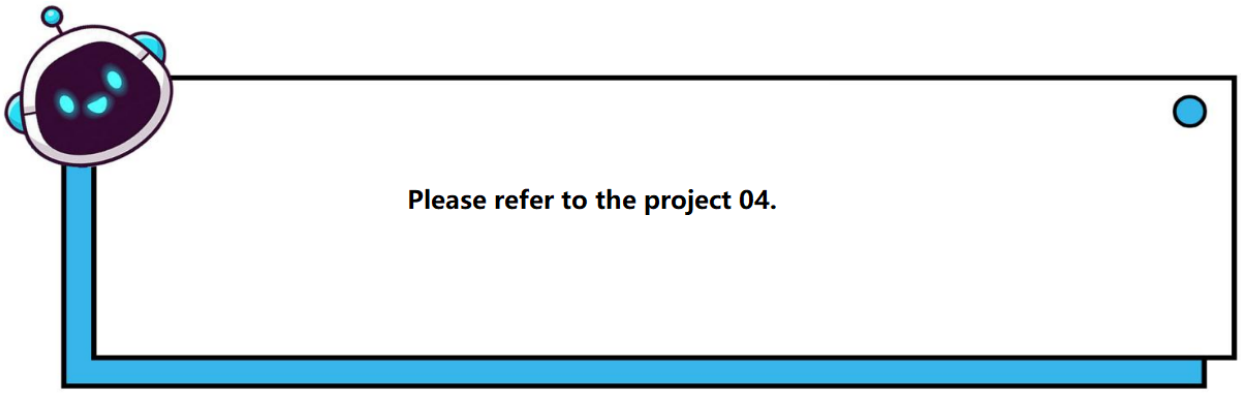


(1). Programming Steps

Step 1Flow Chart

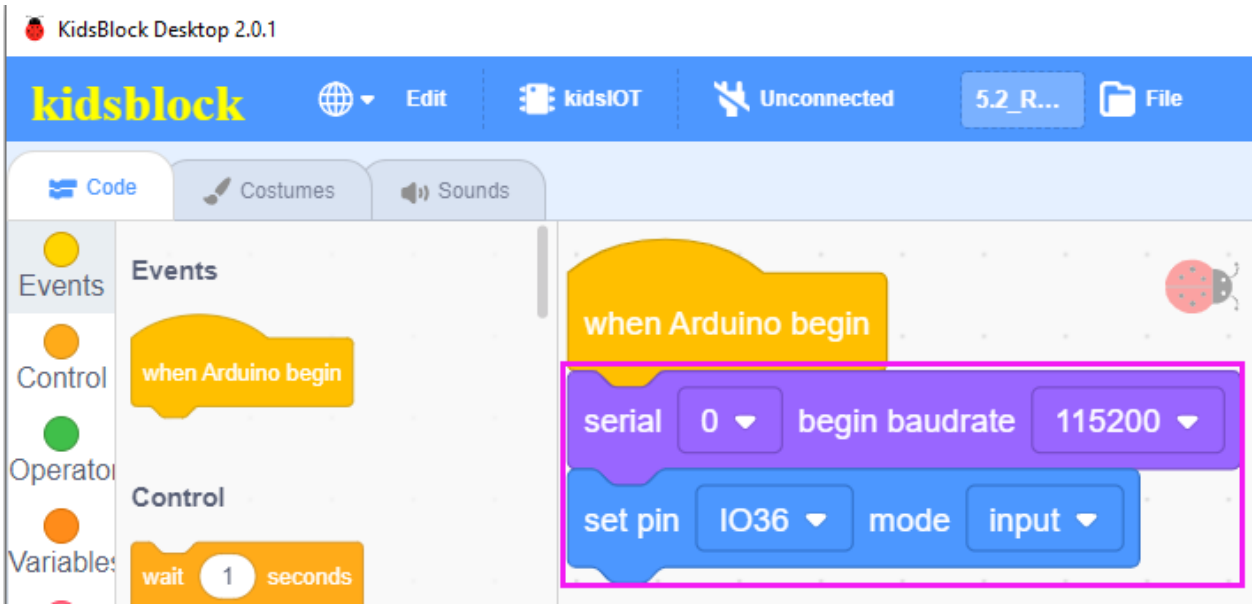


Step 2Add "passive buzzer"

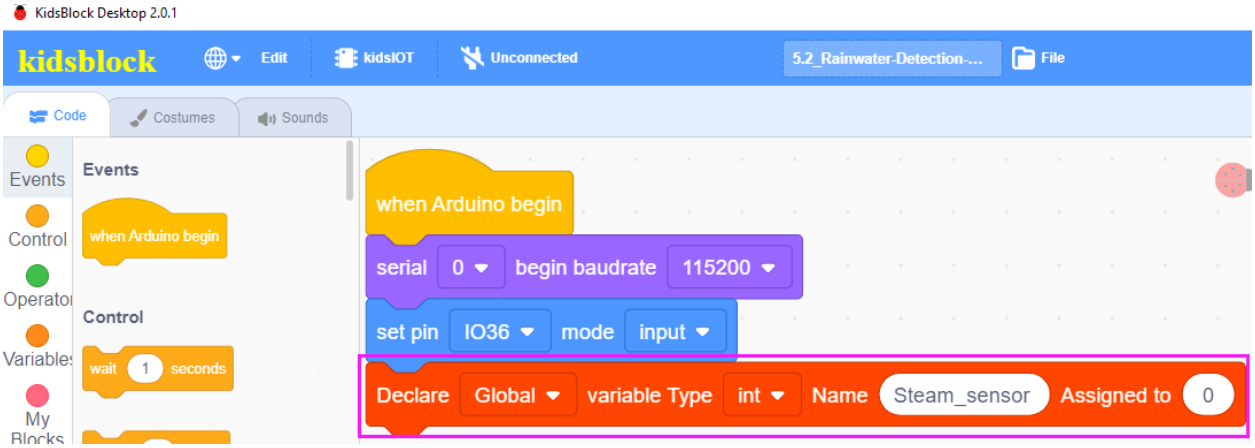


Step 3Write the Program

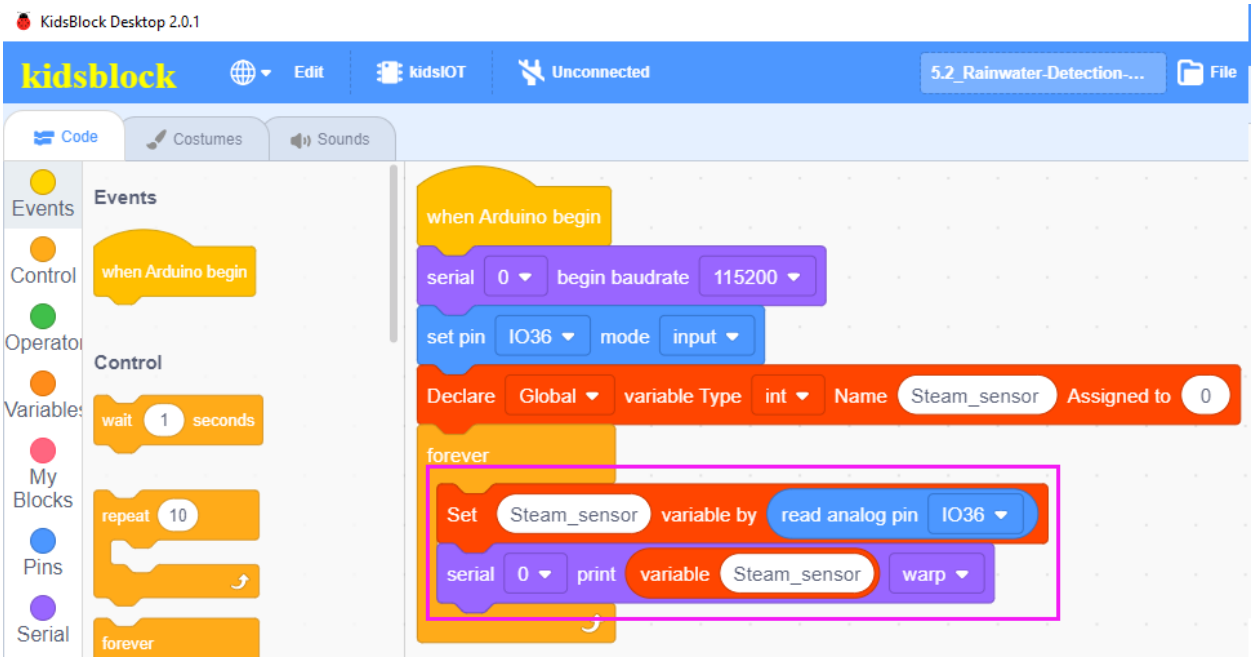
Set the baud rate to 15200, the IO36 pin of steam sensor to "**input**" mode.



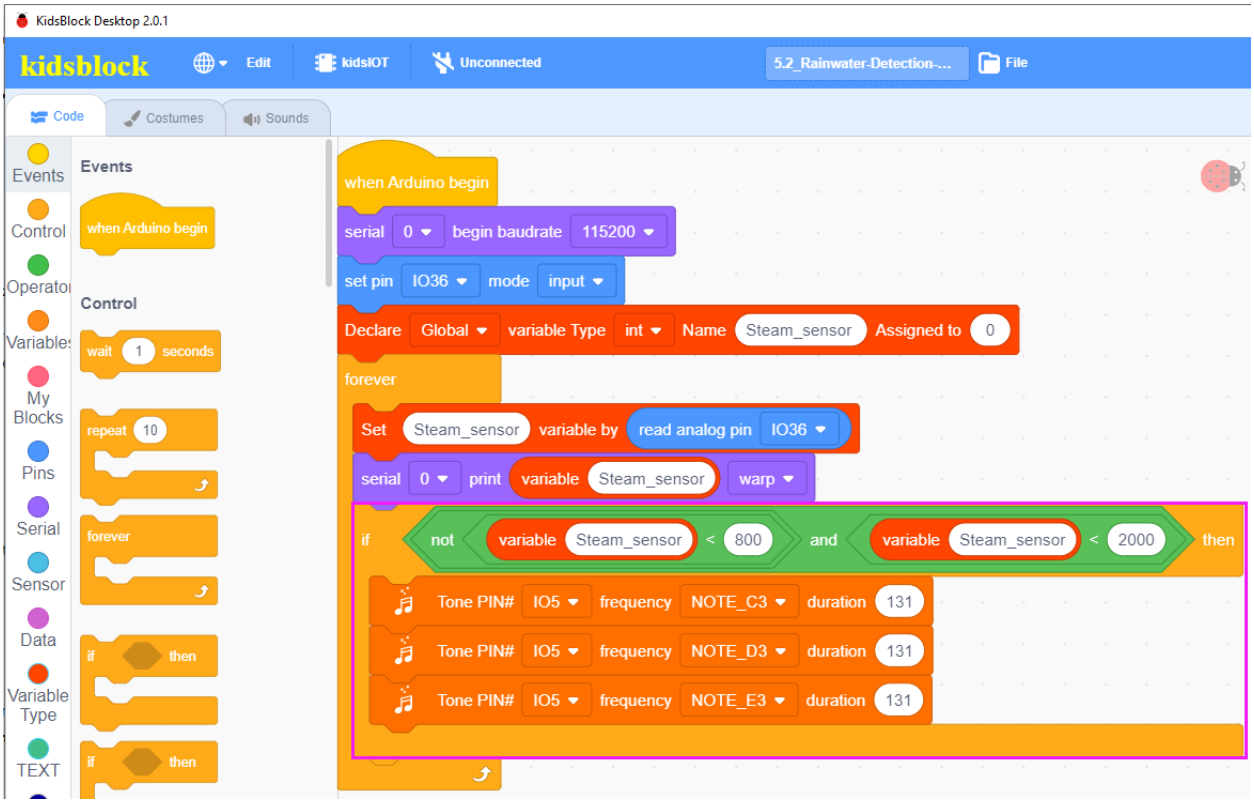
Define a "Steam_sensor" global variable to store the analog value of the steam sensor.



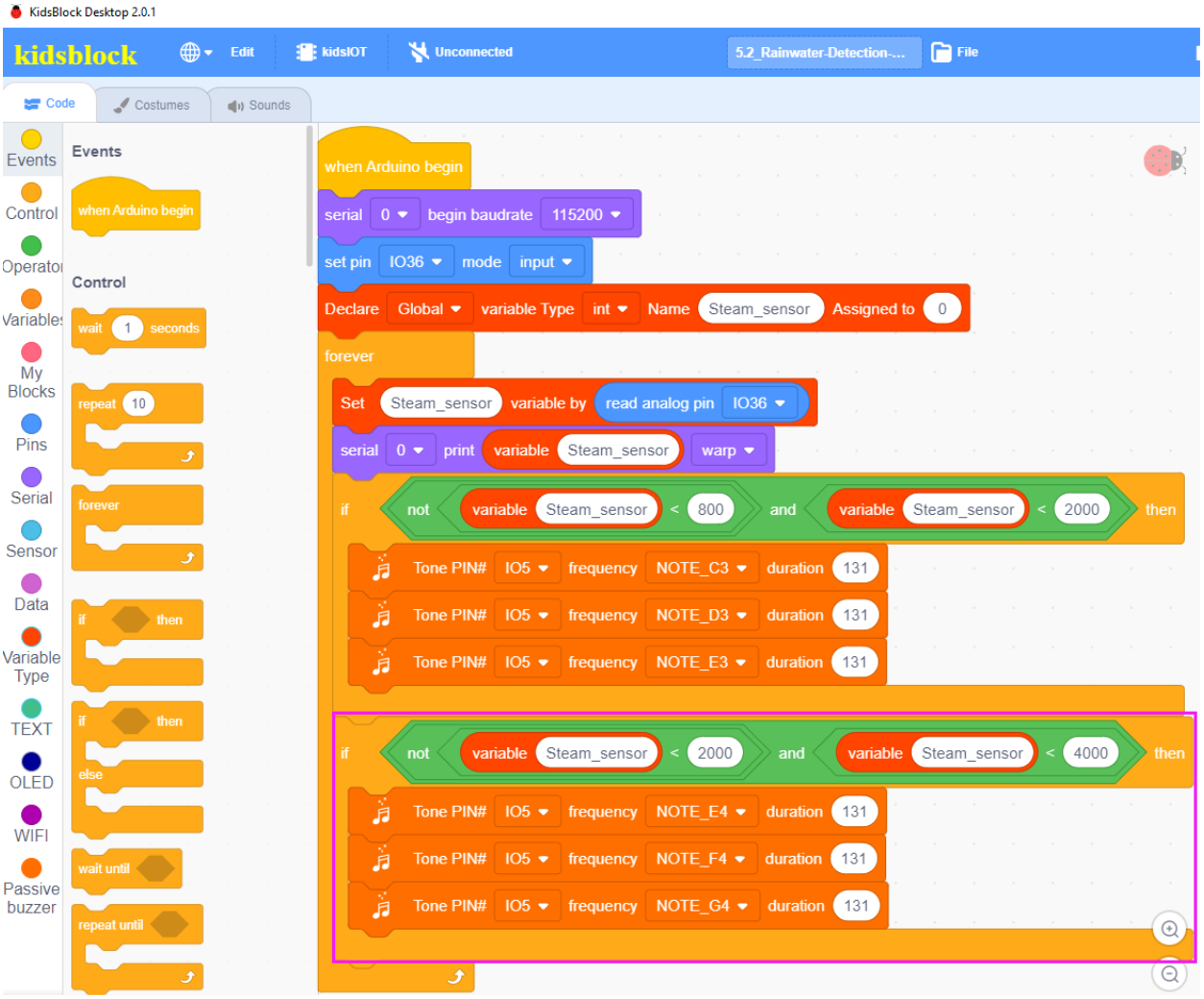
Receive analog value of the sensor and print it on the serial port.



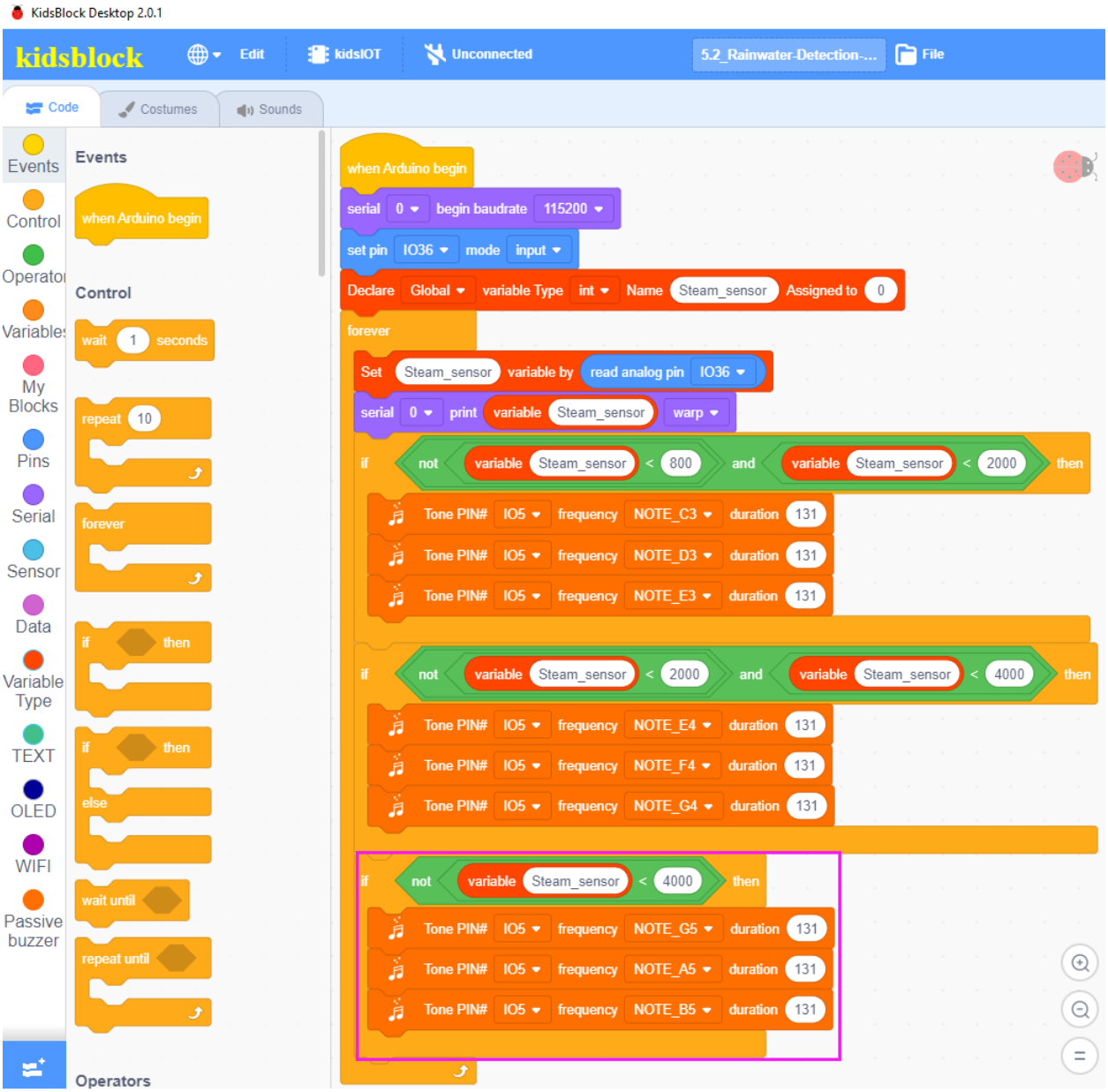
Determine the received analog value of the sensor. When 800analog value<2000, the buzzer will sound alarm 1.



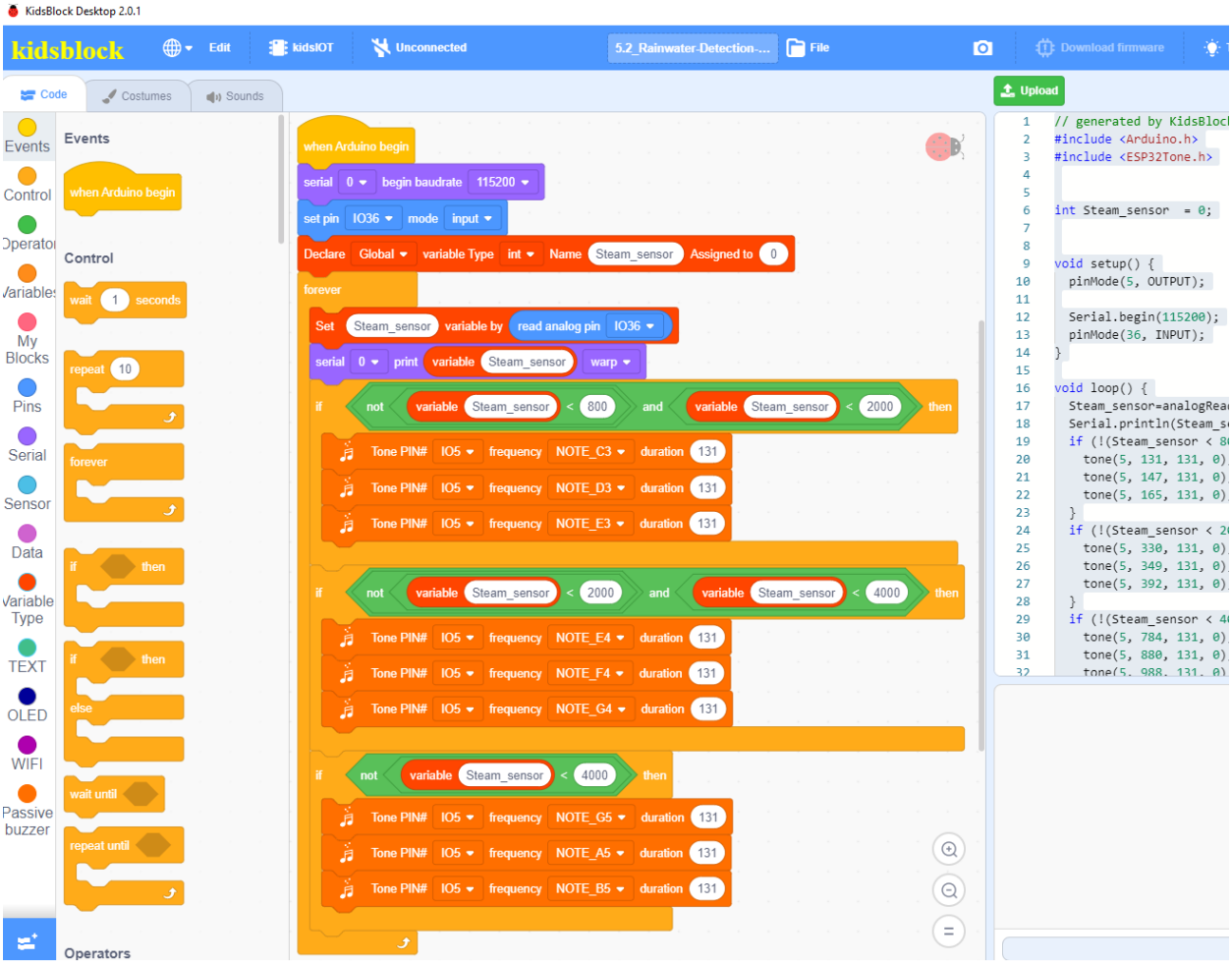
Determine the received analog value of the sensor. When 2000analog value<4000, the buzzer will sound alarm 2.



Determine the received analog value of the sensor. When analog value4000, the buzzer will sound alarm 3.



Complete Program



(2). Test Result

Click 

to upload the above complete code to the kidsIOT mainboard. After powering up via the USB cable, when the rain value detected by the sensor is larger, the buzzer alarm sound will be louder.

7. Common Problems

Q1: Is the steam sensor waterproof?

A: The detection area can be exposed to water, but when detecting water, please be careful not to use too much water.

Q2: After the sensor detects water, the alarm still sounds after a long time?

A: The passive buzzer keeps alarming because there are still water stains in the detection area of the sensor. Just clean it.

4.3.6 Project 06Temperature and Humidity Control System

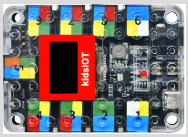
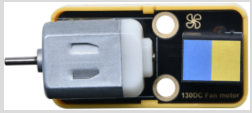
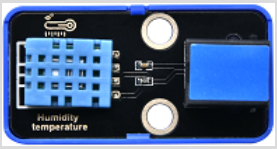
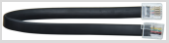
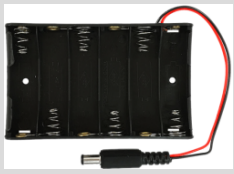
1. Description

This project introduces how to use a kidsIOT mainboard, a temperature and humidity sensor, a fan and an OLED display to build an intelligent temperature and humidity control system.

The system can measure ambient temperature and humidity and control fans to cool down and dehumidify based on demand. When the temperature or humidity exceeds the set threshold, it will automatically turn on the fan to reduce the temperature or humidity in the environment below the set value to protect the animals and plants on the farm.

What's more, it enables to adjust the ambient temperature and humidity and display them on the OLED display.

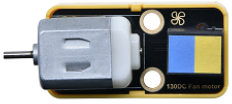
2. Components

				
kidsIOT Mainboard×1	Motor×1	Temperature and Humidity Sensor×1	Wire×2	Battery Holder×1
				
Fan×1	USB Cable×1	Temperature and Humidity System LEGO Pieces×1	AA BatteryNot provide×6	

About motor, temperature and humidity sensor and OLED display

Motor: It is a module that can control the rotation of the motor via the voltage direction of the output signal ends IN+ and IN-, and it can control the output PWM signal to adjust the speed of the motor. It is suitable for applications where fan speed and rotation direction need to be adjusted such as computer cooling and industrial production.

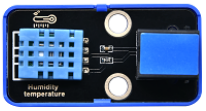
Parameters:



Working voltage: DC 3.3V-5V
Working current: (Max)200mA@5V
Maximum power: 2W
Speed: (2500±10%) rpm @3.3V; (16000±10%) rpm @5V

Temperature and Humidity Sensor: It is a digital output temperature and humidity sensor that uses special analog signal acquisition, conversion technology and temperature and humidity sensing technology to ensure its good stability. It contains a high-precision resistive temperature and humidity sensor and a resistive thermal sensor, and is connected to an 8-bit high-performance microcontroller.

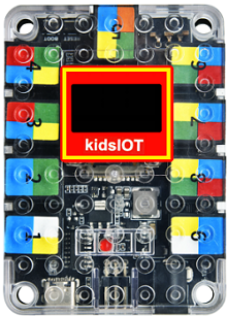
Parameters:



Working voltage: DC 3.3V-5V
Working current: (Max)50mA@5V
Maximum power: 0.25W
Temperature measurement range: -25℃-+60℃
Temperature accuracy: ±2℃
Humidity measurement range: 20-90%RH
Humidity accuracy: ±5%RH

OLED: It is a display featuring clear picture quality, small size and high brightness.

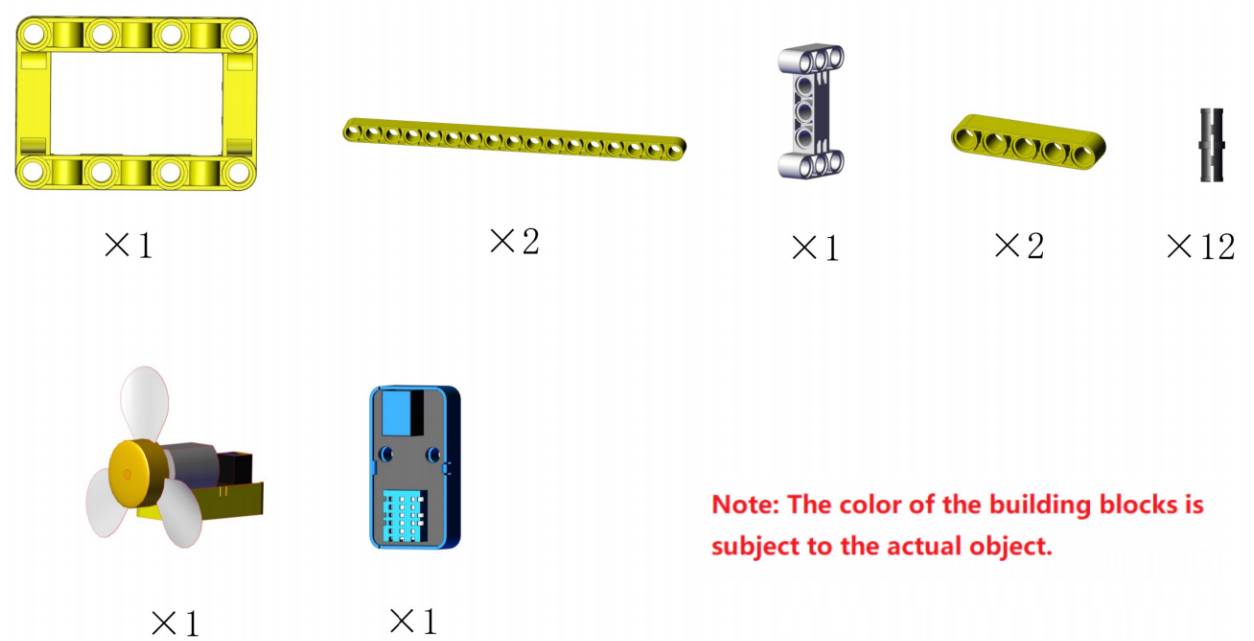
Parameters:



Working voltage: DC 3.3V-5V
Maximum power: 0.06W
Viewing angle: >160°
Resolution: 128*64
Communication mode: IIC
Driver chip: SSD1306

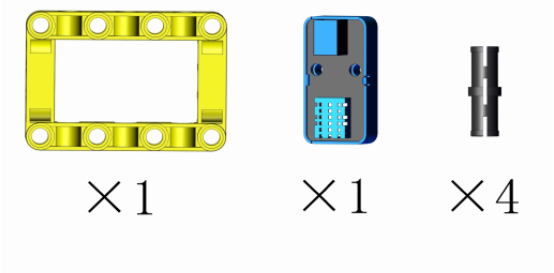
3. Assembly Steps

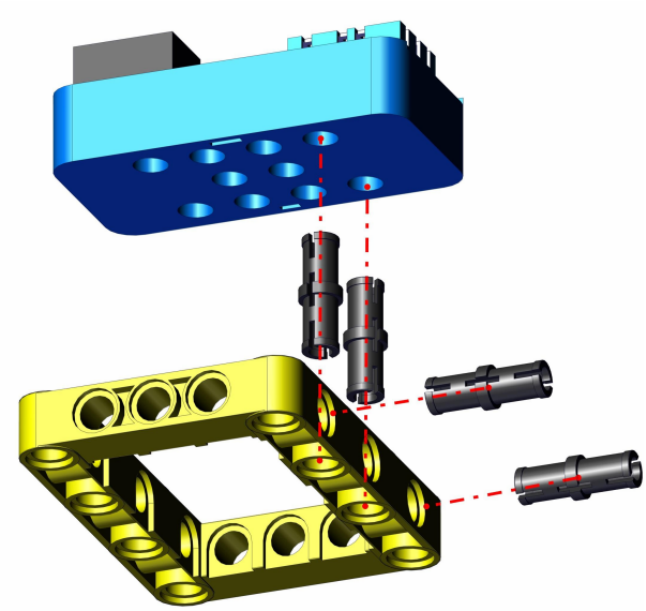
Step 1Components Needed

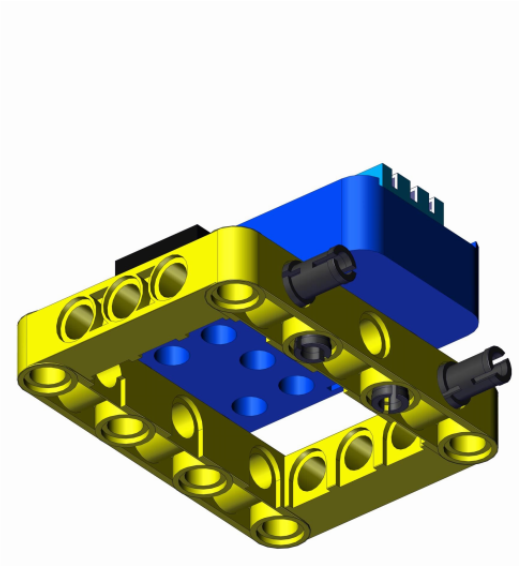


Step 2Process

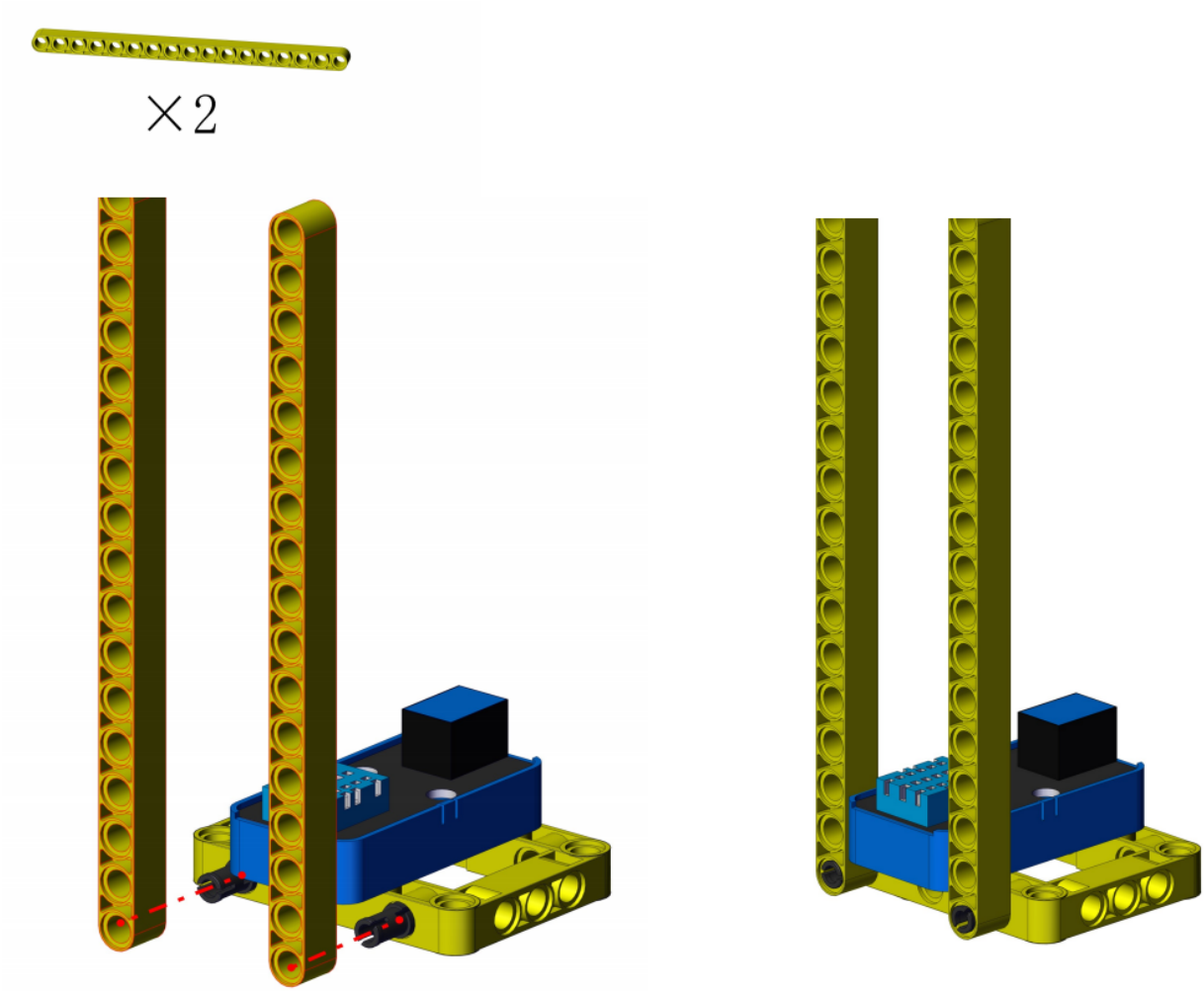
Process 1



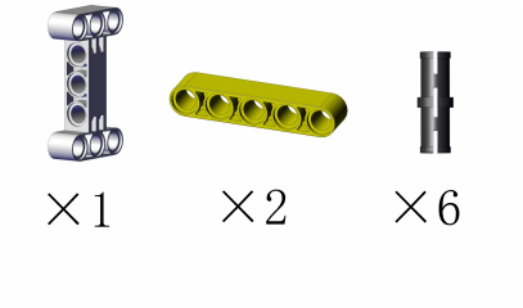


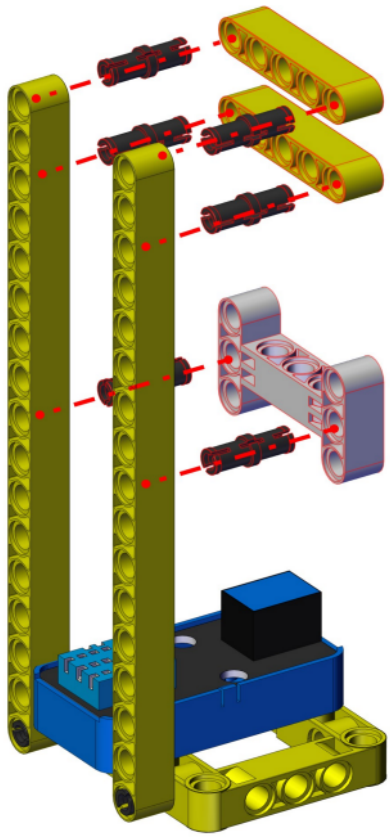


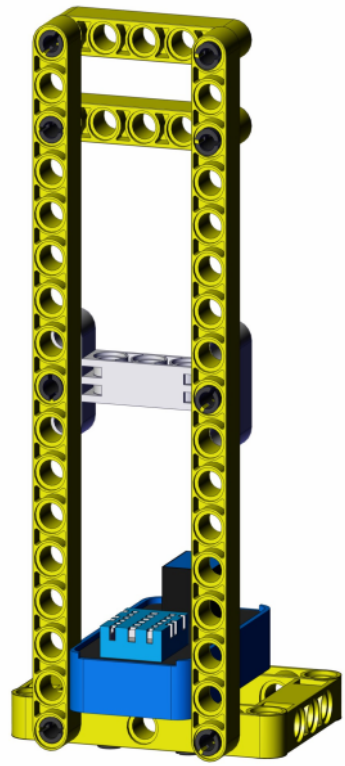
Process 2



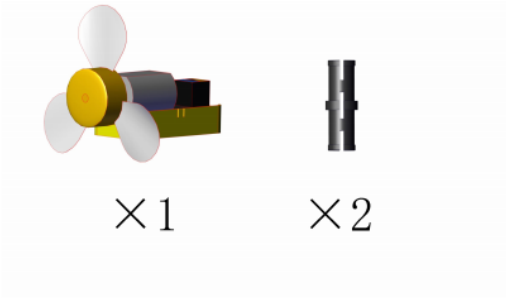
Process 3

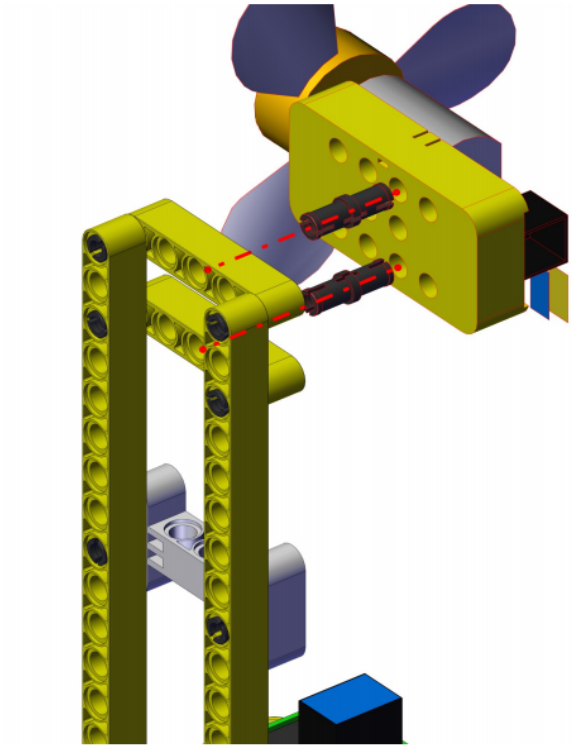


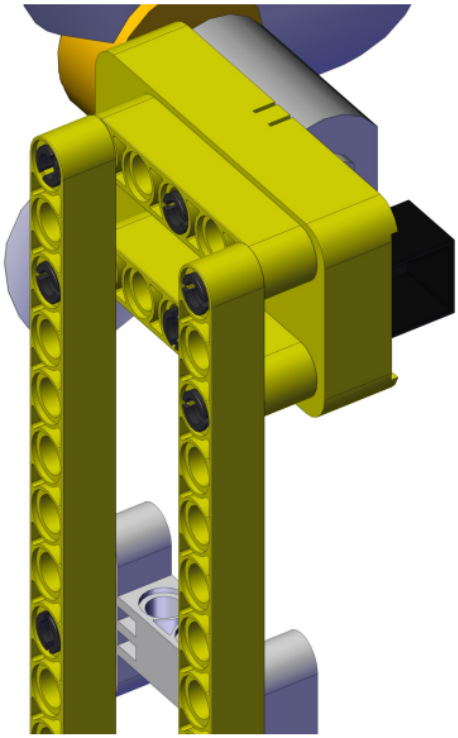




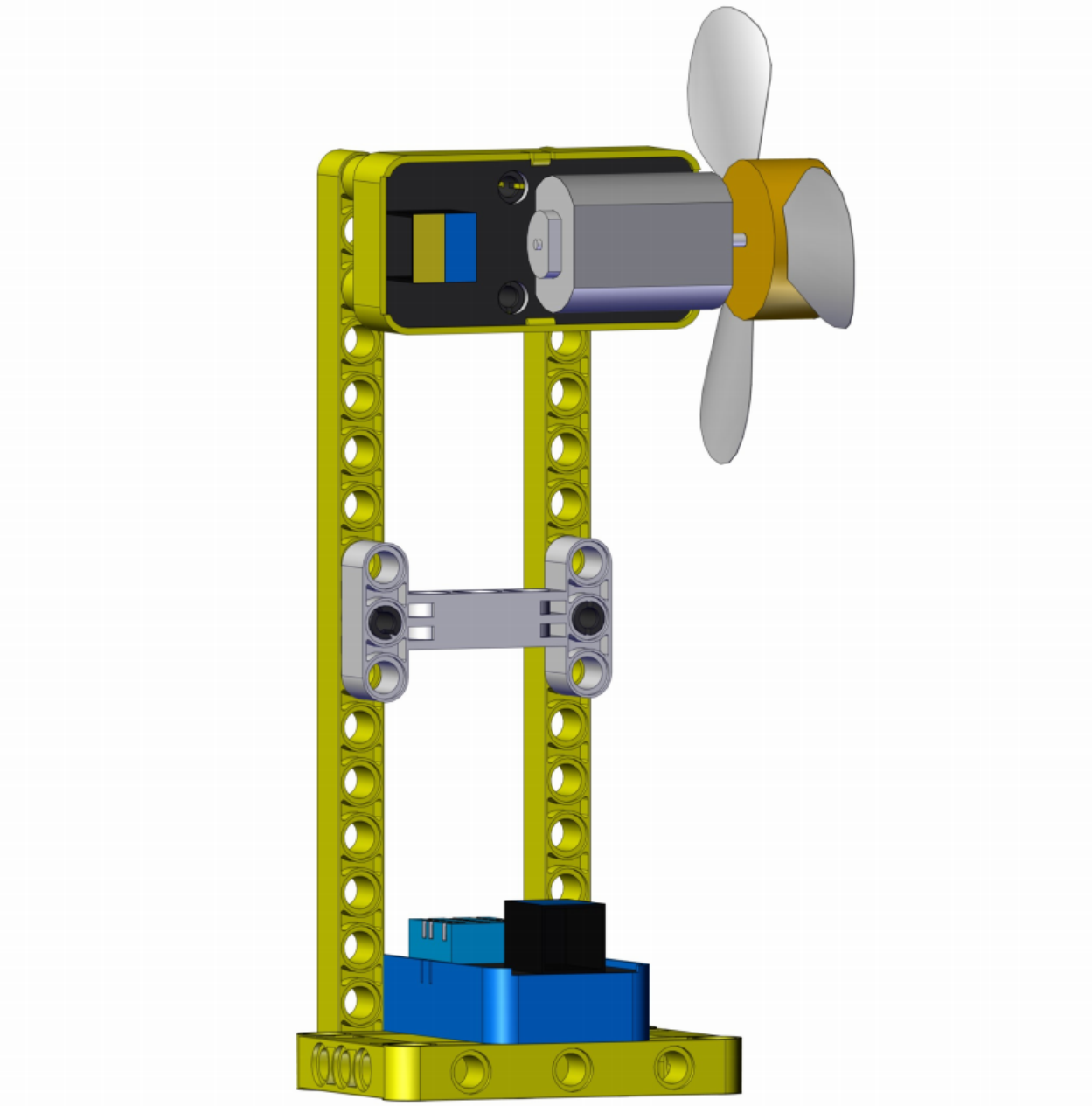
Process 4







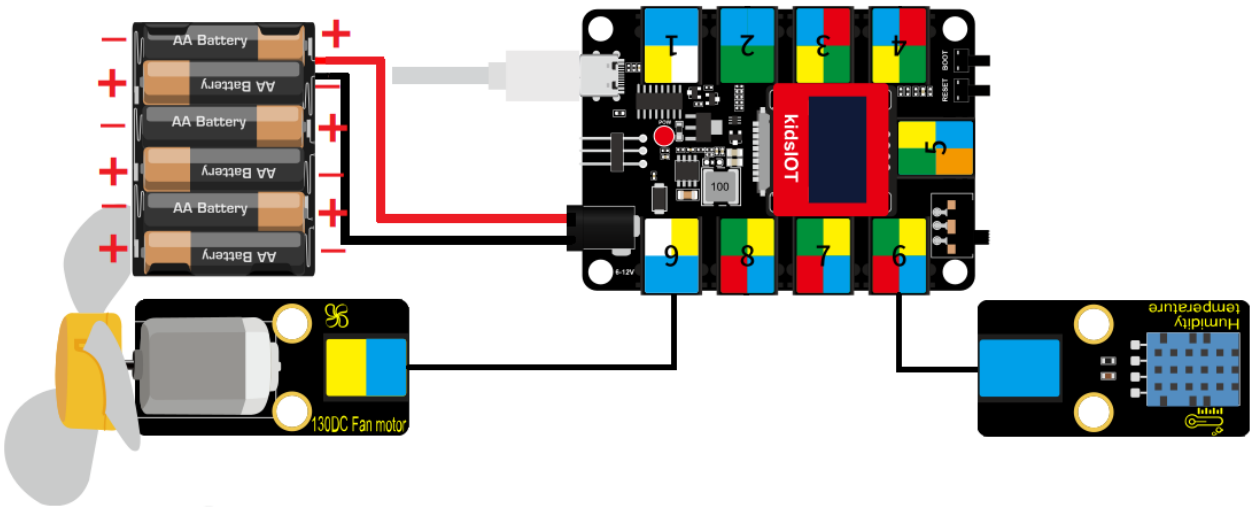
Complete



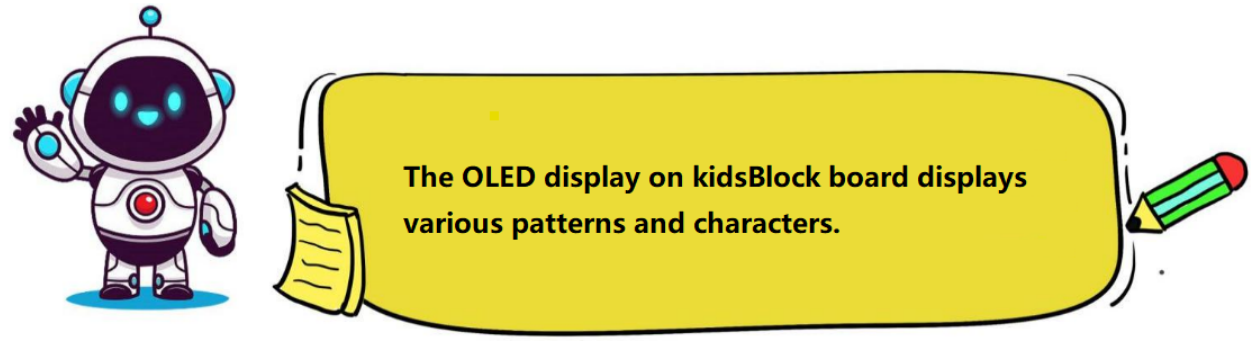
4. Wiring Diagram

Module	kidsIOT Mainboard
Temperature and Humidity Sensor	No.6 portcontrol pin is io23
Motor	No.9 portIN+control pin is io18IN-control pin is io19

Connect the kidsIOT mainboard to your computer via USB cable, connect the external power supply and turn the DIP switch on the mainboard to ON end.



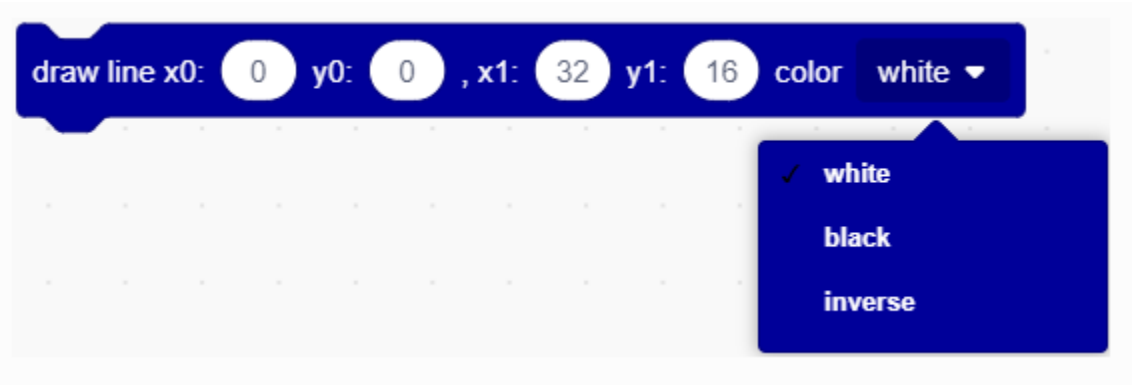
5. OLED Display



(1). Programming Steps

Step 1Description of the Building Block

This block is used to initialize the OLED's width, height and an I2C address.



This is a command block for drawing a straight line from the initial position x0:0 y0:0 to the final position x1:32 y1:16. The number in the block can be changed.

This is a command block that draws a recta with a width of 32 and a height of 16 from the initial position x:0 y:0, the numbers can be changed.

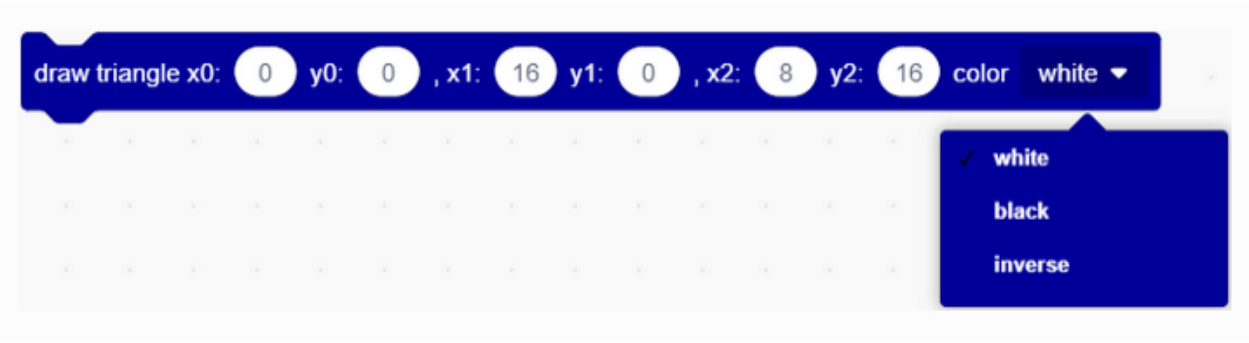
This is a command block that draws a rectangle with a width of 32 and a height of 16 from the initial position x:0 y:0, the numbers can be changed.

This is the command block that draws a circle with a radius of 8 starting from the initial position x:16 y:16.

This is the command block that fills a circle with a radius of 8 starting at an initial position of x:16 y:16.

This is the command block that draws a round rectangle with width 32, height 16, and radius 4 starting from an initial position of x:16 y:16.

This is the command block that fills a round rectangle with width 32, height 16 and radius 4 starting from initial position x:16 y:16.



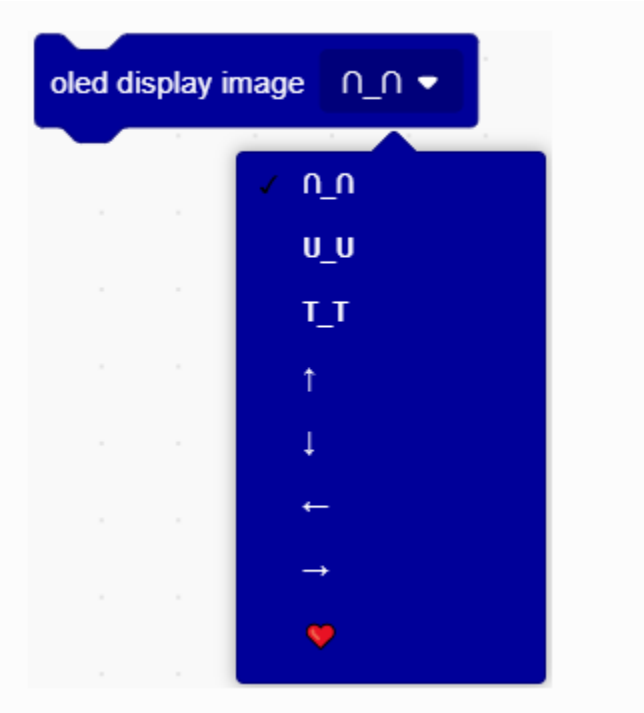
This is the command block to draw a triangle from three positions x0:0 y0:0, x1:16 y1:0 and x2:8 y2:16.

This is the command block that fills the triangle between the three positions x0:0 y0:0, x1:16 y1:0 and x2:8 y2:16.

This is a command block for setting text size and color and background color.

This is the command block that sets the cursor position.

This is a command block for setting the way of printing strings on the OLED screen. "**warp"** means newline printing, "**no-warp**" means no newline printing.



This is the command block to set the OLED display pattern.

This is the command block to clear the OLED screen.

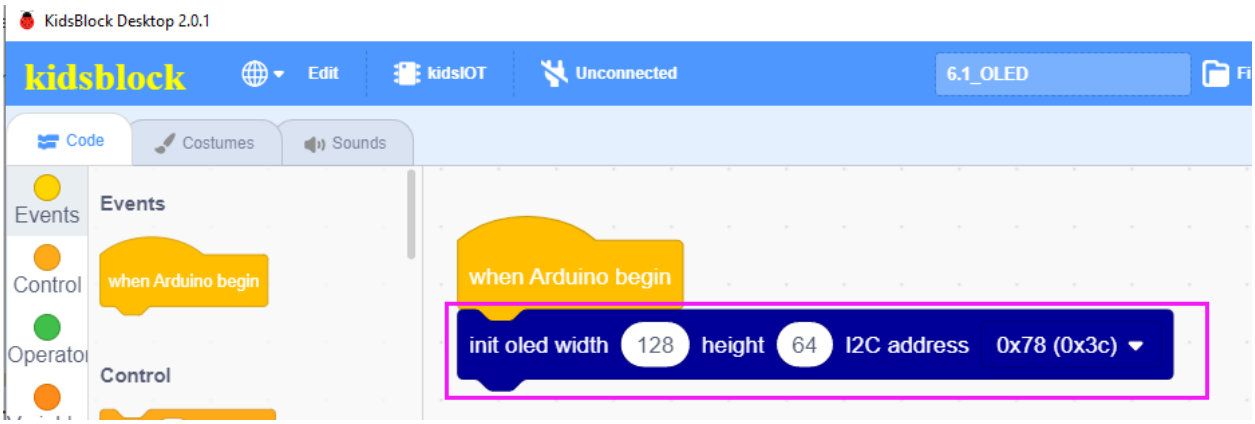
This is the command block to refresh the OLED screen and display the next content.

This is a command block that sets strings to start scrolling in a certain direction.

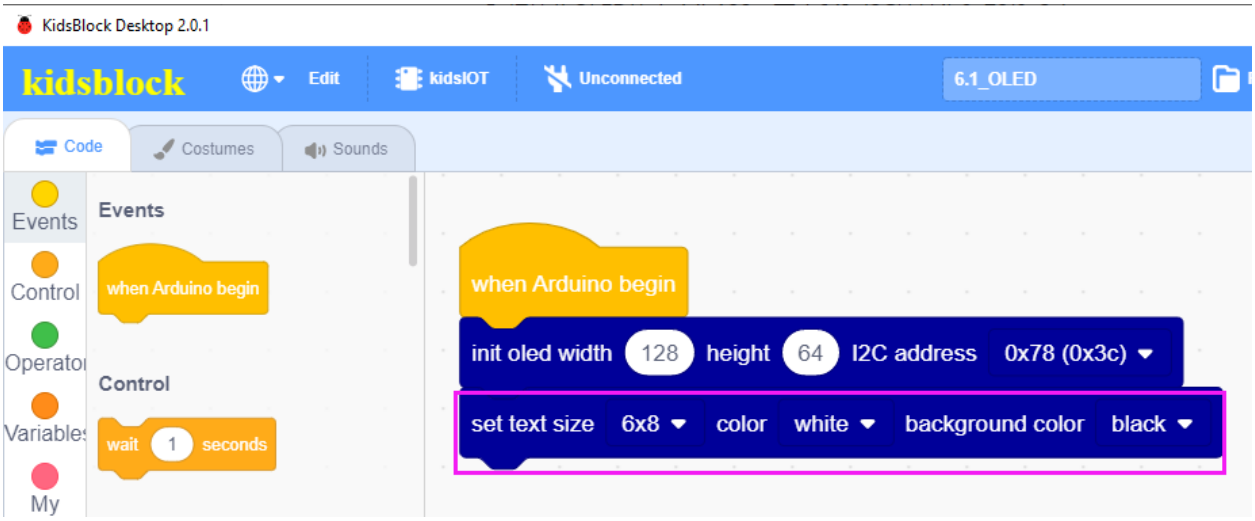
This is the command block to set stop scrolling.

Step 2Write the Program

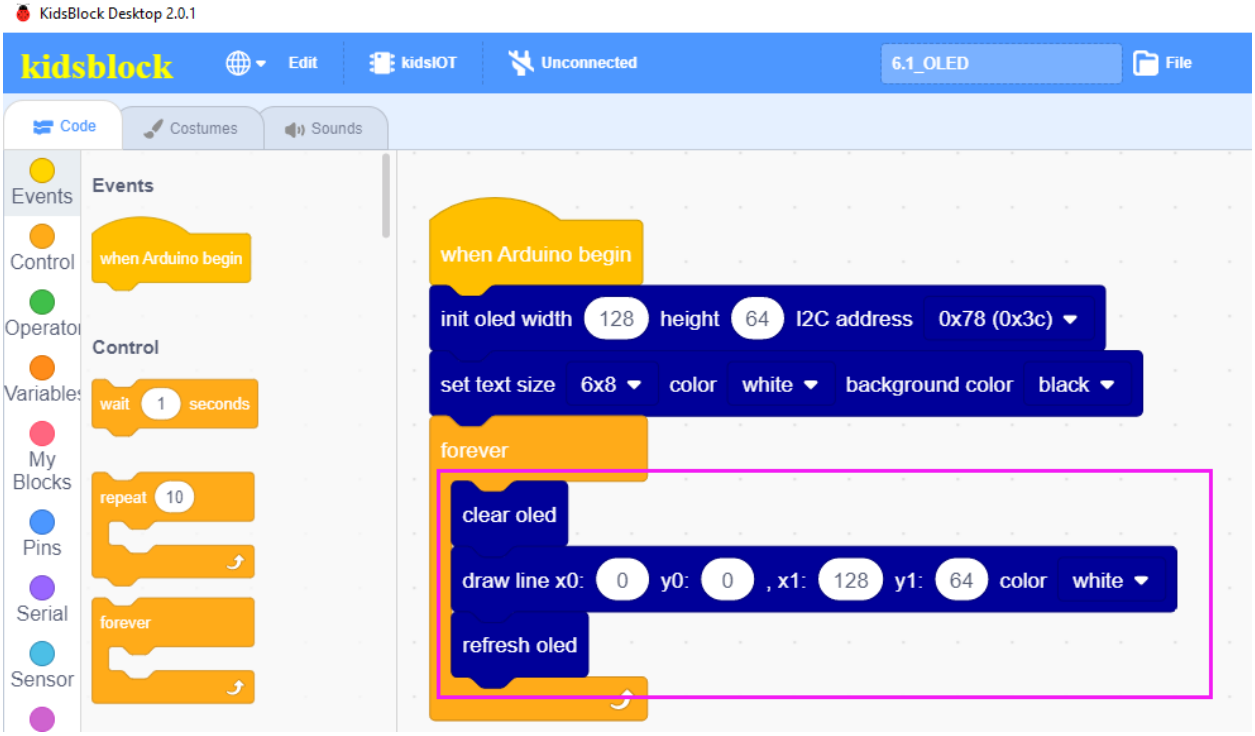
Initialize the OLED with width 128, height 64 and I2C address 0x78 (0x3c).



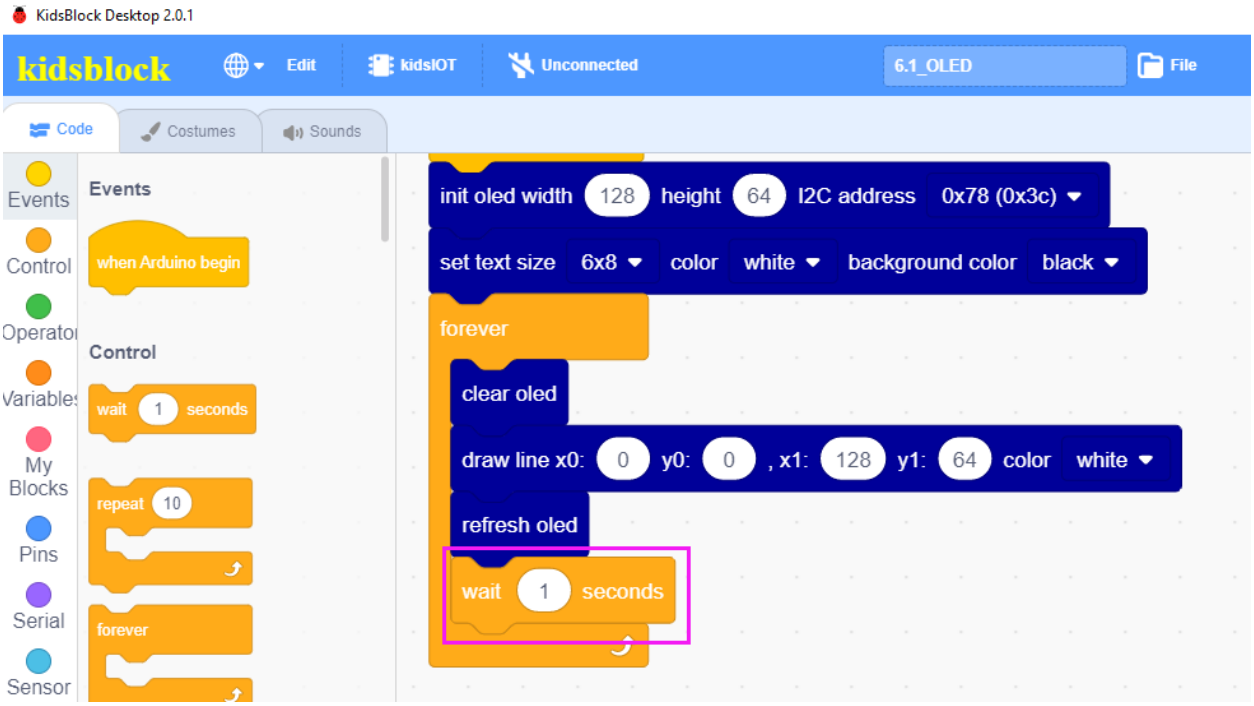
Set the text size displayed on the OLED to 6x8, the text color to white and the background color to black.



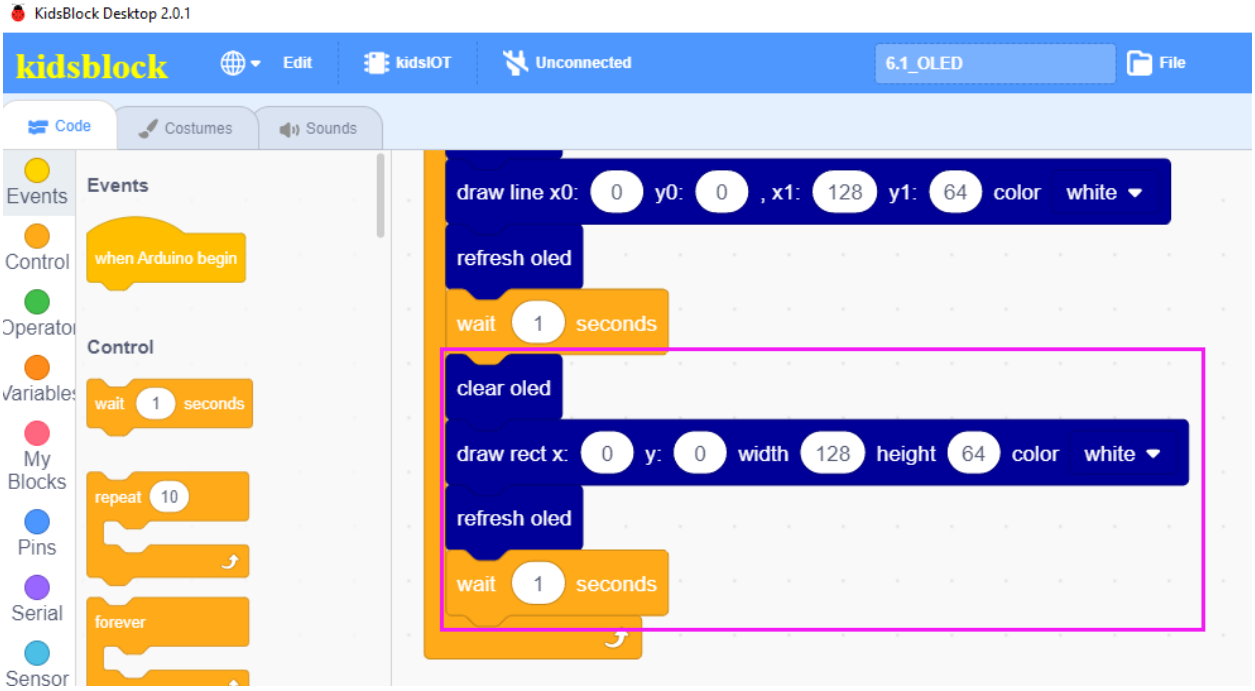
OLED displays straight lines.



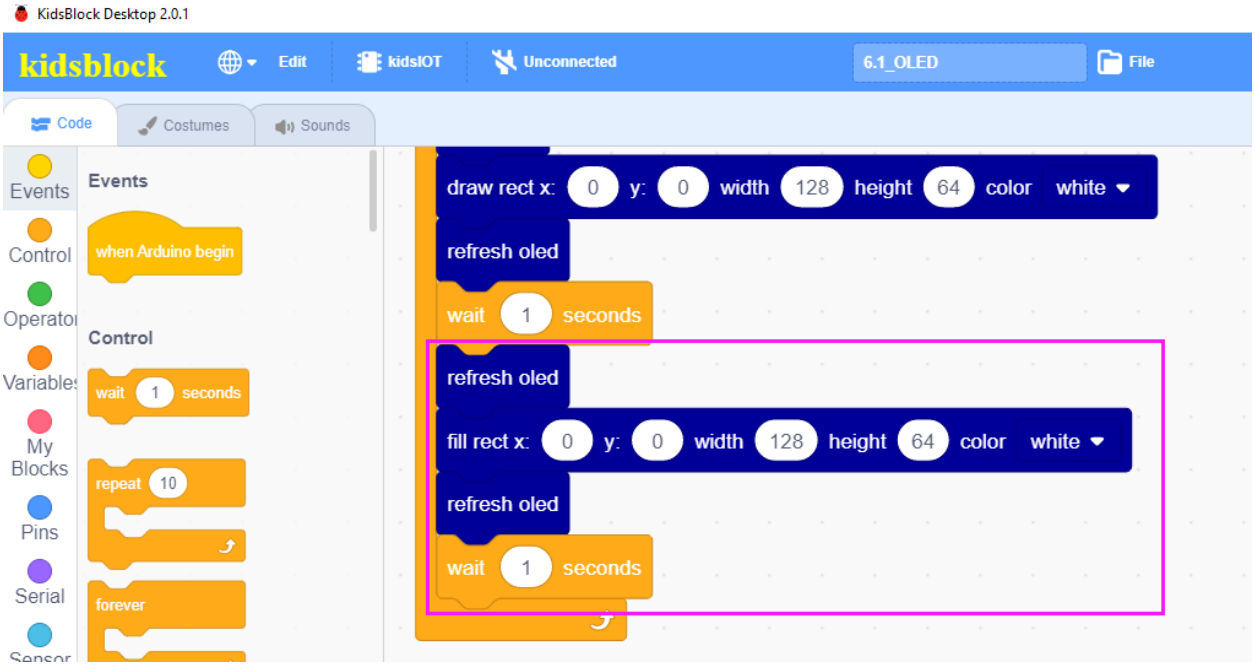
The OLED displays the straight line and delays 1 second.



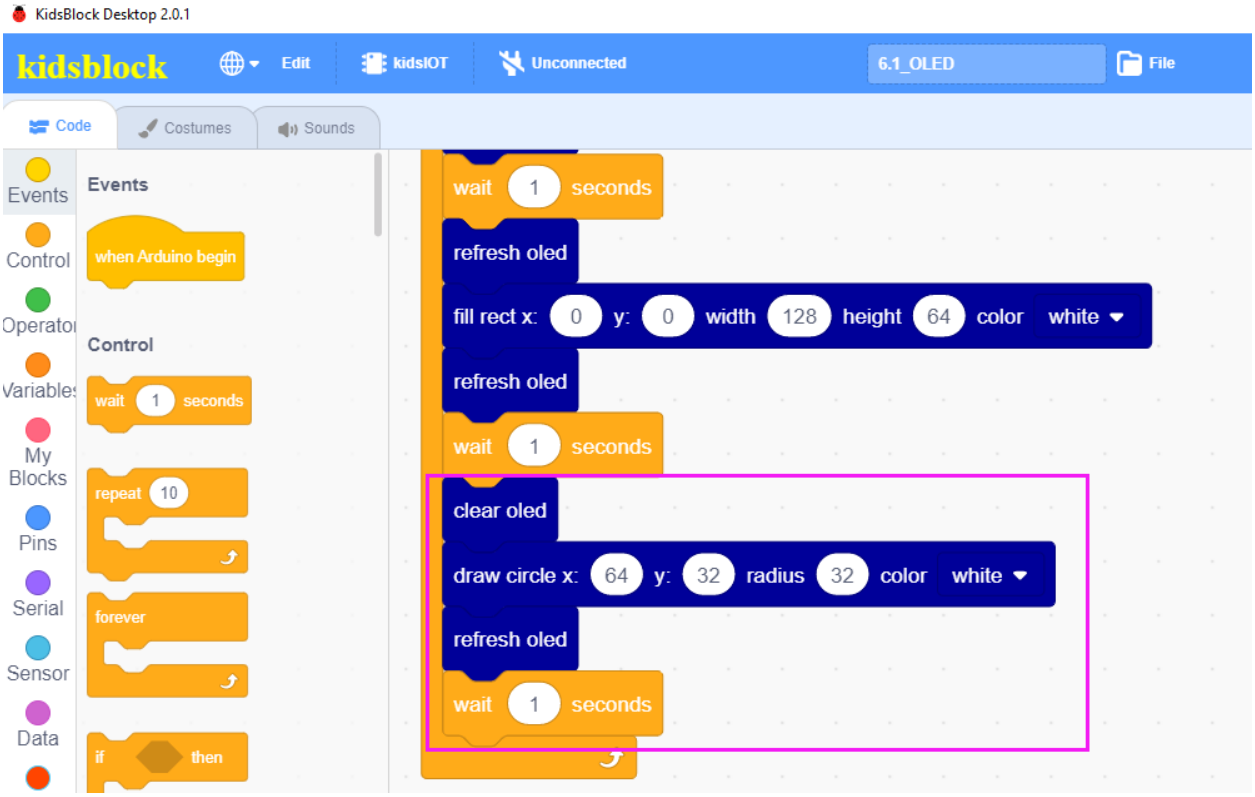
The OLED displays rectangle and delays 1 second.



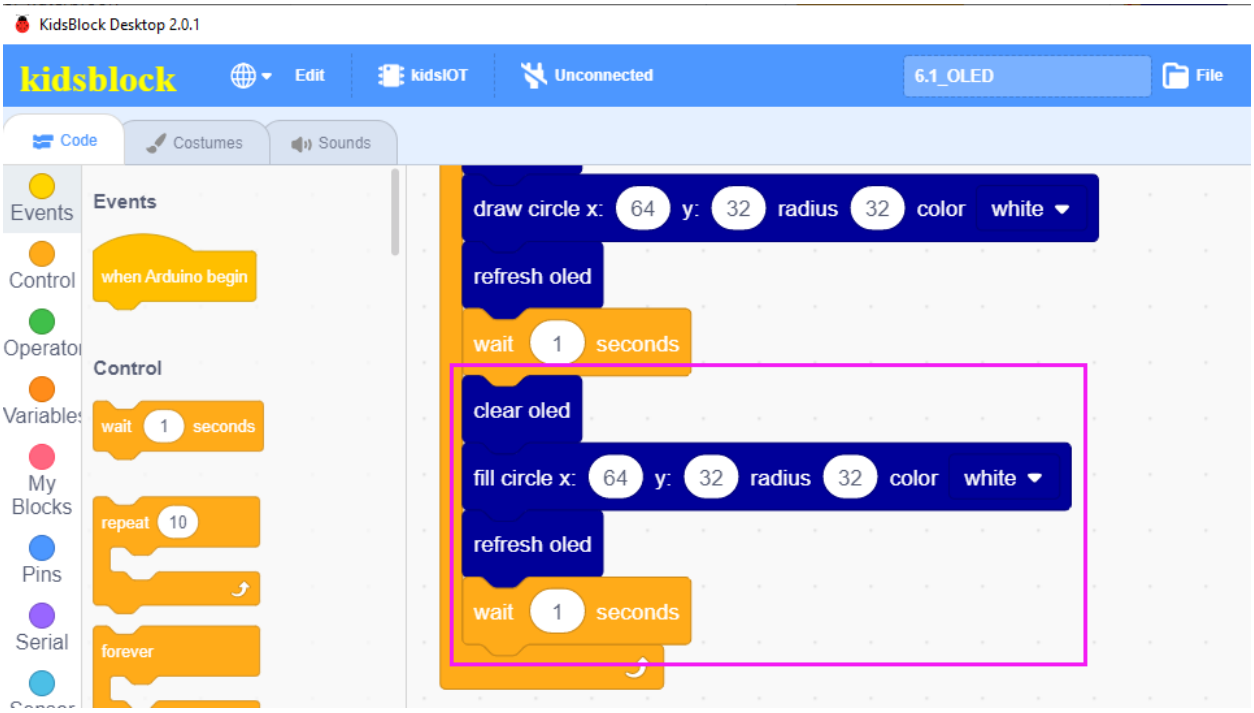
The OLED displays fill rectangle and delays 1 second.



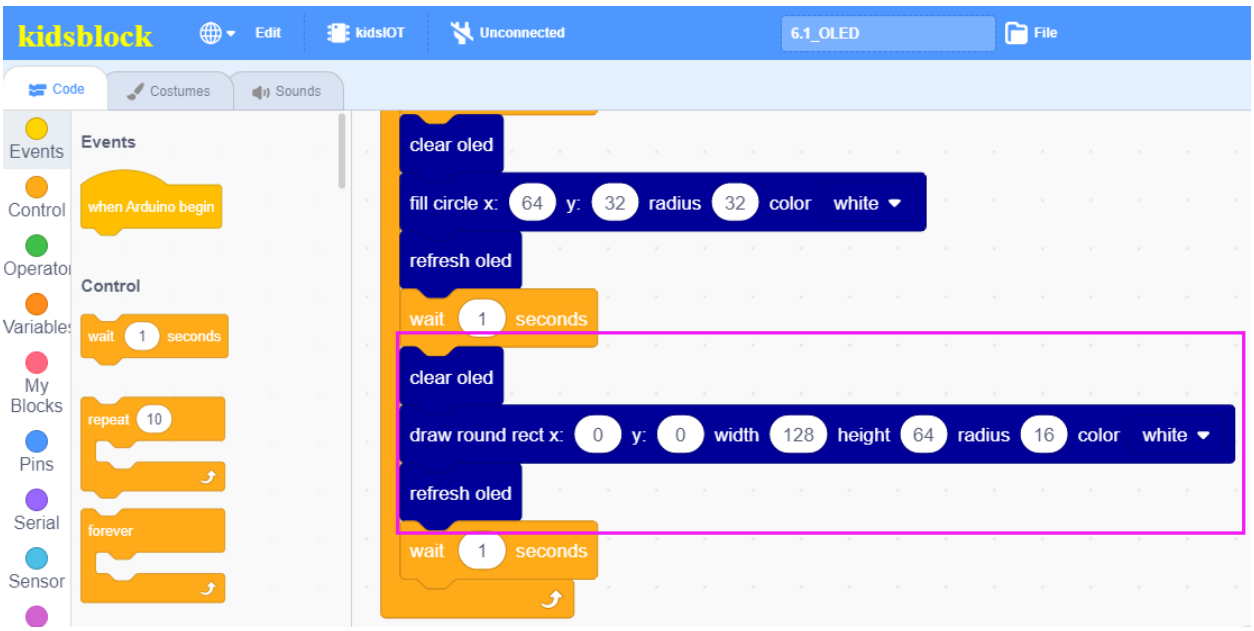
The OLED displays circle and delays 1 second.



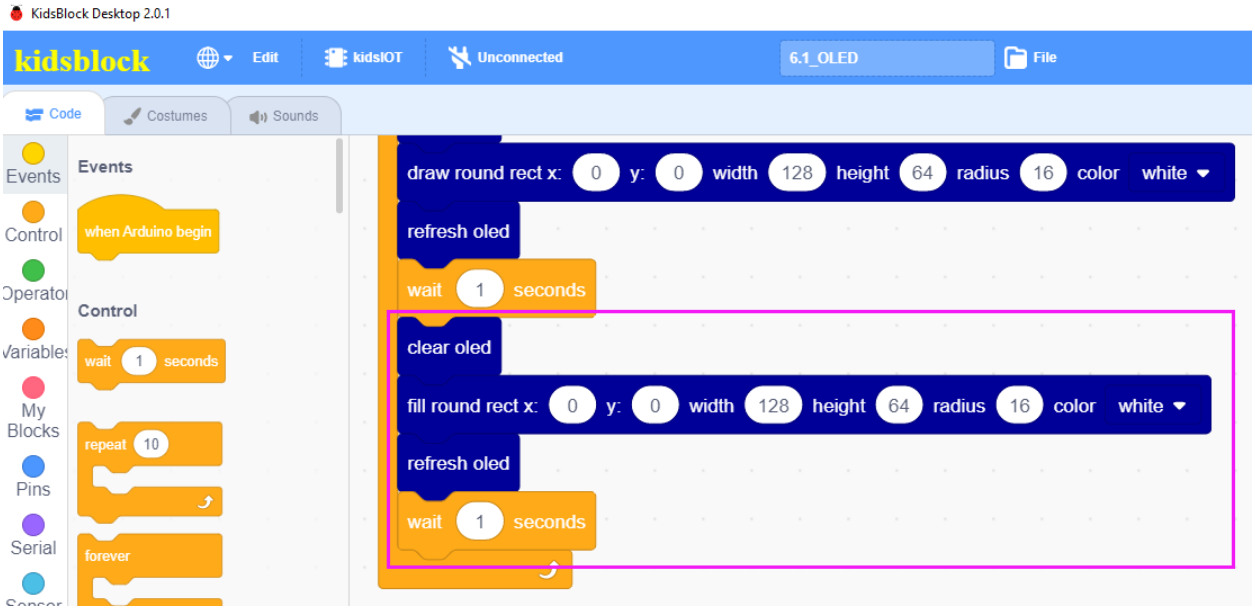
The OLED displays fill circle and delays 1 second.



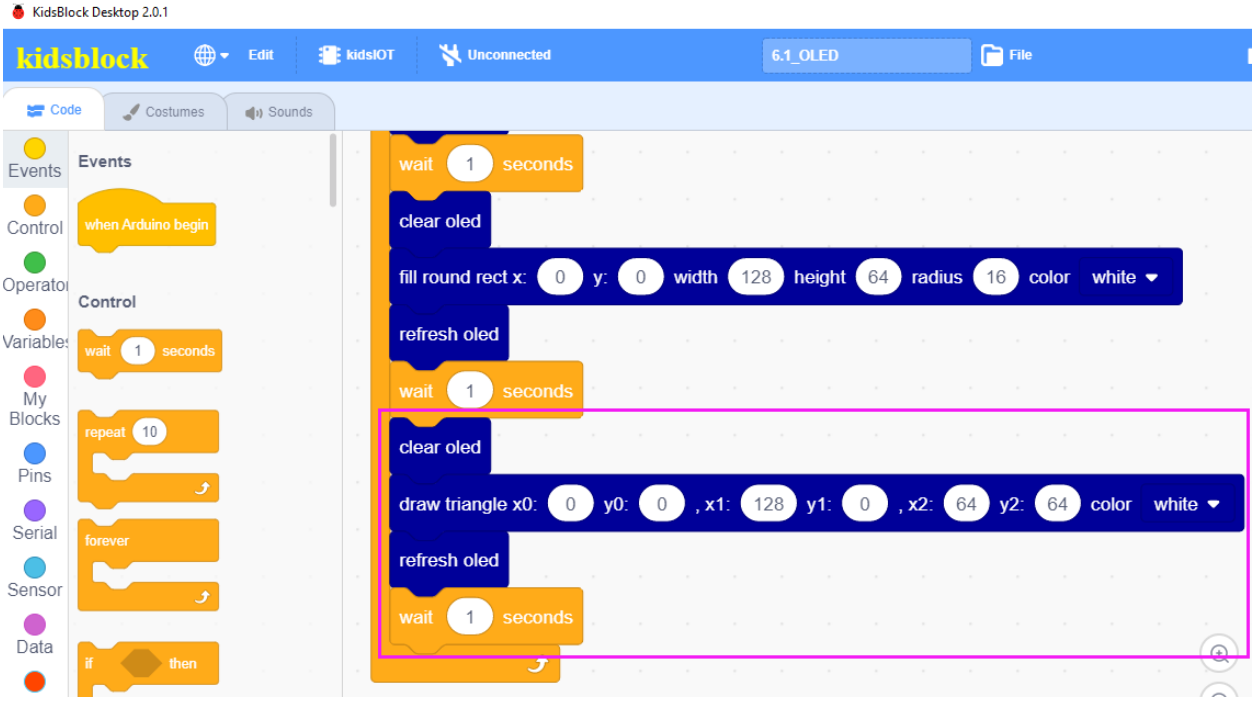
The OLED displays round rectangle and delays 1 second.



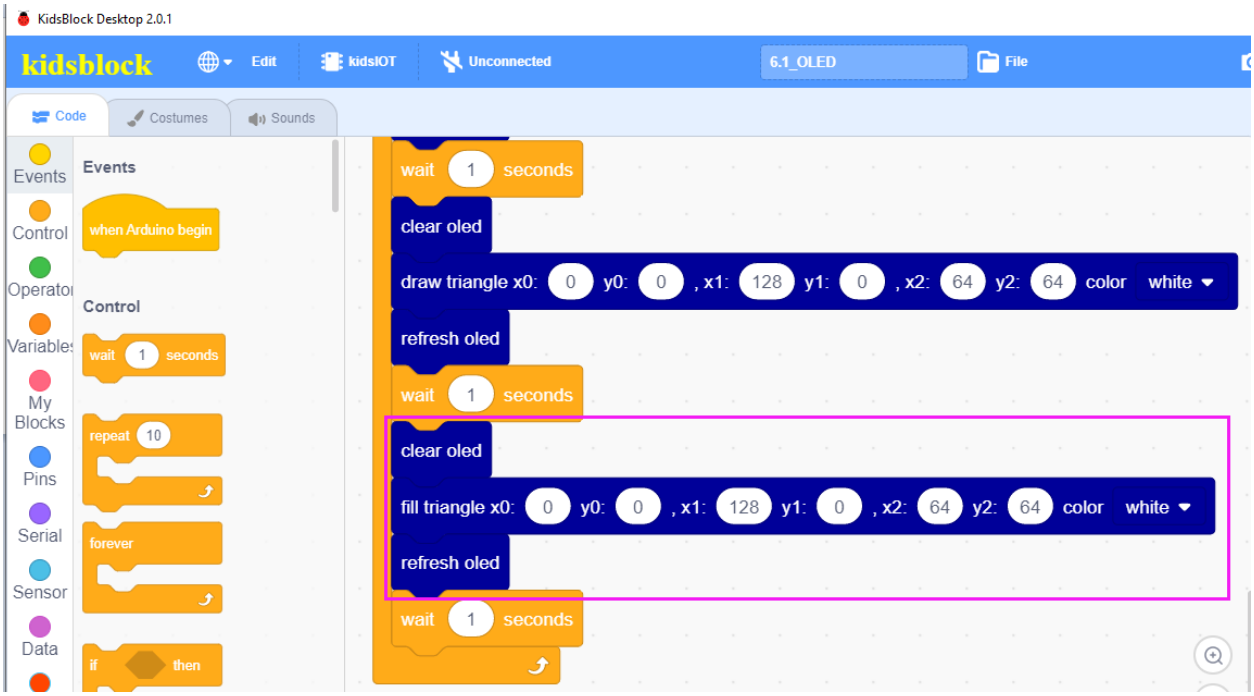
The OLED displays fill round rectangle and delays 1 second.



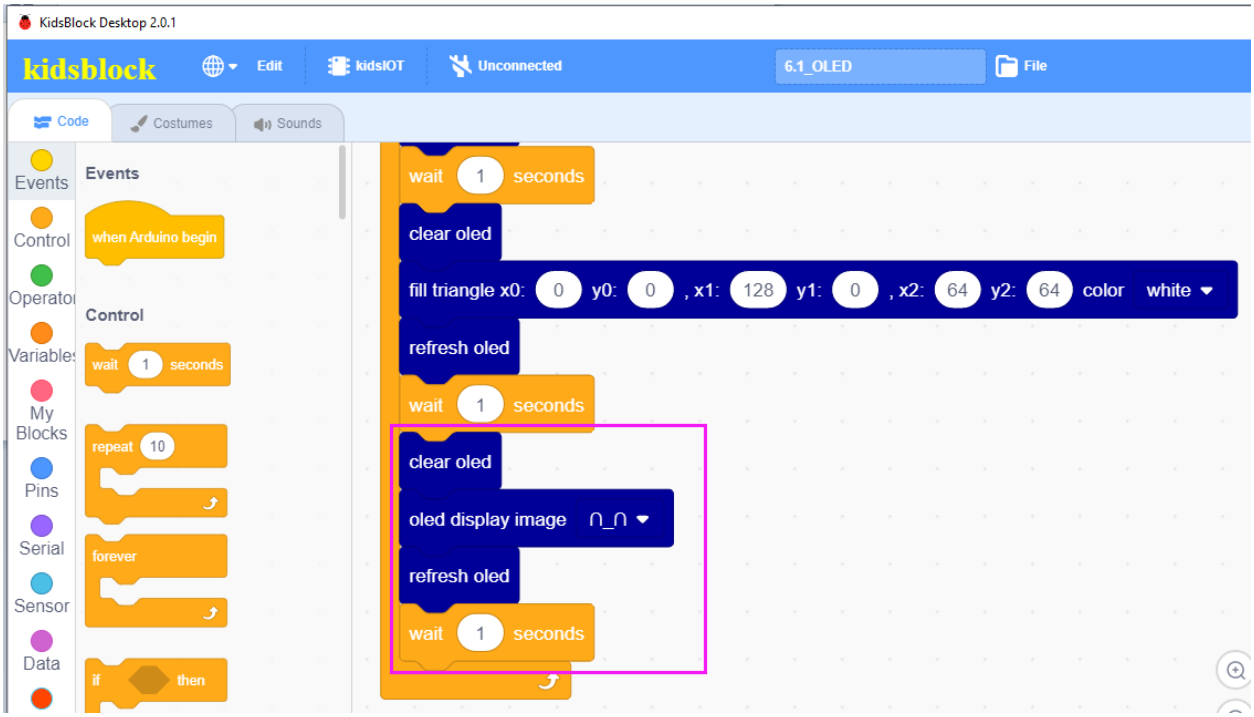
The OLED displays triangle and delays 1 second.



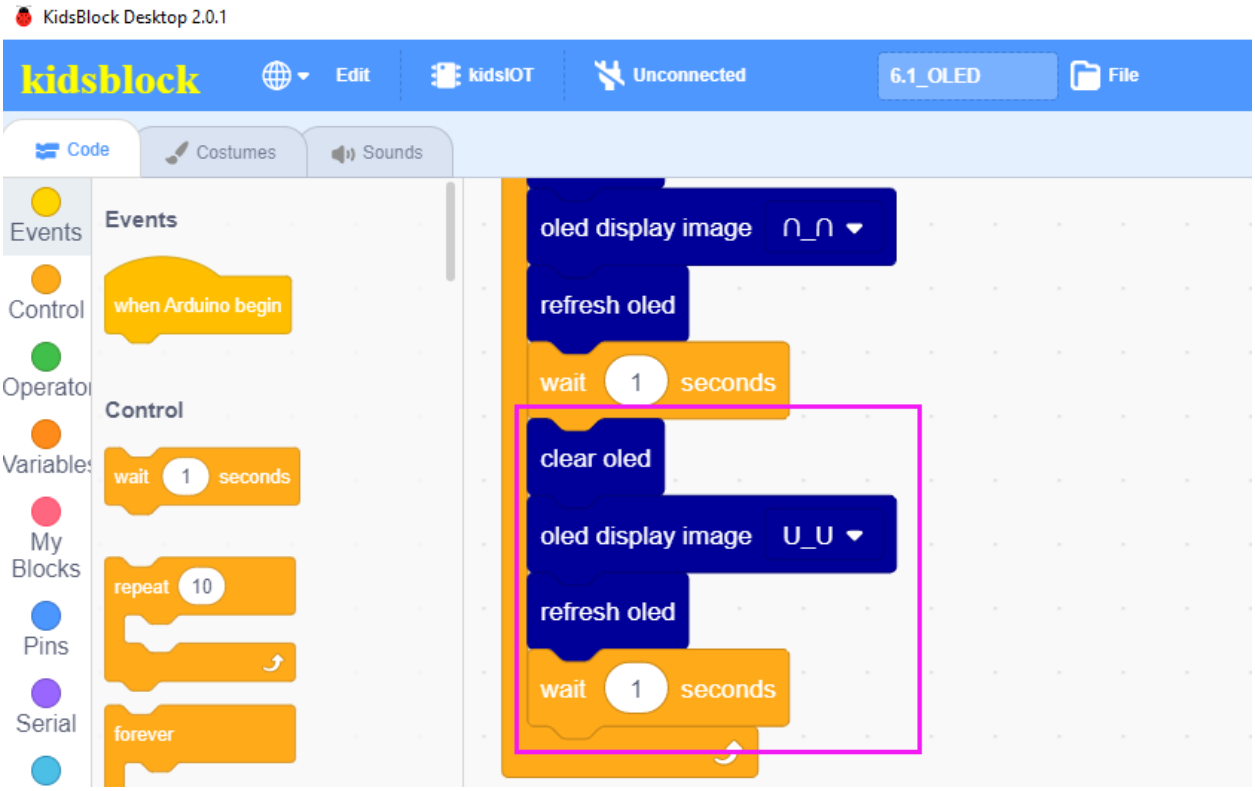
The OLED displays fill triangle and delays 1 second.



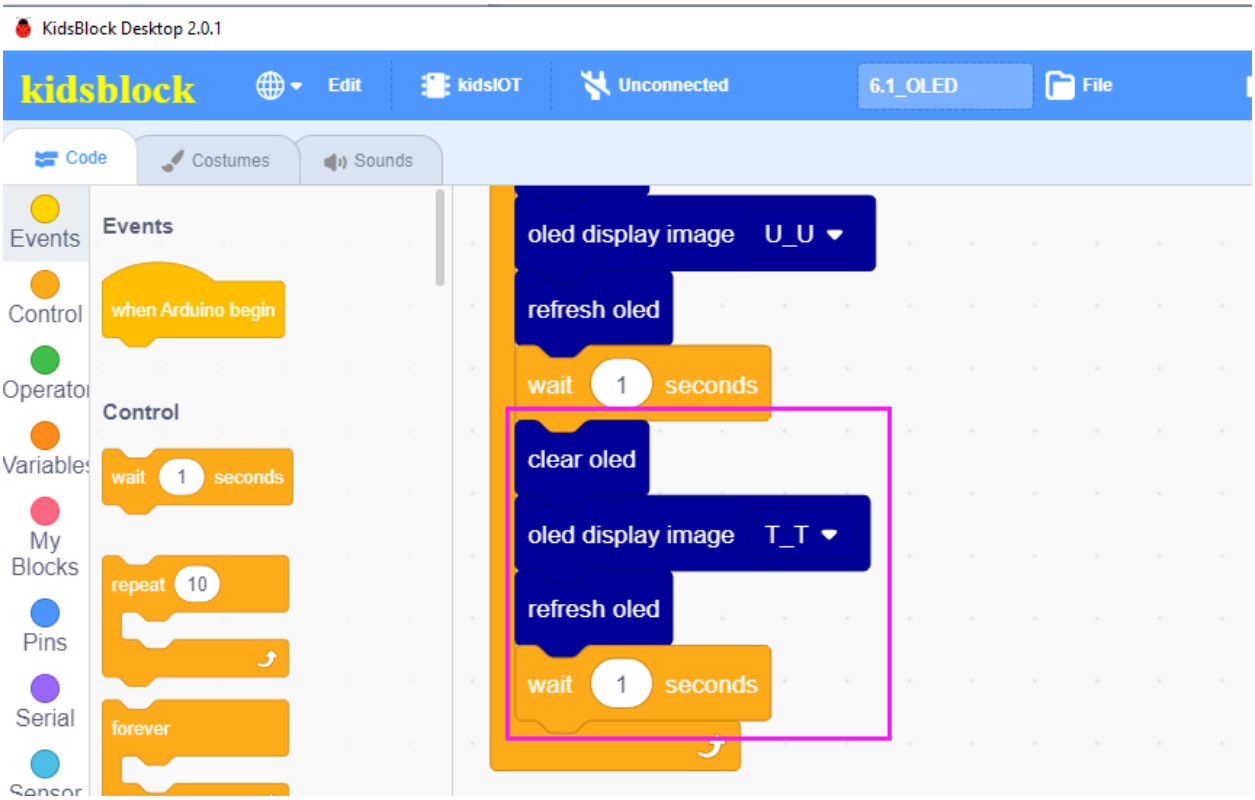
The OLED displays smile face and delays 1 second.



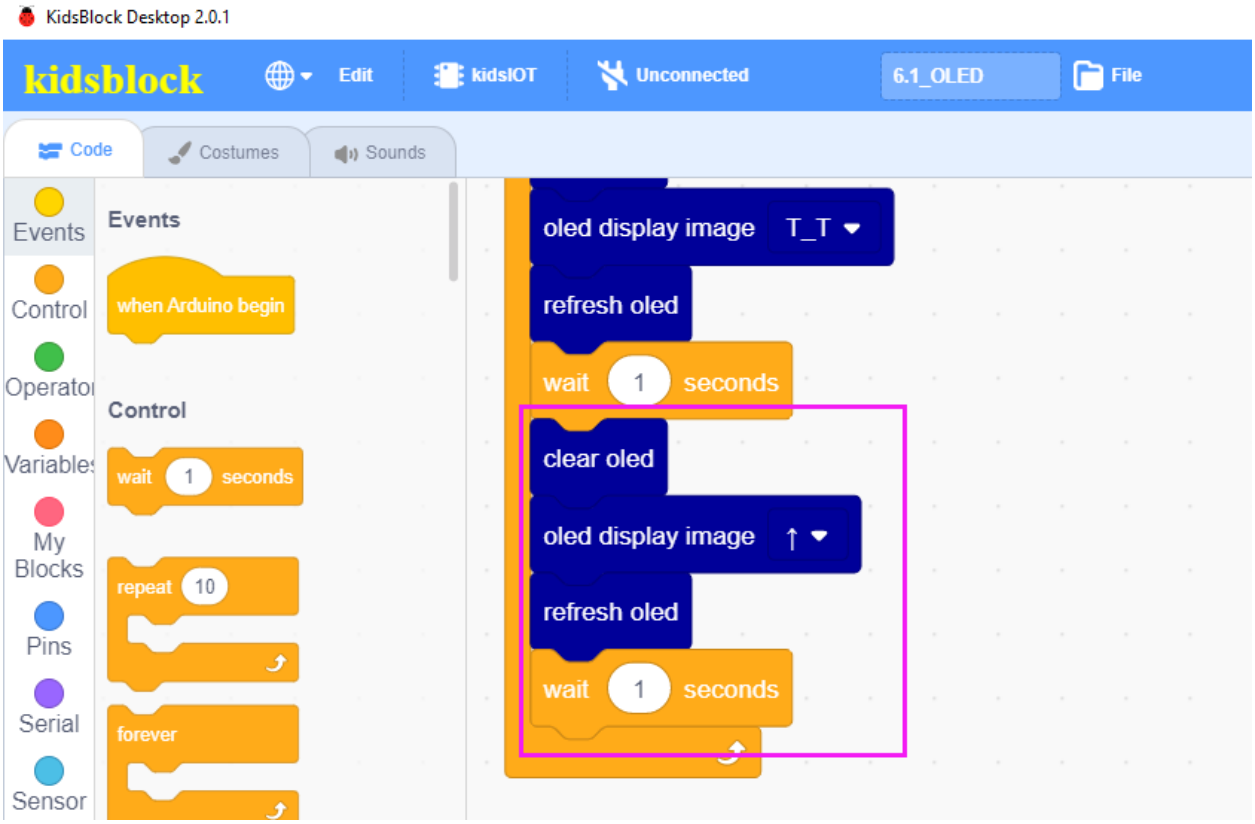
The OLED displays angry face and delays 1 second.



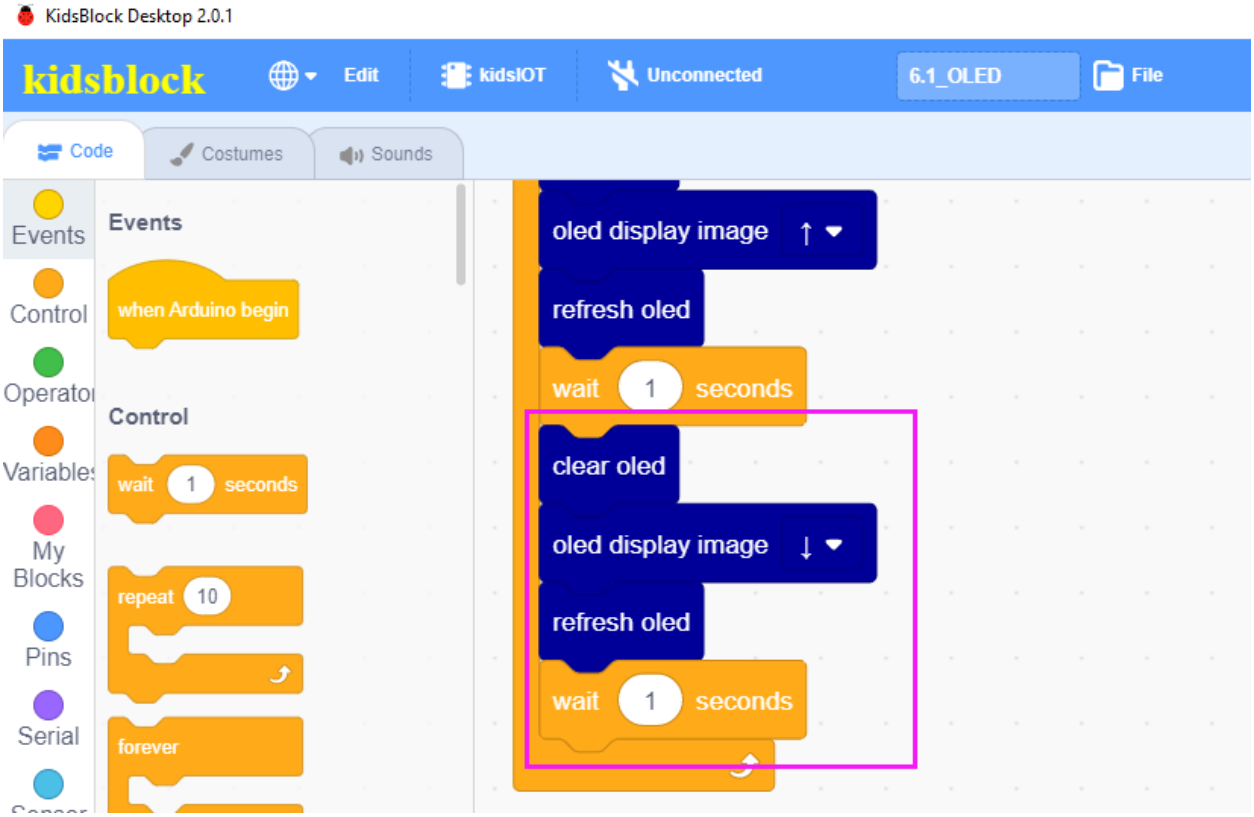
The OLED displays cry face and delays 1 second.



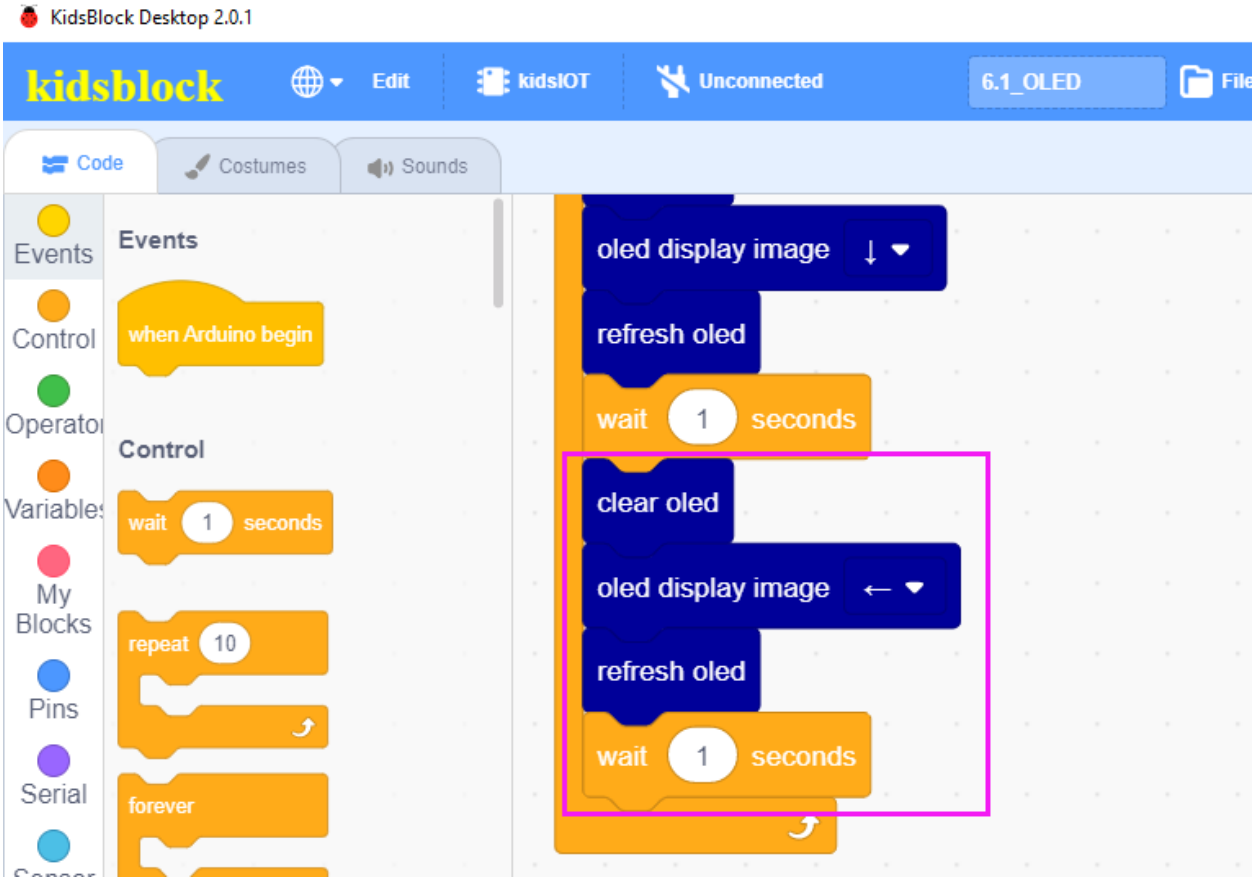
The OLED displays "↑" and delays 1 second.



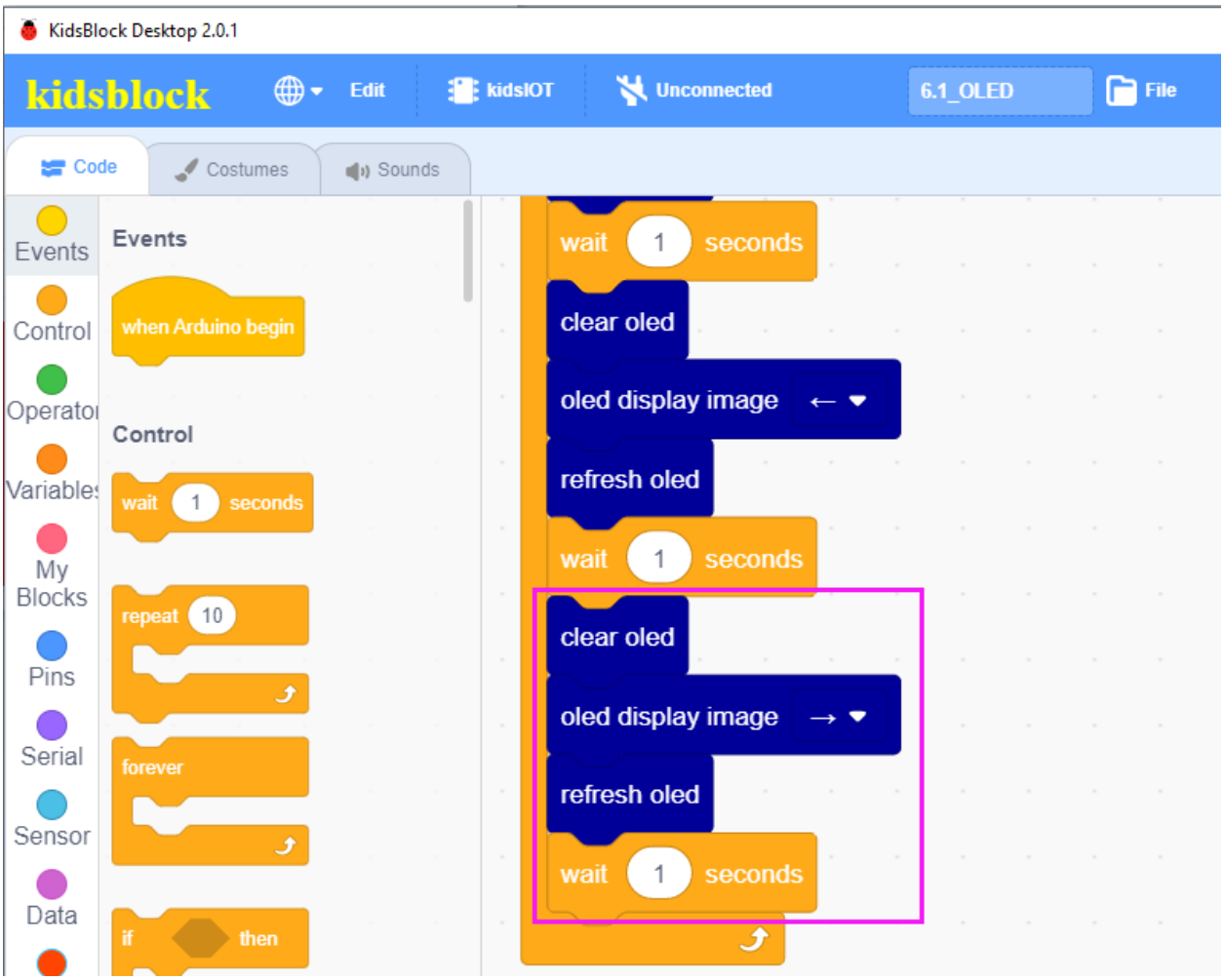
The OLED displays "↓" and delays 1 second.



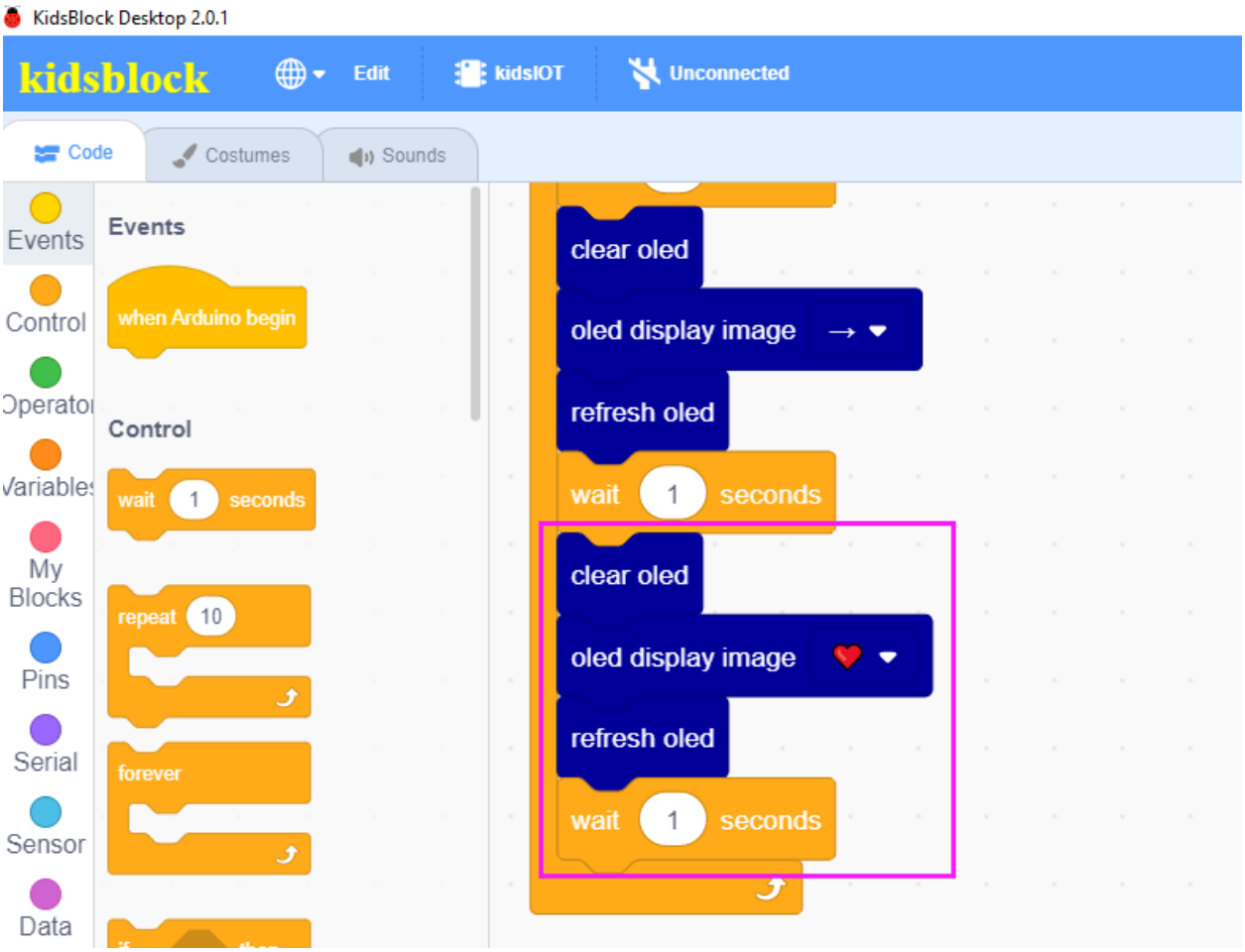
The OLED displays "←" and delays 1 second.



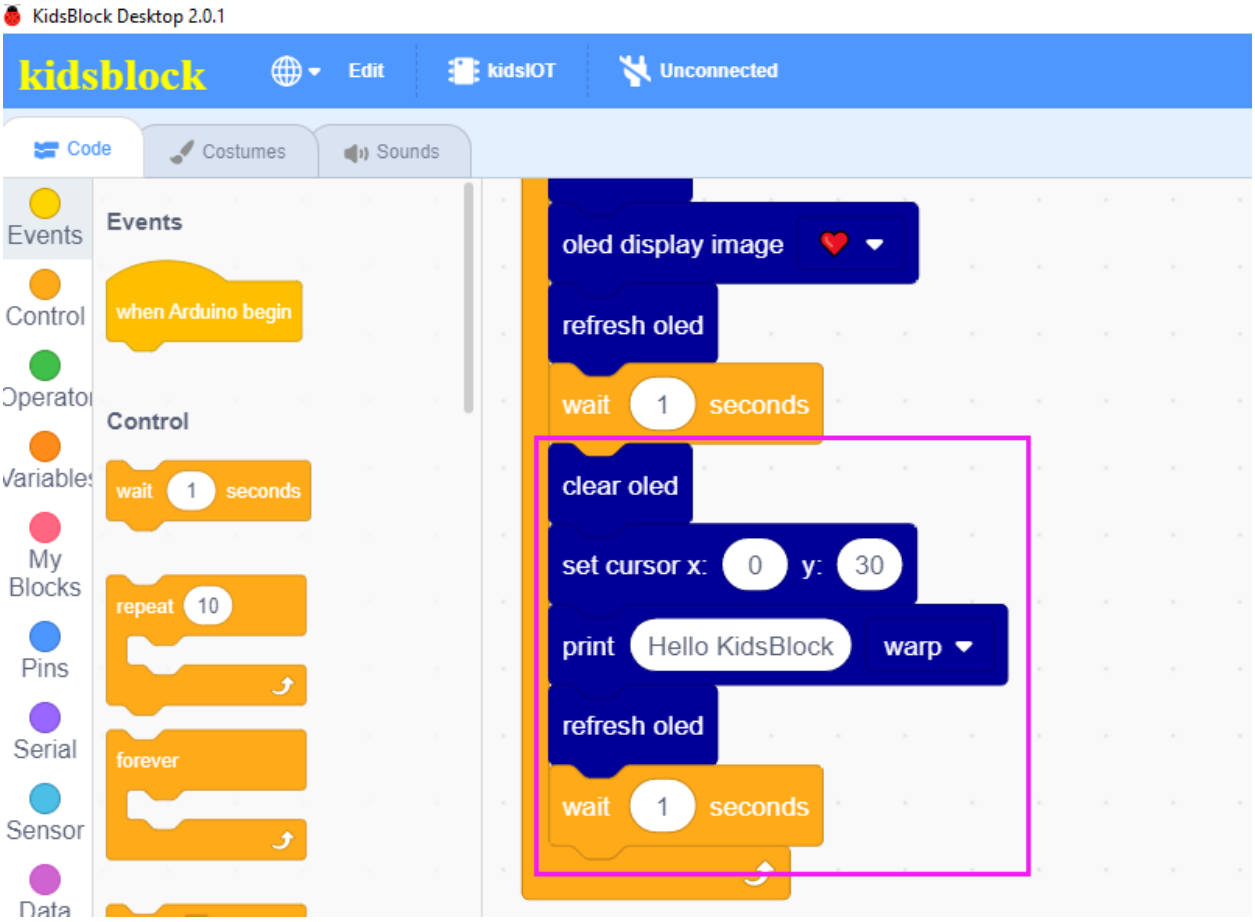
The OLED displays "→" and delays 1 second.



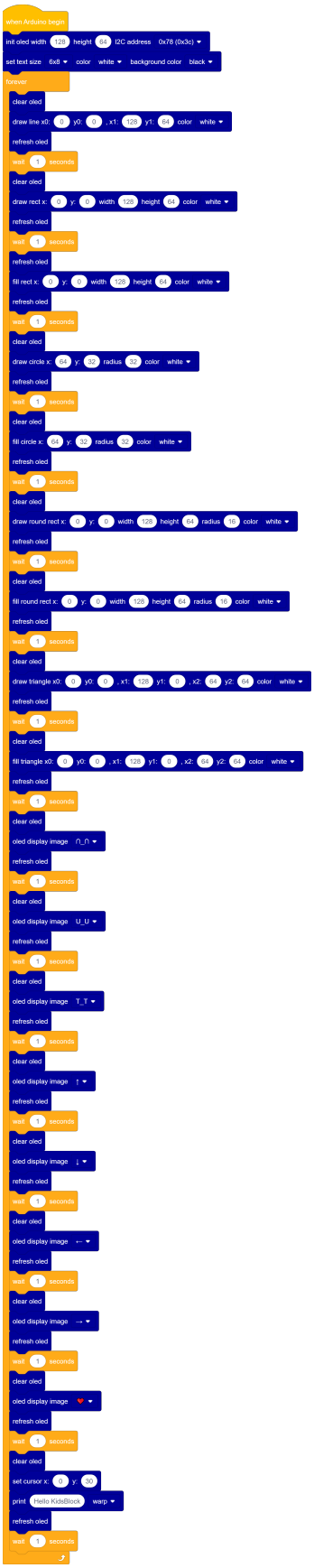
The OLED displays “” and delays 1 second.



Set the cursor position of the OLED to display the "Hello, KidsBlock" string at x:0 y:30 with a delay of 1 second.



Complete Program

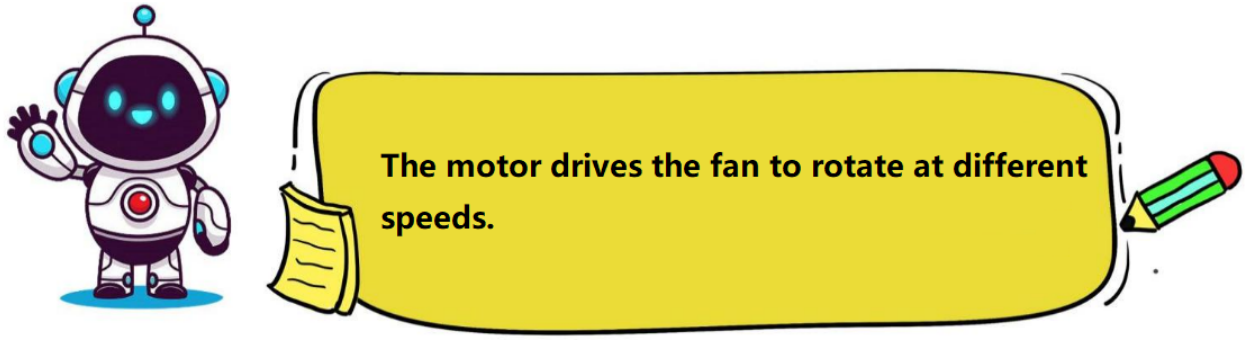

when Arduino begin
init oled width 128 height 64 I2C address 0x78 (0x3c)
set text size 6x8 color white background color black
forever
clear oled
draw line x0: 0 y0: 0 x1: 128 y1: 64 color white
refresh oled
wait 1 seconds
clear oled
draw rect x: 0 y: 0 width 128 height 64 color white
refresh oled
wait 1 seconds
refresh oled
fill rect x: 0 y: 0 width 128 height 64 color white
refresh oled
wait 1 seconds
clear oled
draw circle x: 64 y: 32 radius 32 color white
refresh oled
wait 1 seconds
clear oled
fill circle x: 64 y: 32 radius 32 color white
refresh oled
wait 1 seconds
clear oled
draw round rect x: 0 y: 0 width 128 height 64 radius 16 color white
refresh oled
wait 1 seconds
clear oled
fill round rect x: 0 y: 0 width 128 height 64 radius 16 color white
refresh oled
wait 1 seconds
clear oled
draw triangle x0: 0 y0: 0 x1: 128 y1: 0 x2: 64 y2: 64 color white
refresh oled
wait 1 seconds
clear oled
fill triangle x0: 0 y0: 0 x1: 128 y1: 0 x2: 64 y2: 64 color white
refresh oled
wait 1 seconds
clear oled
oled display image ∩_∩
refresh oled
wait 1 seconds
clear oled
oled display image U_U
refresh oled
wait 1 seconds
clear oled
oled display image T_T
refresh oled
wait 1 seconds
clear oled
oled display image ↑
refresh oled
wait 1 seconds
clear oled
oled display image ↓
refresh oled
wait 1 seconds
clear oled
oled display image ←
refresh oled
wait 1 seconds
clear oled
oled display image →
refresh oled
wait 1 seconds
clear oled
oled display image ♥
refresh oled
wait 1 seconds
clear oled
set cursor x: 0 y: 30
print Hello KidsBlock warp
refresh oled
wait 1 seconds

(2). Test Result

Click 

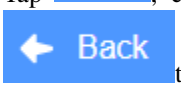
to upload the above complete code to the kidsIOT motherboard. After powering up via the external power supply, the OLED display on the kidsIOT board displays various patterns and English letters.

6. Fan rotates

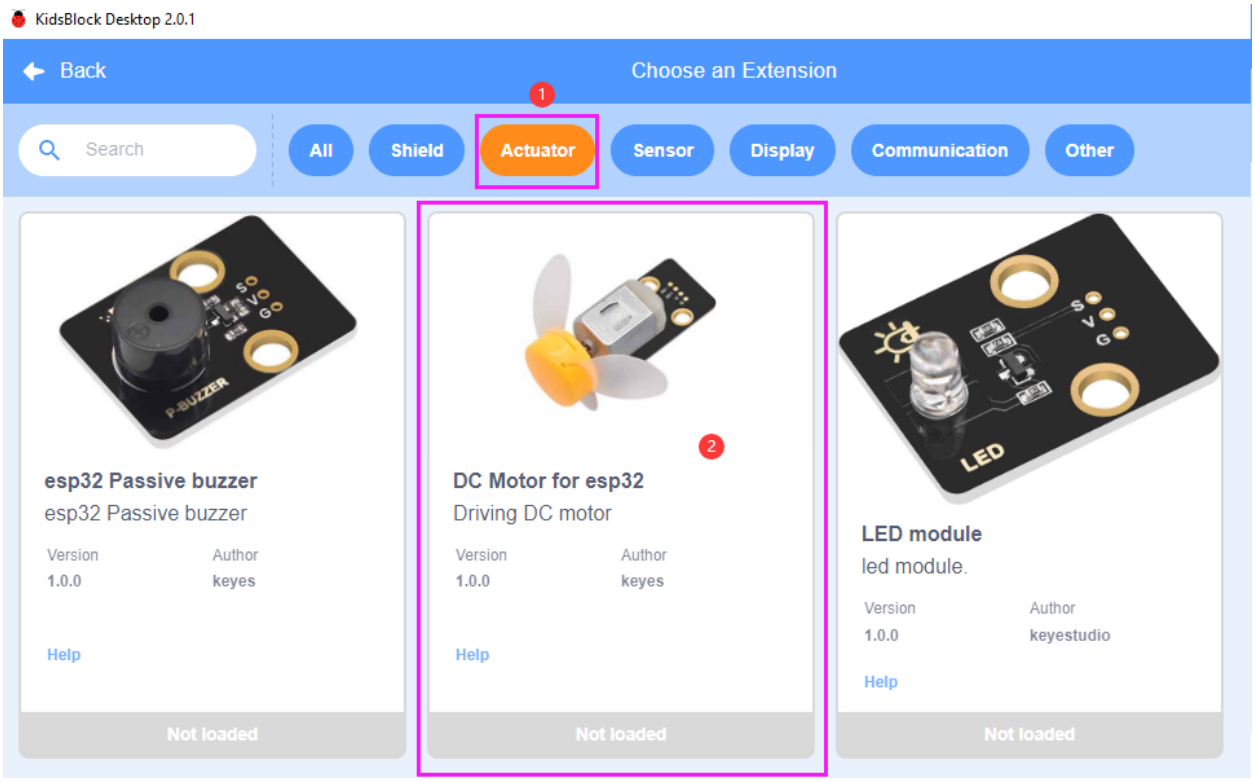


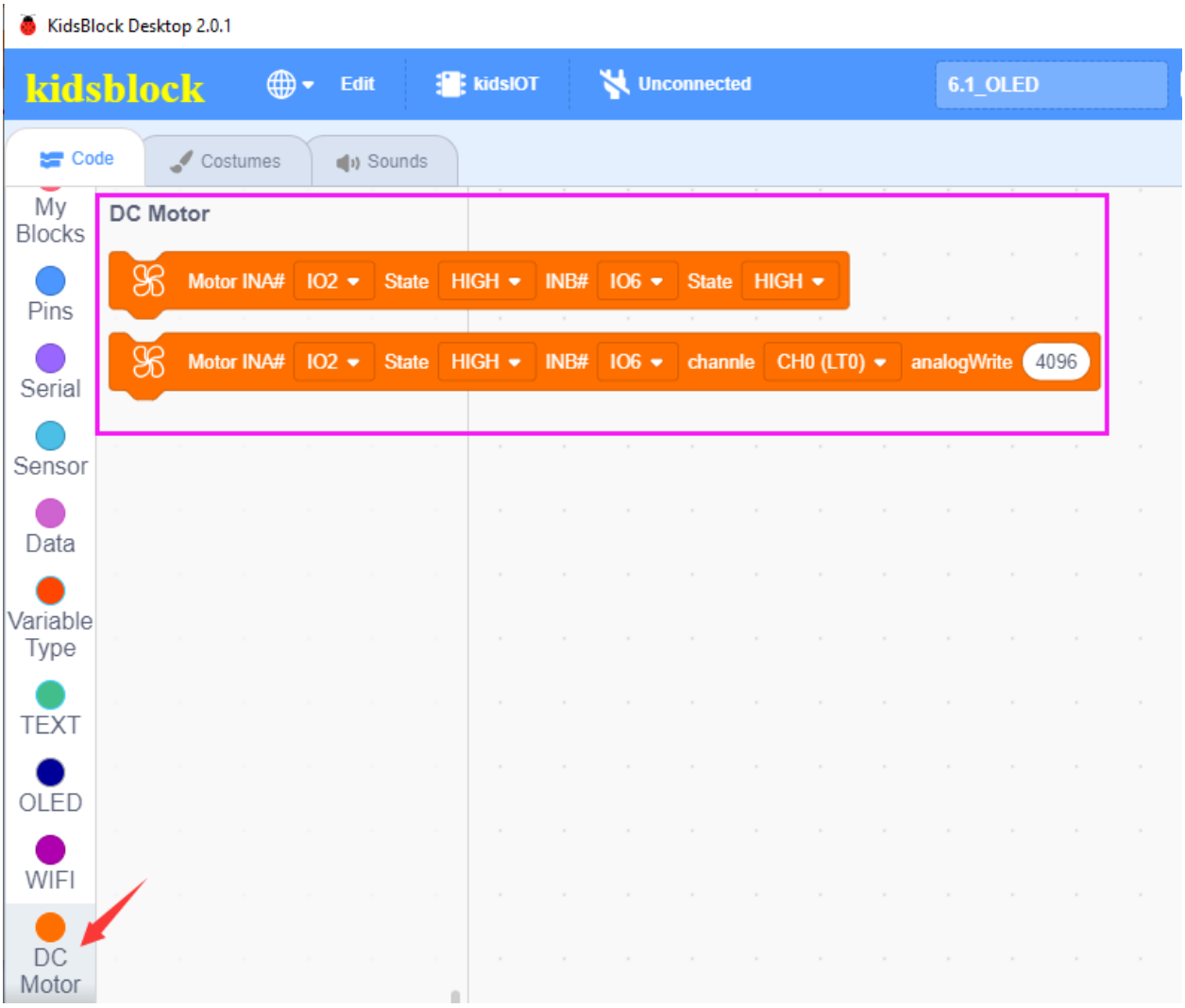
(1). Programming Steps

Step 1Add "DC Motor"

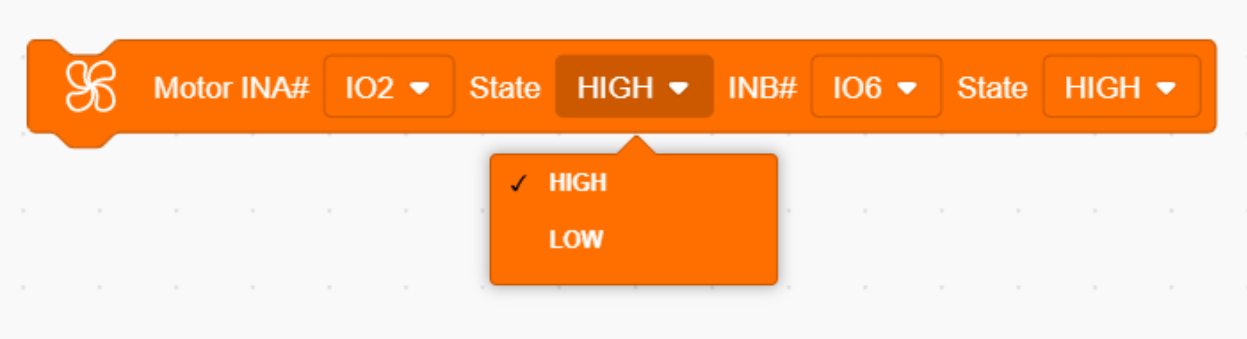
Tap  , click the "Actuator" module in the "Extension" , then select "**DC Motor for esp32**" and click 

to return to the programming interface.

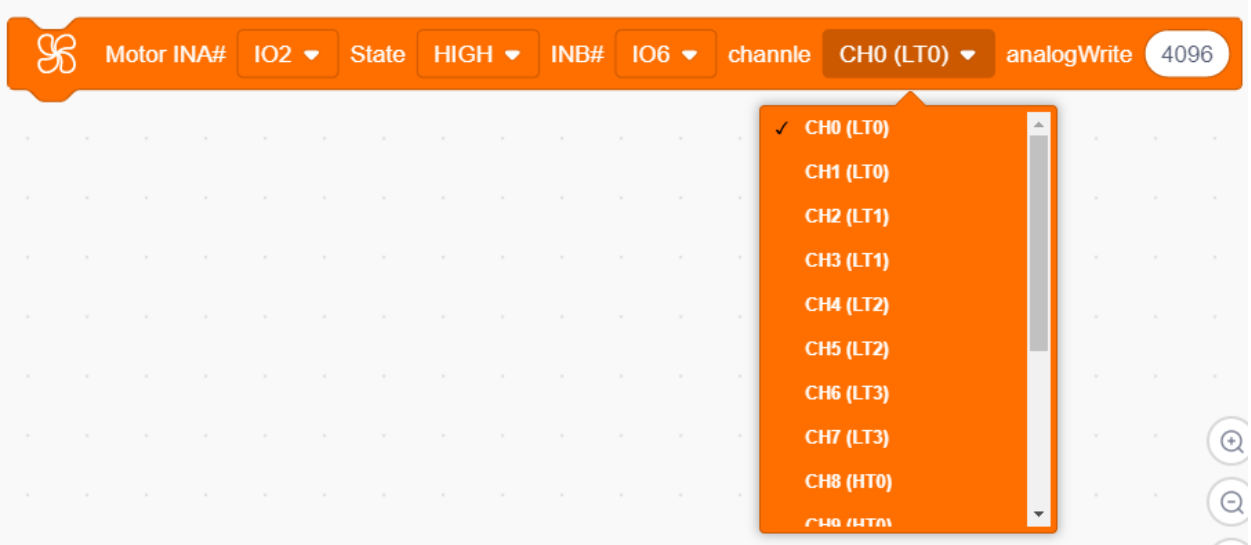

KidsBlock Desktop 2.0.1
Back
Choose an Extension
1
Search
All
Shield
Actuator
Sensor
Display
Communication
Other
esp32 Passive buzzer
esp32 Passive buzzer
Version
1.0.0
Author
keyes
Help
Not loaded
2
DC Motor for esp32
Driving DC motor
Version
1.0.0
Author
keyes
Help
Not loaded
LED module
led module.
Version
1.0.0
Author
keyestudio
Help
Not loaded



Step 2Description of the Building Block



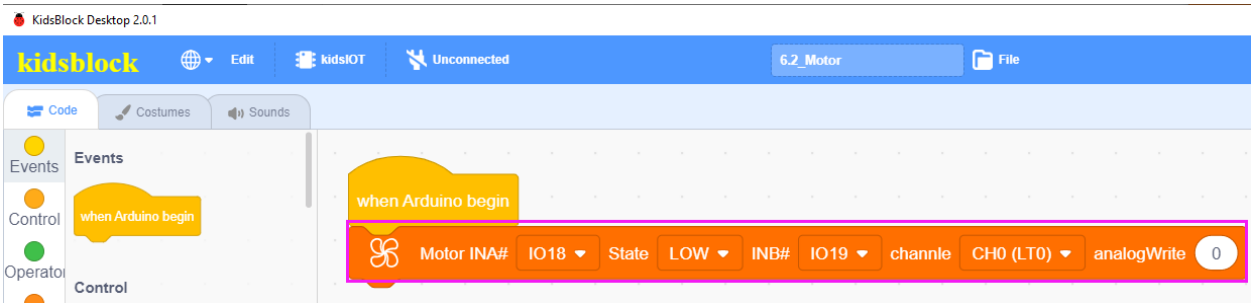
Set the high and low level states of the motor INA pin and INB pin.



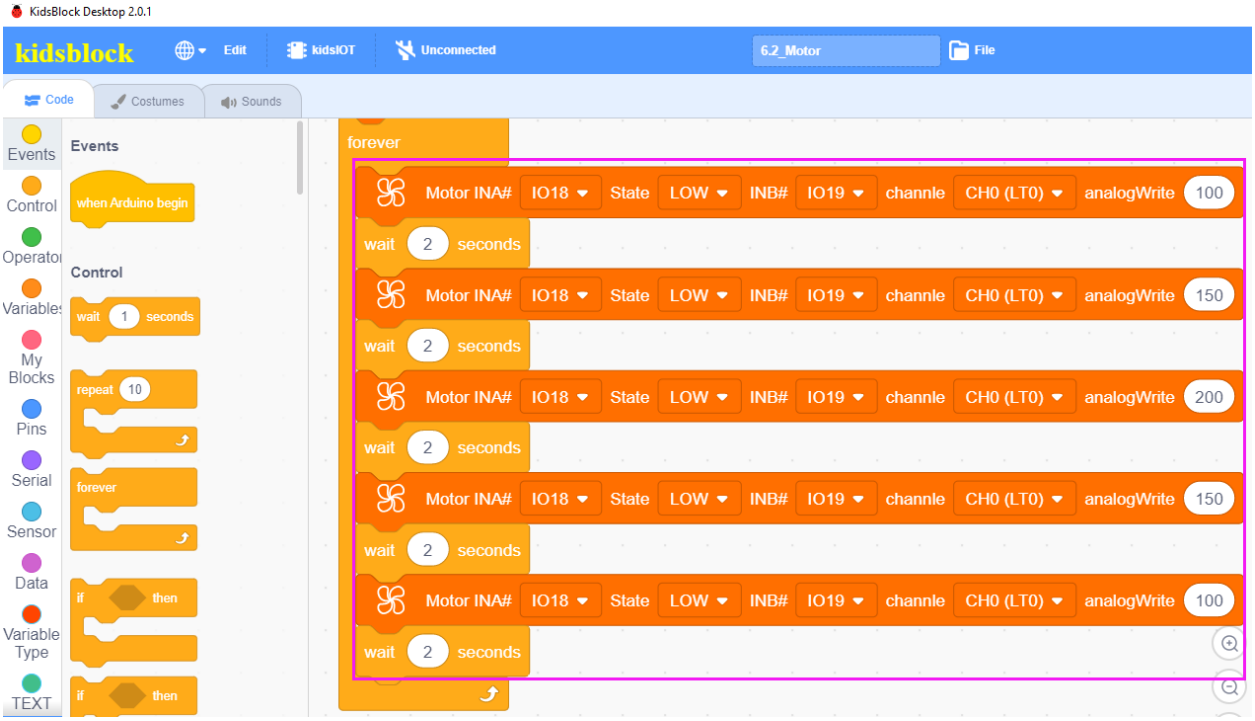
Set the high and low level status of the motor INA pin and the analog output value of the INB pin in certain channels. If the INA pin is in a high-level state, the smaller the INB analog output value, the faster the fan rotates; and if the INA pin is in a low-level state, the larger the INB analog output value, the faster the fan rotates.

Step 3Write the Program

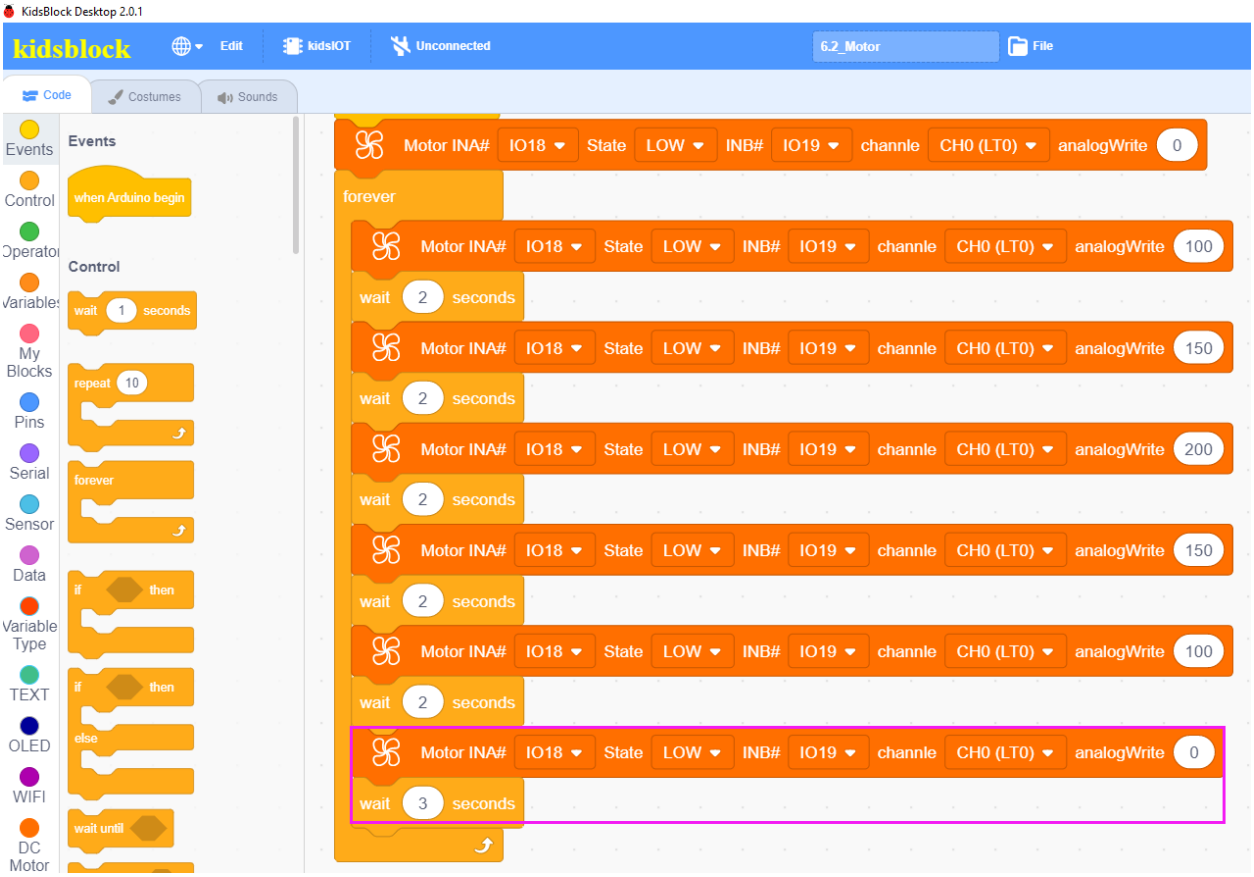
The pin INA of the motor module is IO18, the level is low, the INB pin is IO19, the channel is CH0 (LT0), and the analog output value is 0, then the motor does not rotate.



Set the motor pin INA to low level and the analog output value of the INB pin to different values, then the motor rotates clockwise at different speeds.

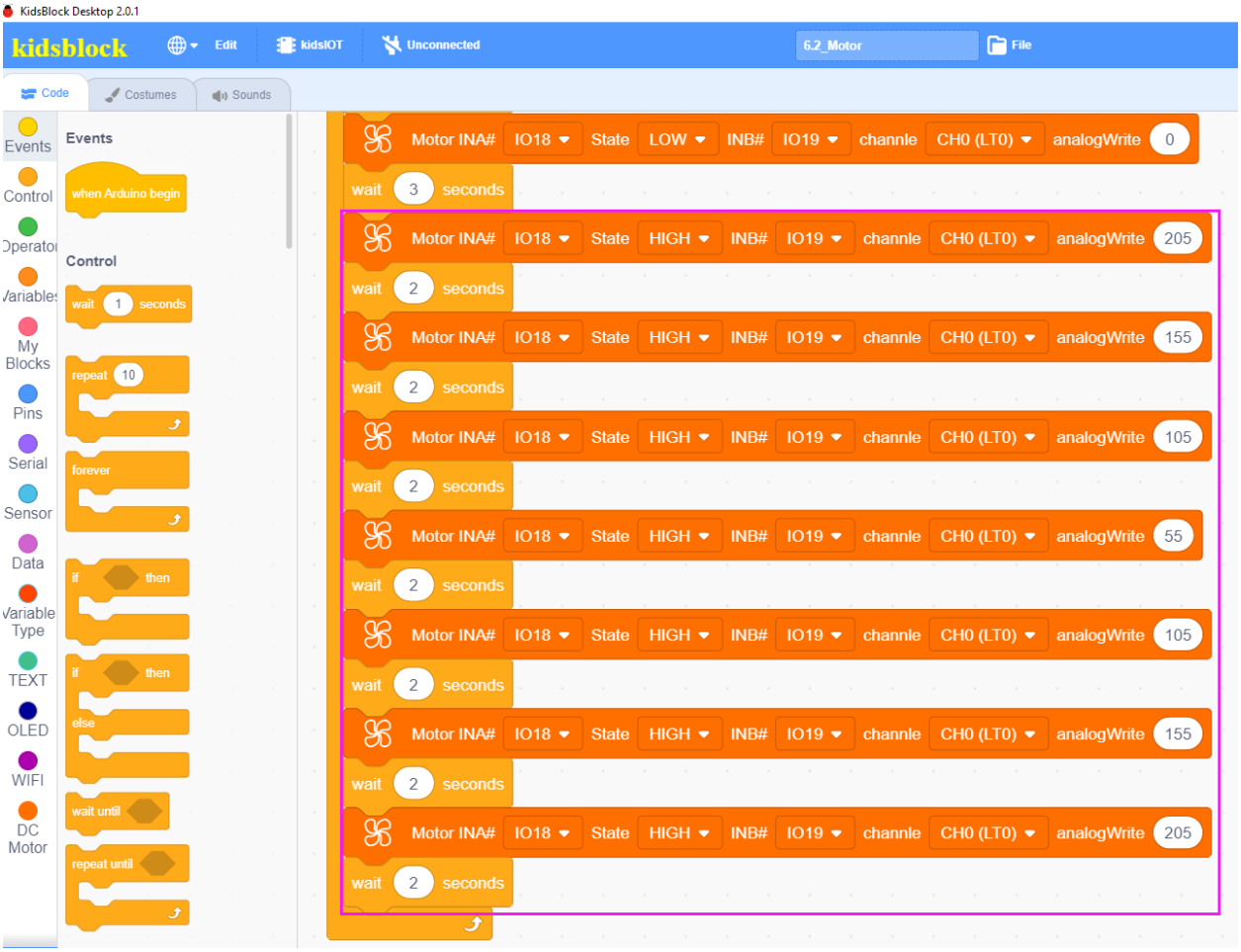


Set the motor to stop rotating for 3 seconds.

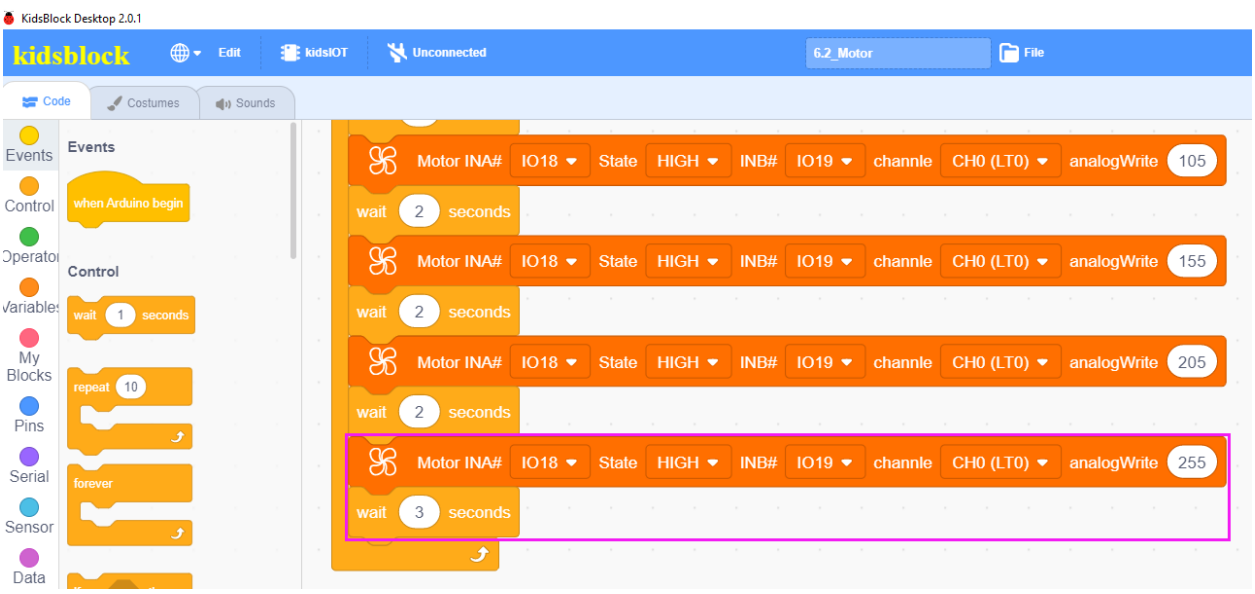


Set the motor pin INA to high level and the analog output value of the INB pin to different values, then the motor

rotates anticlockwise at different speeds.



Set the motor to stop rotating for 3 seconds.



Complete Program

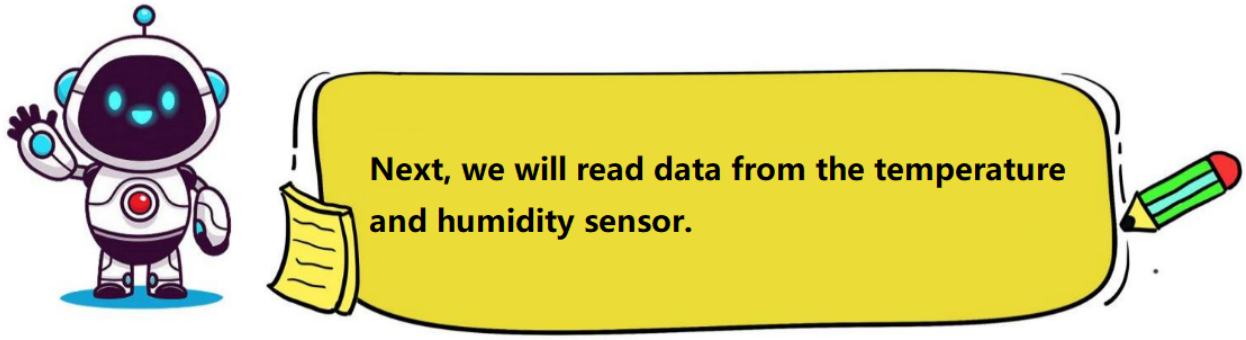
```
when Arduino begin
Motor INA# IO18 State LOW INB# IO19 channle CH0 (LT0) analogWrite 0
forever
  Motor INA# IO18 State LOW INB# IO19 channle CH0 (LT0) analogWrite 100
  wait 2 seconds
  Motor INA# IO18 State LOW INB# IO19 channle CH0 (LT0) analogWrite 150
  wait 2 seconds
  Motor INA# IO18 State LOW INB# IO19 channle CH0 (LT0) analogWrite 200
  wait 2 seconds
  Motor INA# IO18 State LOW INB# IO19 channle CH0 (LT0) analogWrite 150
  wait 2 seconds
  Motor INA# IO18 State LOW INB# IO19 channle CH0 (LT0) analogWrite 100
  wait 2 seconds
  Motor INA# IO18 State LOW INB# IO19 channle CH0 (LT0) analogWrite 0
  wait 3 seconds
  Motor INA# IO18 State HIGH INB# IO19 channle CH0 (LT0) analogWrite 205
  wait 2 seconds
  Motor INA# IO18 State HIGH INB# IO19 channle CH0 (LT0) analogWrite 155
  wait 2 seconds
  Motor INA# IO18 State HIGH INB# IO19 channle CH0 (LT0) analogWrite 105
  wait 2 seconds
  Motor INA# IO18 State HIGH INB# IO19 channle CH0 (LT0) analogWrite 55
  wait 2 seconds
  Motor INA# IO18 State HIGH INB# IO19 channle CH0 (LT0) analogWrite 105
  wait 2 seconds
  Motor INA# IO18 State HIGH INB# IO19 channle CH0 (LT0) analogWrite 155
  wait 2 seconds
  Motor INA# IO18 State HIGH INB# IO19 channle CH0 (LT0) analogWrite 205
  wait 2 seconds
  Motor INA# IO18 State HIGH INB# IO19 channle CH0 (LT0) analogWrite 255
  wait 3 seconds
```

(2). Test Result

Click 

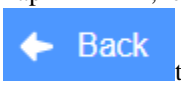
to upload the above complete code to the kidsIOT motherboard. After powering up via the external power supply, the motor rotates clockwise at different speeds and stops for 3 seconds, and then rotates counterclockwise at different speeds and stops for 3 seconds.

7. Read data from the temperature and humidity sensor

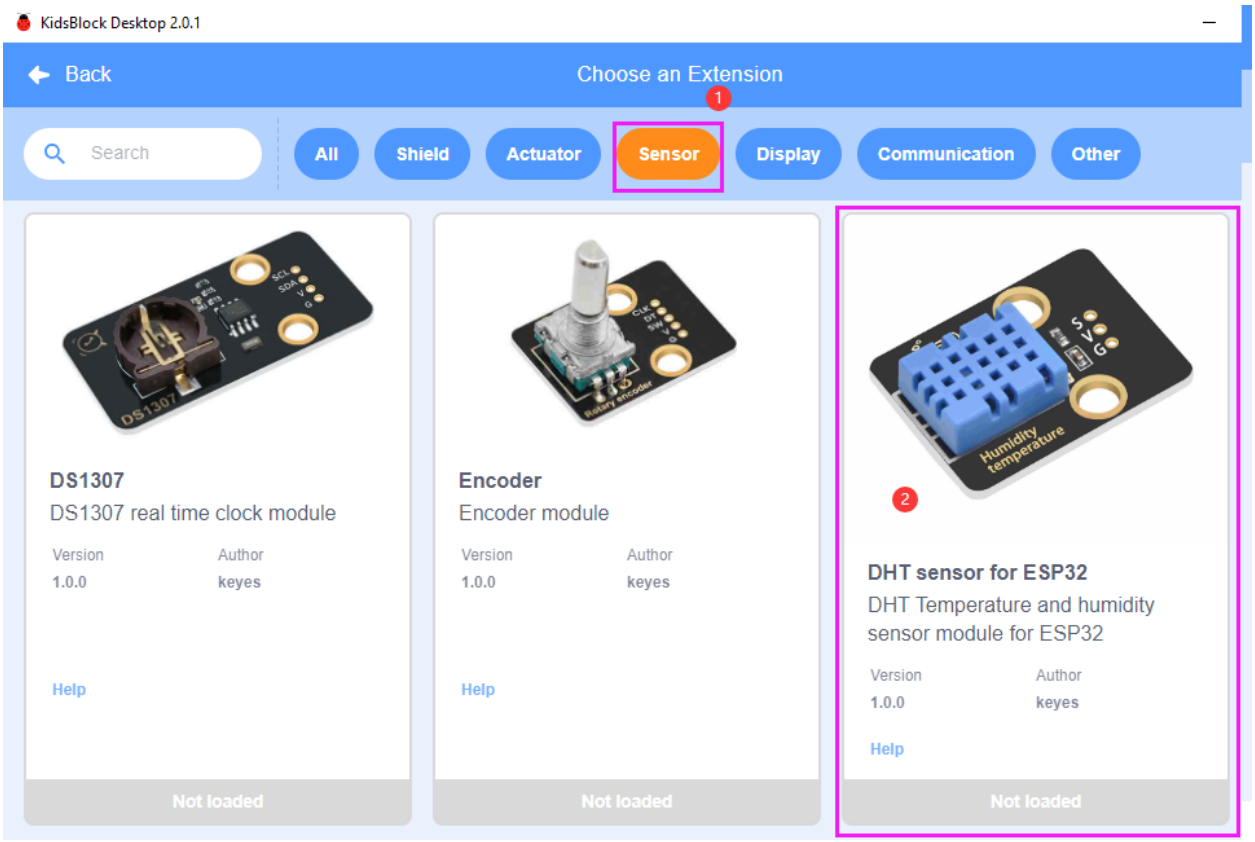


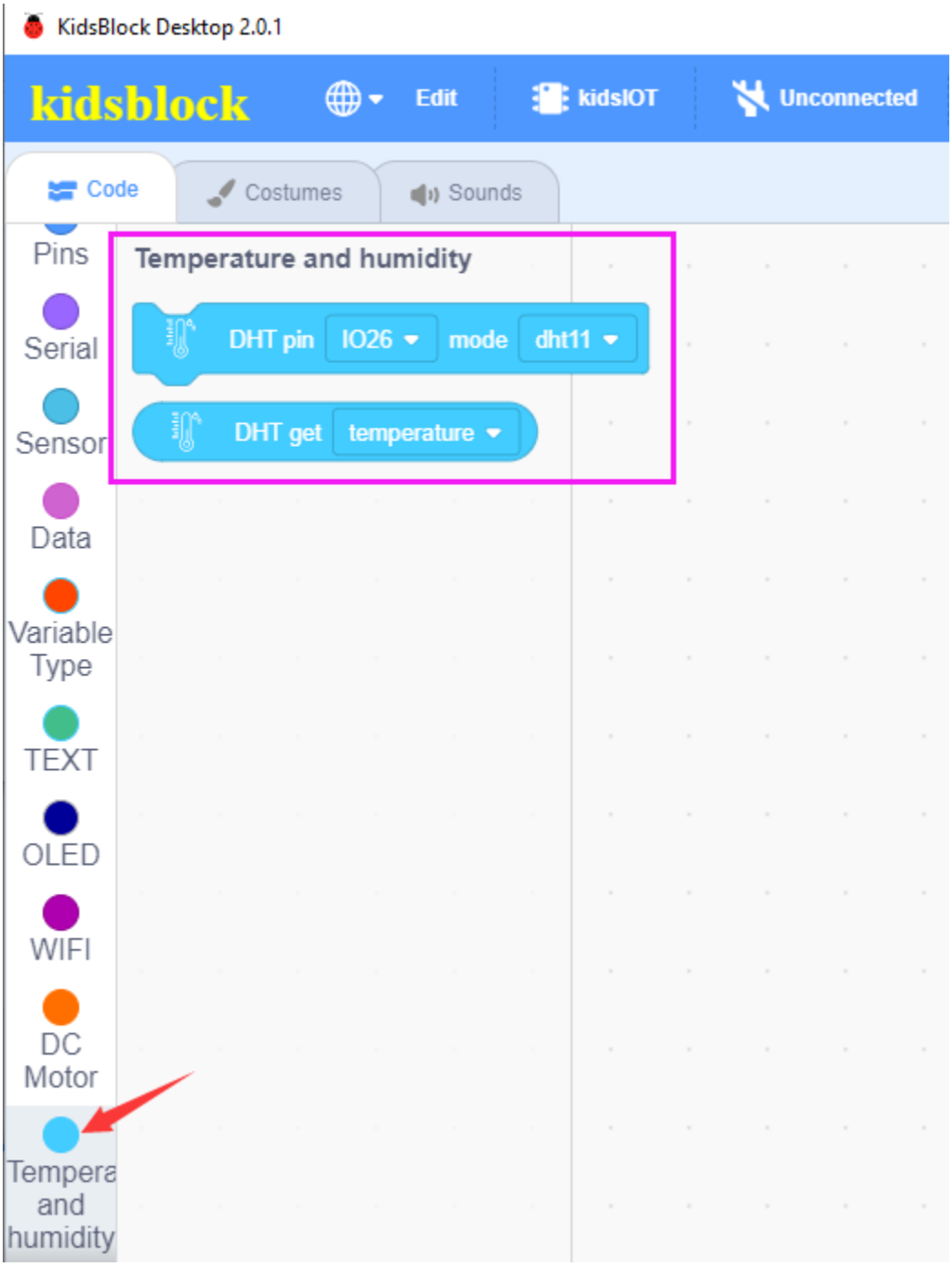
(1). Programming Steps

Step 1Add "temperature and humidity sensor"

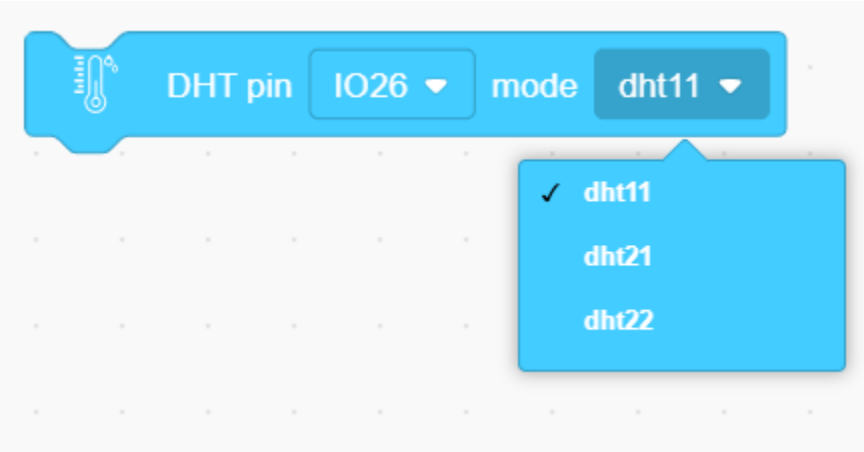
Tap  , click the "Sensor" module in the "Extension" , then select "**DHT sensor for esp32**" and click 

to return to the programming interface.

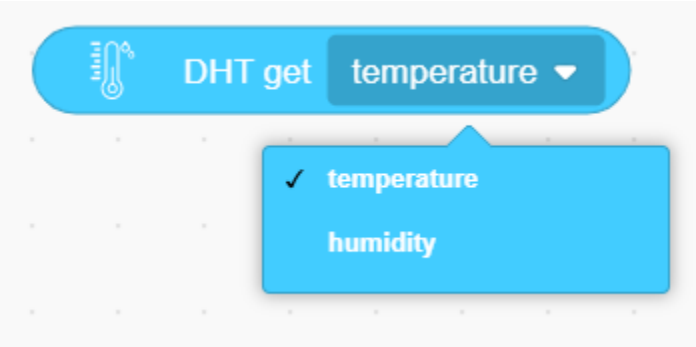

KidsBlock Desktop 2.0.1
Back
Choose an Extension
1
Search
All
Shield
Actuator
Sensor
Display
Communication
Other
DS1307
DS1307 real time clock module
Version
1.0.0
Author
keyes
Help
Not loaded
Encoder
Encoder module
Version
1.0.0
Author
keyes
Help
Not loaded
2
DHT sensor for ESP32
DHT Temperature and humidity sensor module for ESP32
Version
1.0.0
Author
keyes
Help
Not loaded


KidsBlock Desktop 2.0.1
kidsblock
Edit
kidsIOT
Unconnected
Code
Costumes
Sounds
Pins
Serial
Sensor
Data
Variable Type
TEXT
OLED
WIFI
DC Motor
Tempera and humidity
Temperature and humidity
DHT pin
IO26
mode
dht11
DHT get
temperature

Step 2Description of the Building Block



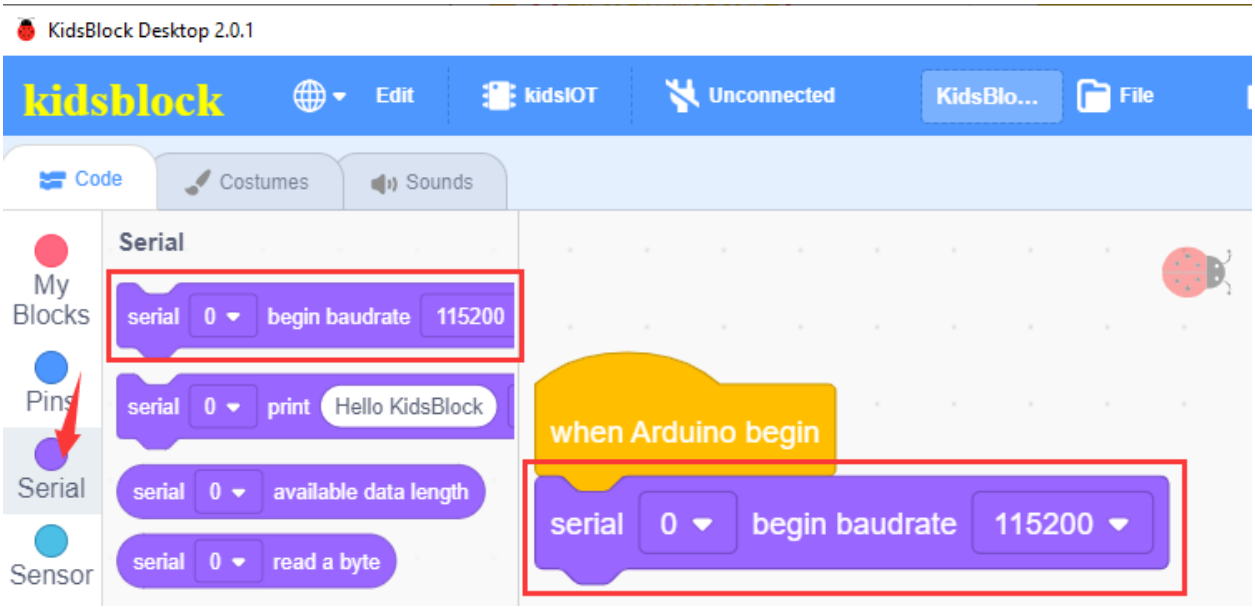
Initialize the pin and mode of the temperature and humidity sensor (dht11, dht21 or dht22).



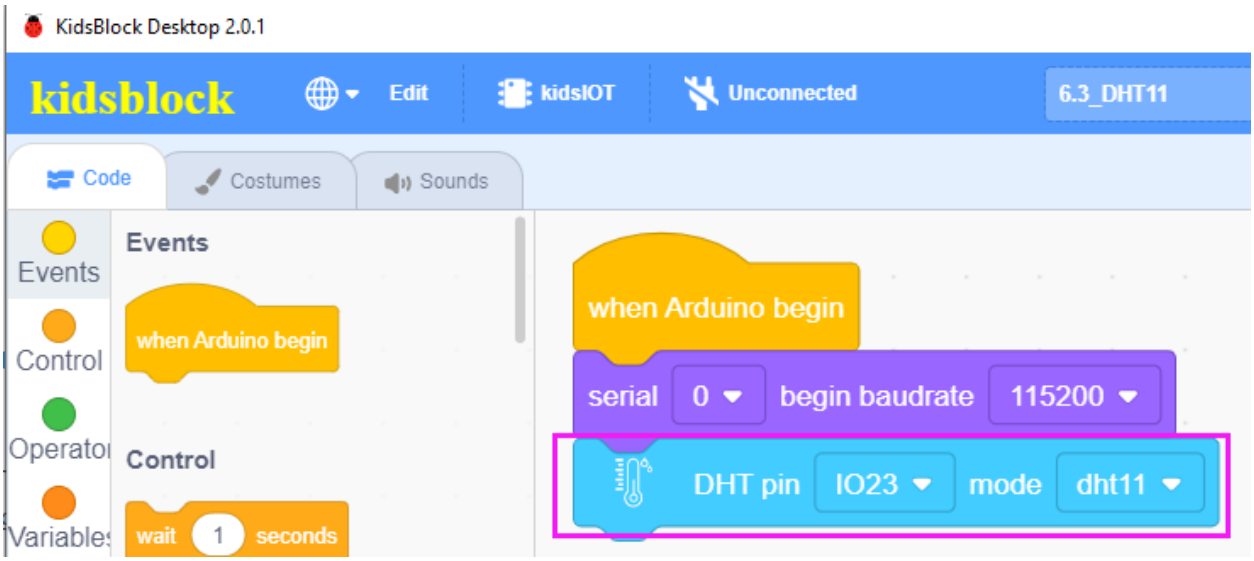
Read the temperature and humidity from the temperature and humidity sensor.

Step 3Write the Program

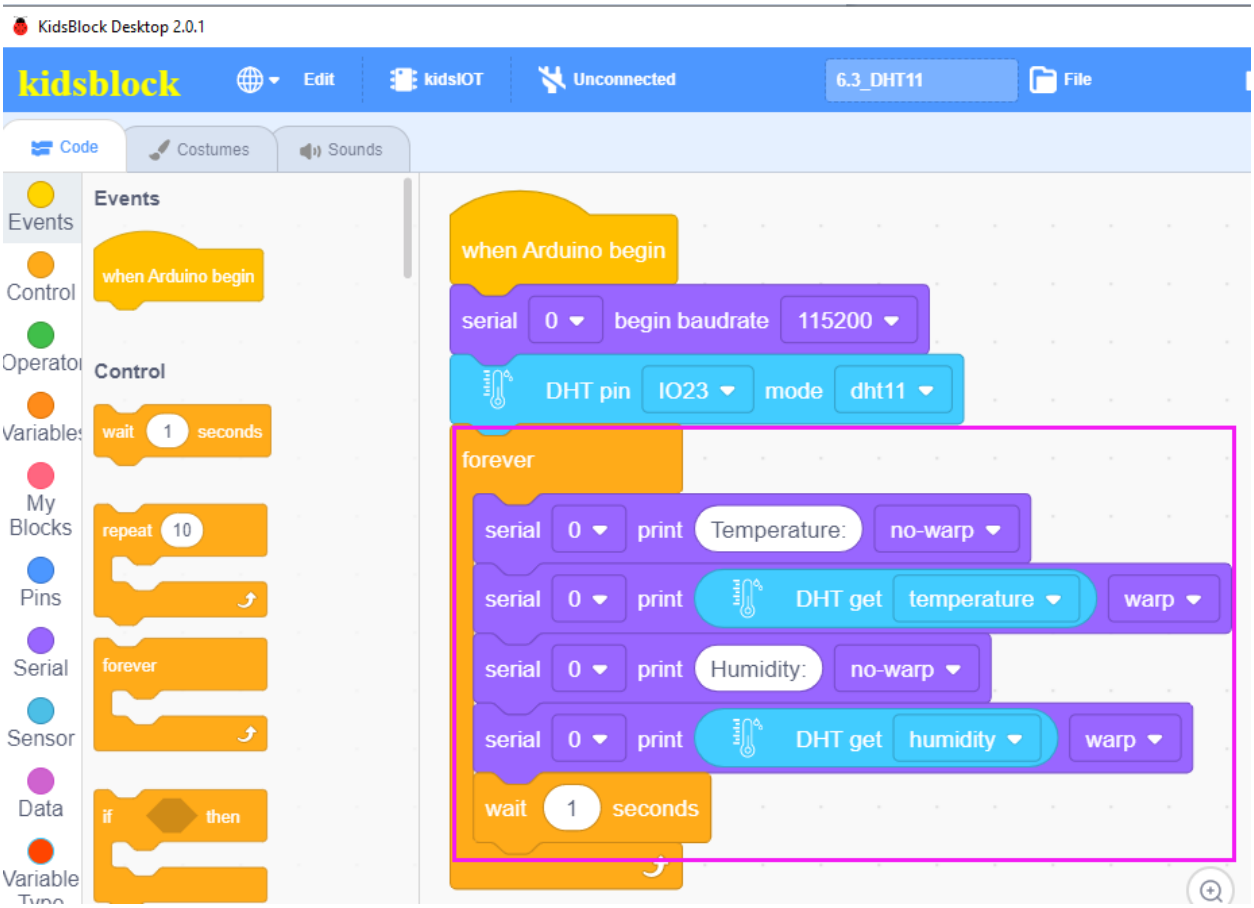
Set the baud rate to 15200.



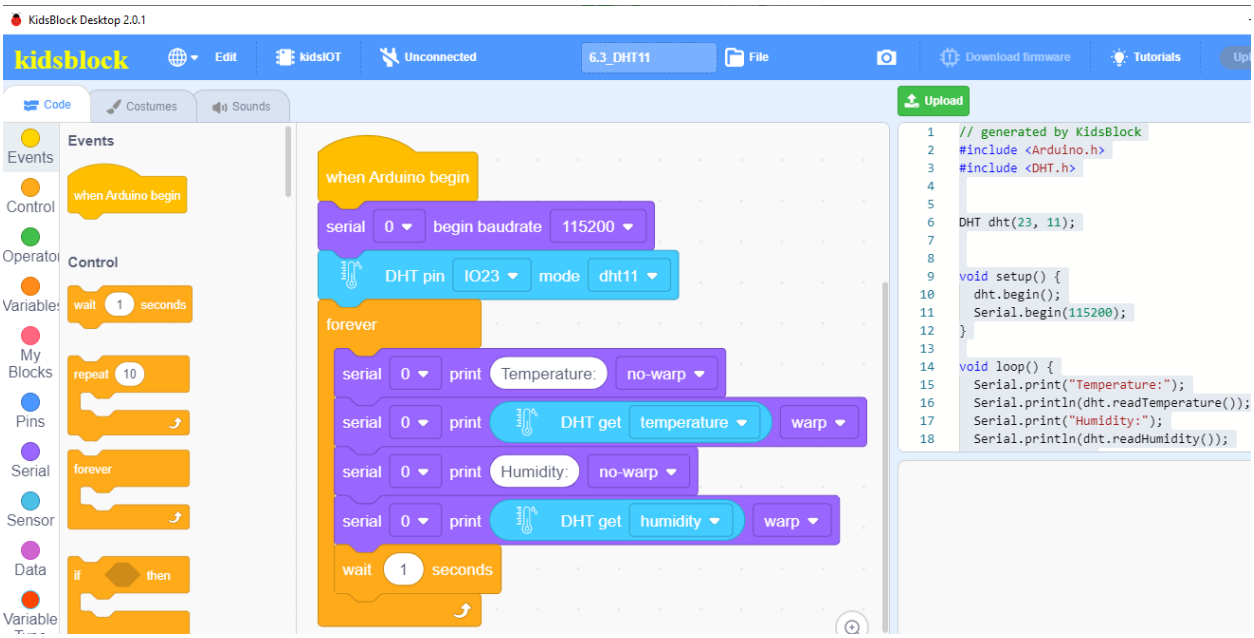
Initialize pin IO23 of the temperature and humidity sensor, and select the dht11 mode.



The serial port prints the read temperature and humidity in the current environment every 1 second.



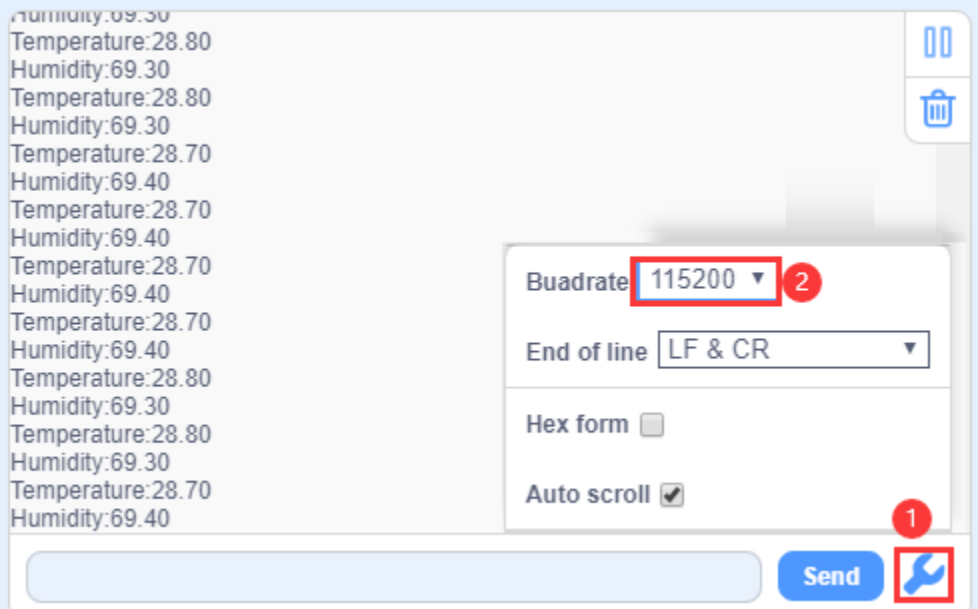
Complete Program



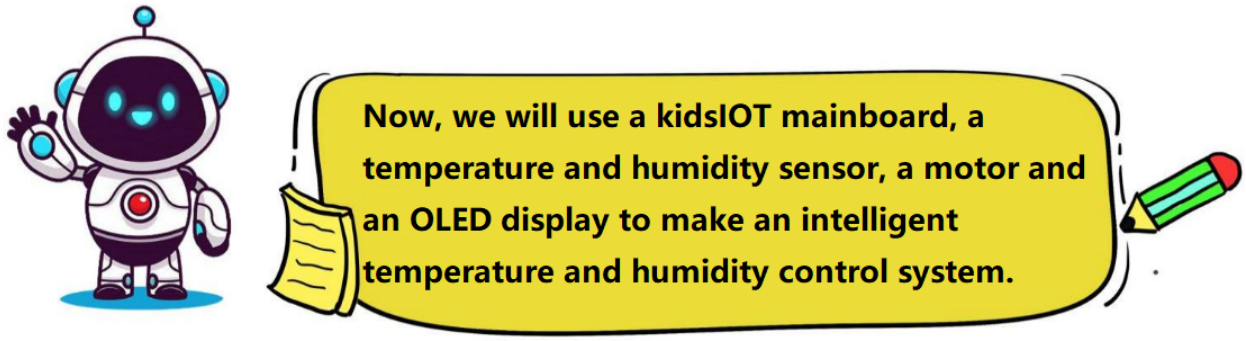
(2). Test Result

Click 

to upload the above complete code to the kidsIOT mainboard. After powering up via the USB cable, click in the serial monitor and set the baud rate to 15200. Then the serial port prints the temperature and humidity in the current environment.

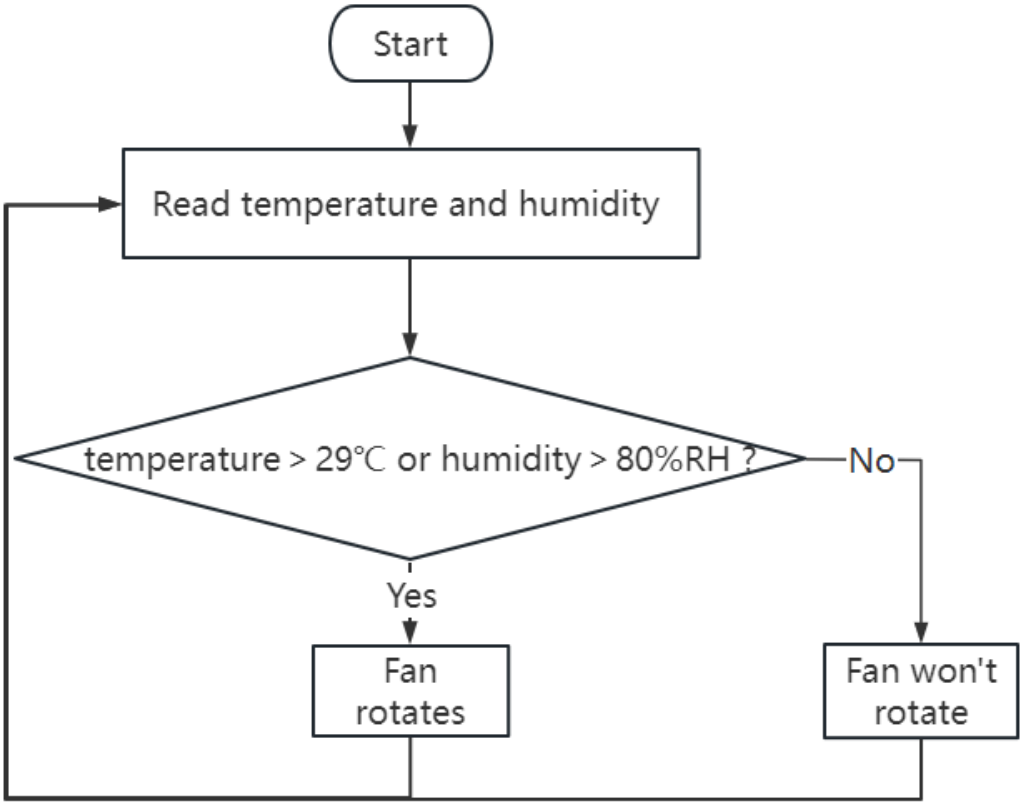


8. Temperature and Humidity Control System



(1). Programming Steps

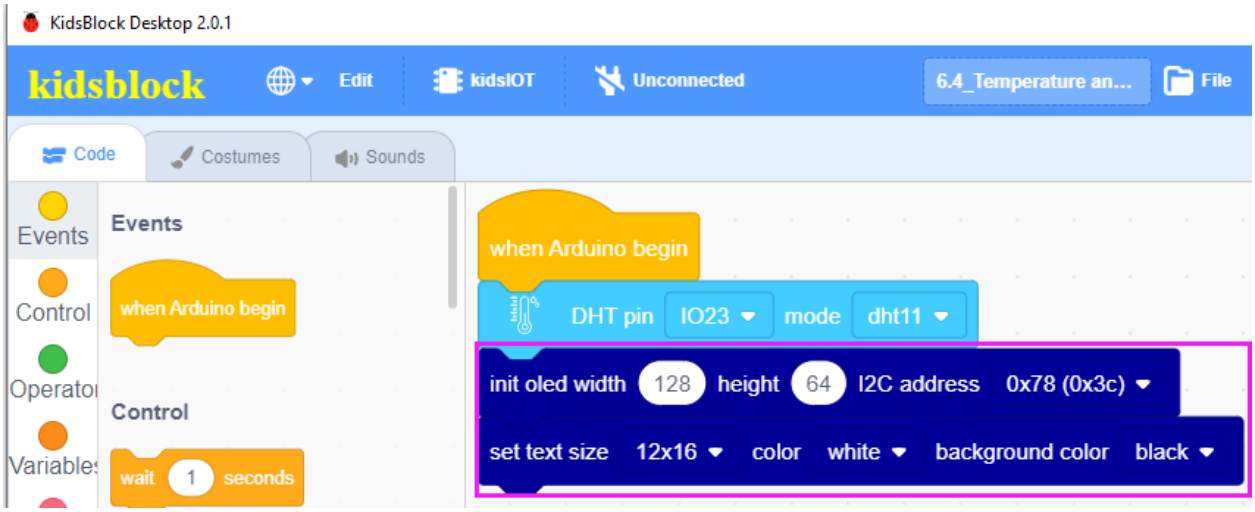
Step 1Flow Chart



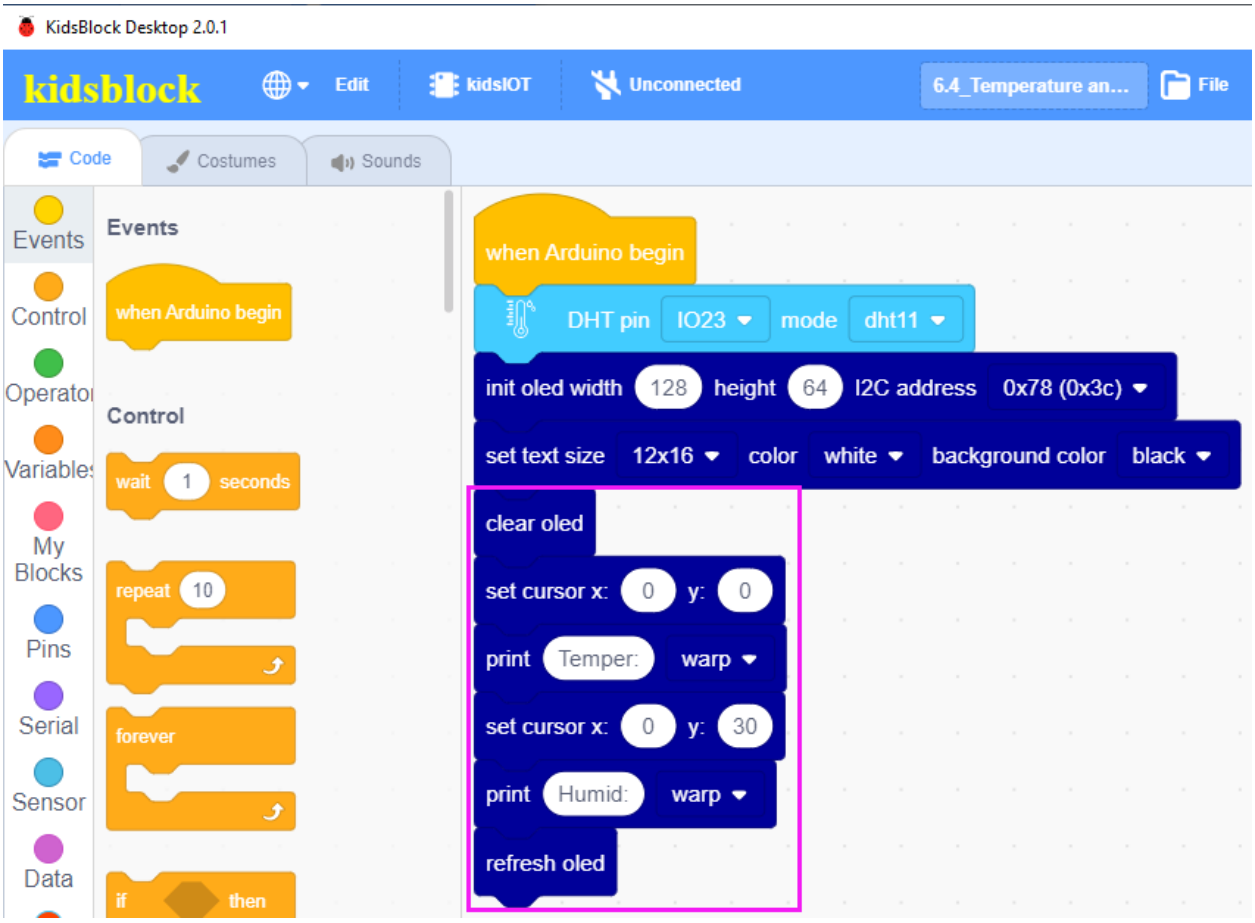
Step 2Write the Program

Initialize pin IO23 of the temperature and humidity sensor, and select the dht11 mode.

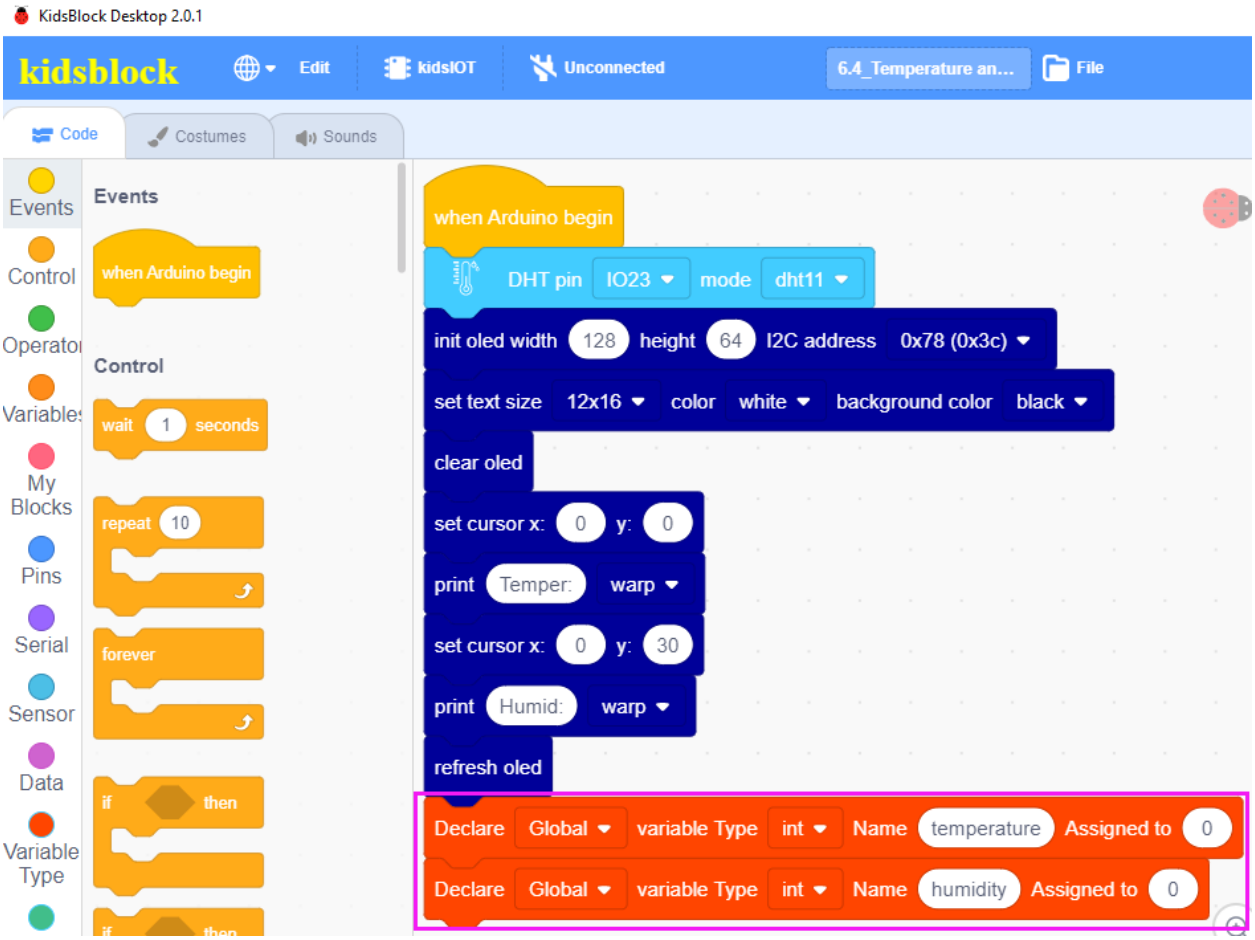
Initialize the width, height, I2C address, text size and color as well as background color of the OLED display.



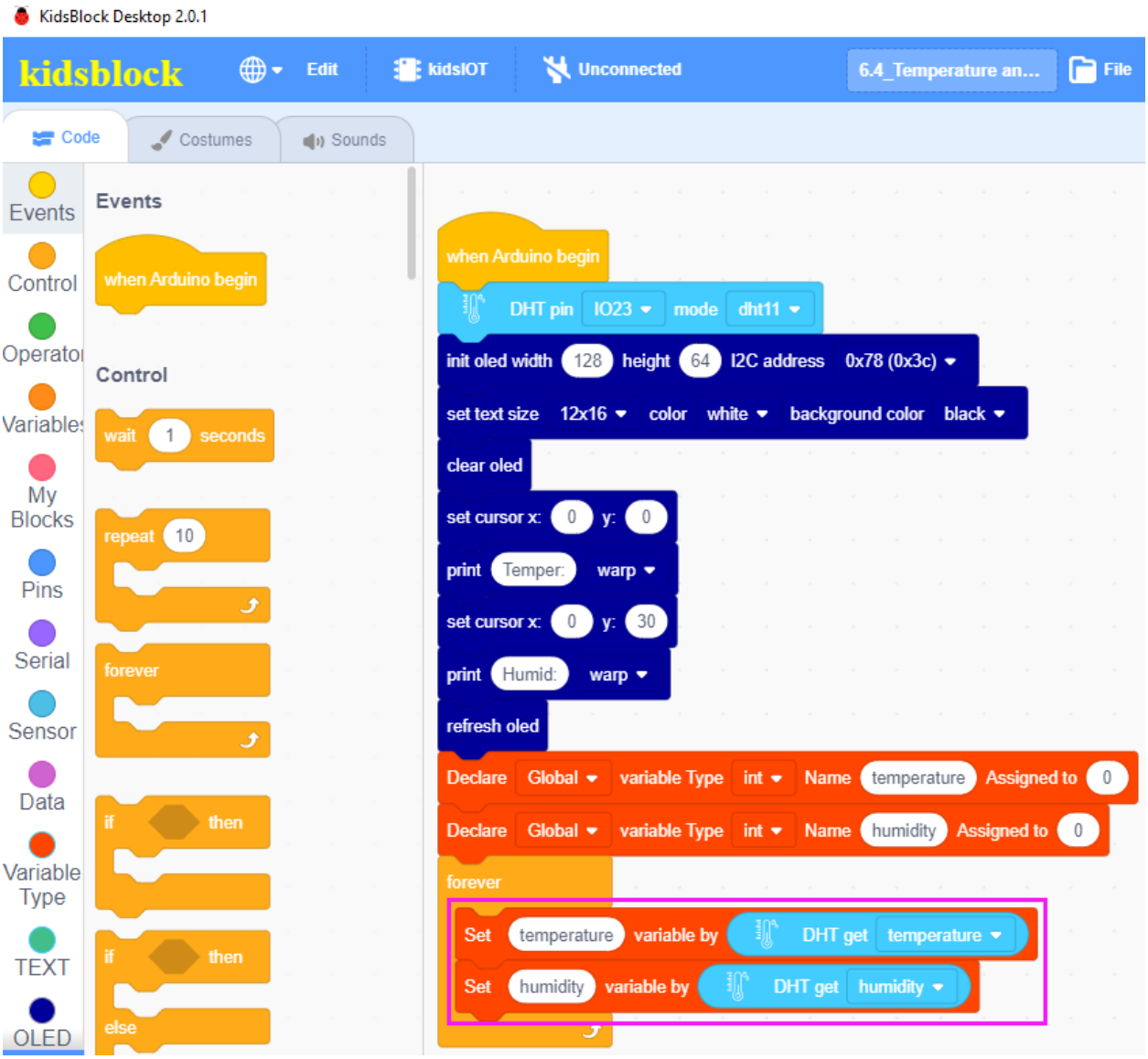
Set the strings "Temper:" and "Humid:" to be displayed on the OLED.



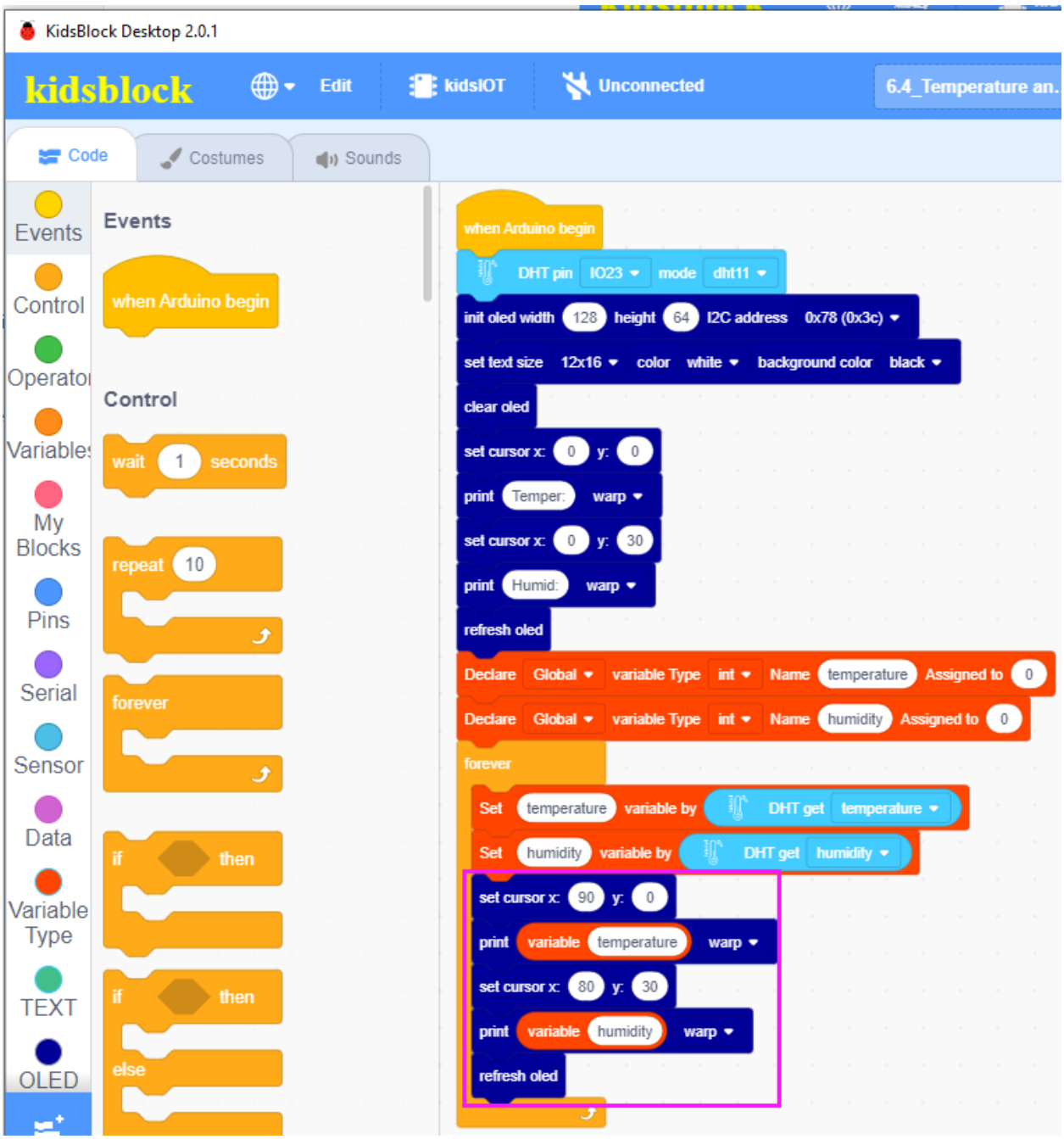
Define variables "**temperature**" and "**humidity**".



Store the read temperature data into the "temperature" variable. The read humidity data is stored in the "humidity" variable.



Display temperature data and humidity data at the corresponding position on the OLED.



Determine the temperature and humidity value in the environment detected by the temperature and humidity sensor. When the temperature is greater than 29°C, or the humidity is greater than 80%RH, the fan will be turned on.

Complete Program

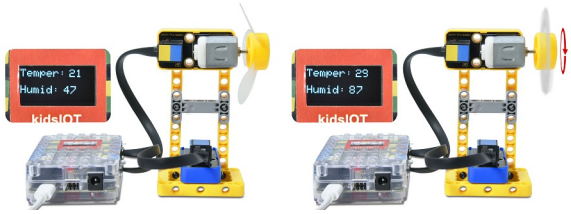

when Arduino begin
DHT pin IO23 mode dht11
init oled width 128 height 64 I2C address 0x78 (0x3c)
set text size 12x16 color white background color black
clear oled
set cursor x: 0 y: 0
print Temper: warp
set cursor x: 0 y: 30
print Humid: warp
refresh oled
Declare Global variable Type int Name temperature Assigned to 0
Declare Global variable Type int Name humidity Assigned to 0
forever
Set temperature variable by DHT get temperature
Set humidity variable by DHT get humidity
set cursor x: 90 y: 0
print variable temperature warp
refresh oled
if variable temperature > 29 or variable humidity > 80 then
Motor INA# IO18 State LOW INB# IO19 channle CH0 (LT0) analogWrite 150
else
Motor INA# IO18 State LOW INB# IO19 channle CH0 (LT0) analogWrite 0

(2). Test Result

Click 

to upload the above complete code to the kidsIOT mainboard. After powering up via the external power, the OLED displays the temperature and humidity in the current environment.

When the temperature reaches 29°C or the humidity reaches 80%RH, the motor will turn on to dissipate heat or dehumidify (the fan simulates heat dissipation and dehumidification, and the heat dissipation and dehumidification effect is average); otherwise, the motor will turn off to achieve automatic farm temperature and humidity control effect.



9. Common Problems

Q1: Is the temperature and humidity sensor waterproof?

A: The sensor detects the temperature and humidity in the air, which is not waterproof. Please do not put the module into water.

Q2: Does the rotation of the motor cause the kidsIOT mainboard to reset?

A: When the motor rotates, it requires a larger current than other sensors, which will cause voltage and current fluctuations in the circuit. Especially when the motor rotates forward and reverse, the voltage and current fluctuations are too large, causing the voltage and current of the kidsIOT mainboard to be too low, thus causing a reset.

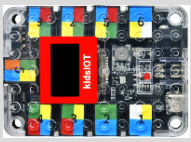
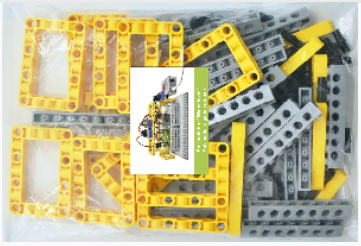
4.3.7 Project 07Soil Moisture Detection System

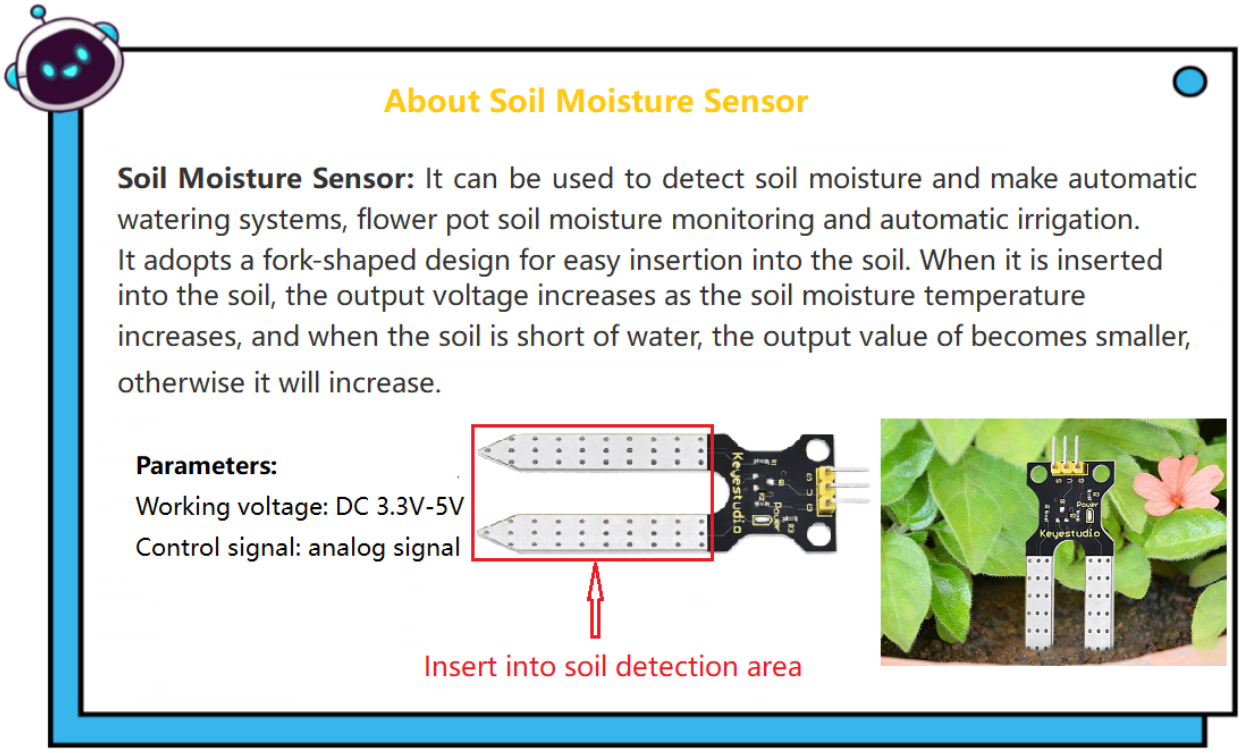
Note: Do not allow water to overflow from sinks and soil troughs when using the device. Sprinkling water on other sensors will cause a short circuit and device failure. Sprinkling water on batteries will cause heating and explosion. Please be careful when using the device, especially when used by young children, it must be under the supervision of parents. To ensure safe operation of the device, please follow relevant usage guidelines and safety regulations.

1. Description

This project introduces how to use a kidsIOT mainboard, a soil moisture sensor, a passive buzzer and an OLED display to make an intelligent soil moisture detection system. The system can display the value of the soil moisture sensor in real time through the OLED display. When the soil moisture is lower than the set value, the buzzer will sound an alarm to remind you that it is time to water the land.

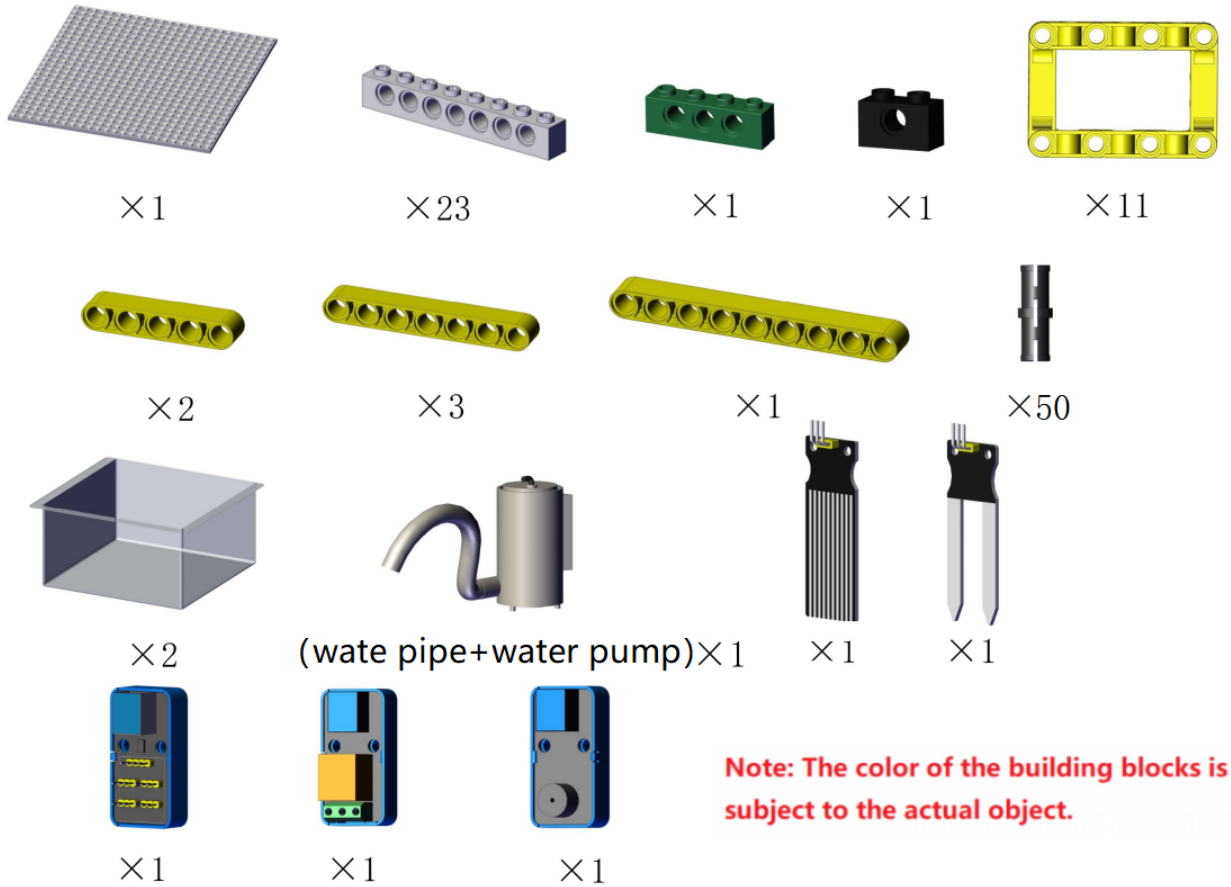
2. Components

				
kidsIOT Mainboard×1	Soil Moisture Sensor×1	Passive Buzzer×1	GPIO Shield×1	Wire×2
				
F-F DuPont Wires	USB Cable×1	Soil Moisture/Water Level/Automatic Irrigation System LEGO Pieces×1	Sink×1	



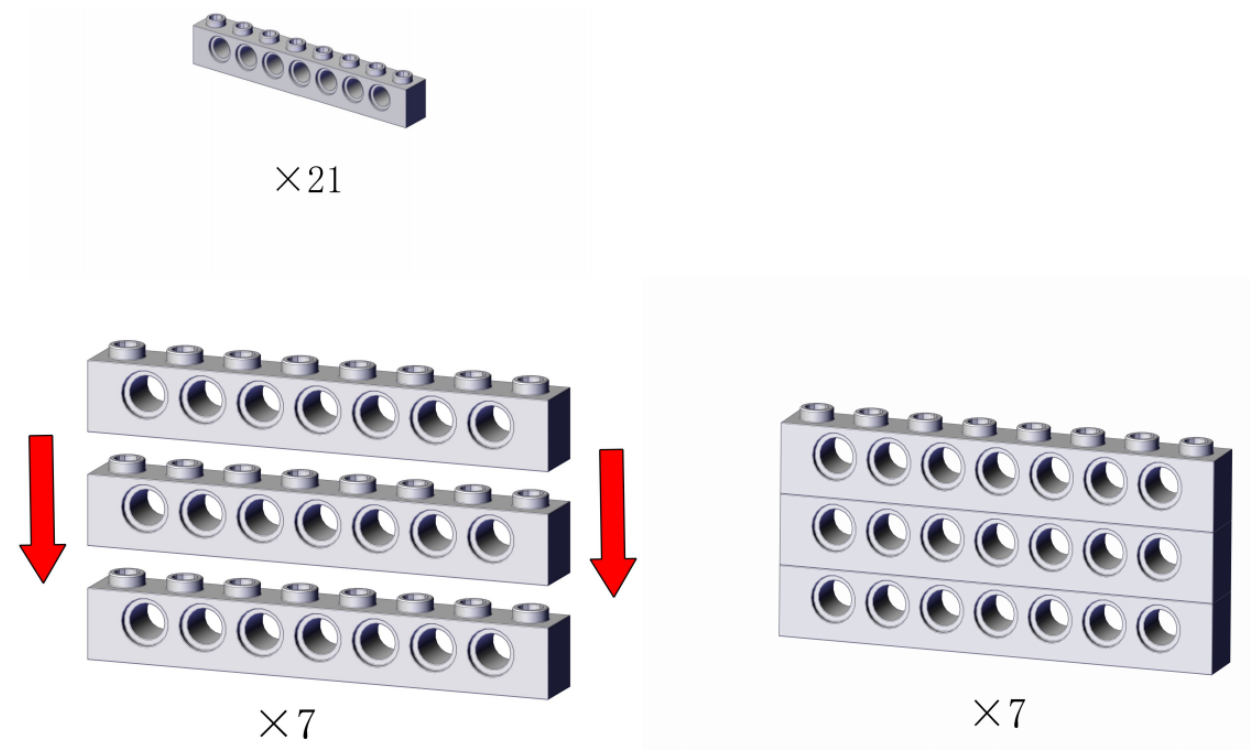
3. Assembly Steps

Step 1Components Needed

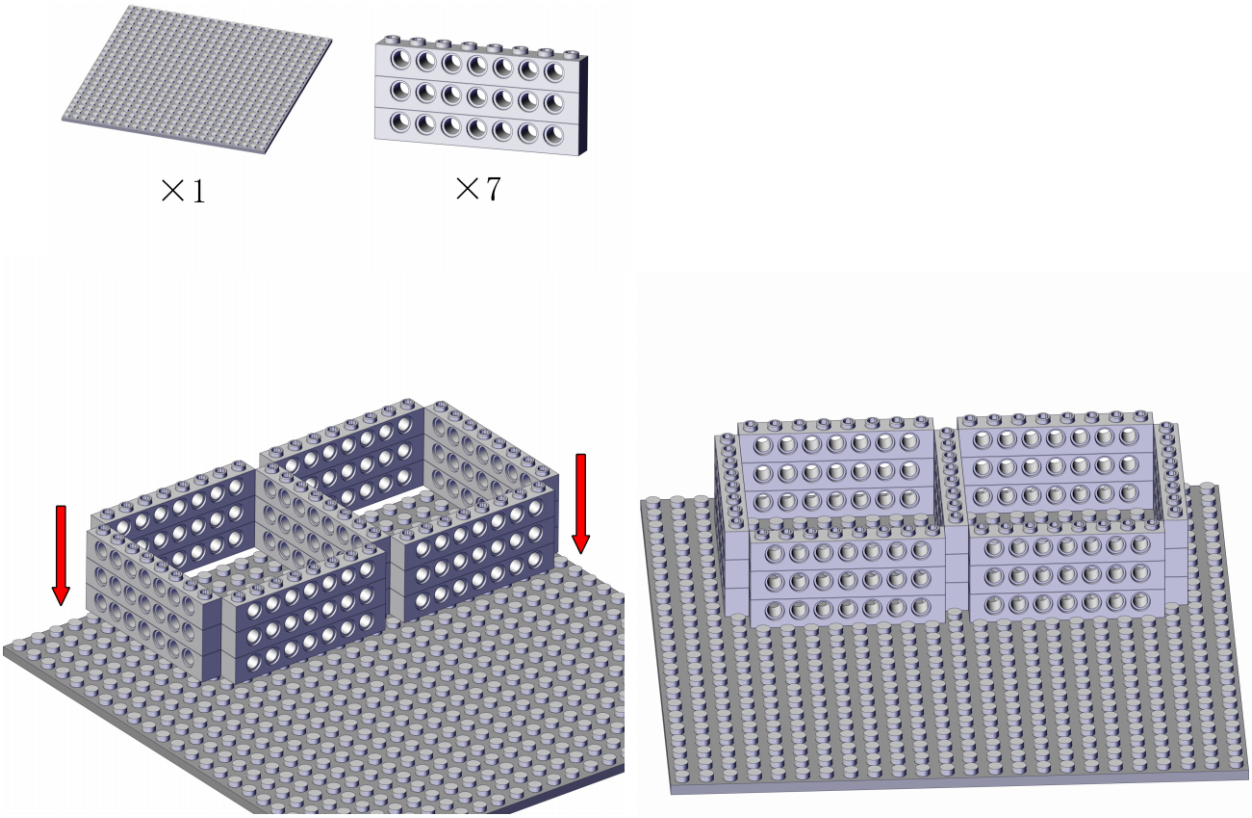


Step 2Process

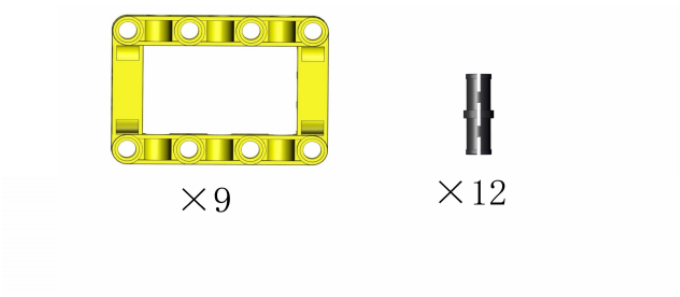
Process 1

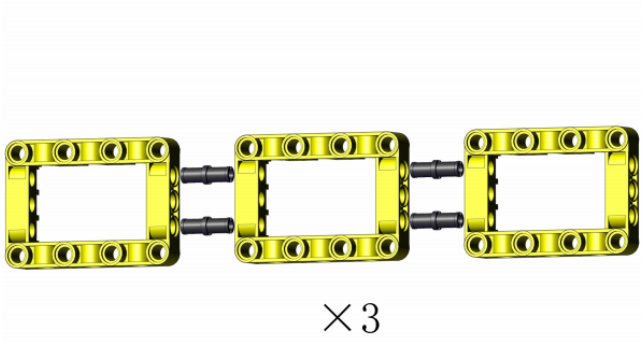


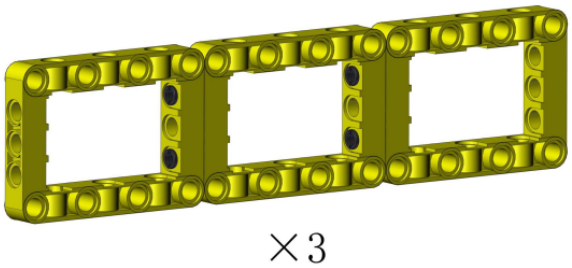
Process 2



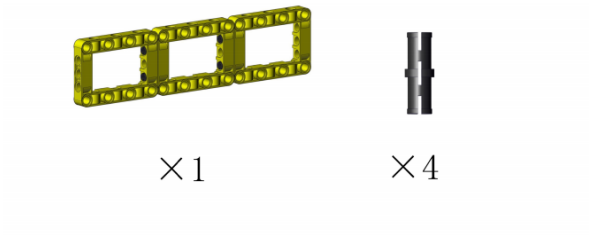
Process 3

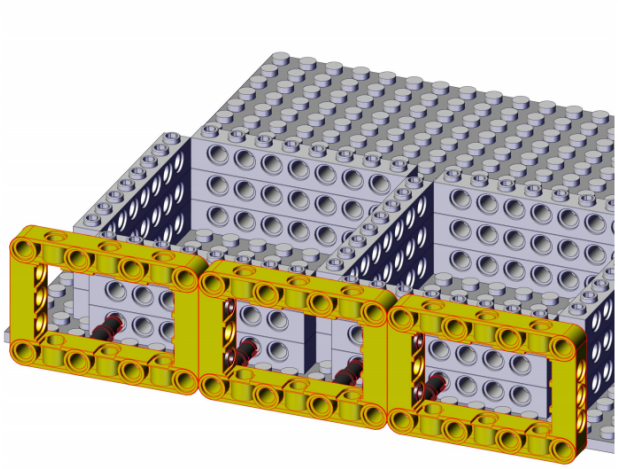


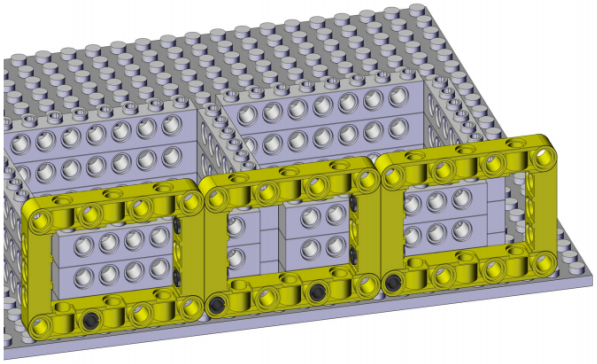




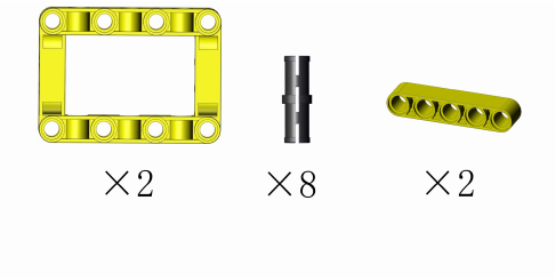
Process 4

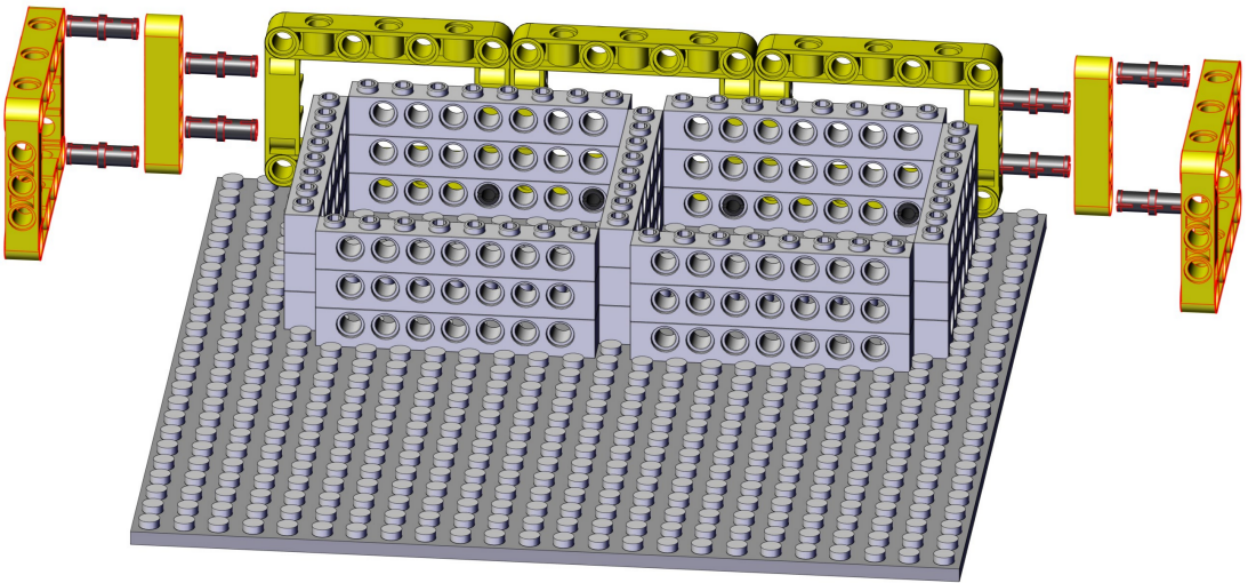


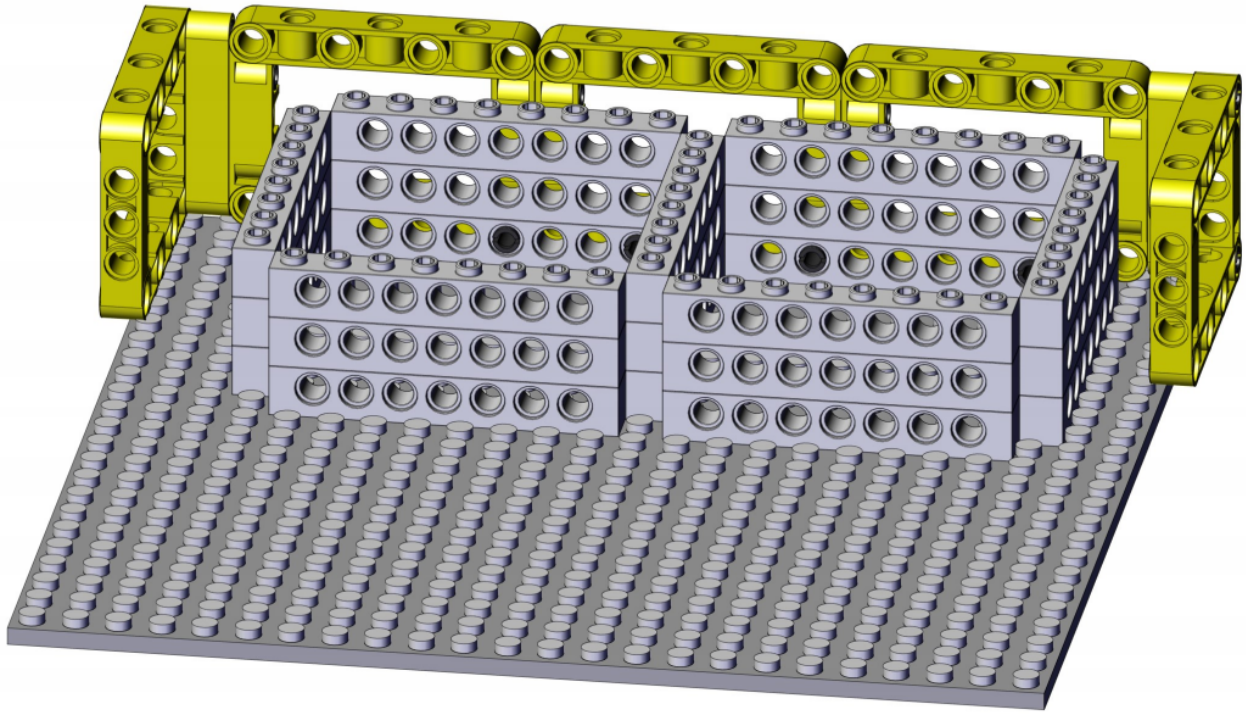




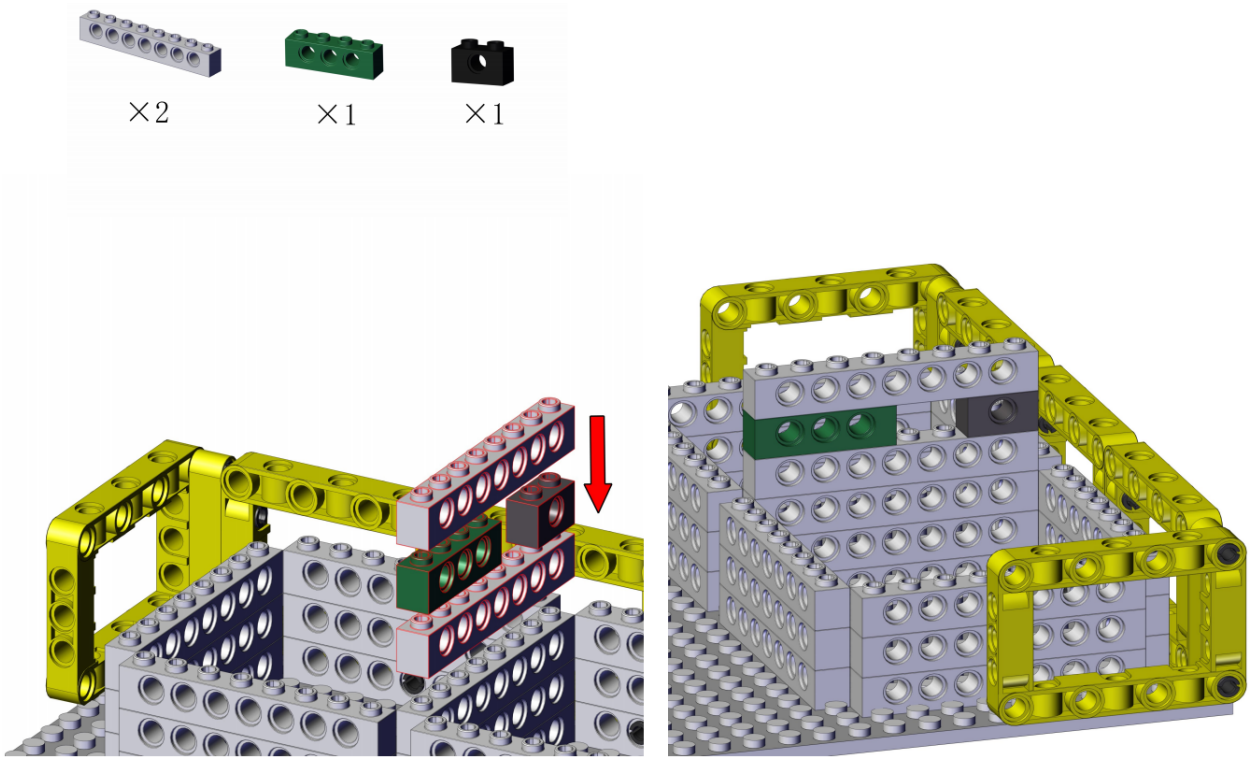
Process 5


×2
×8
×2

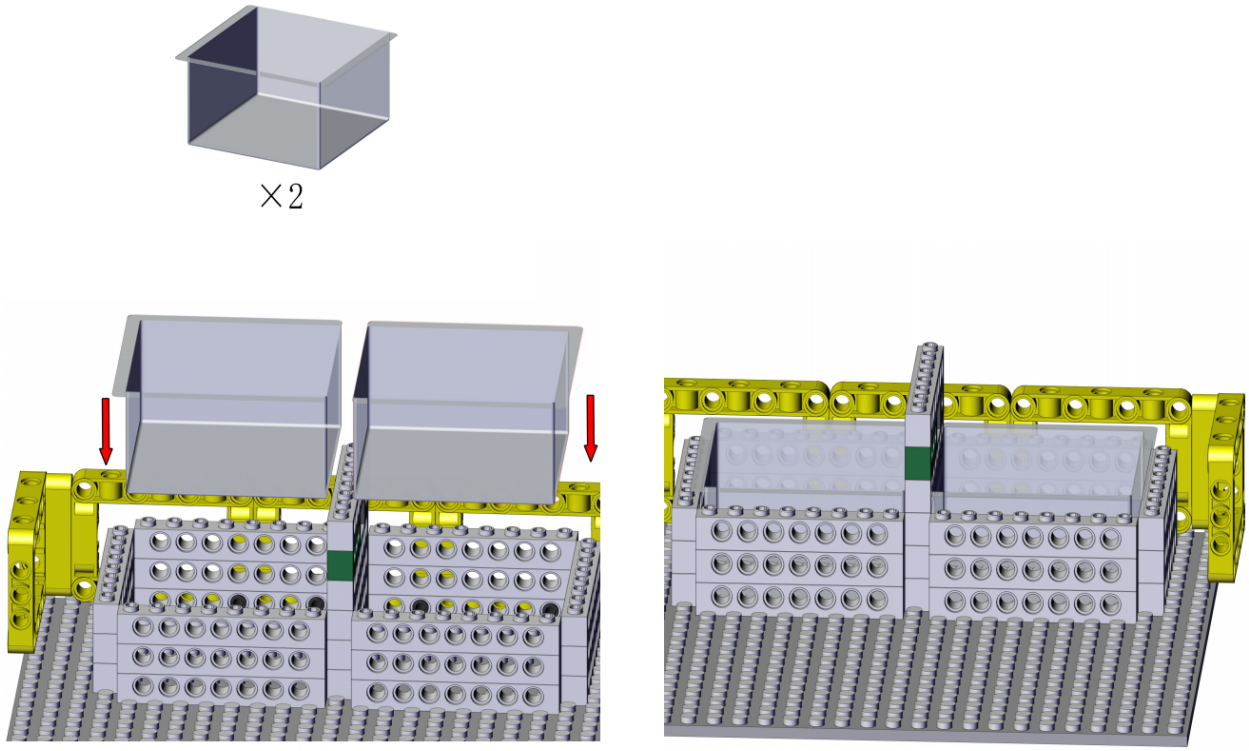




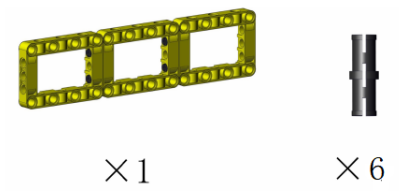
Process 6

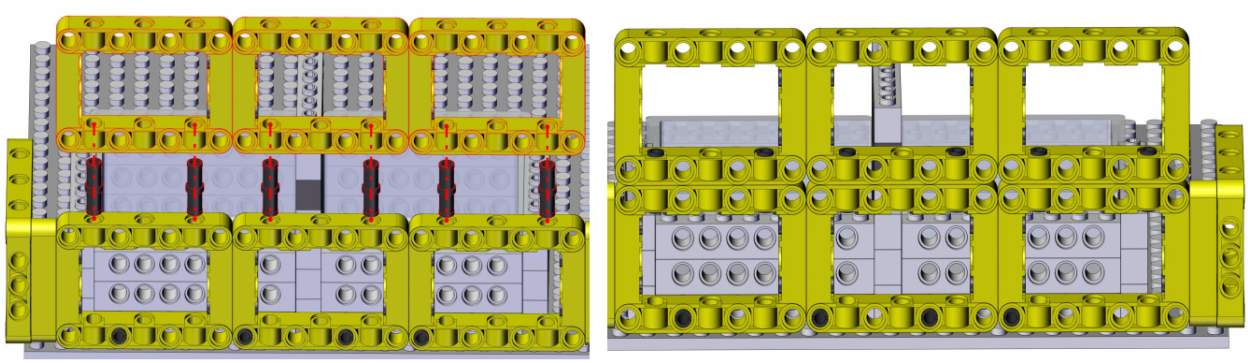


Process 7

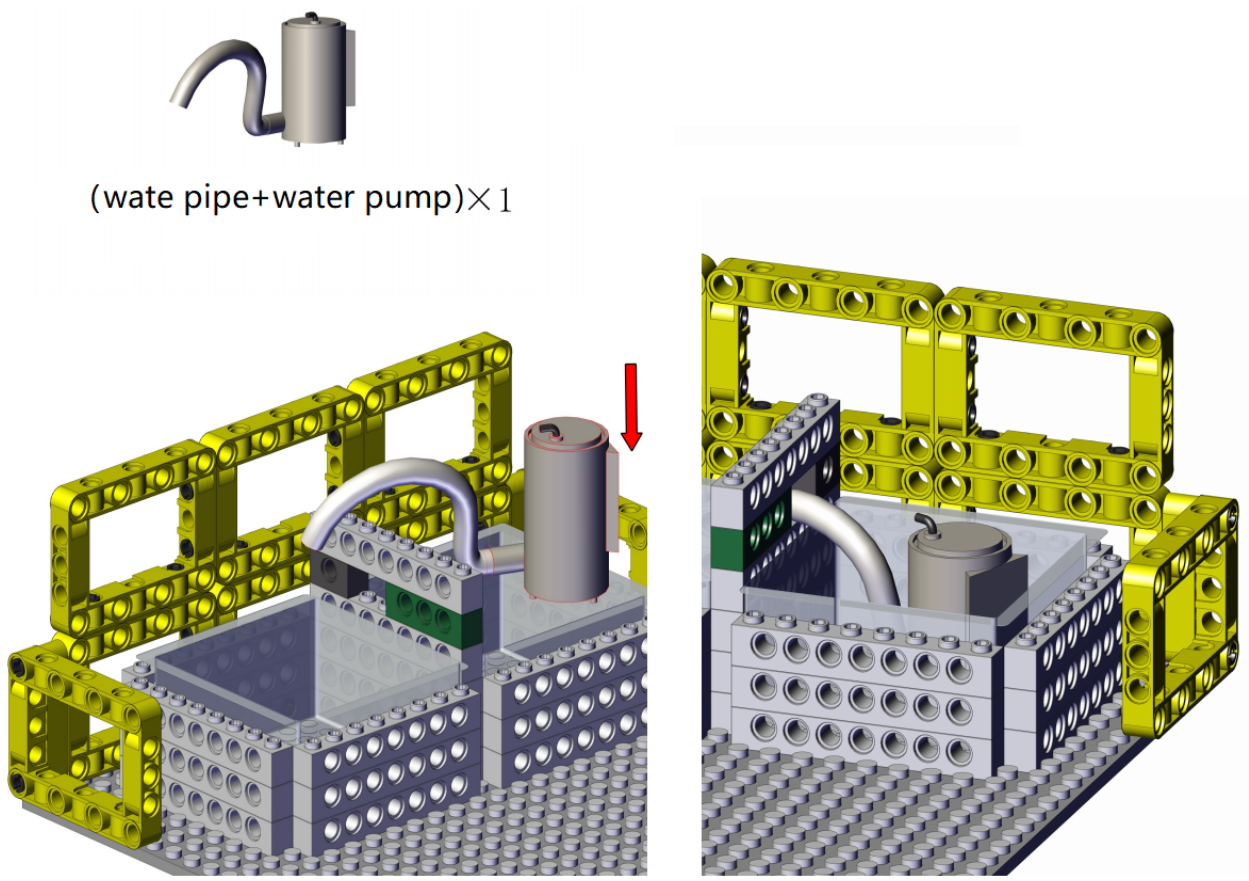


Process 8

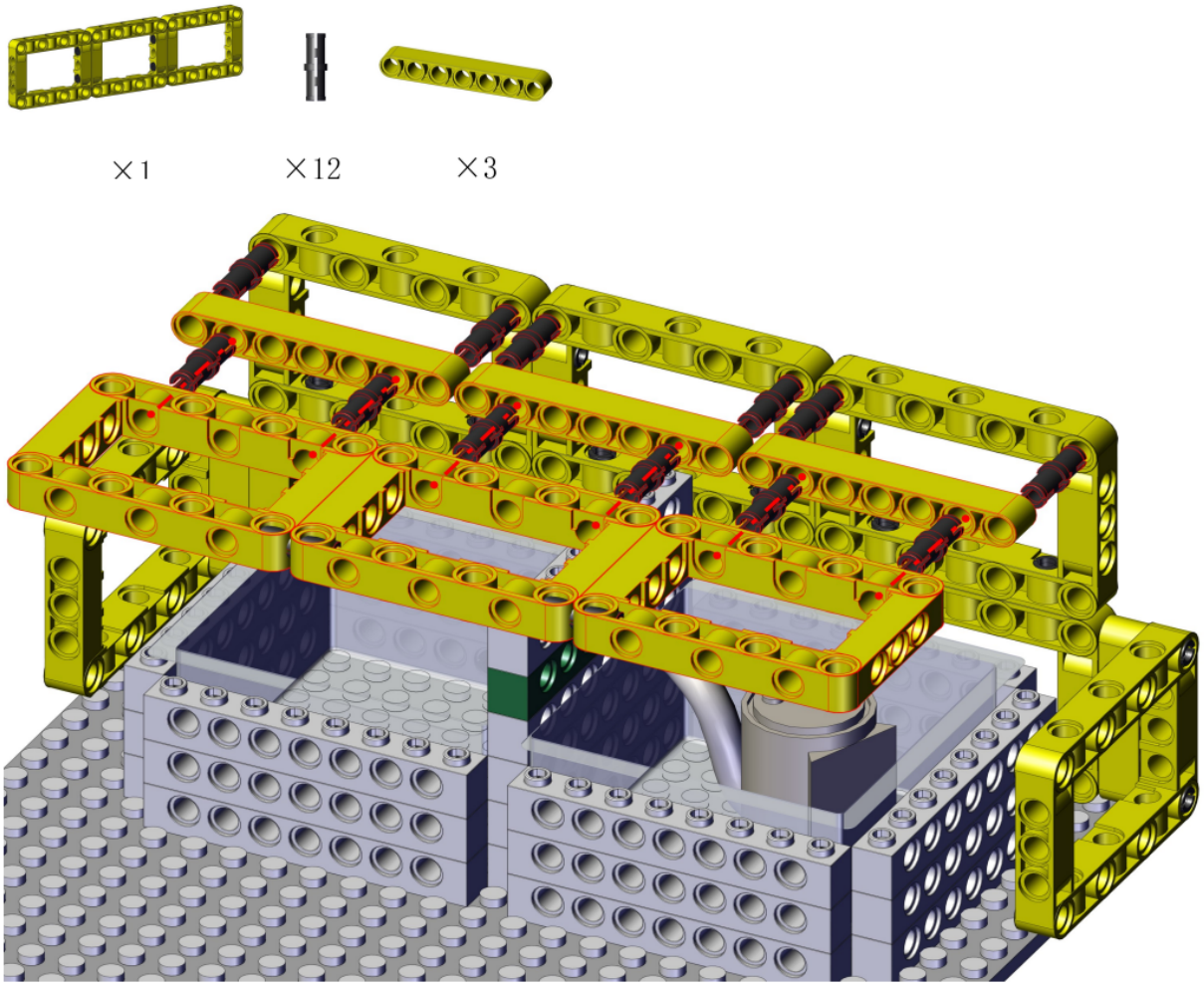


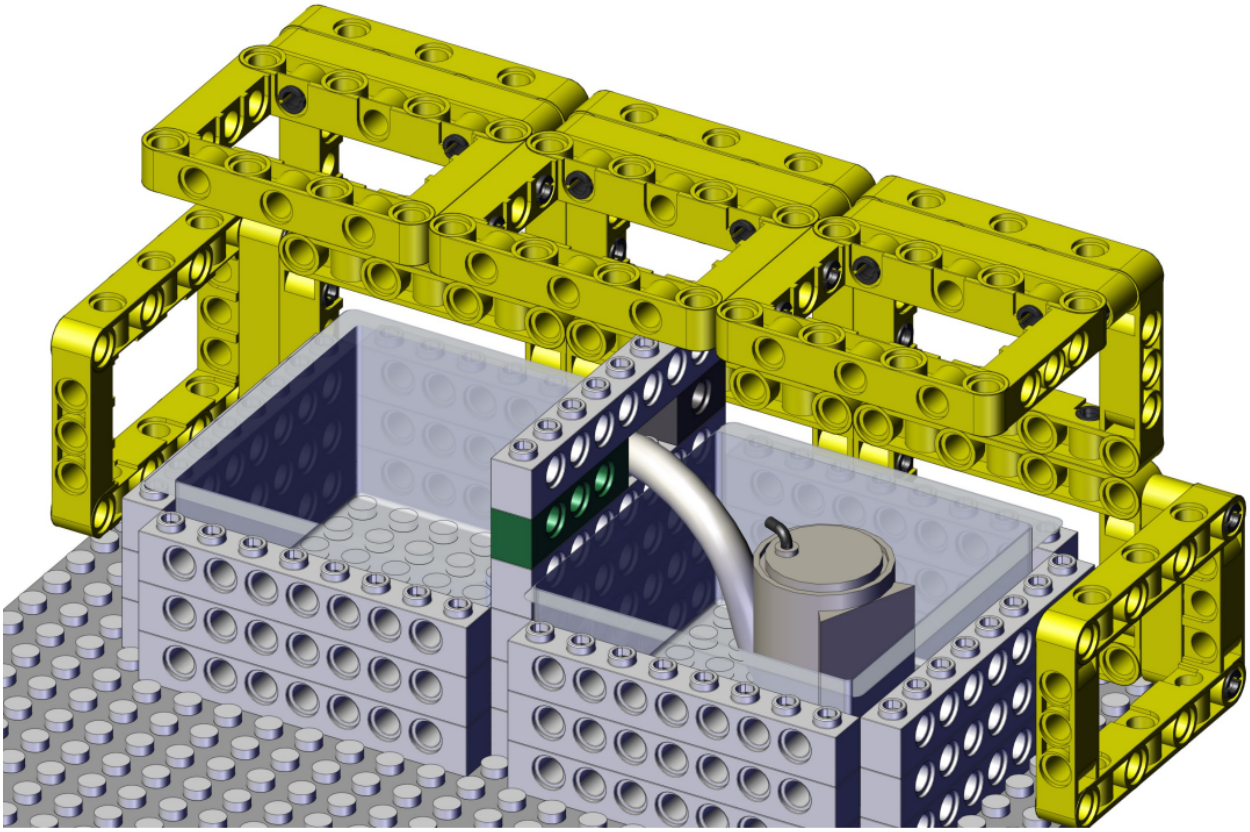


Process 9

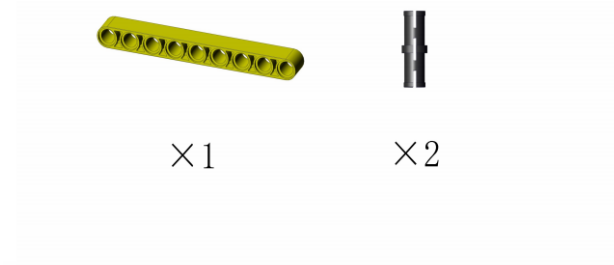


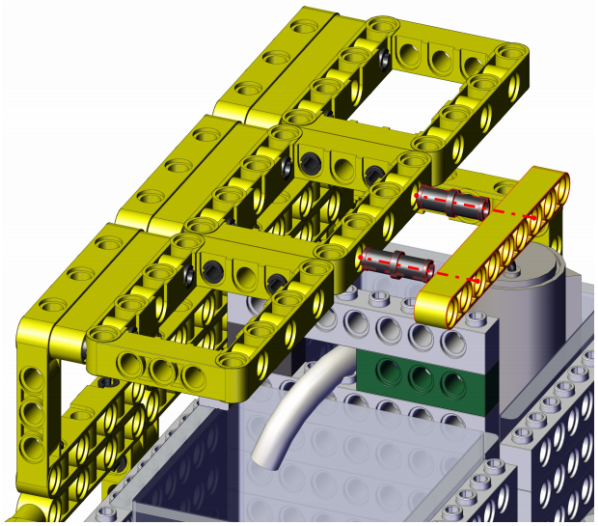
Process 10

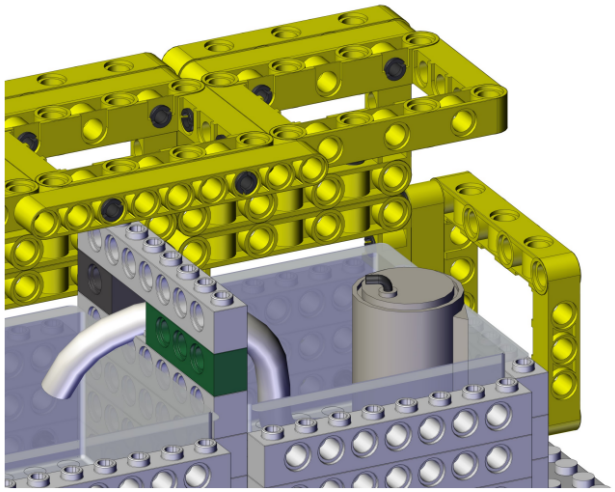

×1
×12
×3



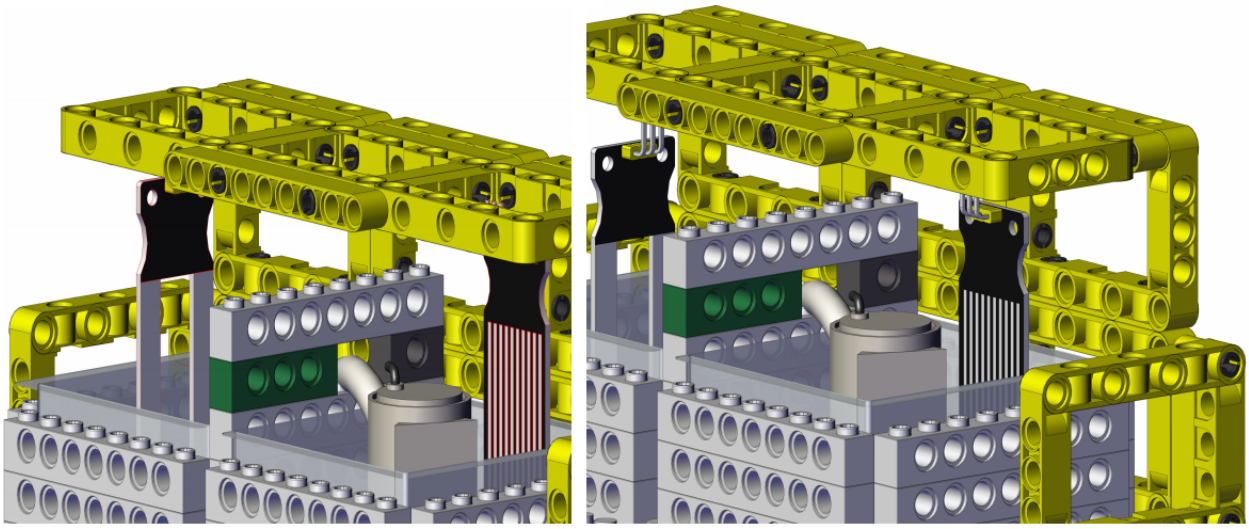
Process 11



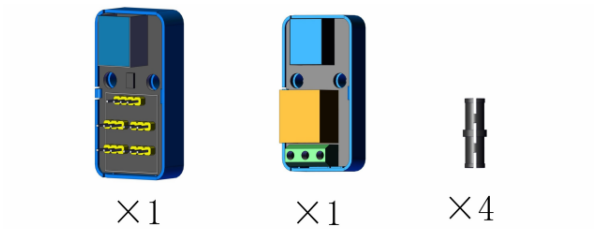


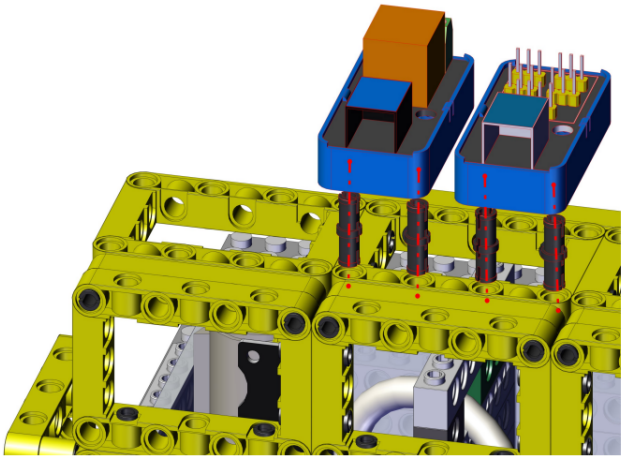


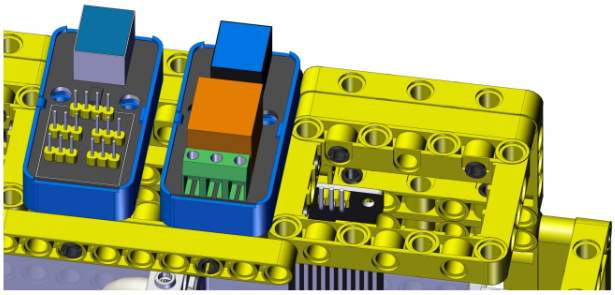
Process 12



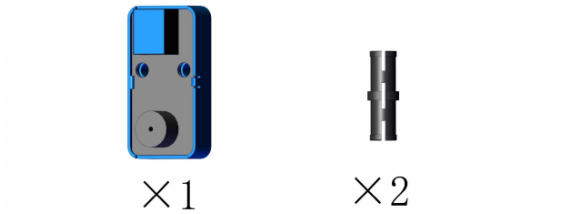
Process 13

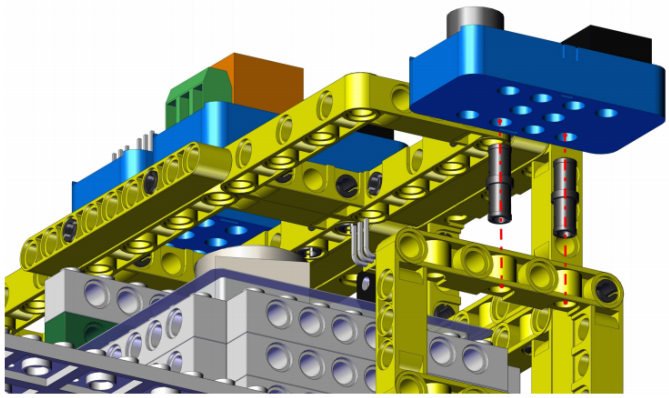


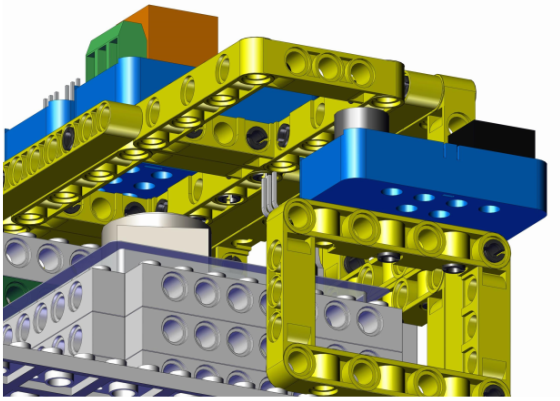




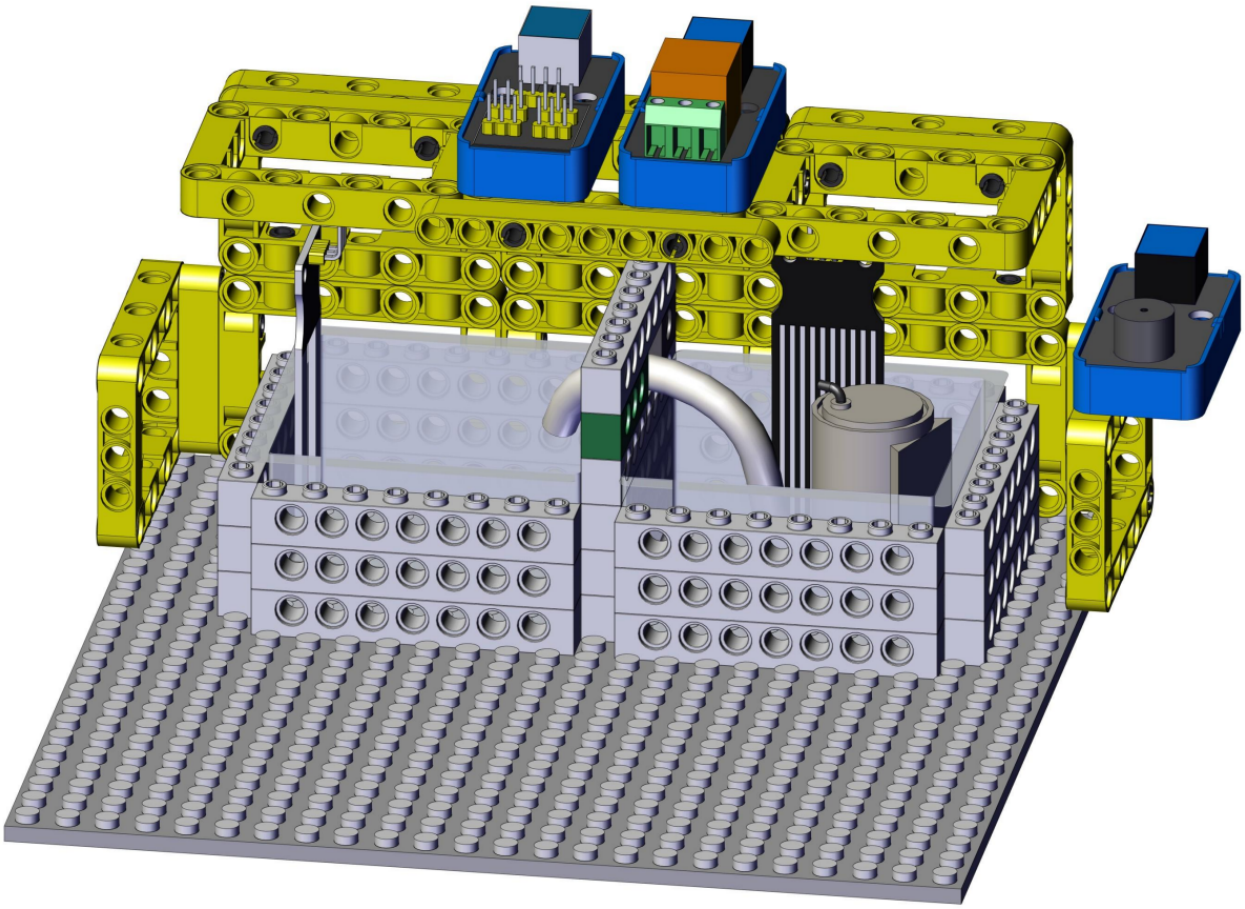
Process 14







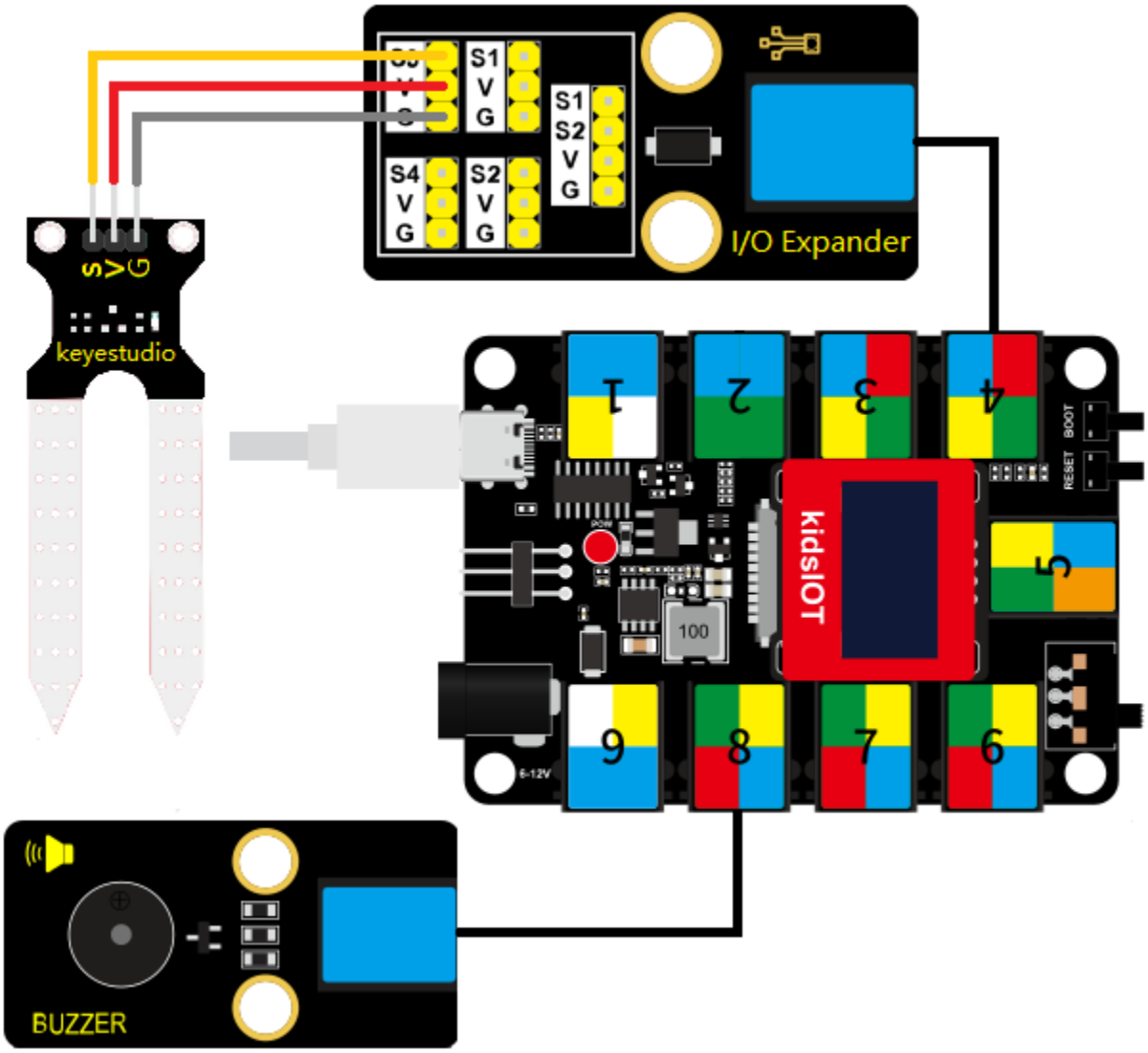
Complete



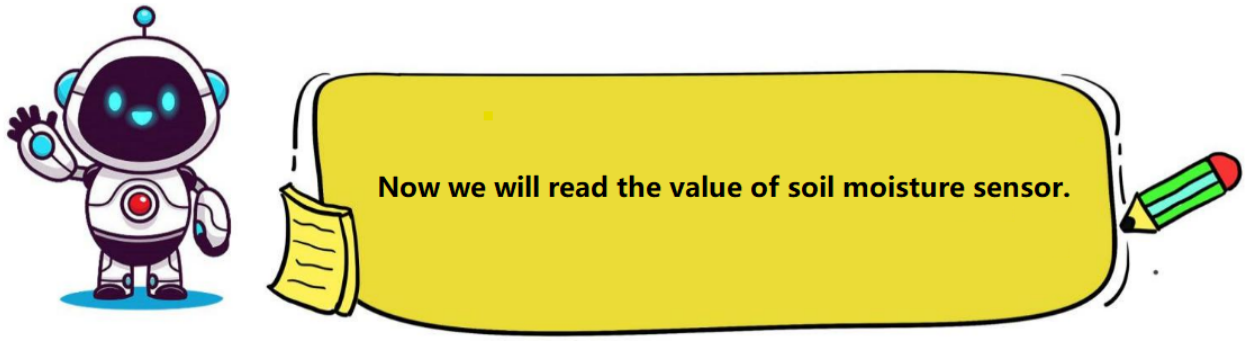
4. Wiring Diagram

Module	GPIO Shield	kidsIOT Mainboard
Soil Moisture Sensor	G→GV→VS→S3	No.4 portcontrol pin of S3 is io27
Passive Buzzer		No.8 portcontrol pin is io5

Connect the kidsIOT mainboard to your computer via USB cable.

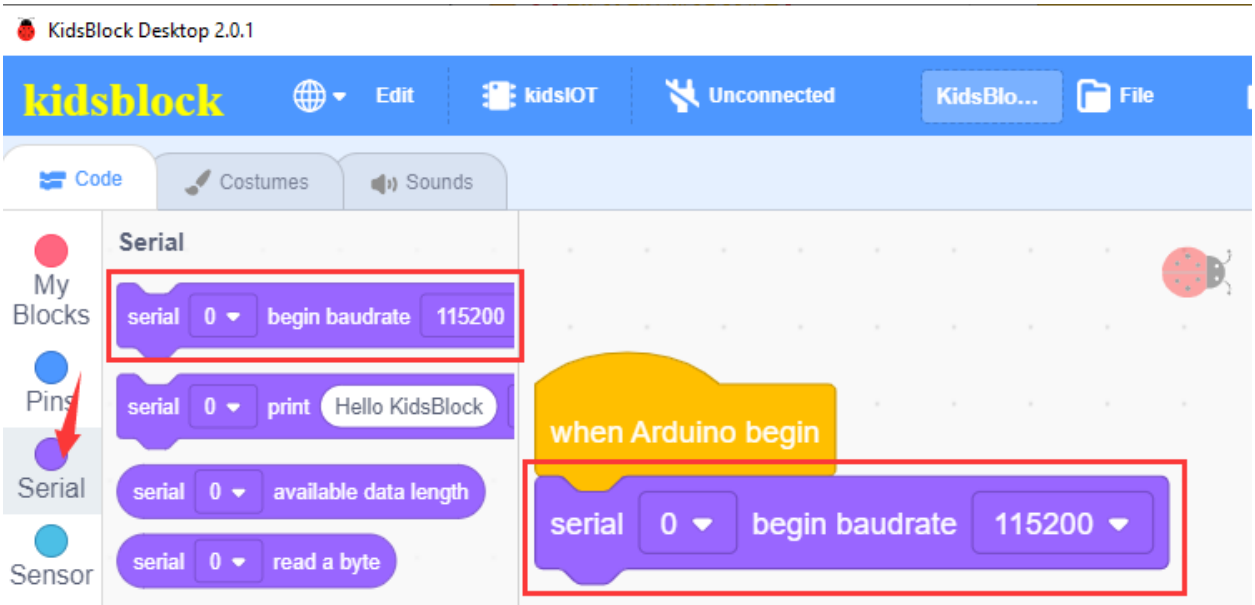


5. Read the value of soil moisture sensor

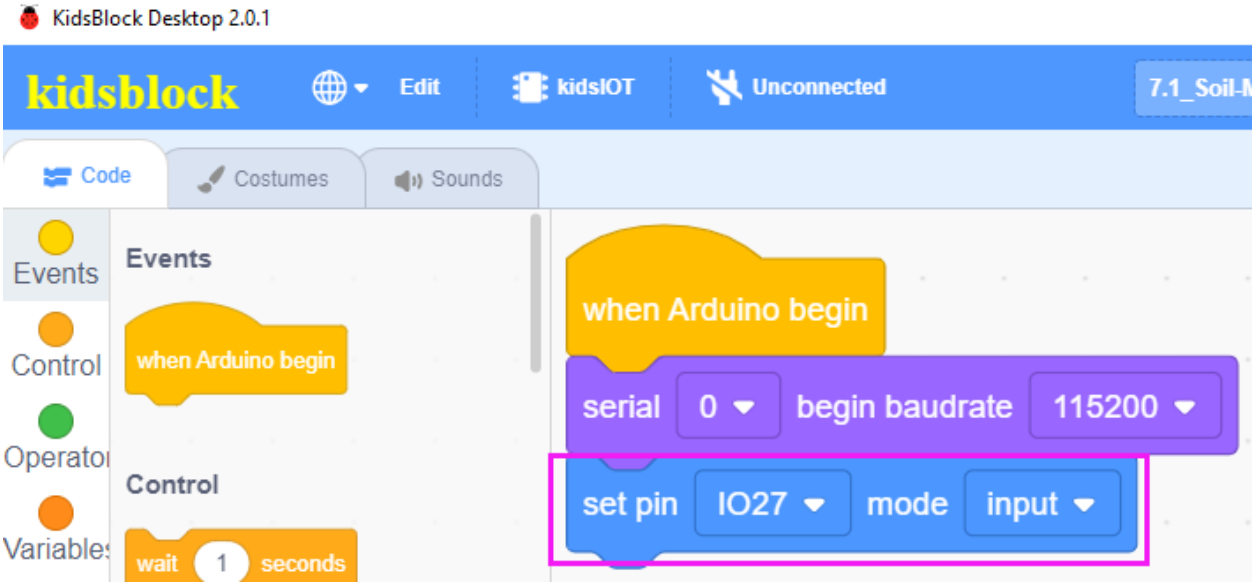


(1). Write the Program

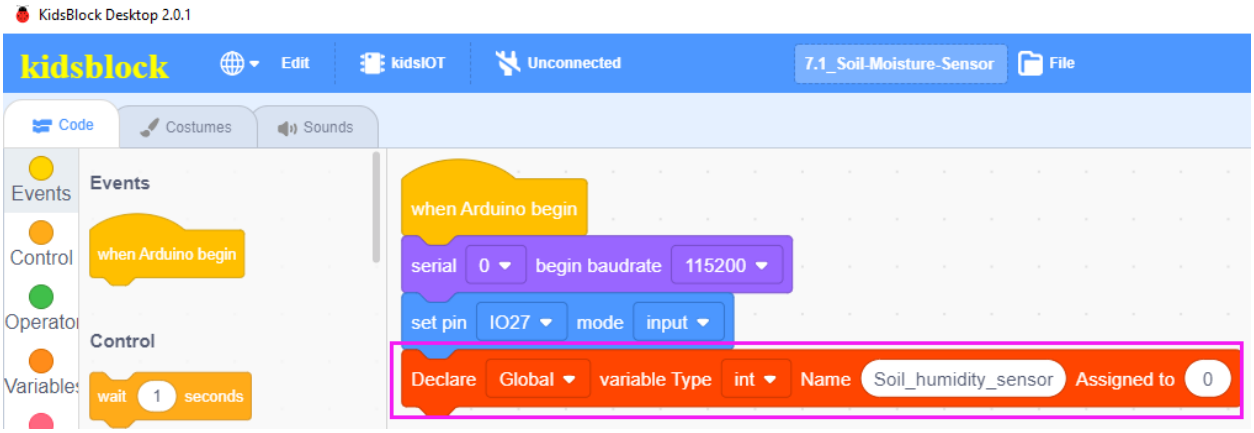
Set the baud rate to 15200.



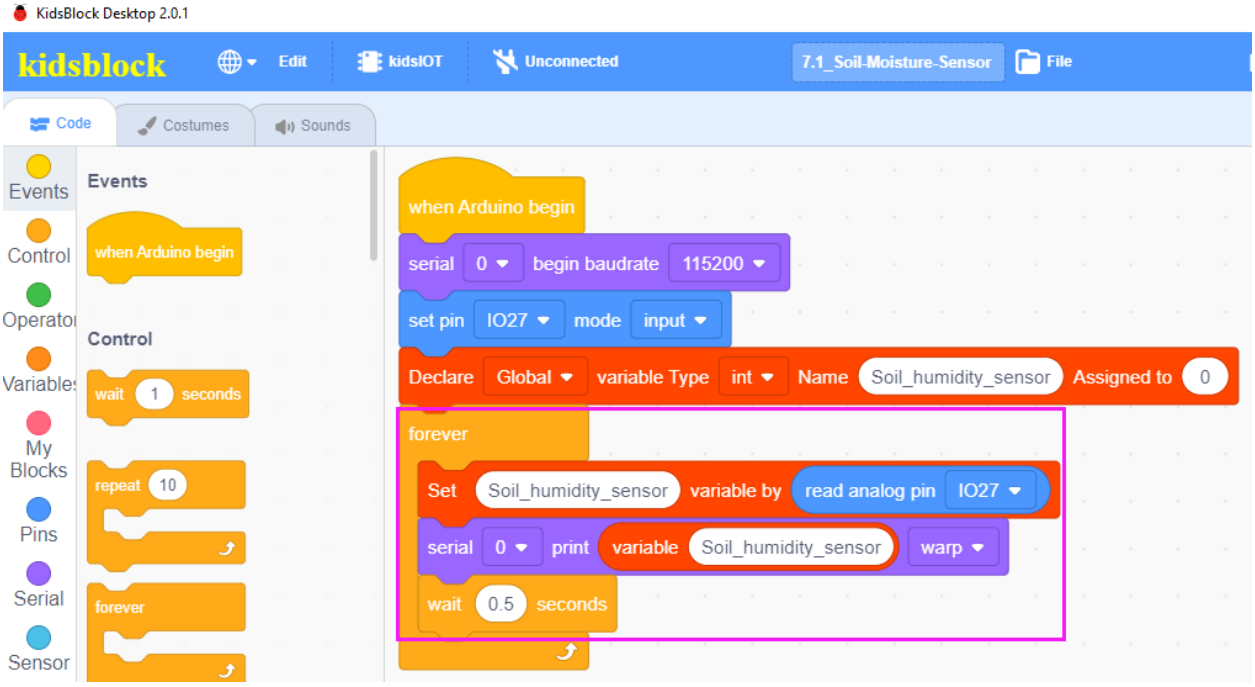
Set the pin IO27 connected to the soil moisture sensor to "**input**" mode.



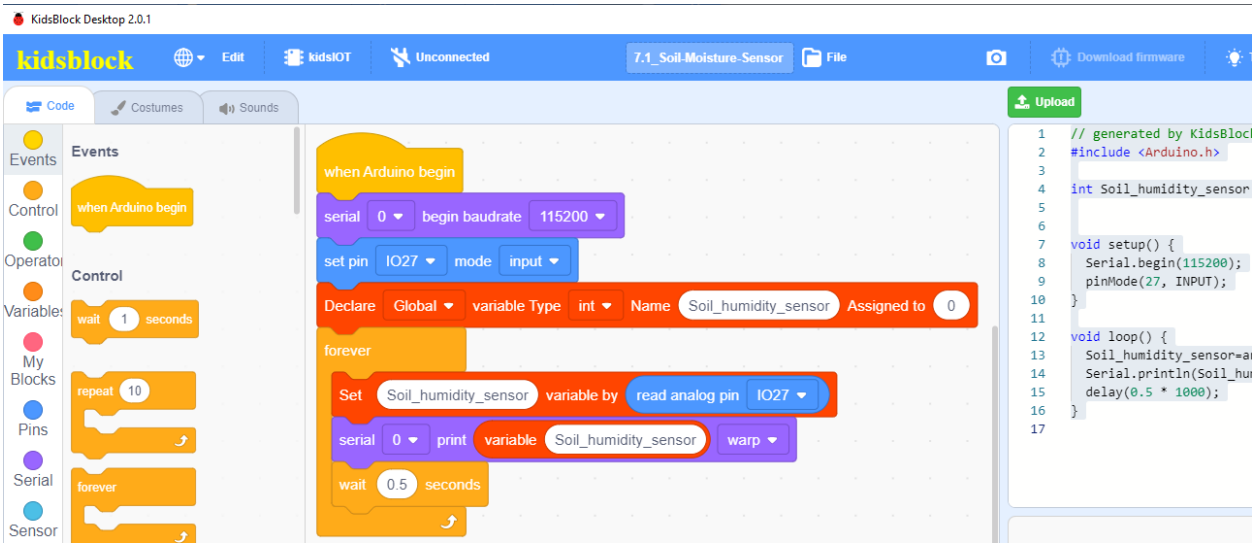
Define a "Soil_humidity_sensor" global variable to store the analog value of the soil moisture sensor.



Store the read analog value of the soil moisture sensor in the "Soil_humidity_sensor" variable and print it on the serial port.



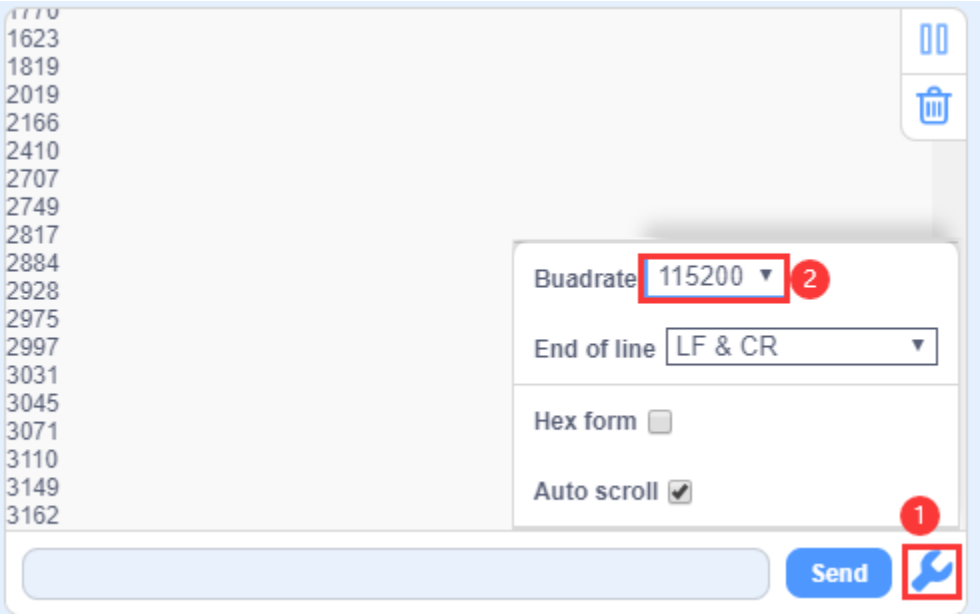
Complete Program



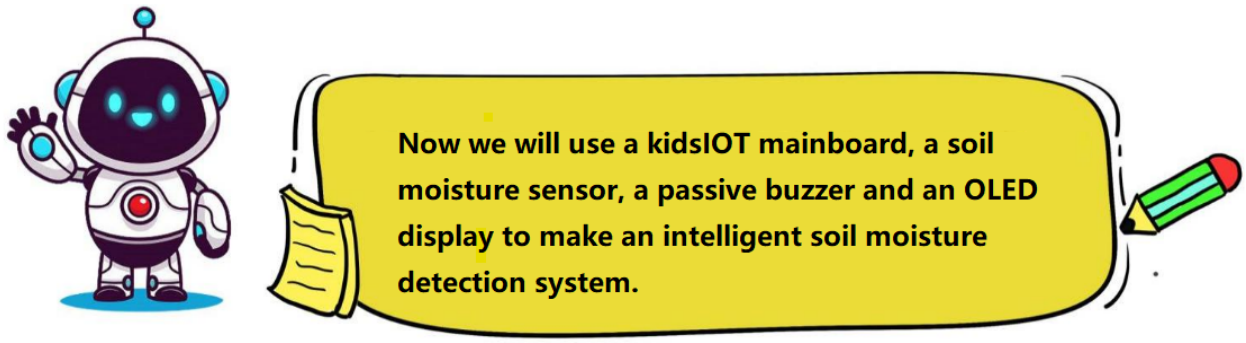
(2). Test Result

Click  to upload the above complete code to the kidsIOT board. After powering up via the USB cable, click  in the serial monitor and set the baud rate to 15200.

Insert the soil moisture sensor into the soil (or touch the sensor with wet hands), then the serial monitor will print the analog value of the soil moisture sensor (range: 0~4095). The greater the soil moisture (or the wetter the hand), the greater the value!

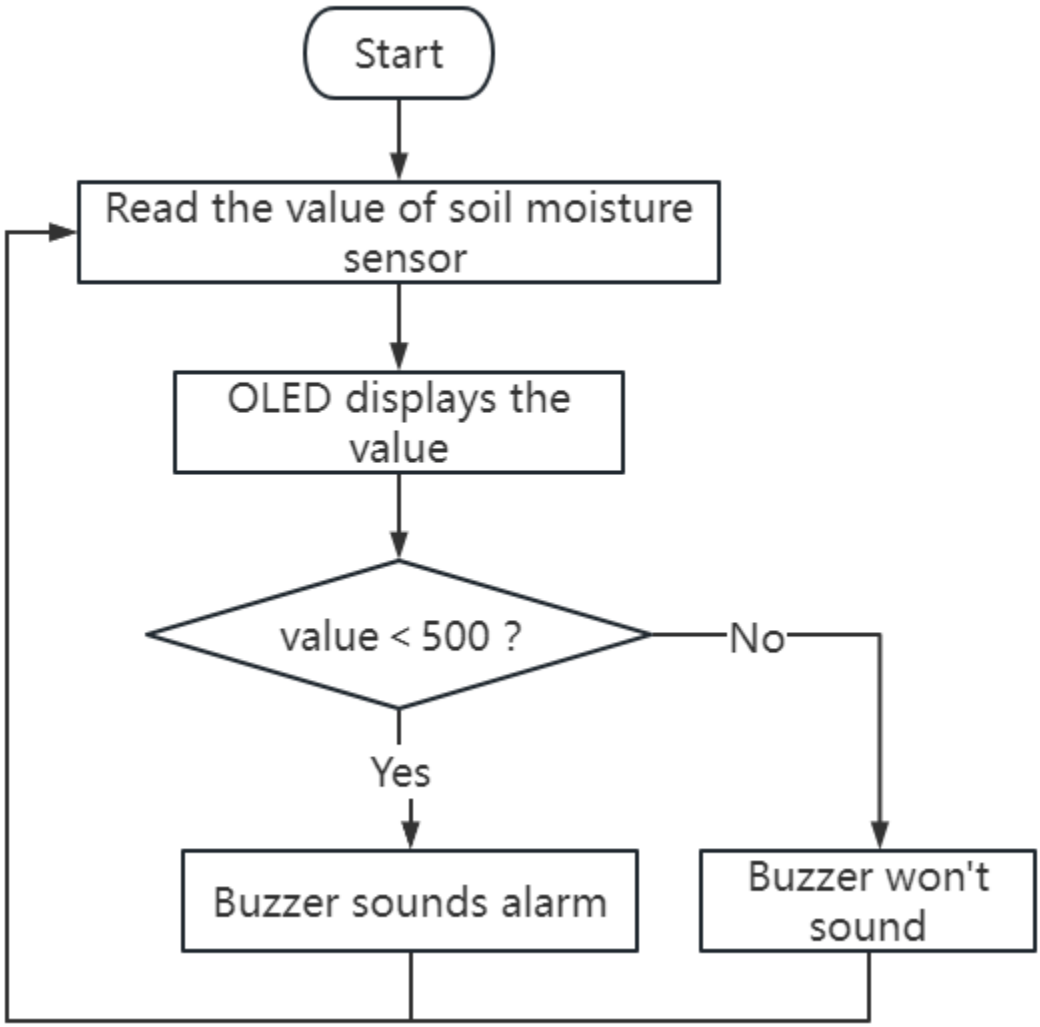


6. Soil Moisture Detection System

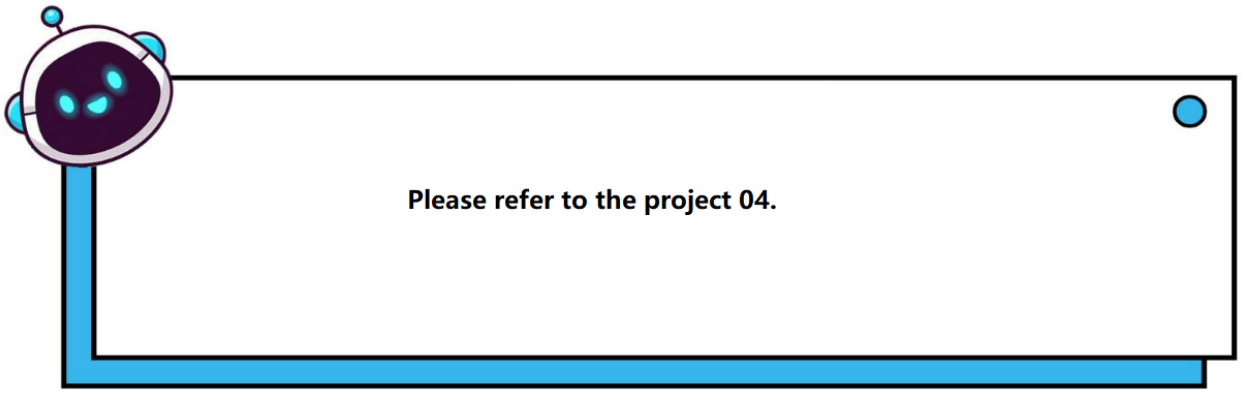


(1). Programming Steps

Step 1Flow Chart

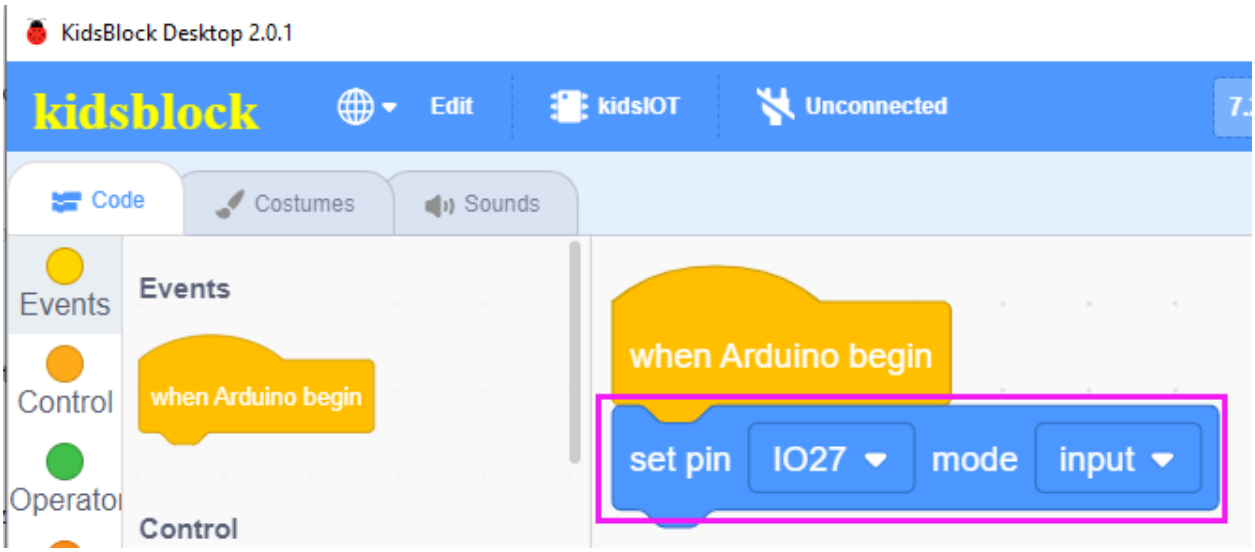


Step 2Add "passive buzzer"

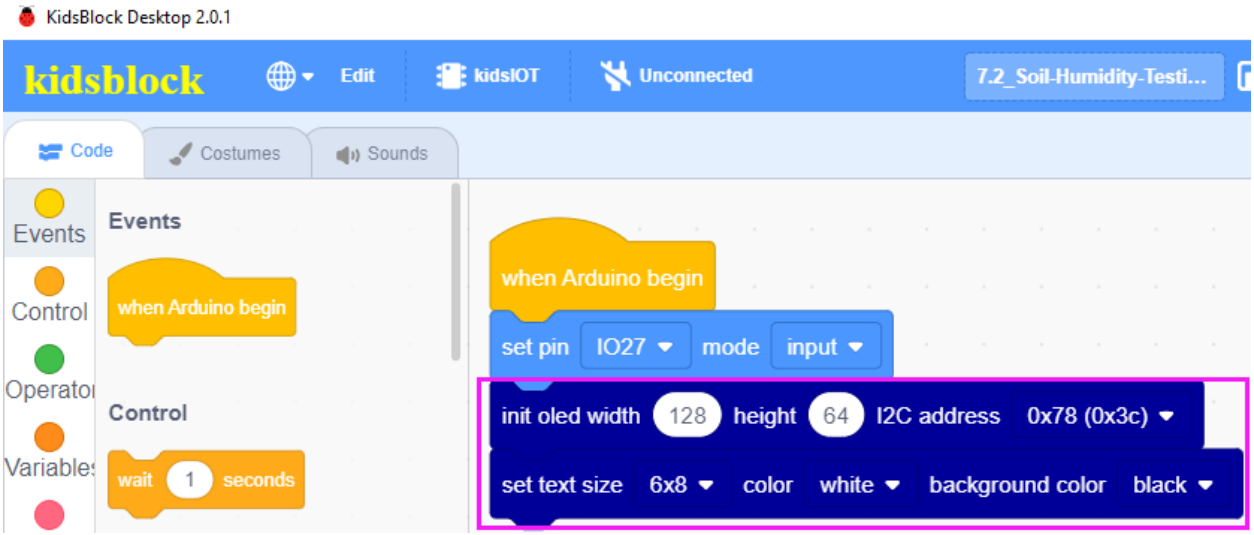


Step 3Write the Program

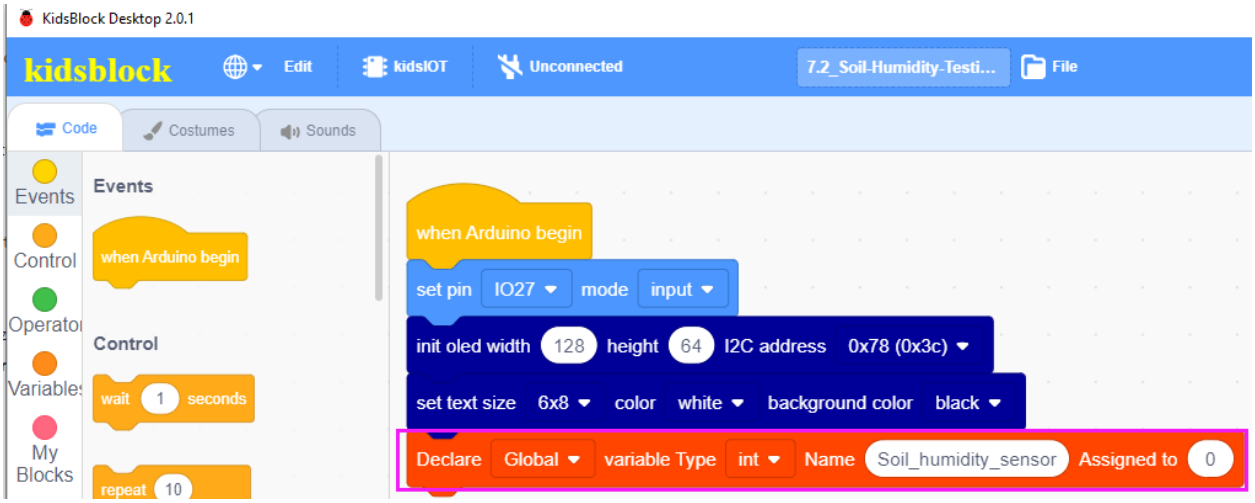
Set the pin IO27 connected to the soil moisture sensor to "**input**" mode.



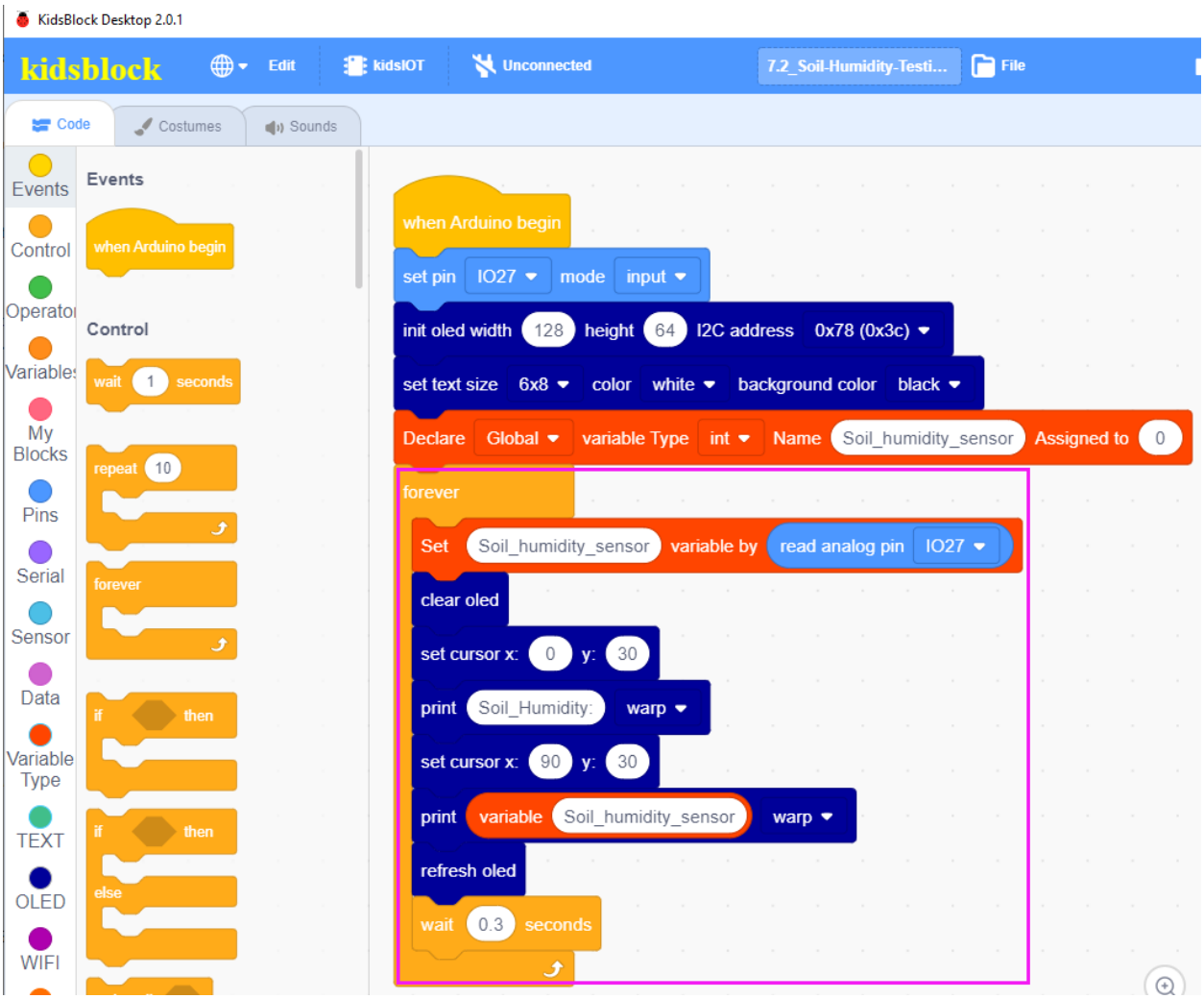
Initialize the width, height, I2C address, text size and color as well as background color of the OLED display.



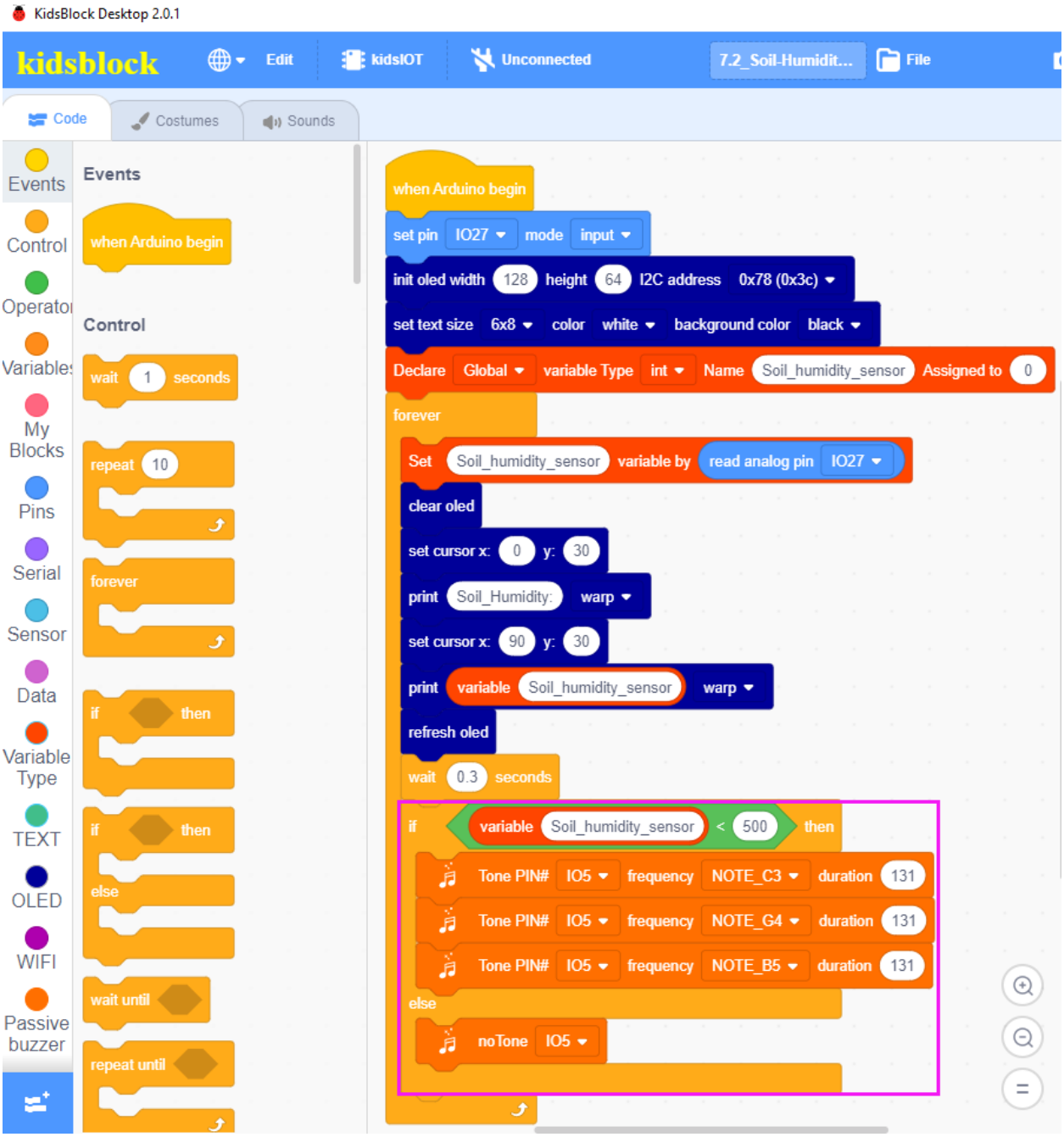
Define a "Soil_humidity_sensor" global variable to store the analog value of the soil moisture sensor.



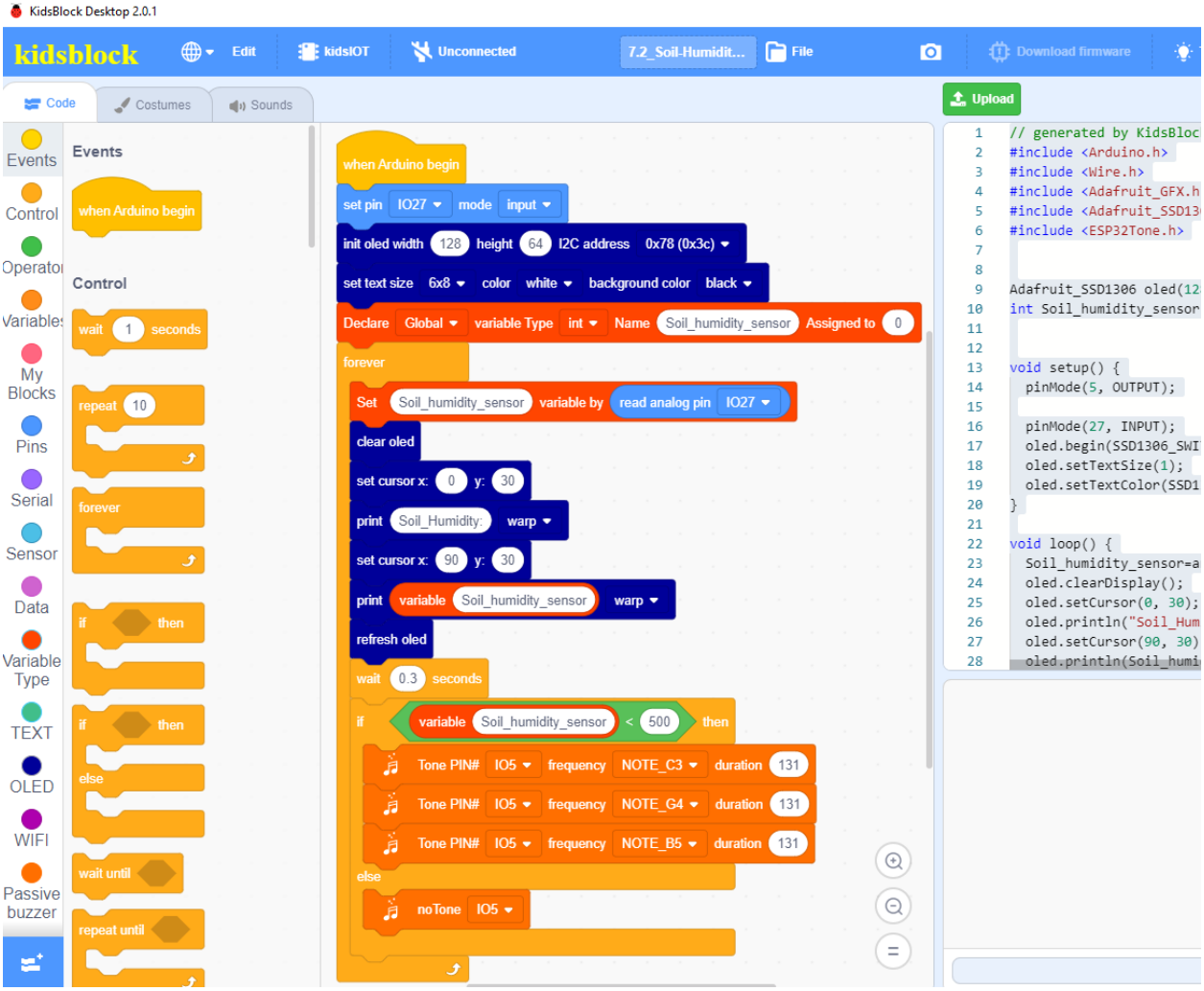
Store the read analog value of the soil moisture sensor in the "Soil_humidity_sensor" variable and display it on the OLED.



Determine the analog value of the sensor. If it is less than 500, the buzzer will sound an alarm; otherwise, the buzzer will not sound.



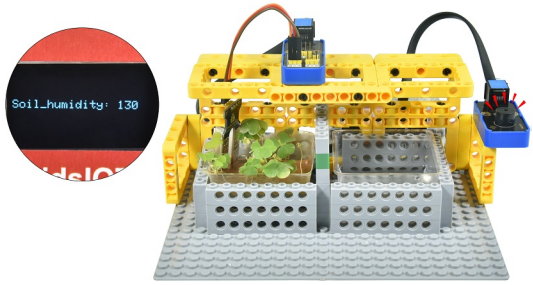
Complete Program



(2). Test Result

Click 

to upload the above complete code to the kidsIOT mainboard. After powering up via the USB cable, insert the soil moisture sensor into the soil (or touch the sensor with wet hands), when the sensor detects that the moisture of the soil (or the moisture of your hands) is less than the set threshold, the buzzer will sound an alarm.



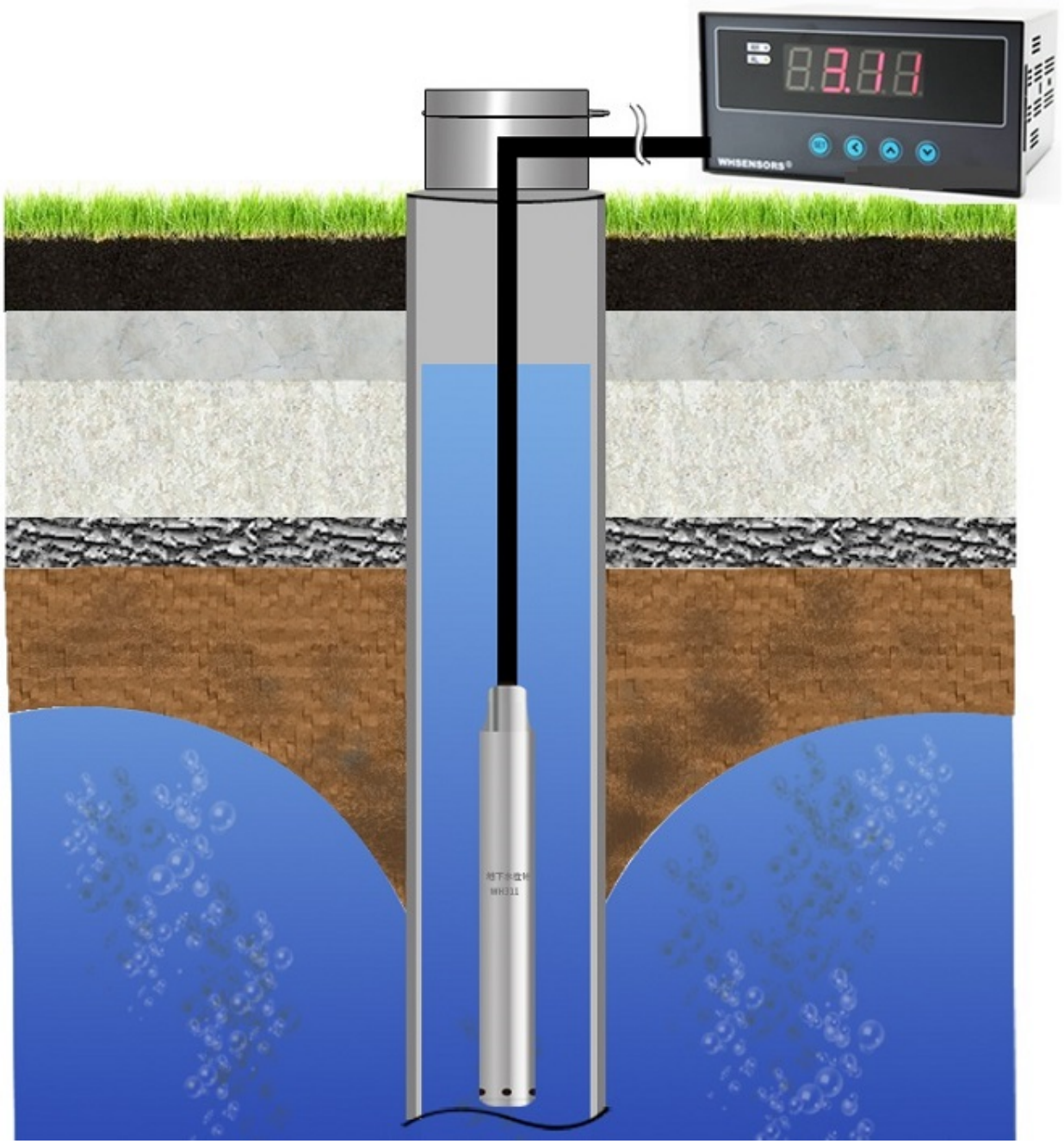
7. Common Problems

Q: Is the sensor waterproof?

A: The detection area of the soil moisture sensor is waterproof. Exceeding the detection area will cause a short circuit.

4.3.8 Project 08Water Level Detection System

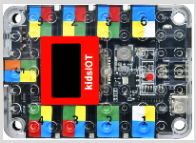
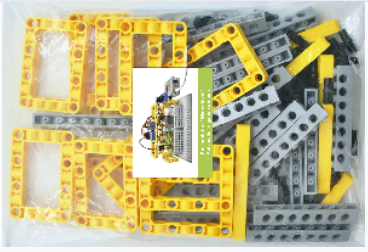
Note: Do not allow water to overflow from sinks and soil troughs when using the device. Sprinkling water on other sensors will cause a short circuit and device failure. Sprinkling water on batteries will cause heating and explosion. Please be careful when using the device, especially when used by young children, it must be under the supervision of parents. To ensure safe operation of the device, please follow relevant usage guidelines and safety regulations.

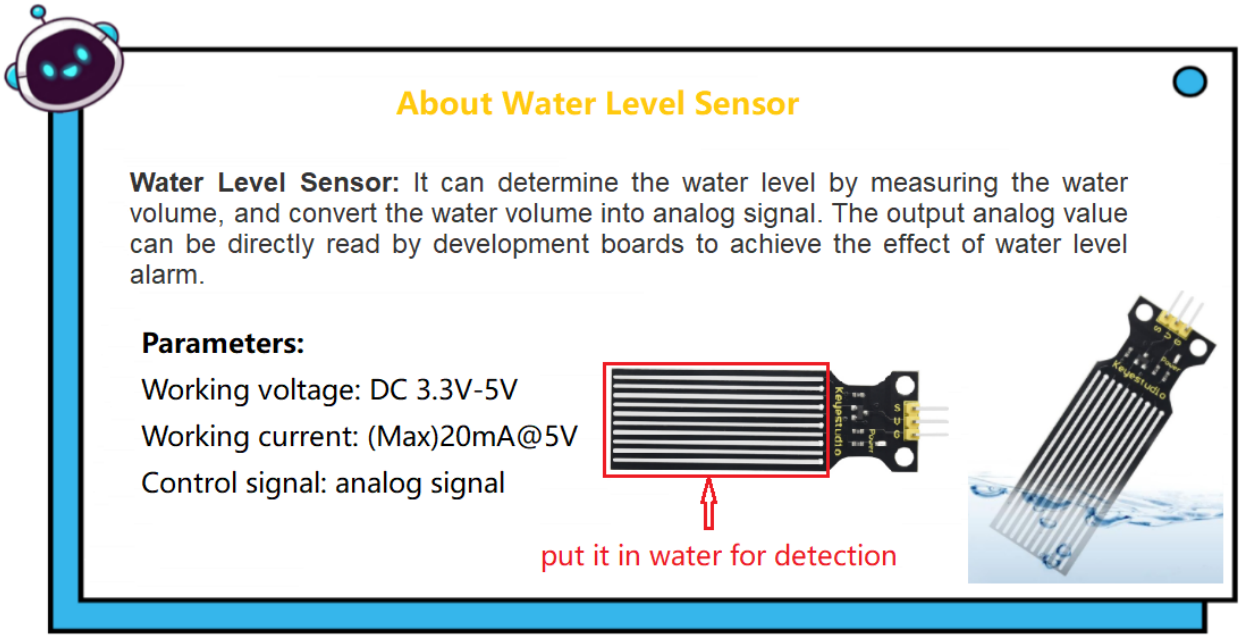

WHSENSORS

1. Description

This project introduces how to use a kidsIOT mainboard, a water level sensor, a passive buzzer and an OLED display to make an intelligent automatic water level detection system. It is able to monitor water level changes, detect problems in time and take measures to avoid disasters, and is widely used in water conservancy projects, urban drainage as well as environment monitoring.

2. Components

				
kidsIOT Mainboard×1	Water Level Sensor×1	Passive Buzzer×1	GPIO Shield×1	Wire×2
				
USB Cable×1	Soil Moisture/Water Level/Automatic Irrigation System LEGO Pieces×1	F-F DuPont Wires	Sink×1	



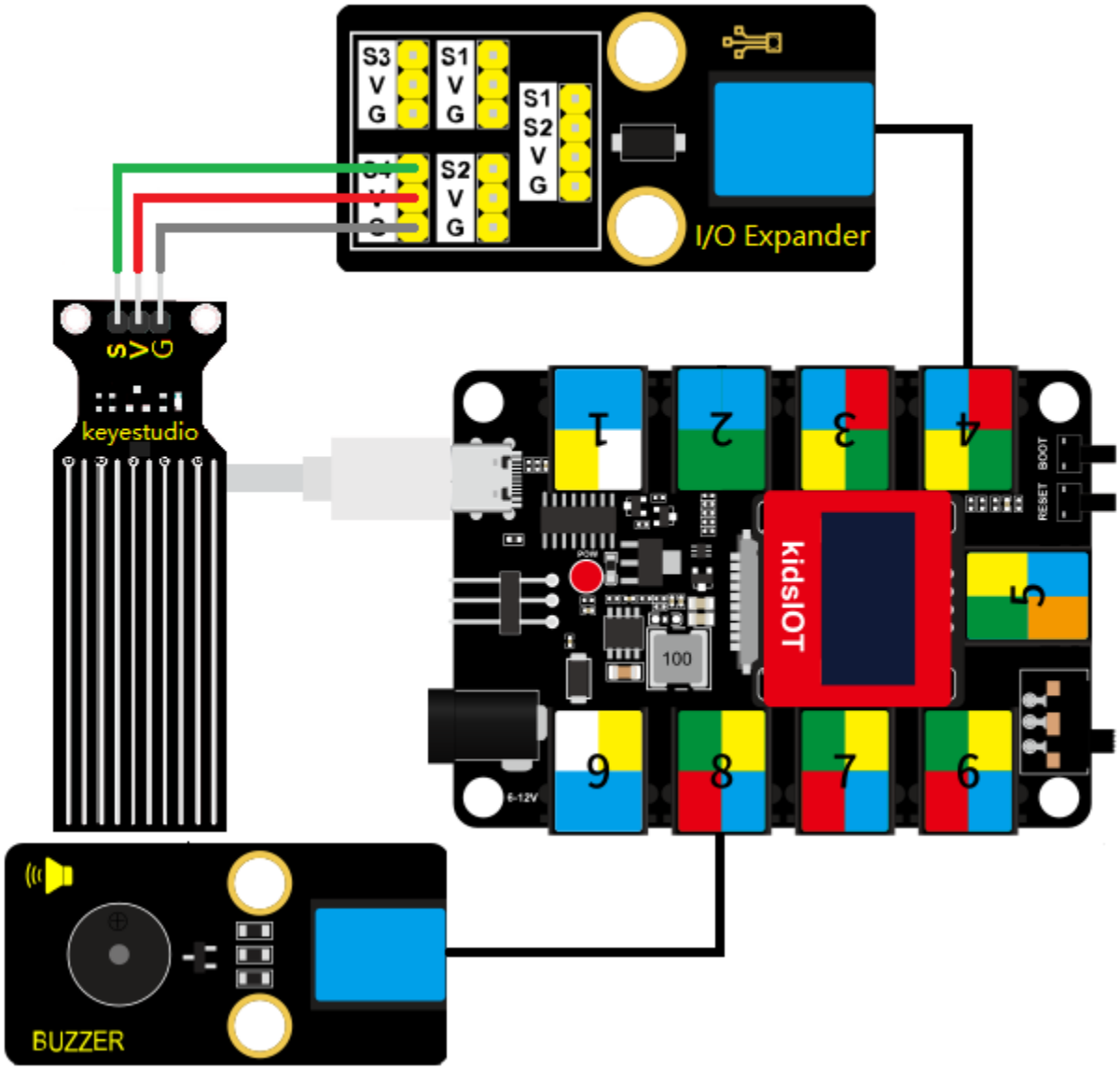
3. Assembly Steps

It shares the same structural shape with **Project 07**. If the assembly parts of **Project 07** have finished, there is no need to assemble it again.

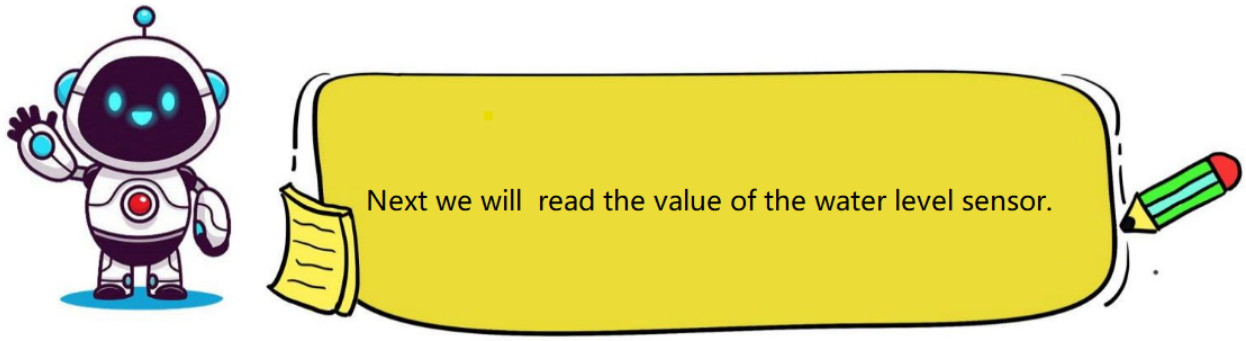
4. Wiring Diagram

Module	GPIO Shield	kidsIOT Mainboard
Water Level Sensor	G→GV→VS→S4	No.4 portcontrol pin of S4 is io39
Passive Buzzer		No.8 portcontrol pin is io5

Connect the kidsIOT mainboard to your computer via USB cable.

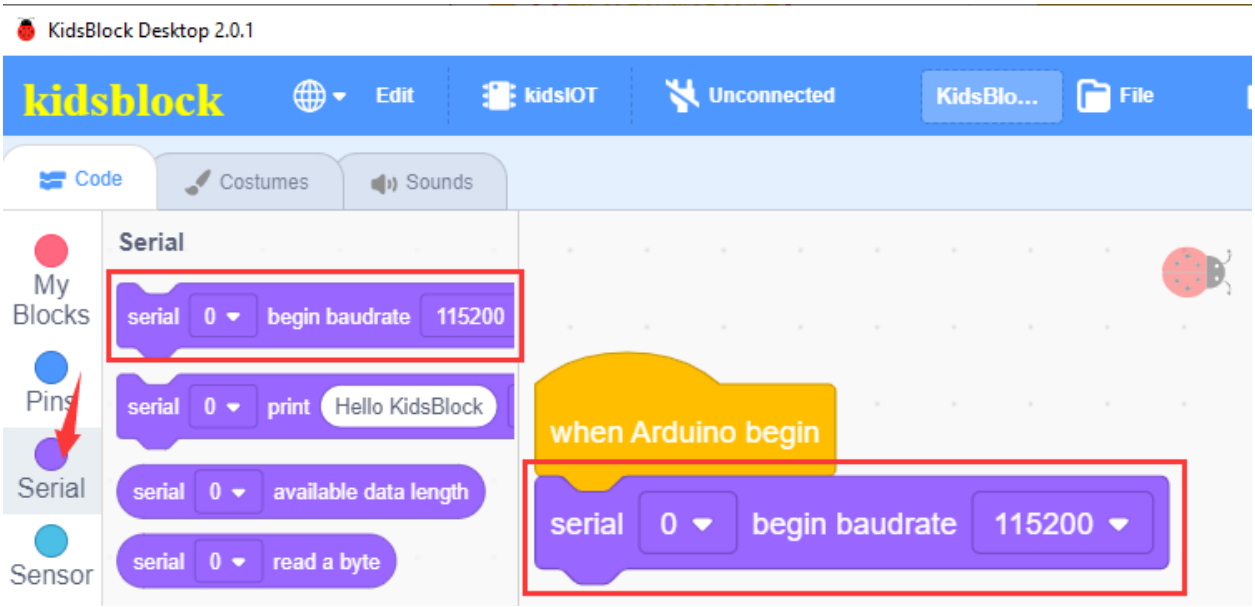


5. Read the value of water level sensor

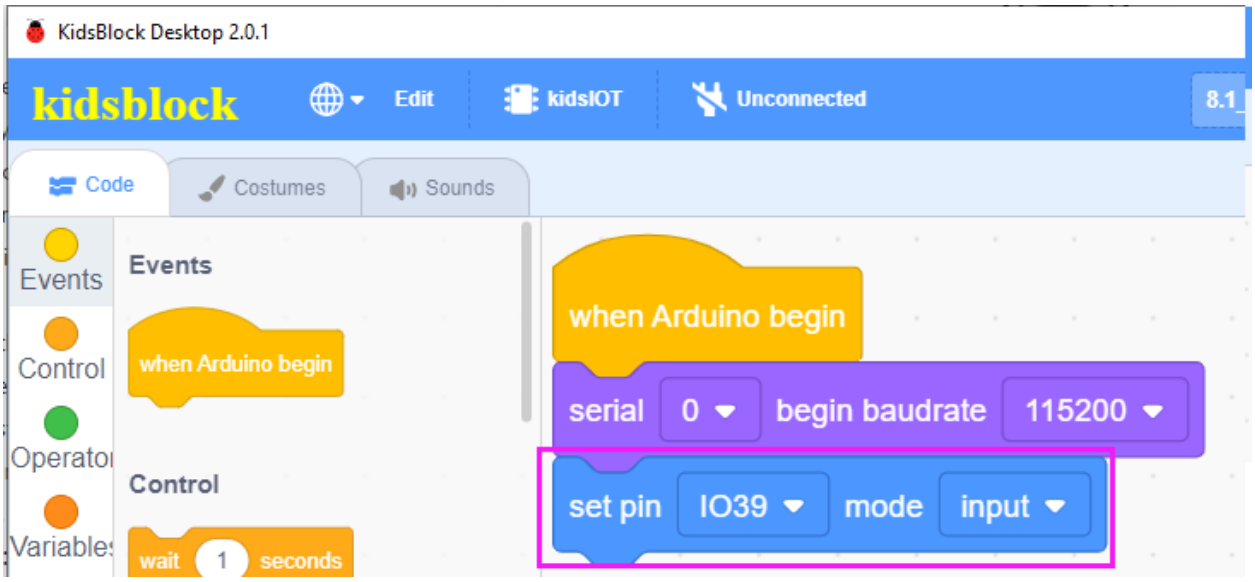


(1). Write the Program

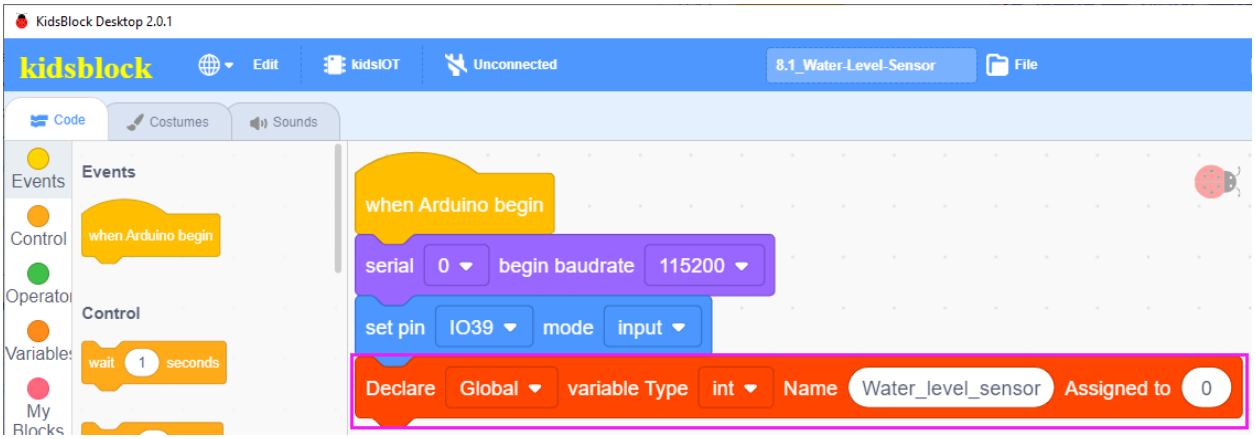
Set the baud rate to 15200.



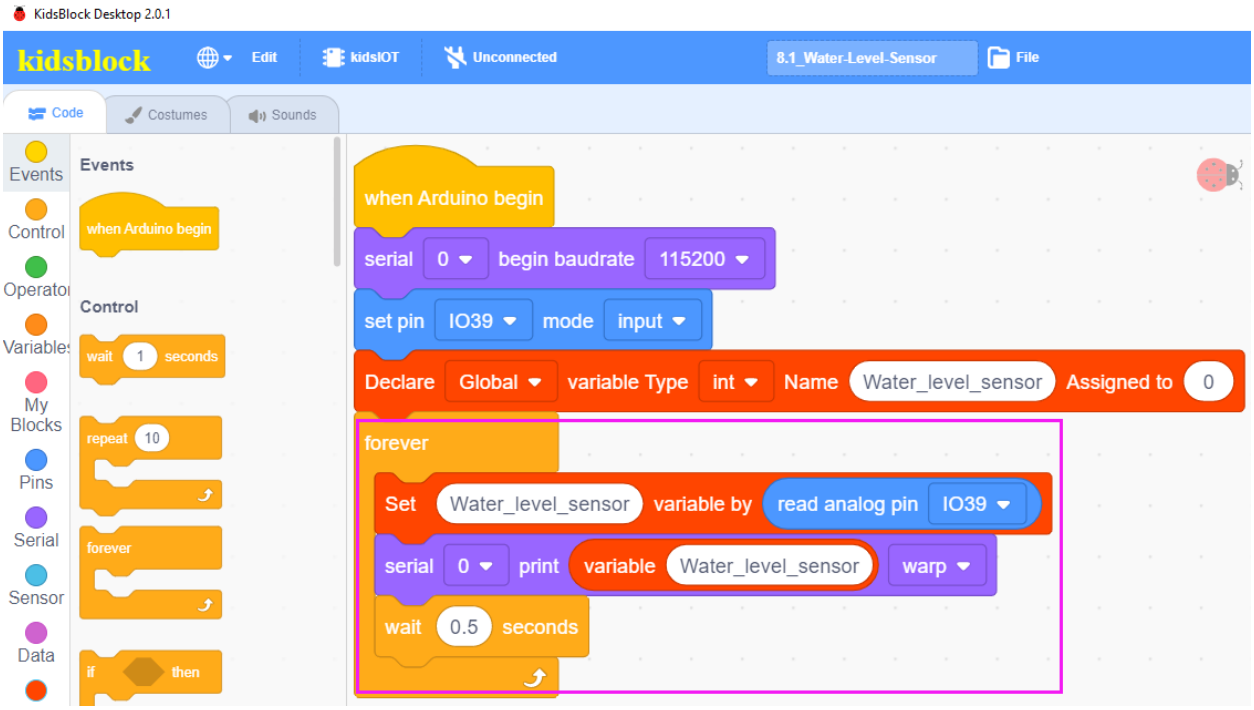
Set the pin IO39 connected to the water level sensor to "**input**" mode.



Define a "Water_level_sensor" global variable to store the analog value of the water level sensor.



Store the read analog value of the water level sensor in the "Water_level_sensor" variable and print it on the serial port.

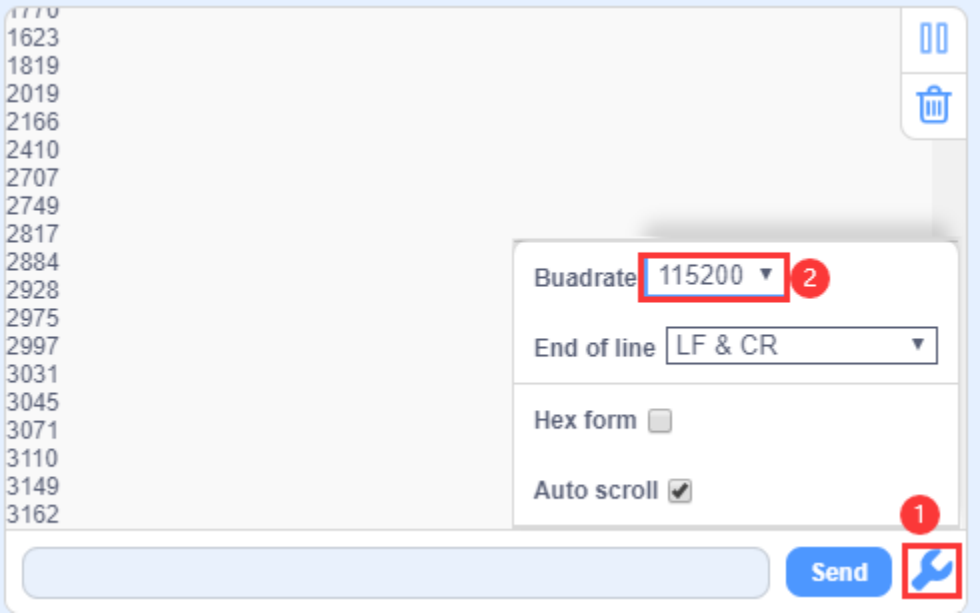


Complete Program

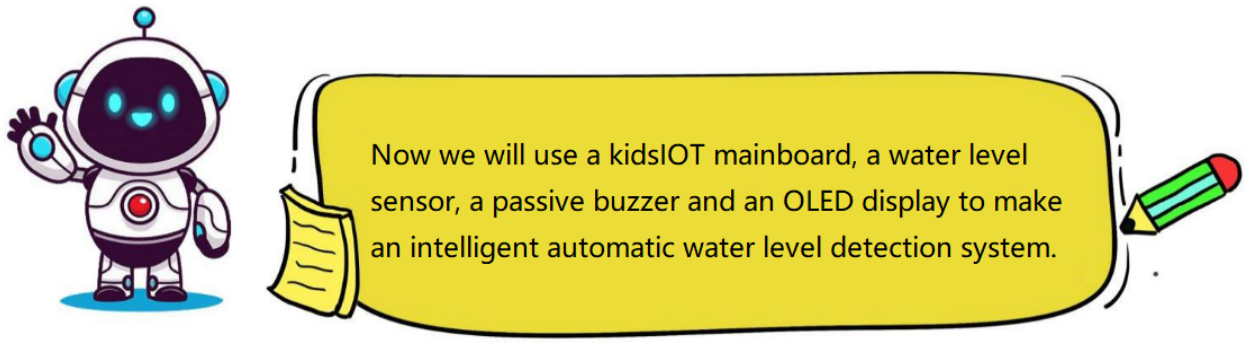
(2). Test Result

Click  to upload the above complete code to the kidsIOT mainboard. After powering up via the USB cable, click  in the serial monitor and set the baud rate to 15200.

Insert the water level sensor in the water (or touch the sensor with wet hands), then the serial monitor will print the analog value of the sensor (range: 0~4095). The greater the water level, the greater the value!

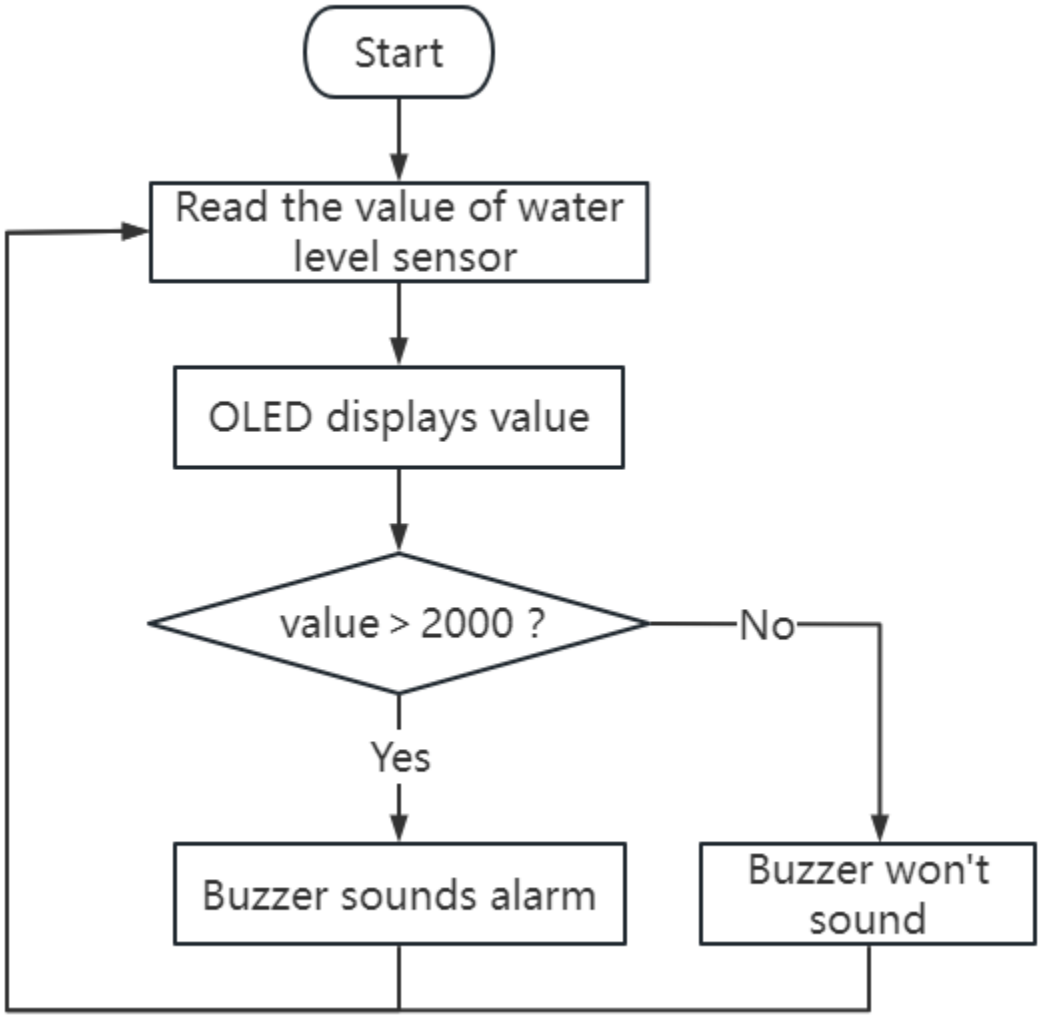


6. Water Level Detection System

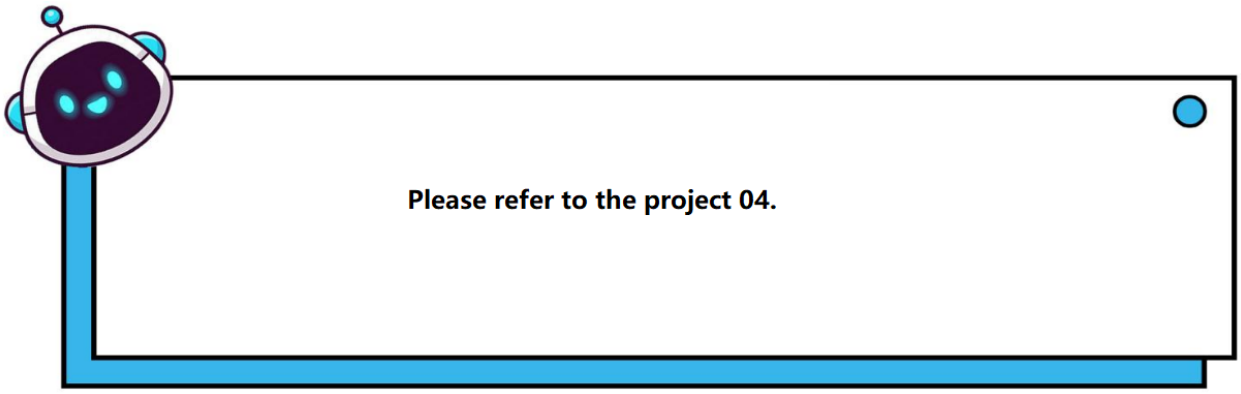


(1). Programming Steps

Step 1Flow Chart

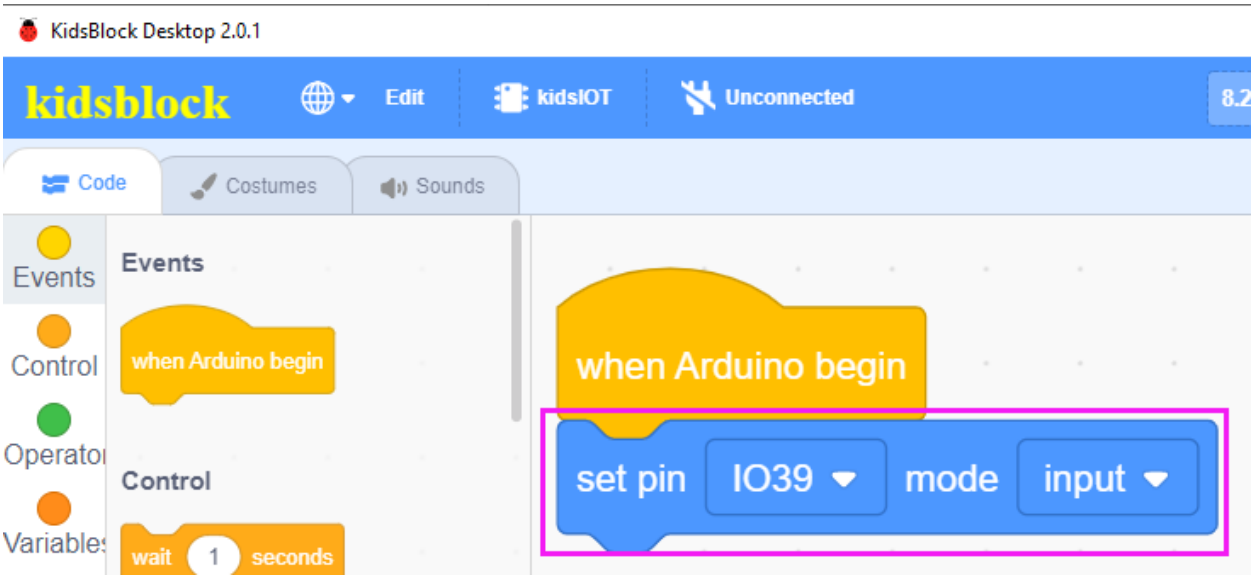


Step 2Add "passive buzzer"

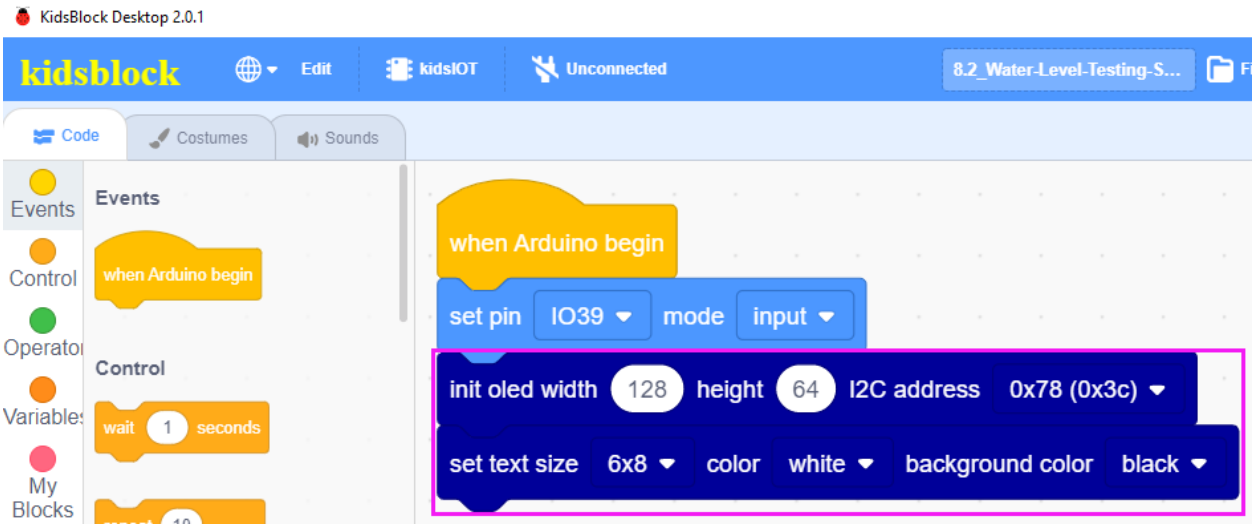


Step 3Write the Program

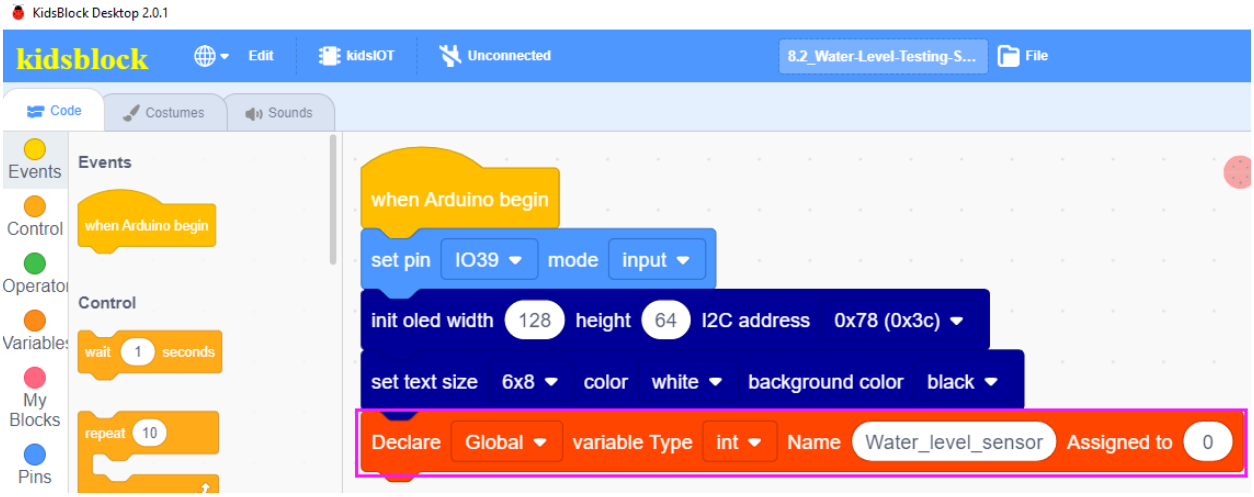
Set the pin IO39 connected to the water level sensor to "**input**" mode.



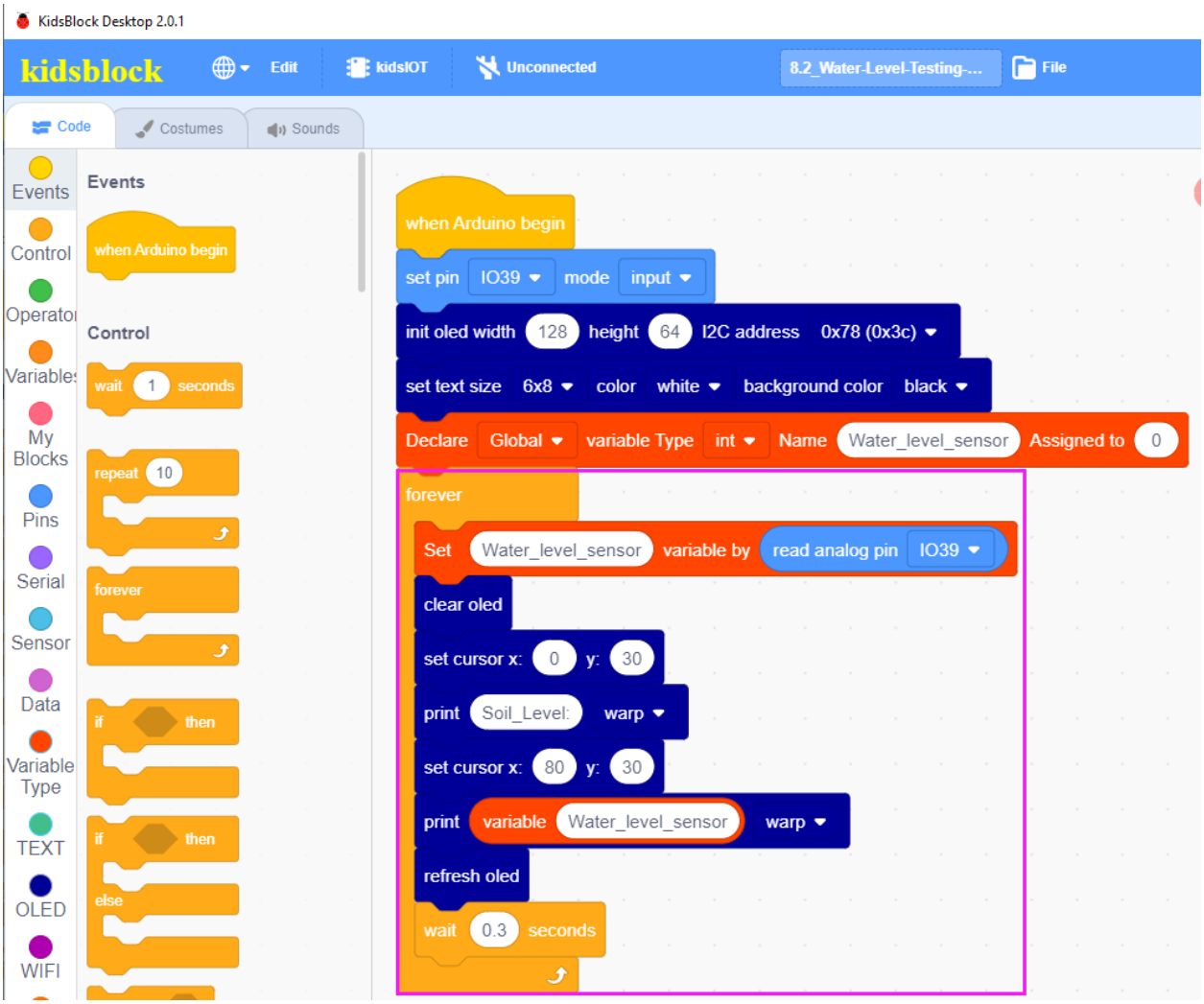
Initialize the width, height, I2C address, text size and color as well as background color of the OLED display.



Define a “Water_level_sensor” global variable to store the analog value of the water level sensor.

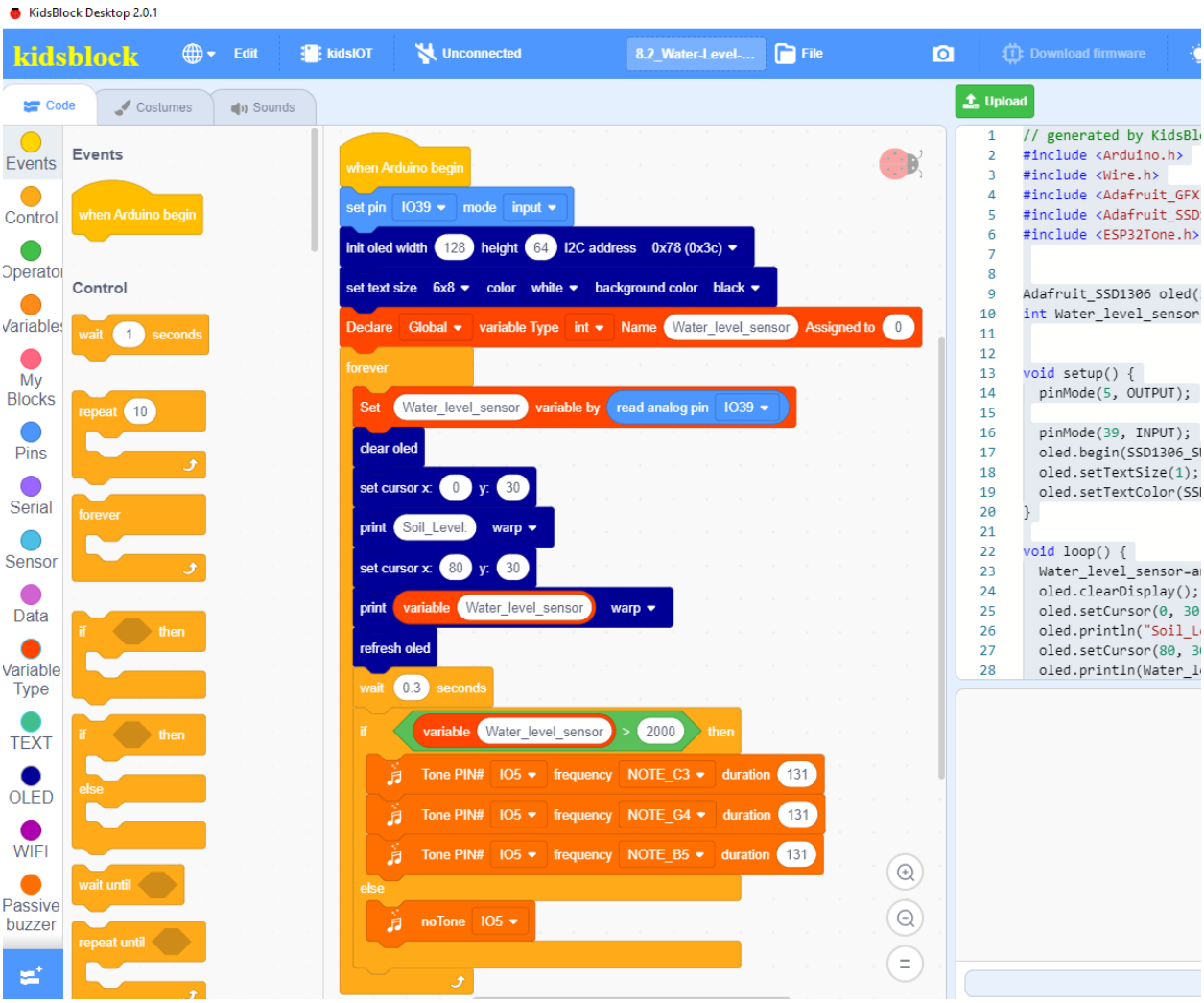


Store the read analog value of the sensor in the“Water_level_sensor” variable and display it on the OLED.



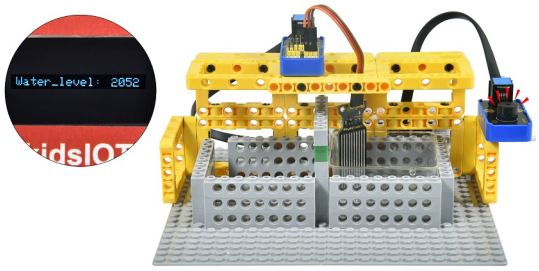
Determine the analog value of the sensor. If it is greater than 2000, the buzzer will sound an alarm; otherwise, the buzzer will not sound.

Complete Program



(2). Test Result

Click  to upload the above complete code to the kidsIOT mainboard. After powering up via the USB cable, insert the water level sensor in the water (or touch the sensor with wet hands), when the sensor detects that the water level (or the moisture of your hands) is greater than the set threshold, the buzzer will sound an alarm.



7. Common Problems

Q: Is the sensor waterproof?

A: The detection area of the sensor is waterproof. Exceeding the detection area will cause a short circuit.

4.3.9 Project 09Automatic Irrigation System

Note: Do not allow water to overflow from sinks and soil troughs when using the device. Sprinkling water on other sensors will cause a short circuit and device failure. Sprinkling water on batteries will cause heating and explosion. Please be careful when using the device, especially when used by young children, it must be under the supervision of parents. To ensure safe operation of the device, please follow relevant usage guidelines and safety regulations.

1. Description

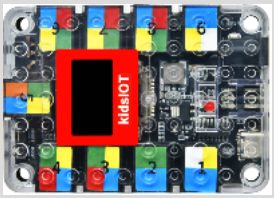
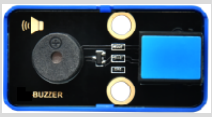
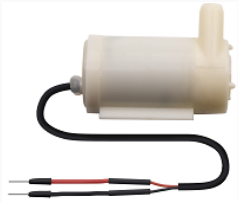
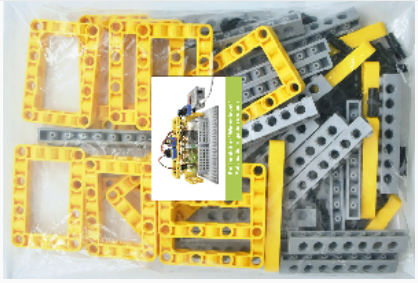
This project introduces how to use a kidsIOT mainboard, a soil moisture sensor, a water level sensor, a passive buzzer, an OLED display, a relay and a water pump to build an automatic irrigation system.

We will read the analog values of the soil moisture sensor and water level sensor by writing code to control the relay and the water pump.

When the soil is too dry, the relay will be turned on to control the water pump to irrigate the plants, when too low, the water pump will not be started and the buzzer will alarm.

At the same time, the OLED display will display the dryness of the soil and the water level, thus realizing automated plant watering and water level control, improving production efficiency while reducing the time and energy costs of manual operations.

2. Components

			
kidsIOT Mainboard×1	Water Level Sensor×1	Soil Moisture Sensor×1	Passive Buzzer×1
			
GPIO Shield×1	Relay Module×1	Water Pump×1	Wire×3
			
Battery Holder×1	Slotted Screwdriver×1	M-F DuPont Wires	F-F DuPont Wires
			
Water Pipe×1	Soil Moisture/Water Level/Automatic Irrigation System LEGO Pieces×1	AA BatteryNot provide×6	Sink×2

About Relay Module

Relay Module: It is an "automatic switch" module that uses small current to control large current, which is usually used in automated control circuits. Its three green binding posts NO, COM and NC are used for external circuits.

When the relay module is not connected to the control signal, COM and NC are connected and COM and NO are disconnected. During control, when the signal terminal is high level, the relay is closed, COM and NC are disconnected, and COM and NO are connected. When it is low level, the relay is disconnected, COM and NC are connected and COM and NO are disconnected.

Parameters:

Working voltage: DC 3.3V-5V

Working current: 125mA@3.3V, 75mA@5V

Access capacity: 250VAC/3A, 30VDC/3A

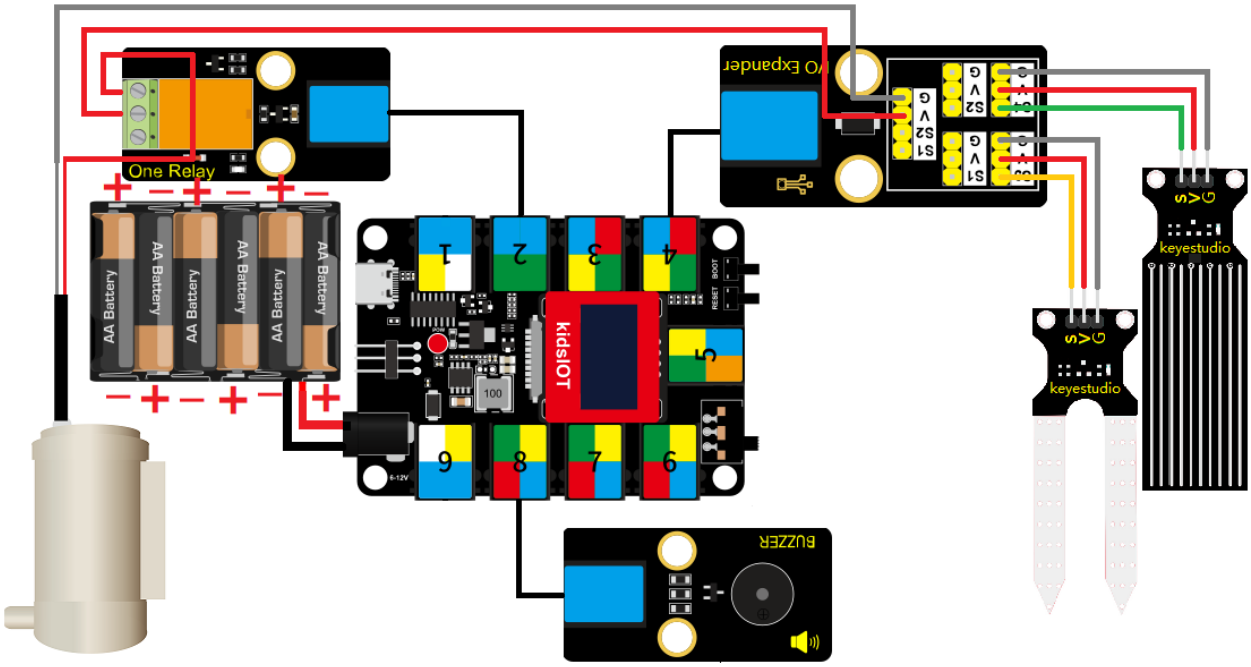
3. Assembly Steps

It shares the same structural shape with **Project 07**. If the assembly parts of **Project 07** have finished, there is no need to assemble it again.

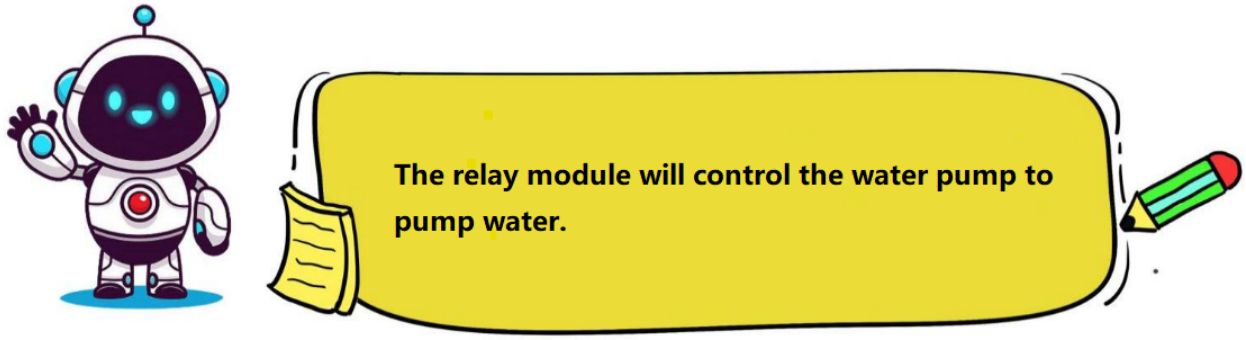
4. Wiring Diagram

Module	GPIO Shield	kidsIOT Mainboard
Soil Moisture Sensor	G→GV→VS→S3	No.4 portcontrol pin of S3 is io27
Water Level Sensor	G→GV→VS→S4	No.4 portcontrol pin of S4 is io39
Passive Buzzer		No.8 portcontrol pin is io5
Relay Module		No.2 portcontrol pin is io2

Connect the kidsIOT mainboard to your computer via USB cable, connect the external power supply and turn the DIP switch on the mainboard to ON end.

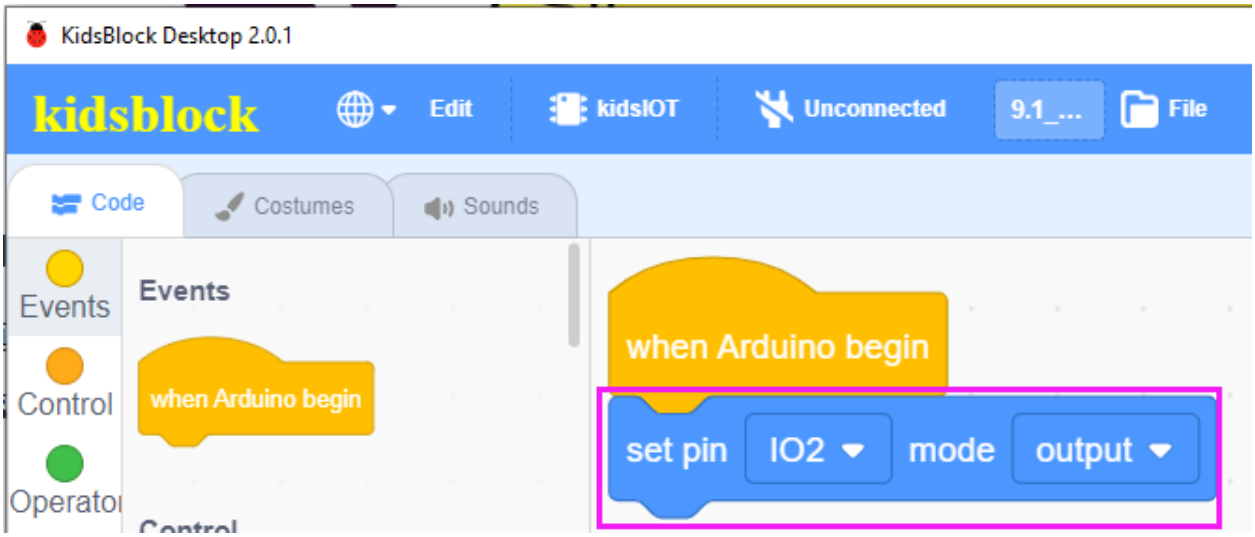


5. Pumping System

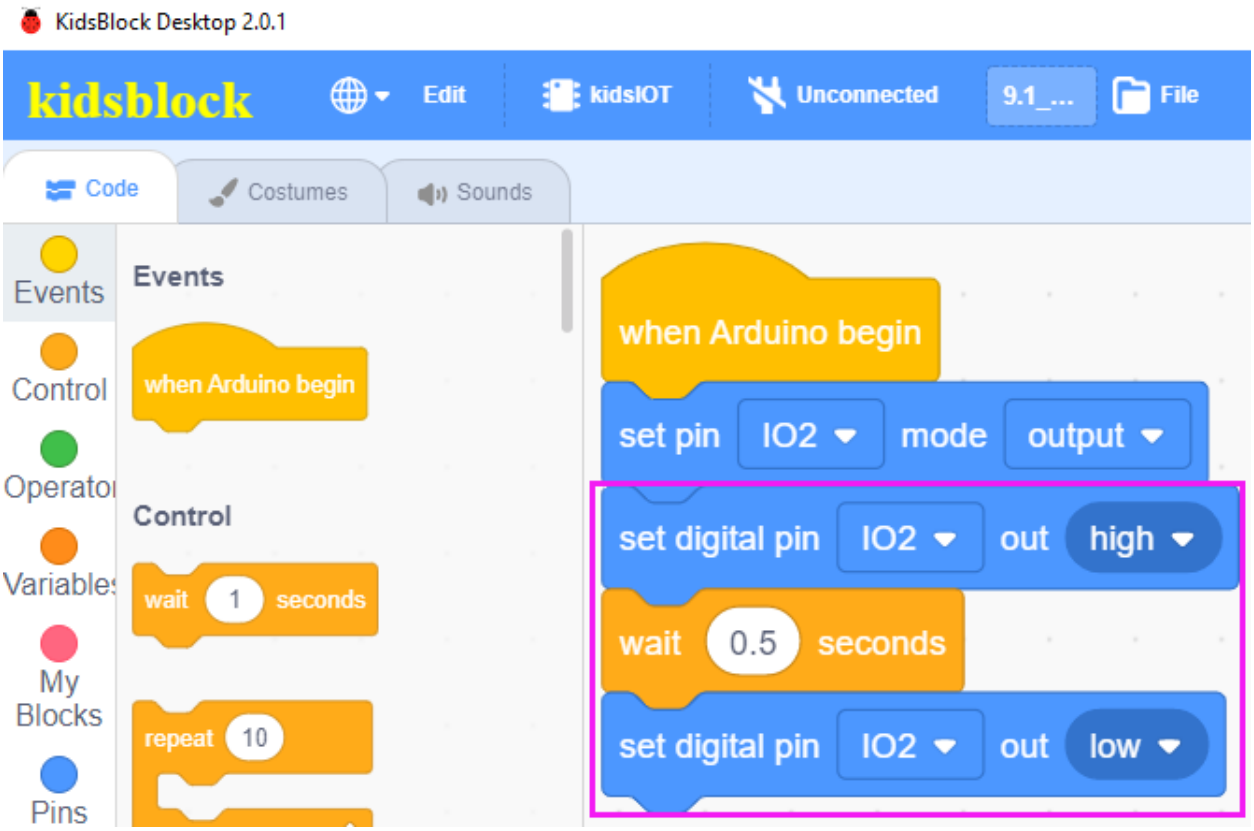


(1). Write the Program

Set the pin IO2 connected to the relay module to "**input**" mode.



Set the relay module to close for 0.5 seconds and then disconnect, controlling the water pump to pump water for 0.5 seconds.



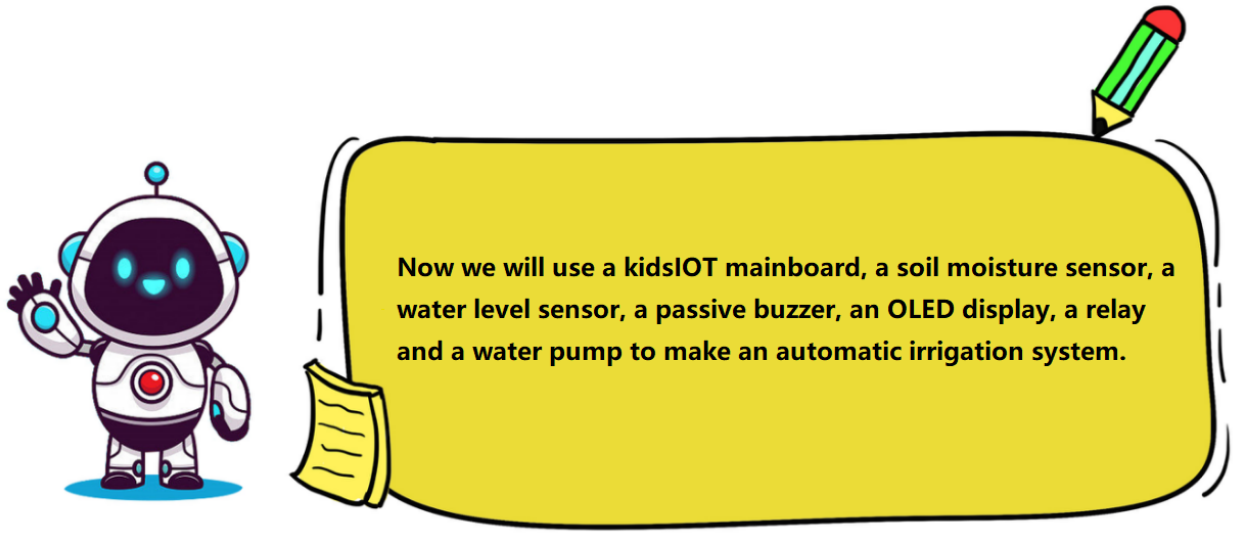
Complete Program

(2). Test Result

Click 

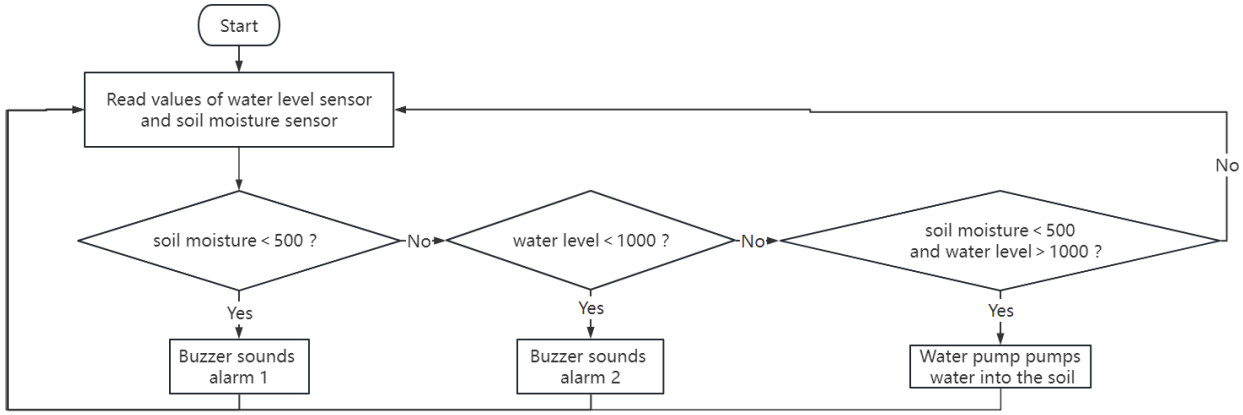
to upload the above complete code to the kidsIOT mainboard. After powering up via the USB cable, the relay module will control the water pump to pump water.

6. Automatic Irrigation System

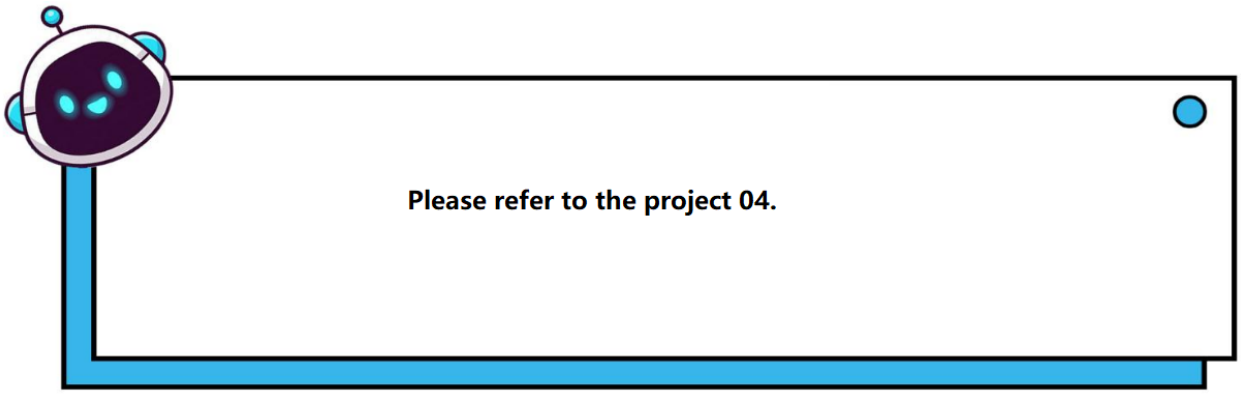


(1). Programming Steps

Step 1Flow Chart

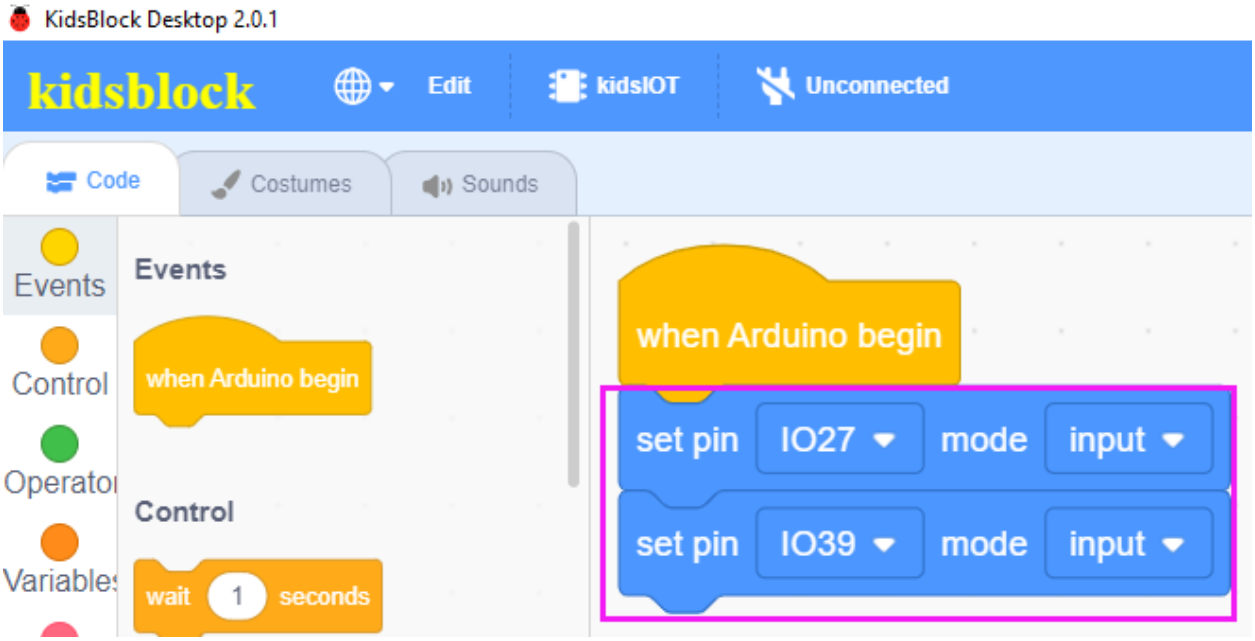


Step 2Add "passive buzzer"

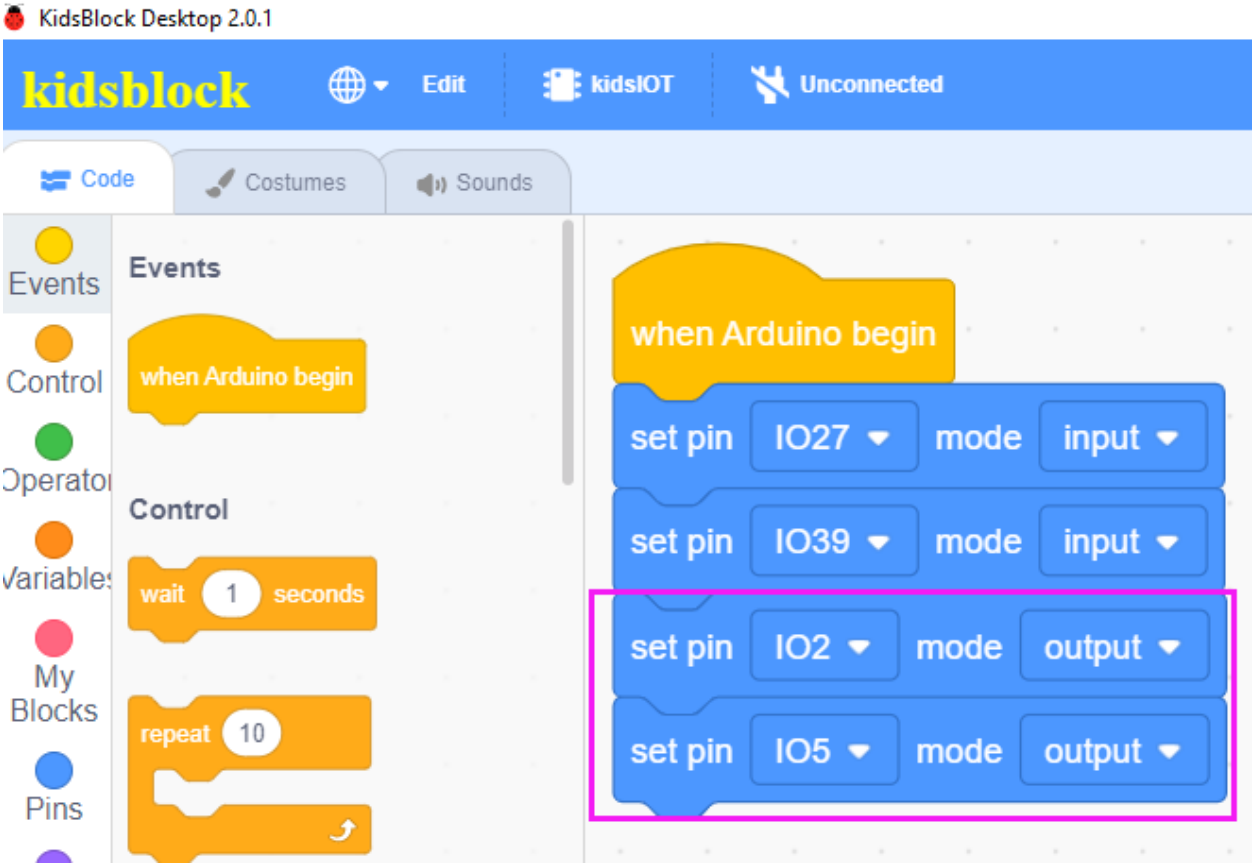


Step 3Write the Program

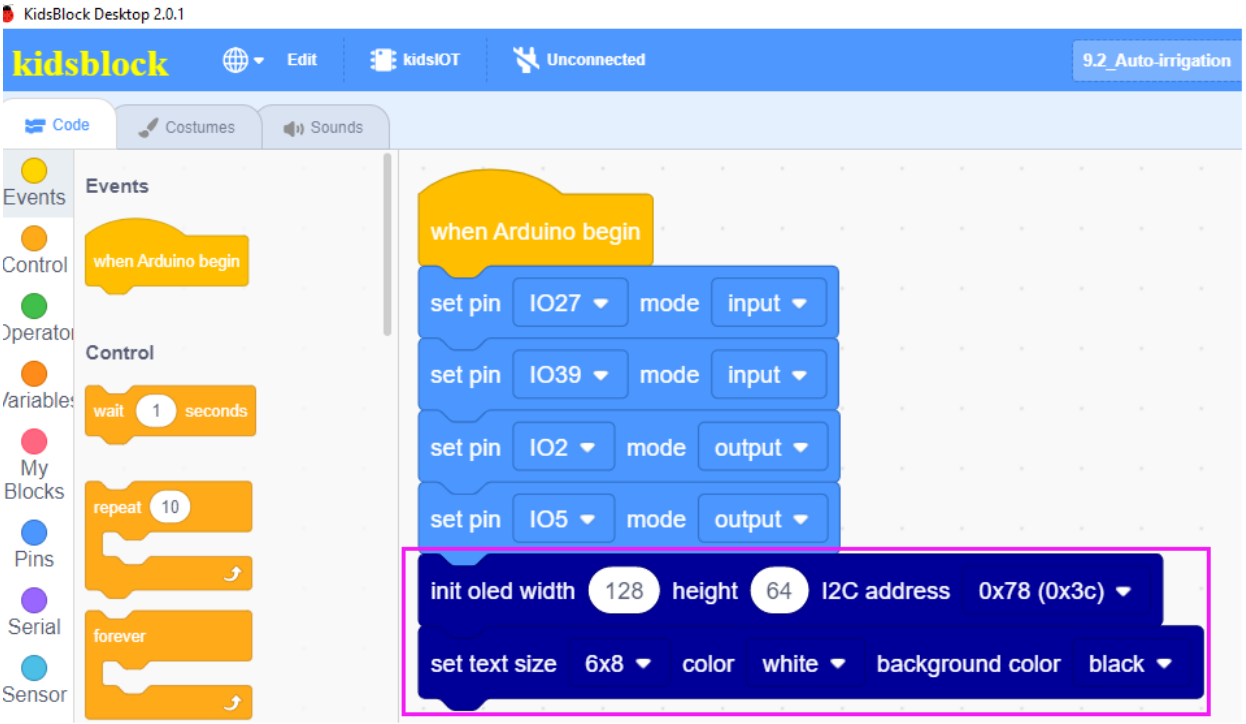
Initialize pin IO27 of the soil moisture sensor and pin IO39 of the water level sensor to "**input**" mode.



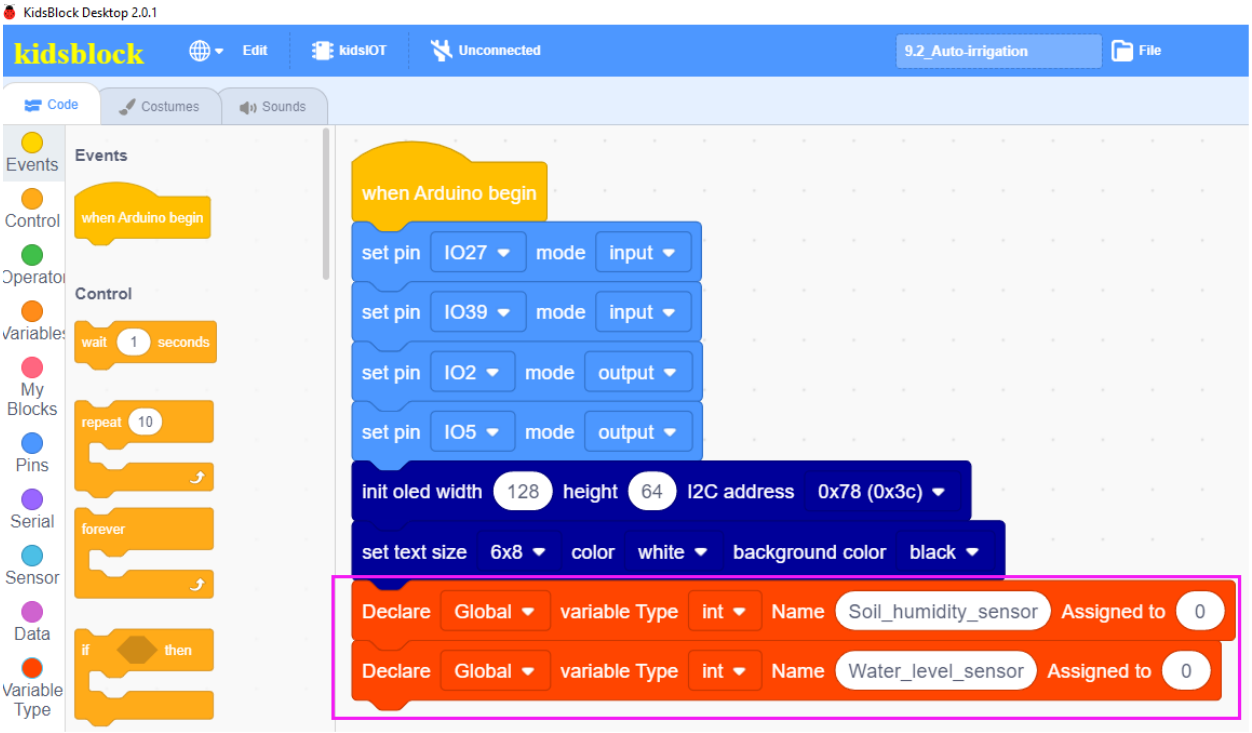
Initialize pin IO2 of the relay module and pin IO5 of the passive buzzer to "**Output**" mode.



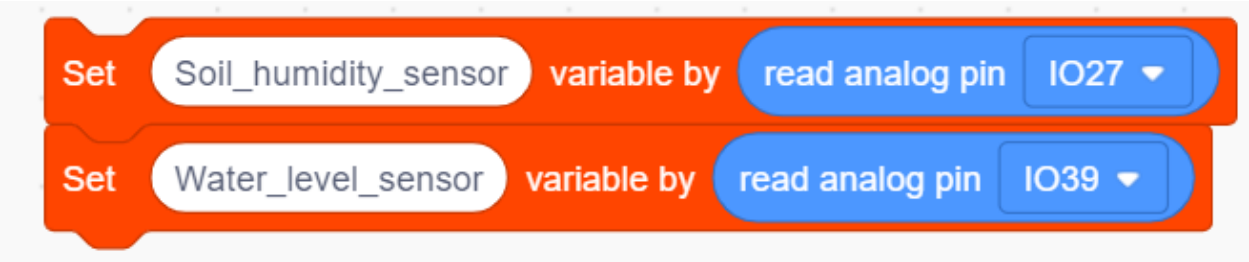
Initialize the width, height, I2C address, text size and color as well as background color of the OLED display.



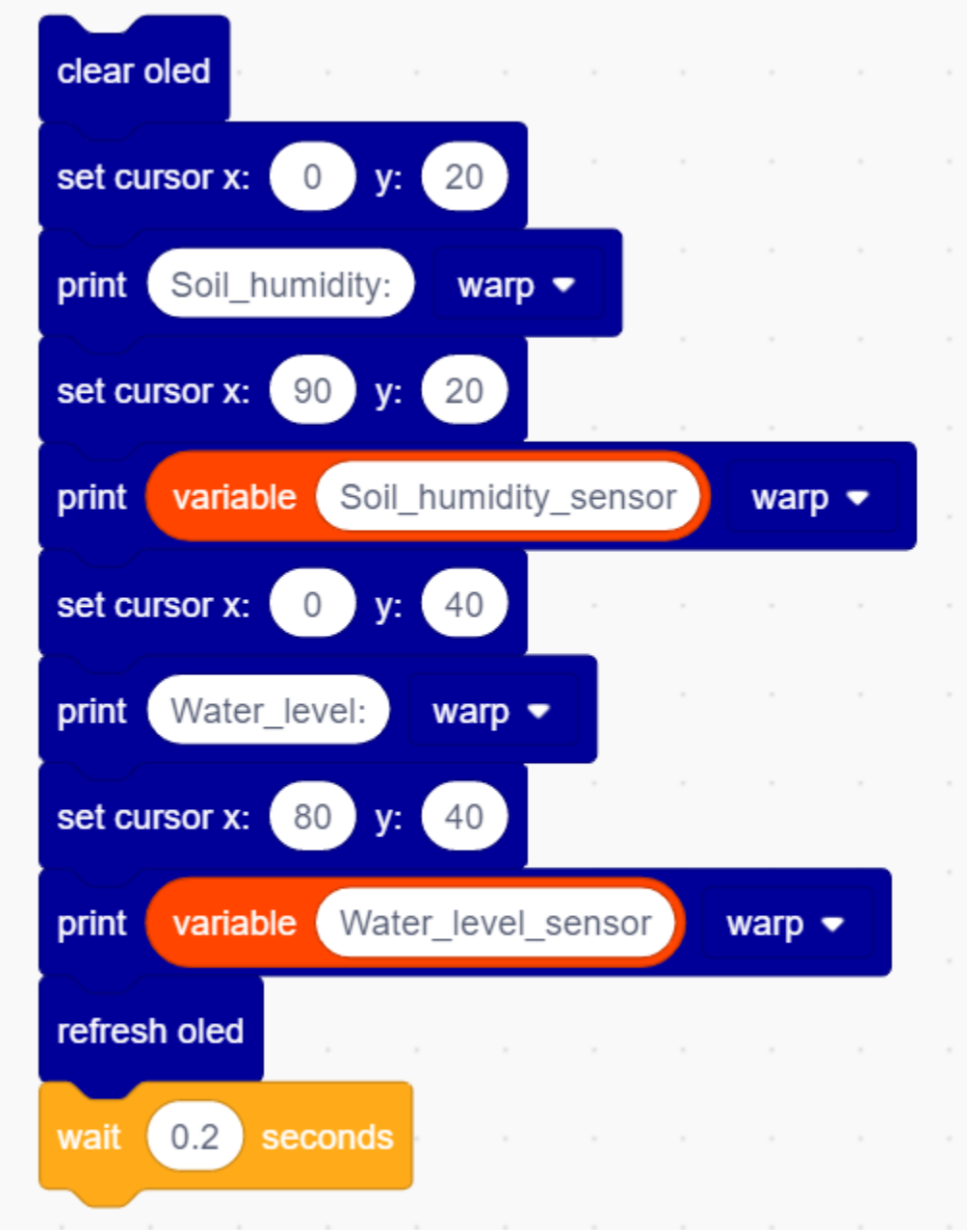
Define a "Soil_humidity_sensor" global variable to store the analog value of the soil moisture sensor and a "Water_level_sensor" global variable to store the analog value of the water level sensor.



Assign sensor data to variables.

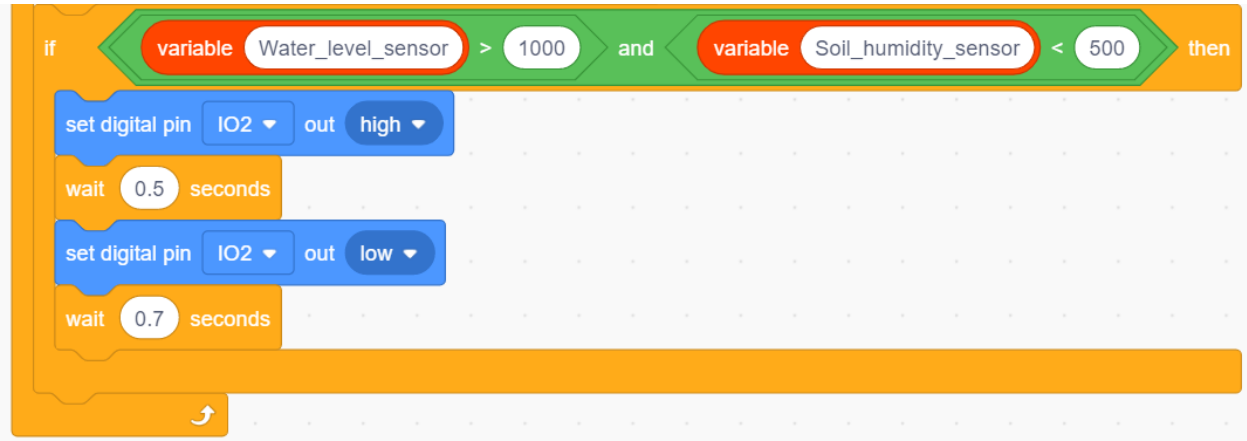


Display the sensor data at the corresponding position on the OLED display.



Judgment: When the water level is less than 1000, or the soil moisture is less than 500, the buzzer will sound an alarm.

When the soil moisture is less than 500 and the water level in the sink is greater than 1000, the relay will drive the water pump for automatic irrigation.



Complete Program

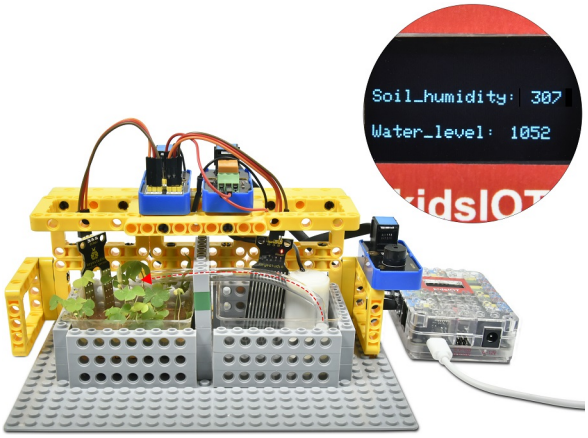
```
when Arduino begin
set pin IO27 mode input
set pin IO39 mode input
set pin IO2 mode output
set pin IO5 mode output
init oled width 128 height 64 I2C address 0x78 (0x3c)
set text size 6x8 color white background color black
Declare Global variable Type int Name Soil_humidity_sensor Assigned to 0
Declare Global variable Type int Name Water_level_sensor Assigned to 0
forever
  Set Soil_humidity_sensor variable by read analog pin IO27
  Set Water_level_sensor variable by read analog pin IO39
  clear oled
  set cursor x: 0 y: 20
  print Soil_humidity: warp
  set cursor x: 90 y: 20
  print variable Soil_humidity_sensor warp
  set cursor x: 0 y: 40
  print Water_level: warp
  set cursor x: 80 y: 40
  print variable Water_level_sensor warp
  refresh oled
  wait 0.2 seconds
  if variable Soil_humidity_sensor < 500 then
    Tone PIN# IO5 frequency NOTE_C3 duration 131
    Tone PIN# IO5 frequency NOTE_D3 duration 131
    Tone PIN# IO5 frequency NOTE_E3 duration 131
  if variable Water_level_sensor < 1000 then
    Tone PIN# IO5 frequency NOTE_G5 duration 131
    Tone PIN# IO5 frequency NOTE_A5 duration 131
    Tone PIN# IO5 frequency NOTE_B5 duration 131
  if variable Water_level_sensor > 1000 and variable Soil_humidity_sensor < 500 then
    set digital pin IO2 out high
    wait 0.5 seconds
    set digital pin IO2 out low
    wait 0.7 seconds
```

(2). Test Result

Click 

to upload the above complete code to the kidsIOT mainboard. After powering up via the USB cable, the OLED displays current soil moisture and sink water level information.

When the soil moisture is lower than the set threshold, then the soil is too dry, the buzzer will sound an alarm. At this time, the pumping system will automatically irrigate the soil. When the water level in the sink is lower than the set threshold, the pumping system will not work and the buzzer will sound an alarm to indicate that there is insufficient water in the sink.



7. Common Problems

Q1: Is the sensor waterproof?

A: The relay is not waterproof.

Q2: Does the rotation of the water pump cause the kidsIOT board to reset?

A: When the water pump rotates, it requires a larger current than other sensors, which will cause voltage and current fluctuations in the circuit. Excessive voltage and current fluctuations will cause the voltage and current of the kidsIOT mainboard to be too low, causing the kidsIOT mainboard to reset.

4.3.10 Project 10 WiFi Web Page Controls Smart Farm System

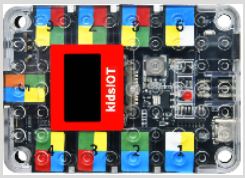
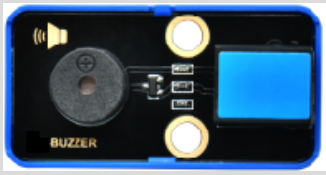
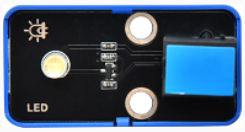
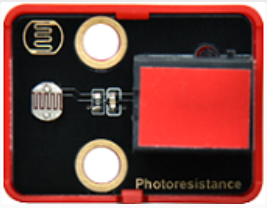
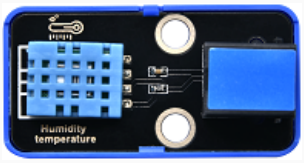
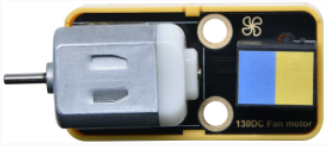
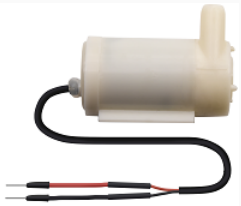
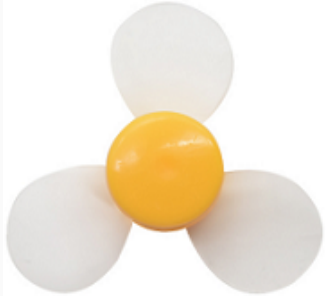
Note: Do not allow water to overflow from sinks and soil troughs when using the device. Sprinkling water on other sensors will cause a short circuit and device failure. Sprinkling water on batteries will cause heating and explosion. Please be careful when using the device, especially when used by young children, it must be under the supervision of parents. To ensure safe operation of the device, please follow relevant usage guidelines and safety regulations.

1. Description

In today's era of rapid technological development, unified control of intelligent devices via mobile phones has gradually gained people's favor. This method uses a microcontroller to establish a connection between a mobile phone and an intelligent device through a WiFi module and the Internet network to achieve remote control of the intelligent device.

In this project, we will focus on the WiFi infrastructure of ESP32 and control the smart farm system via WiFi web page.

2. Components

			
kidsIOT Mainboard×1	Water Level Sensor×1	Soil Moisture Sensor×1	Passive Buzzer×1
			
White LED Module×1	Photoresistor×1	Temperature and Humidity Sensor×1	Motor×1
			
Water Pump×1	GPIO Shield×1	Relay Module×1	Fan×1
			
F-F DuPont Wires	Sink×2	Battery Holder×1	M-F DuPont Wire×1
			
Water Pipe×1	AA BatteryNot provide×6	Slotted Screwdriver×1	Wire×7

3. Assembly Steps

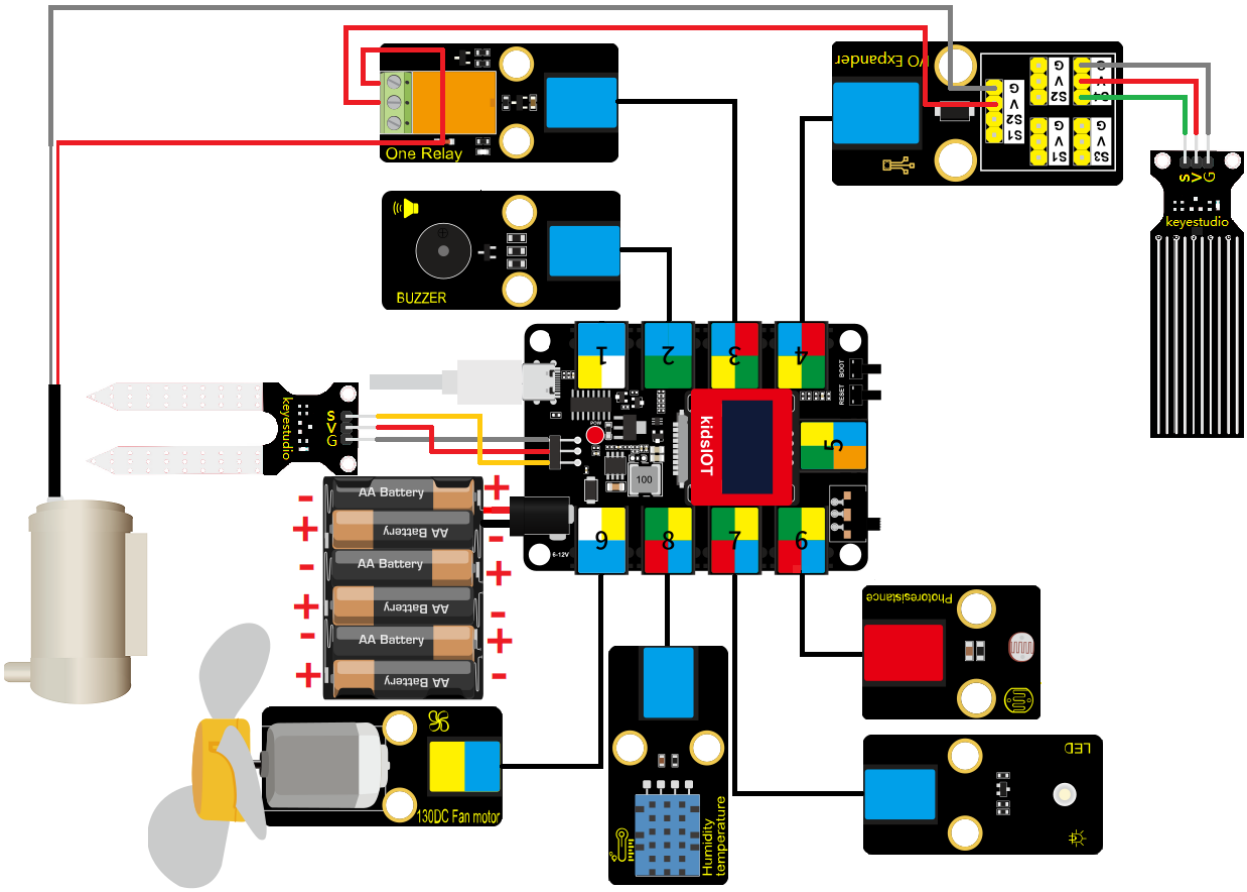
Just put the structural shapes assembled in Project 02, Project 06 and Project 07 together .

docs/KidsBlock_Tutorial/media/10-1.jpg

4. Wiring Diagram

Module	GPIO Shield	kidsIOT Mainboard
Water Level Sensor	G→GV→VS→S4	No.4 portcontrol pin of S4 is io39
Soil Moisture Sensor		G/V/io33 portG→GV→VS→io33
Photoresistor		No.6 portcontrol pin is io36
Temperature and Humidity Sensor		No.8 portcontrol pin is io5
White LED Module		No.7 portcontrol pin is io16
Passive Buzzer		No.2 portcontrol pin is io2
Motor		No.9 portIN+control pin is io18IN-control pin is io19
Relay Module		No.3 portcontrol pin is io26

Connect the kidsIOT mainboard to your computer via USB cable, connect the external power supply and turn the DIP switch on the mainboard to ON end.



5. WiFi Web Page Display

The ESP32 chip on the kidsIOT board comes with Wi-Fi (2.4GHz) and Bluetooth (4.2) functions. It can easily connect to the Wi-Fi network and communicate with other devices on the network. You can use ESP32 to display web pages in the browser.

(1). Knowledge

The Wi-Fi library supports configuring and monitoring ESP32 Wi-Fi networking functions. Supported configurations:

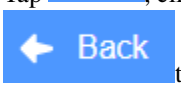
- station mode (STA mode or Wi-Fi client mode), at this time the ESP32 is connected to the access point (AP).
- AP mode (Soft-AP mode or Access Point mode), at this time the base station is connected to the ESP32.
- station/AP coexistence mode (ESP32 is both an access point and a base station connected to another access point).
- Various security modes for the above mentioned modes (WPA, WPA2, WPA3, etc.).
- Scan access points (including active scanning and passive scanning).
- Monitor IEEE802.11 Wi-Fi packets using promiscuous mode.

For details about wifi, tap ithttps://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/network/esp_wifi.html

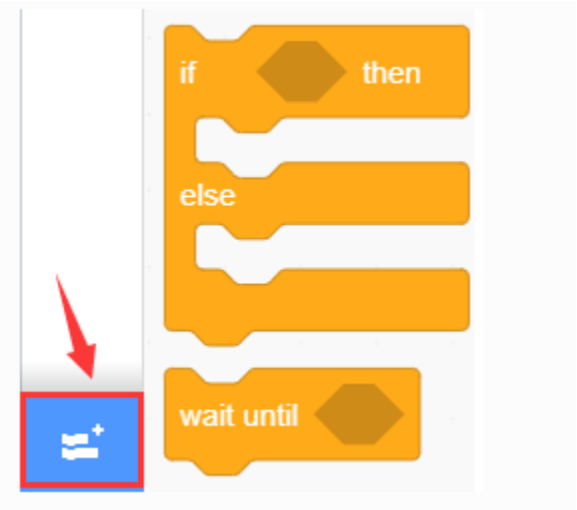
Espressif official websitehttps://www.espressif.com.cn/en/home

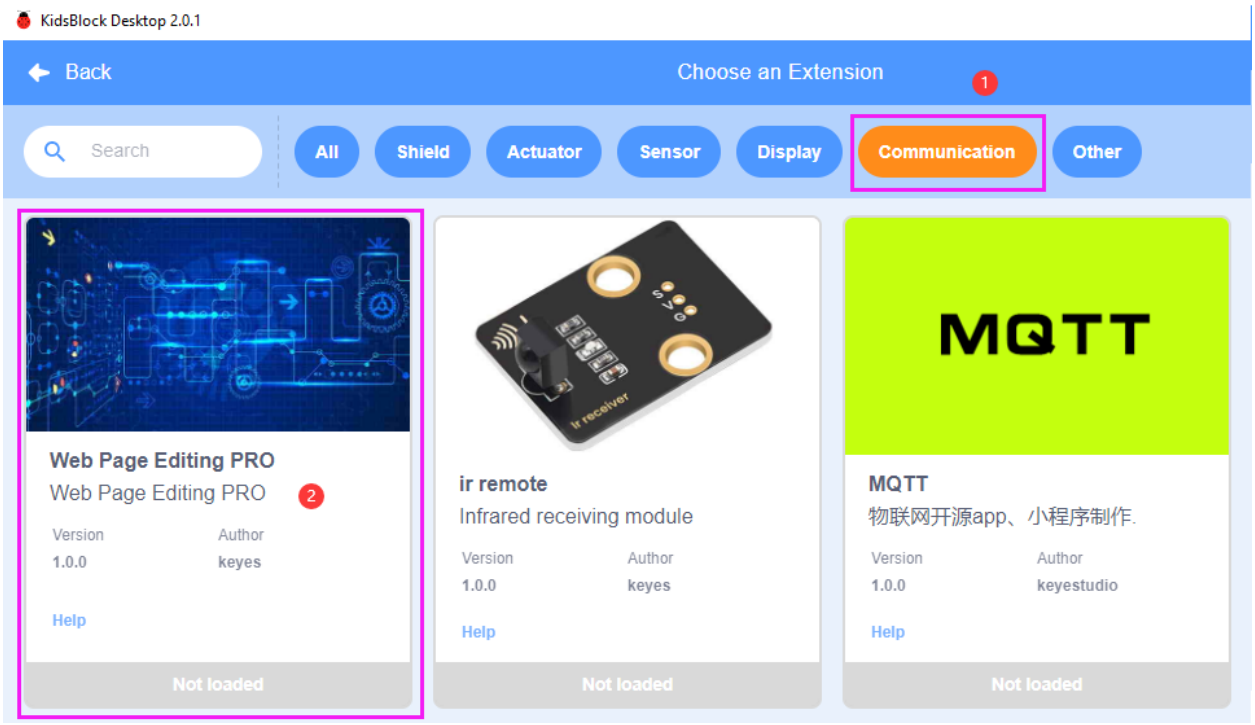
(2). Programming Steps

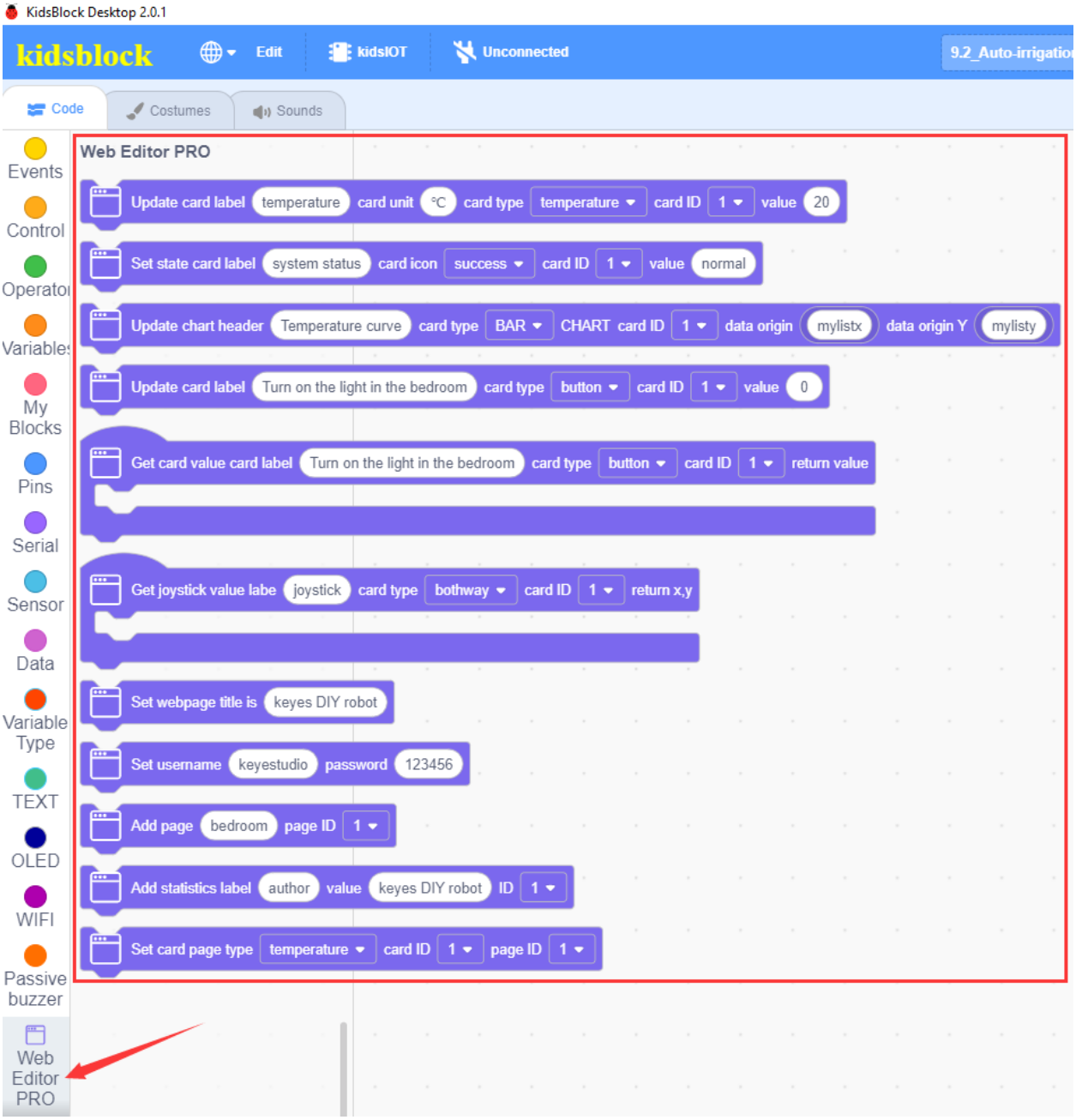
Step 1Add the "Web Page Editing PRO" library

Tap  , click the "communication" module in the "Extension" , then select "**Web Page Editing PRO**" and click 

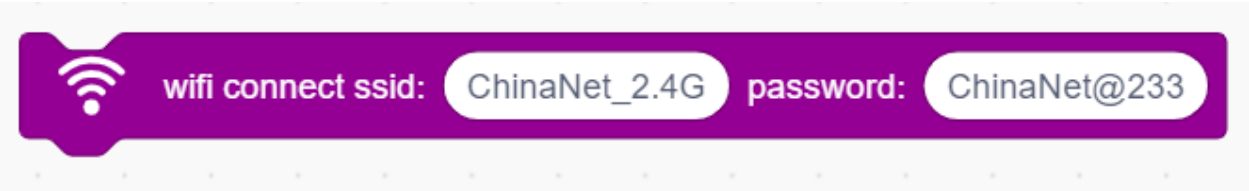
to return to the programming interface.



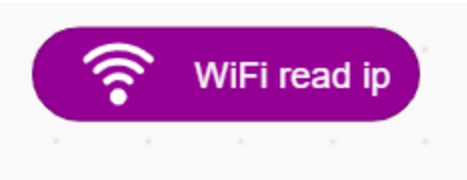




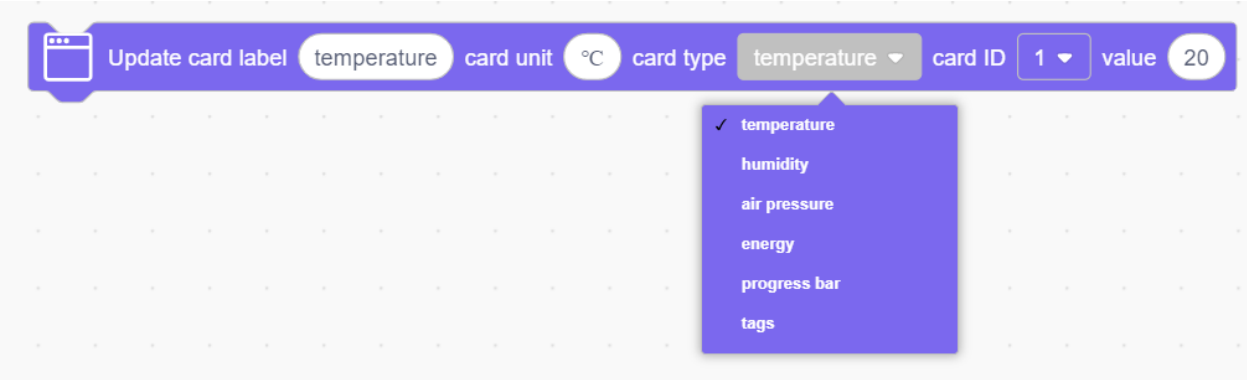
Step 2Description of the Building Block



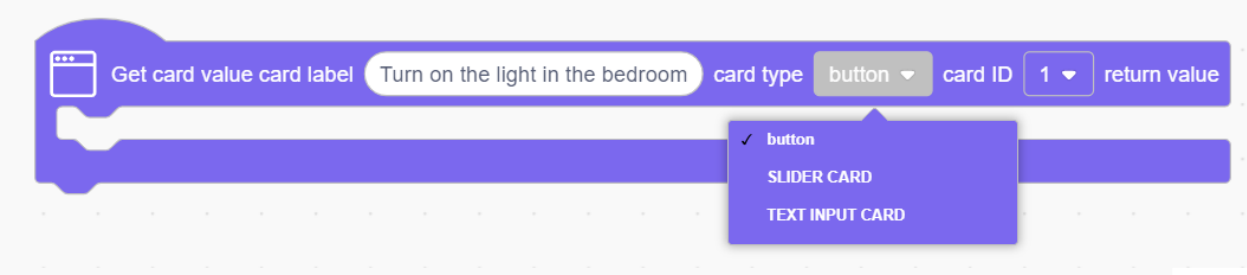
Enter the WiFi name and password to connect to the WiFi hotspot.



Read the WiFi IP address.



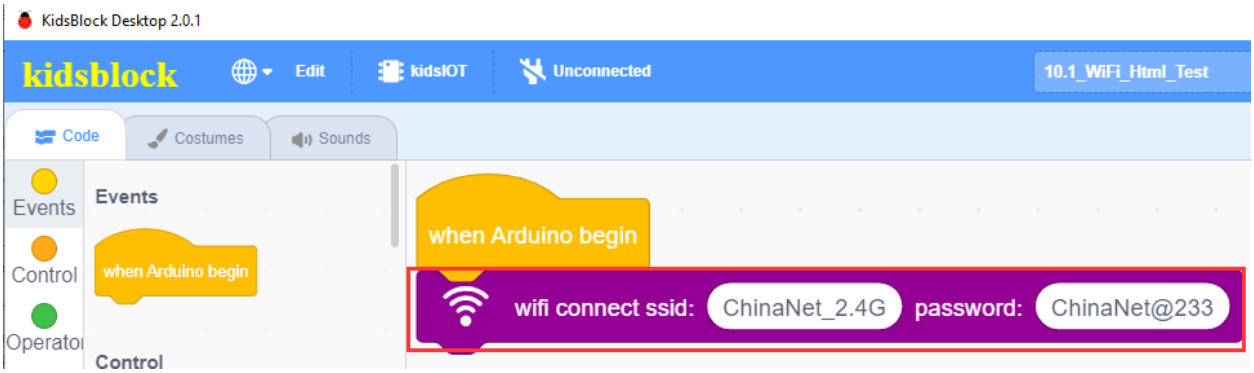
Set up a card on the web page, and its card label, card unit and card type correspond one to one.



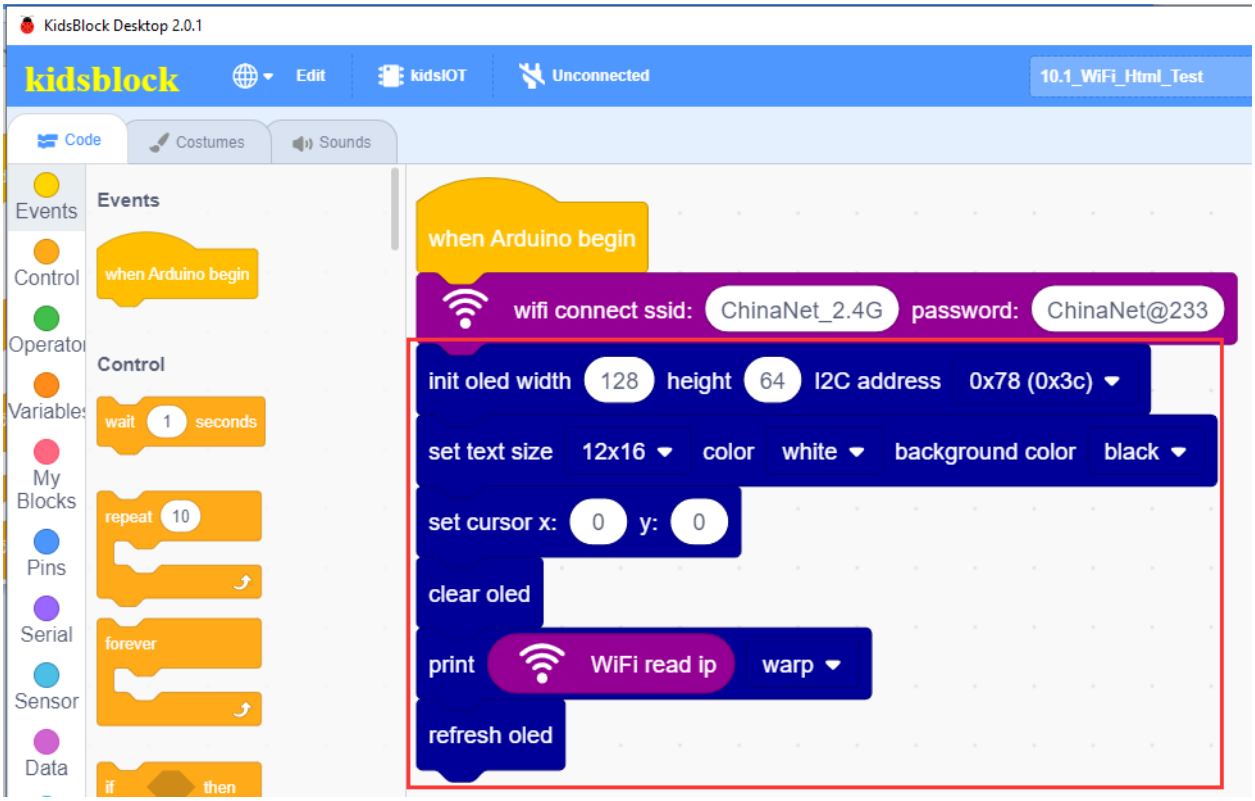
Add a button card to the web page.

Step 3Write the Program

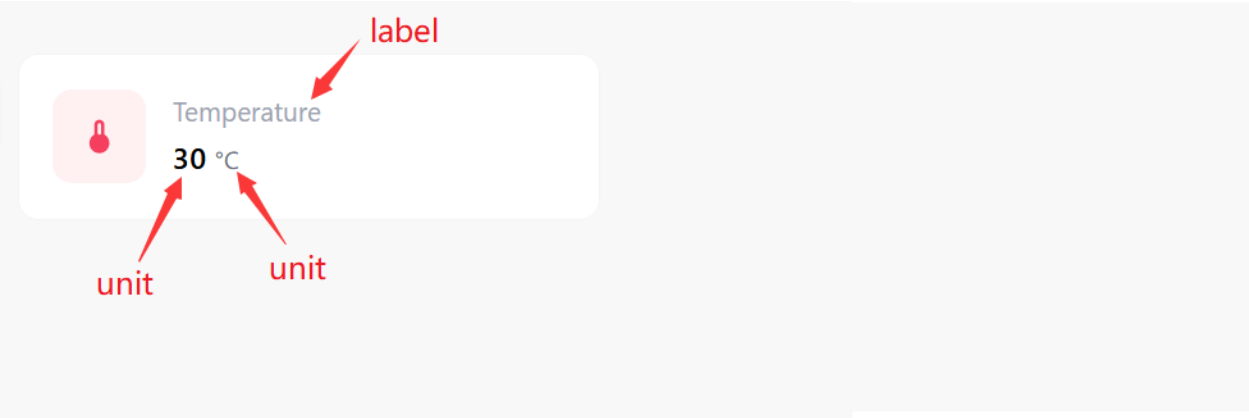
Change the WiFi name and password in the code to your own WiFi name and password, and connect to the WiFi hotspot.



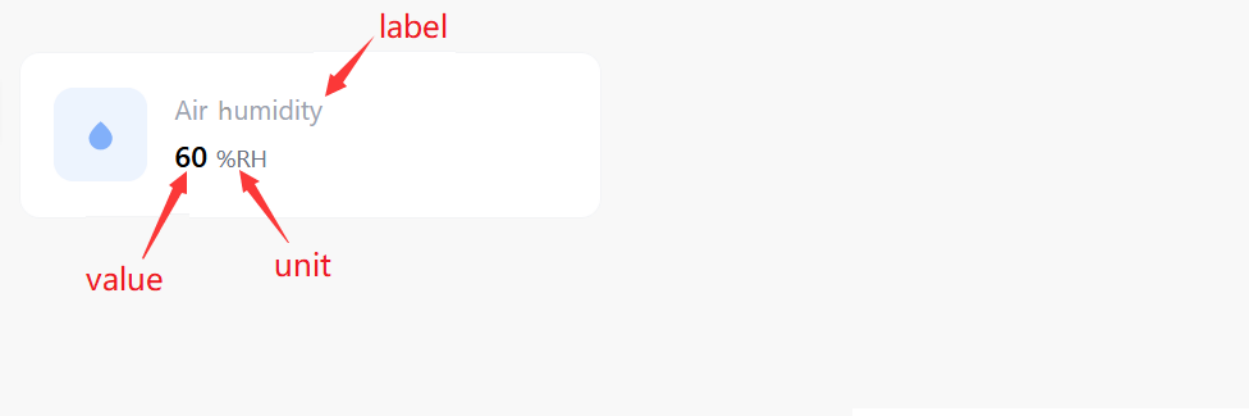
Display the WiFi IP address on the OLED.



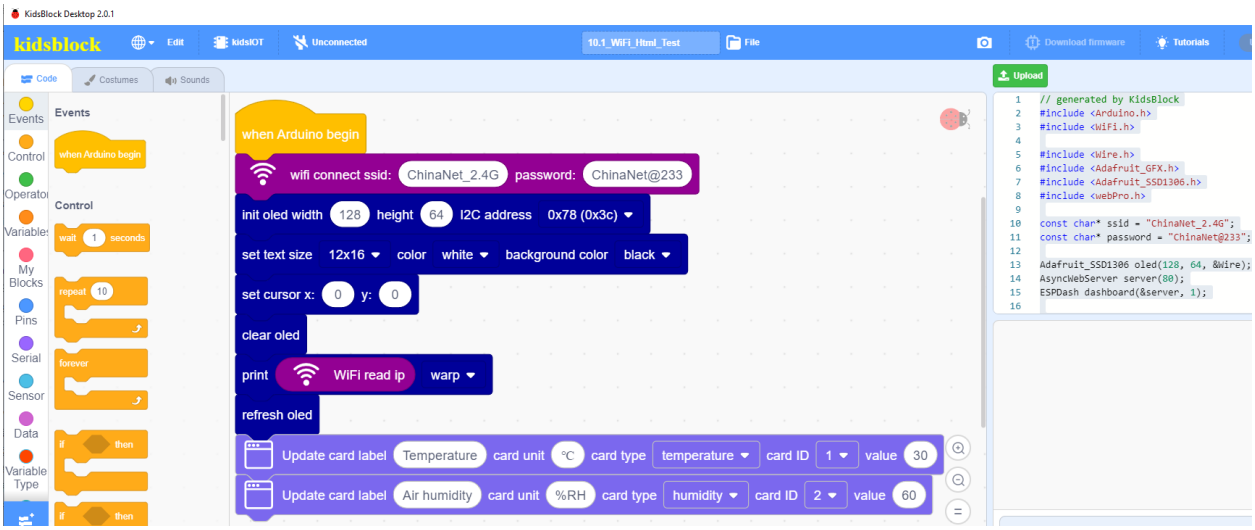
Set a card on the web page, the card label is “Temperature”, the card type is “Temperature”, the card unit is “°C”, the card ID is 1, and the temperature value is 30.



Set up a card on the web page. The card label is “Air humidity”, the card type is “Humidity”, the card unit is “%RH”, the card ID is 2, and the temperature value is 60.



Complete Program



(3). Test Result

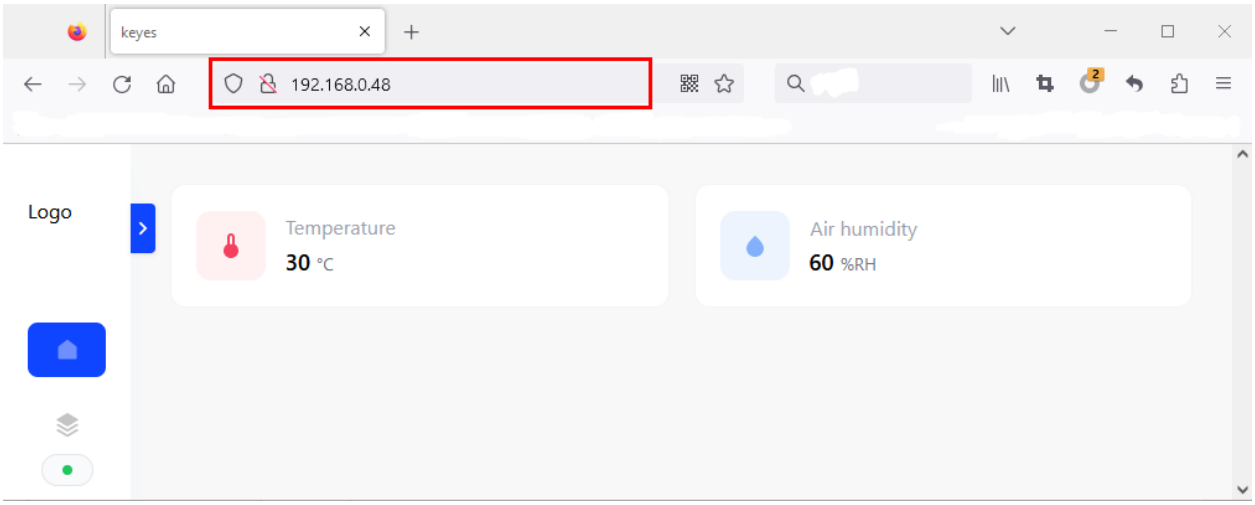
Click 

to upload the above complete code to the kidsIOT mainboard and power on. Once connected to WiFi, the OLED on the board will display the corresponding IP address (Here we take the IP address: 192.168.0.48 as an example).

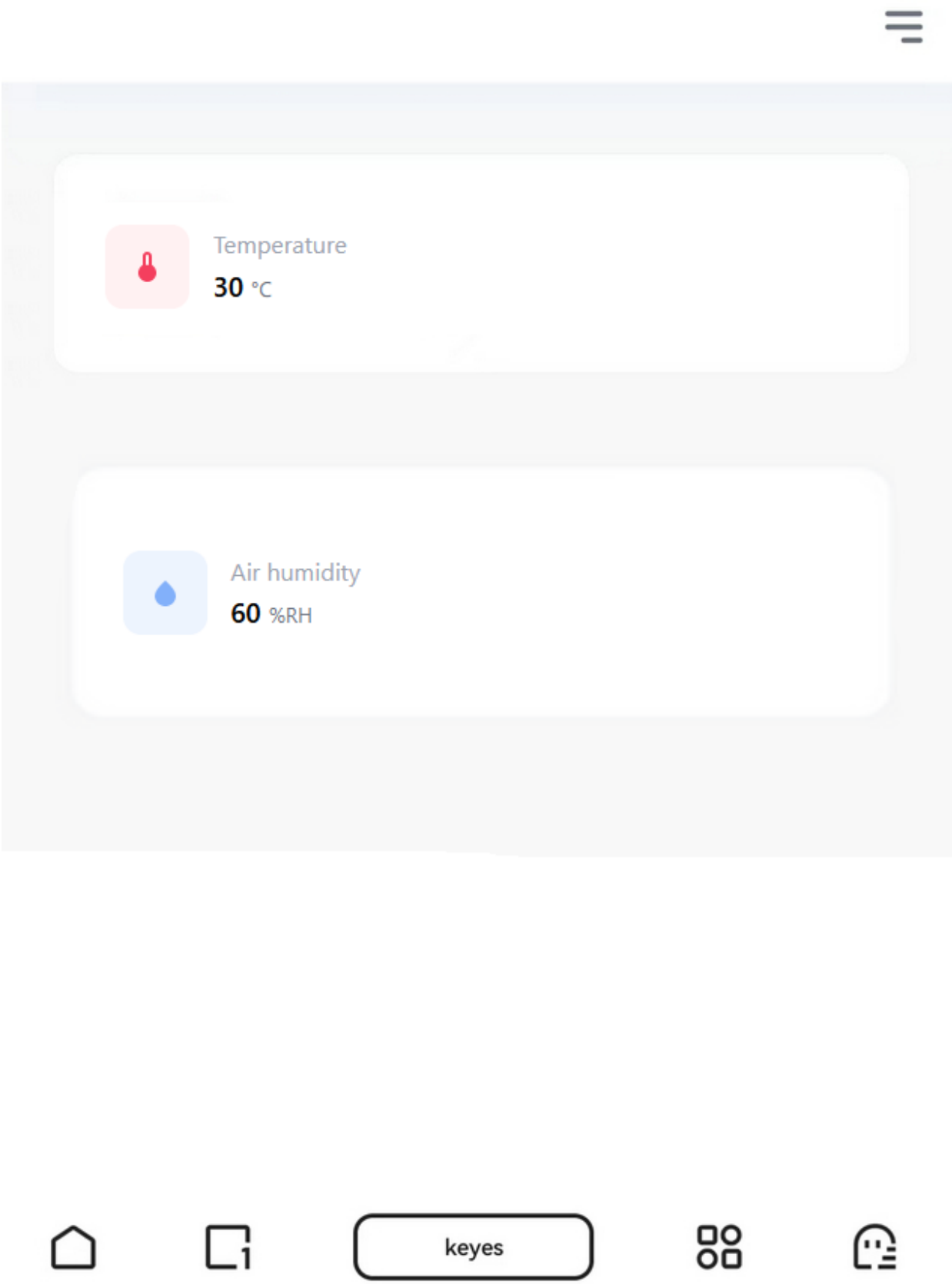
Note: When the PC, mobile phone and kidsIOT mainboard are connected to the same network, this webpage can be opened on the PC and mobile phone at the same time. Here is the IP address displayed on the OLED on your own kidsIOT board

You can enter "**http://[IP address displayed on the OLED display]**" in the browser to view the web page . In this way, you will create a simple web page that displays a fixed temperature information and a fixed humidity information:

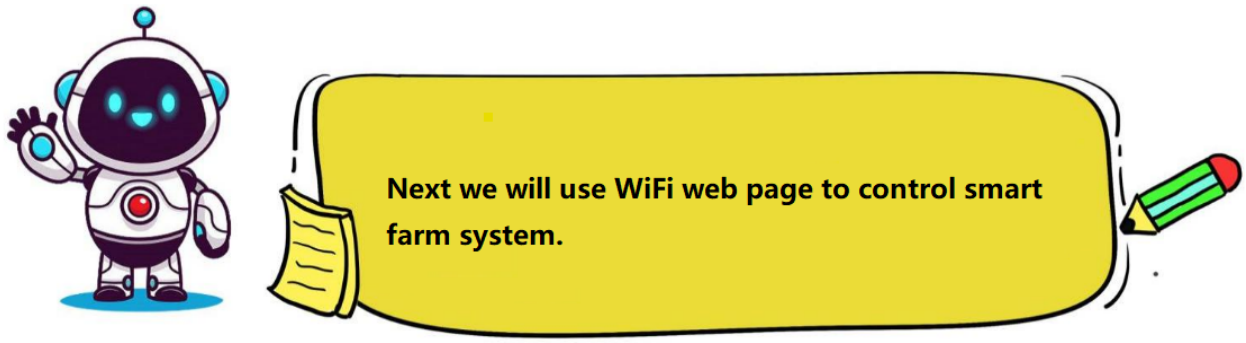
PC



Mobile phone

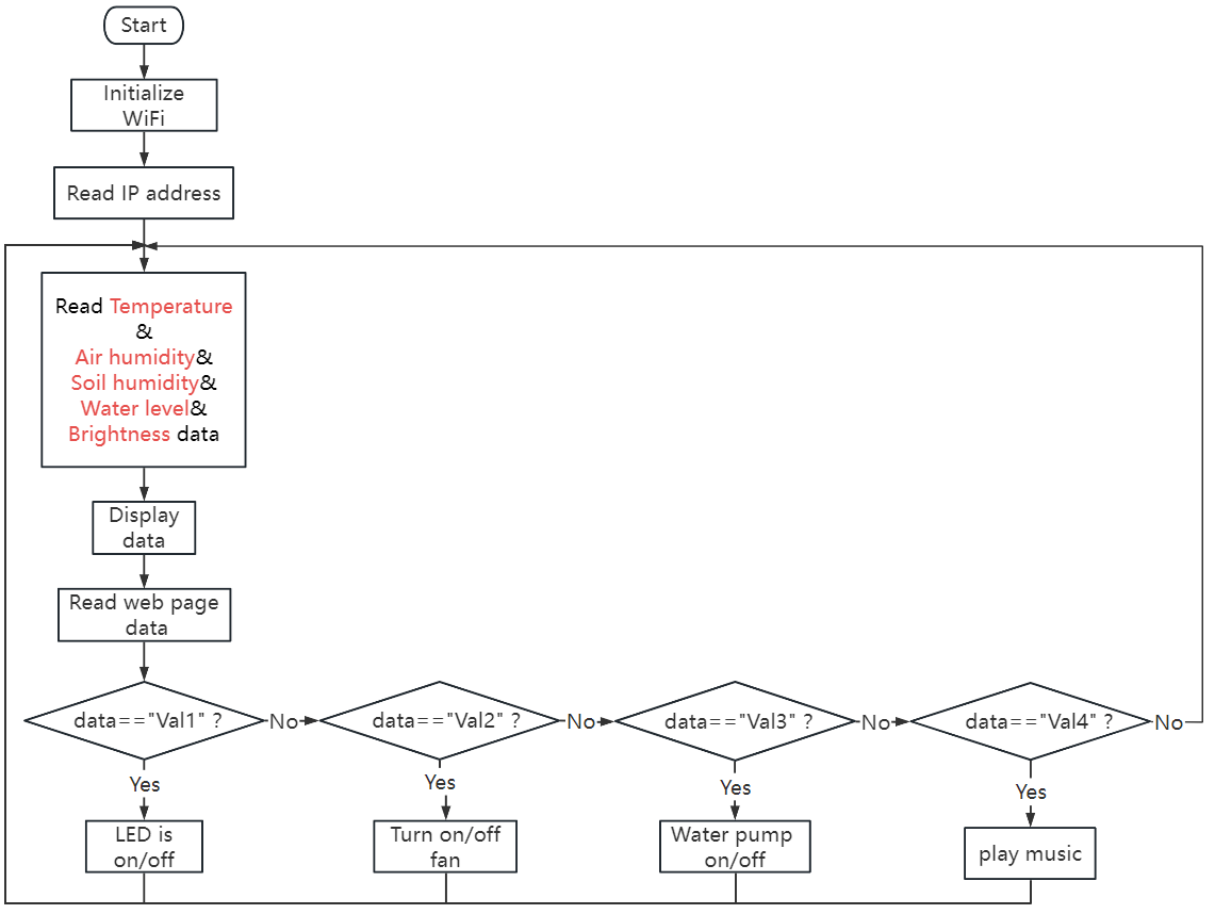

Temperature
30 °C
Air humidity
60 %RH
keyes

6. WiFi Web Page Controls Smart Farm System



(1). Programming Steps

Step 1Flow Chart

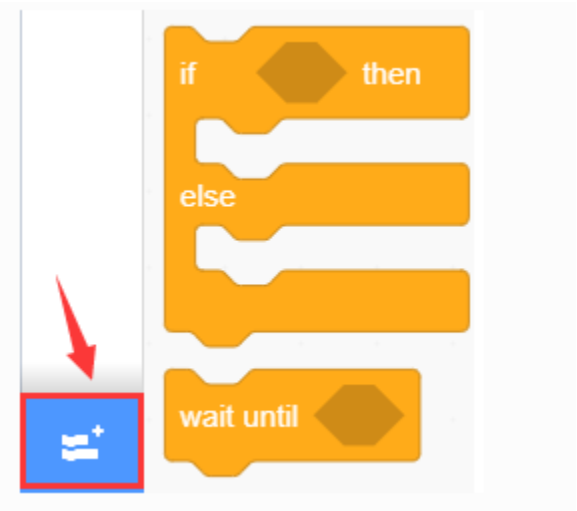


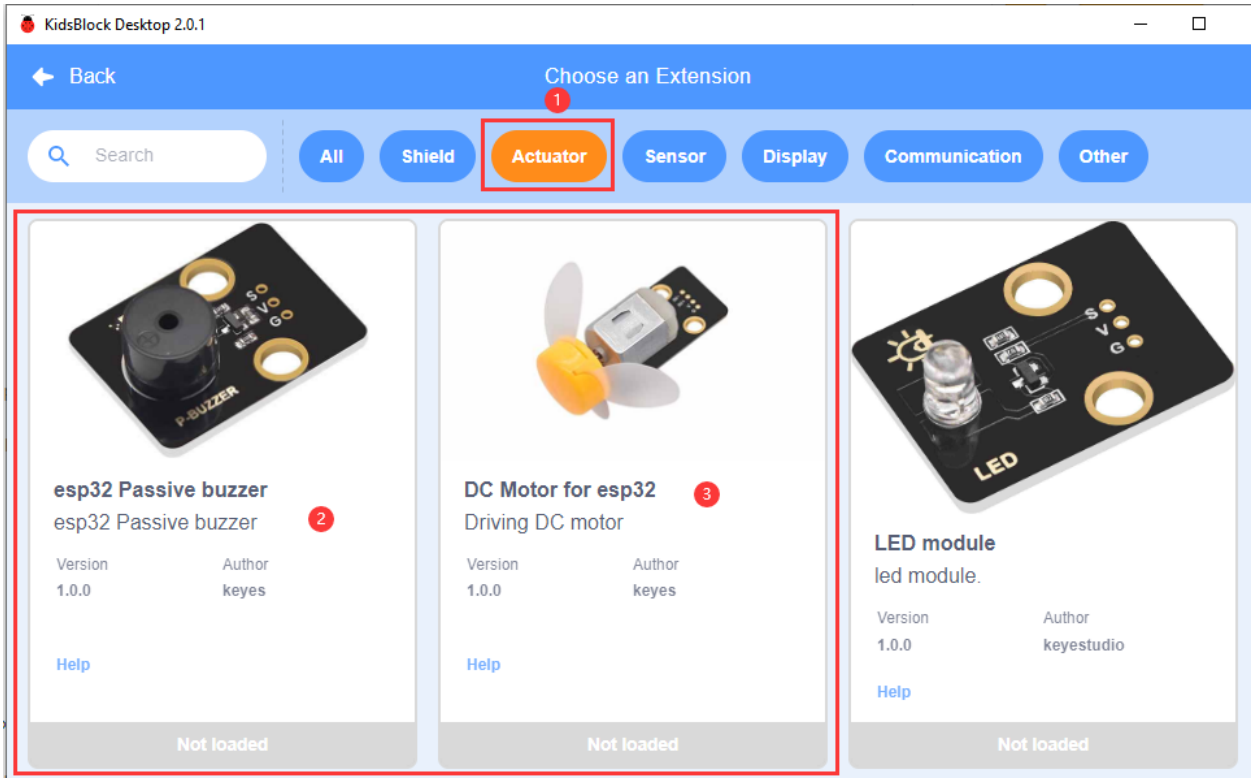
Step 2Add "Passive Buzzer", "DC Motor", "Temperature and Humidity Sensor" and "Web Page Editing PRO" libraries

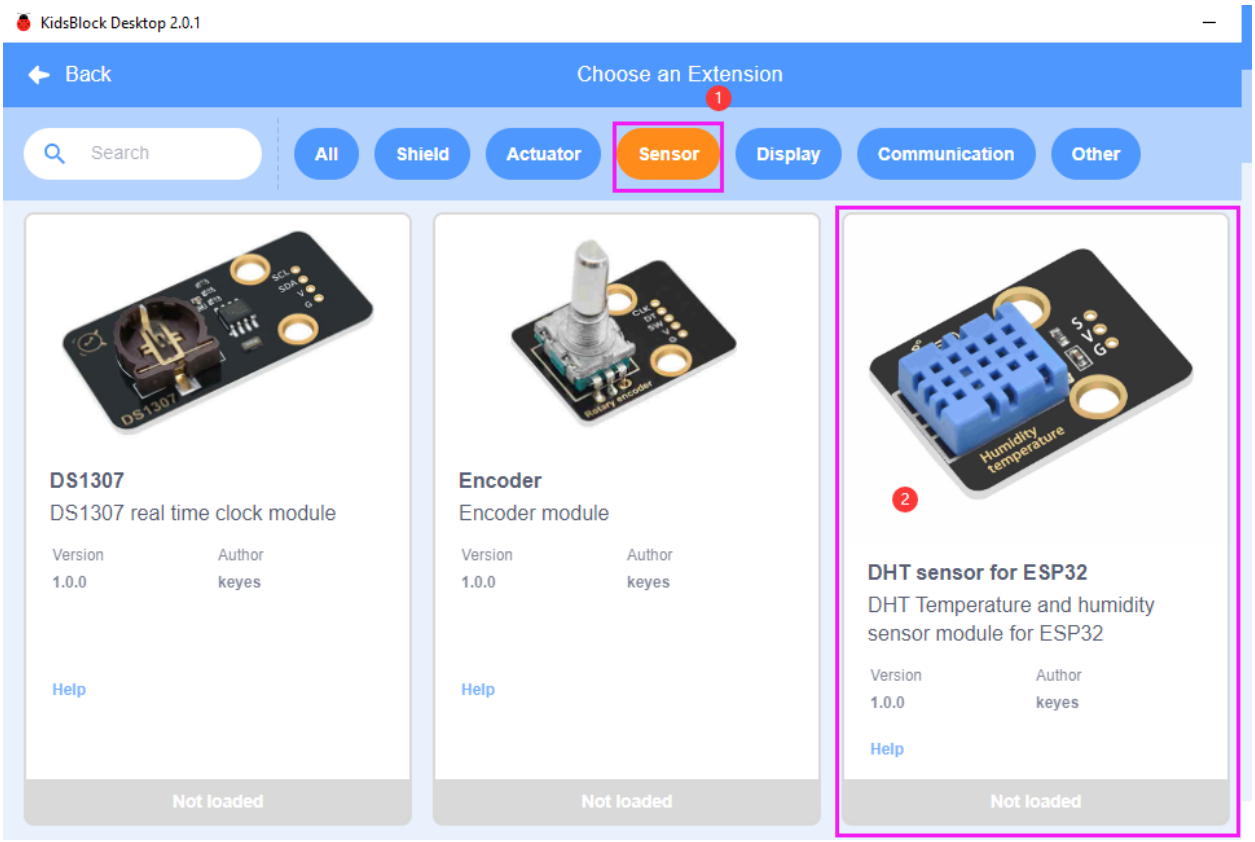
Tap  , click "Actuator" and find "esp32 Passive buzzer" and "DC Motor for esp32" . Click "Sensor" and find "DHT sensor for ESP32" .

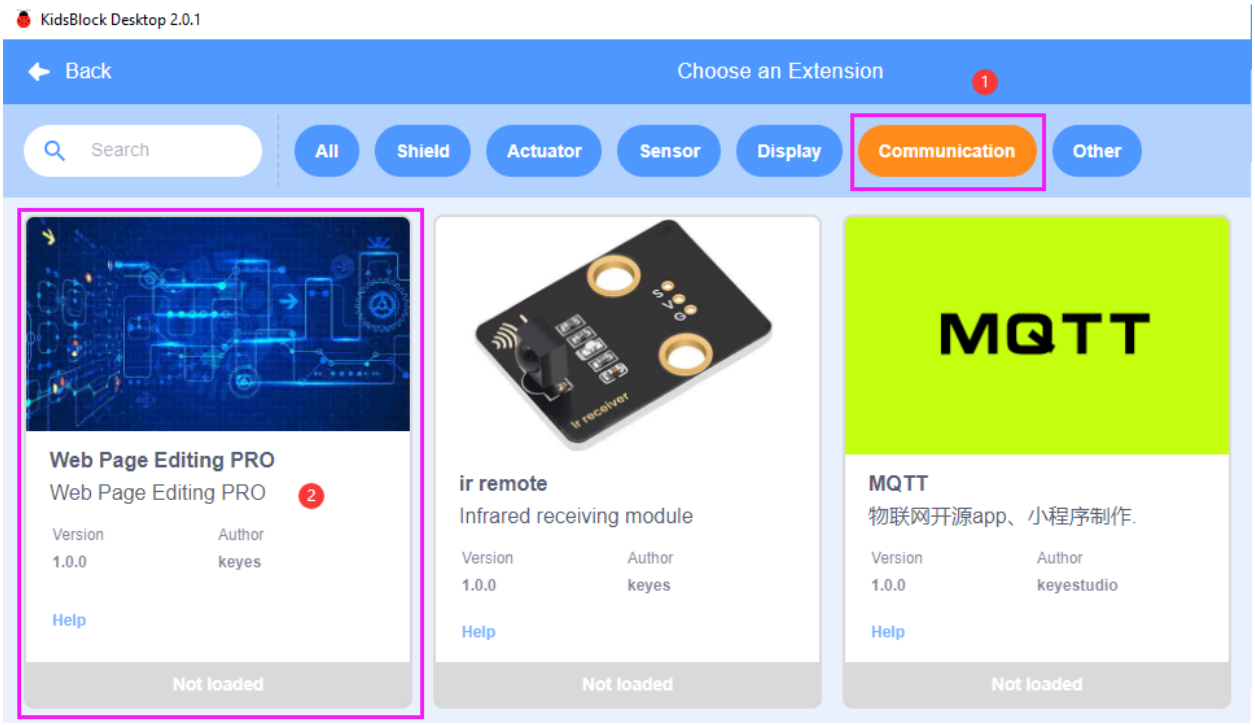
Click the "communication" , then select "**Web Page Editing PRO**" and click 

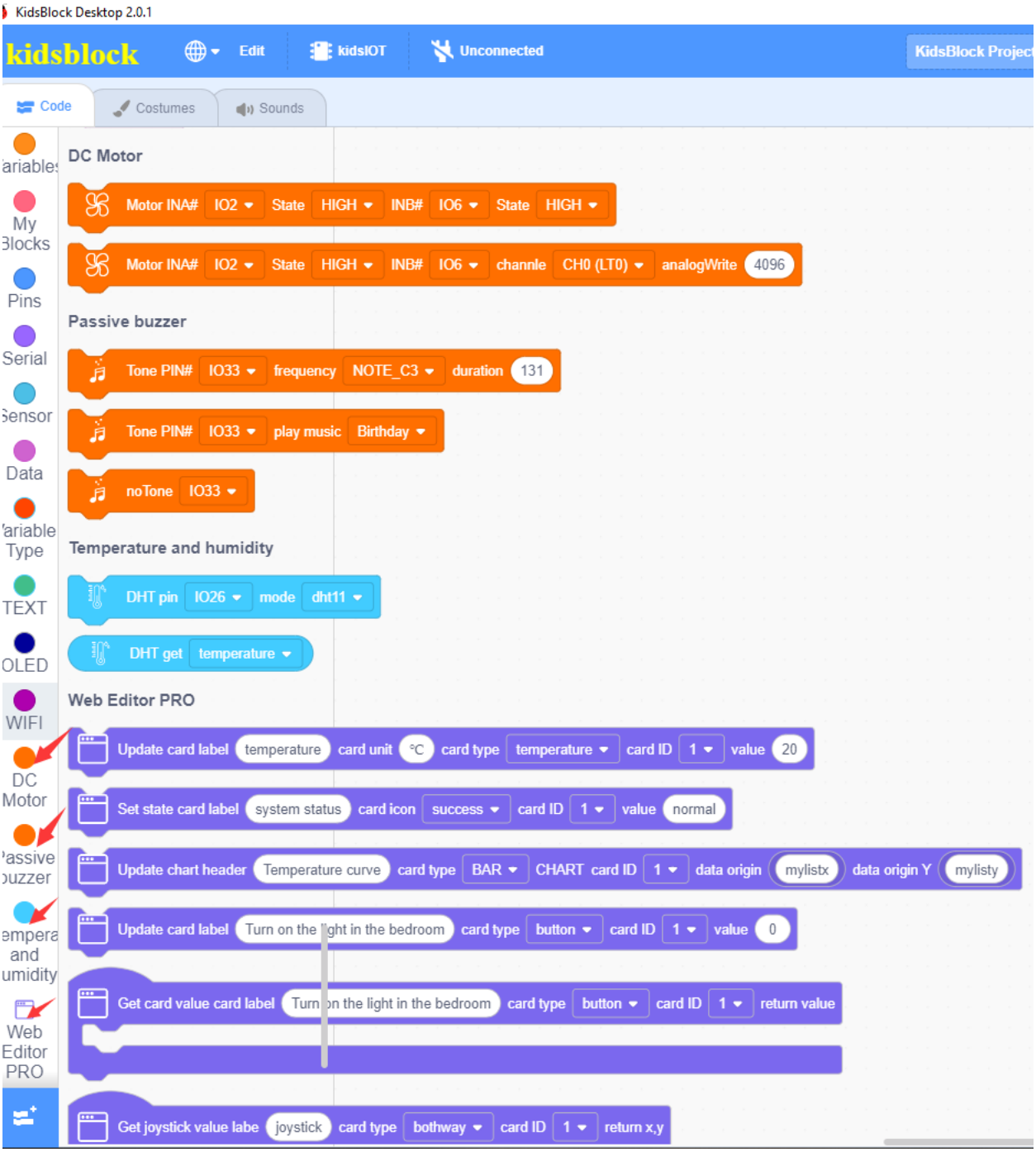
to return to the programming interface.






KidsBlock Desktop 2.0.1
Back
Choose an Extension
1
Search
All
Shield
Actuator
Sensor
Display
Communication
Other
DS1307
DS1307 real time clock module
Version
1.0.0
Author
keyes
Help
Not loaded
Encoder
Encoder module
Version
1.0.0
Author
keyes
Help
Not loaded
2
DHT sensor for ESP32
DHT Temperature and humidity sensor module for ESP32
Version
1.0.0
Author
keyes
Help
Not loaded


KidsBlock Desktop 2.0.1
Back
Choose an Extension
1
Search
All
Shield
Actuator
Sensor
Display
Communication
Other
Web Page Editing PRO
Web Page Editing PRO
2
Version
1.0.0
Author
keyes
Help
Not loaded
ir remote
Infrared receiving module
Version
1.0.0
Author
keyes
Help
Not loaded
MQTT
MQTT
物联网开源app、小程序制作.
Version
1.0.0
Author
keyestudio
Help
Not loaded

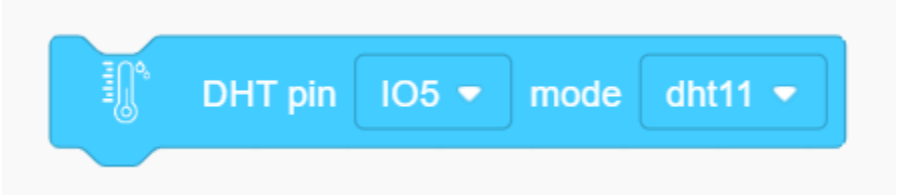

KidsBlock Desktop 2.0.1
kidsblock
Edit
kidsIOT
Unconnected
KidsBlock Projec
Code
Costumes
Sounds
ariables
My Blocks
Pins
Serial
Sensor
Data
ariable Type
TEXT
OLED
WIFI
DC Motor
assive buzzer
empera and umidity
Web Editor PRO
DC Motor
Motor INA# IO2 State HIGH INB# IO6 State HIGH
Motor INA# IO2 State HIGH INB# IO6 channle CH0 (LT0) analogWrite 4096
Passive buzzer
Tone PIN# IO33 frequency NOTE_C3 duration 131
Tone PIN# IO33 play music Birthday
noTone IO33
Temperature and humidity
DHT pin IO26 mode dht11
DHT get temperature
Web Editor PRO
Update card label temperature card unit °C card type temperature card ID 1 value 20
Set state card label system status card icon success card ID 1 value normal
Update chart header Temperature curve card type BAR CHART card ID 1 data origin mylistx data origin Y mylisty
Update card label Turn on the ght in the bedroom card type button card ID 1 value 0
Get card value card label Turn on the light in the bedroom card type button card ID 1 return value
Get joystick value labe joystick card type bothway card ID 1 return x,y

Step 3Write the Program

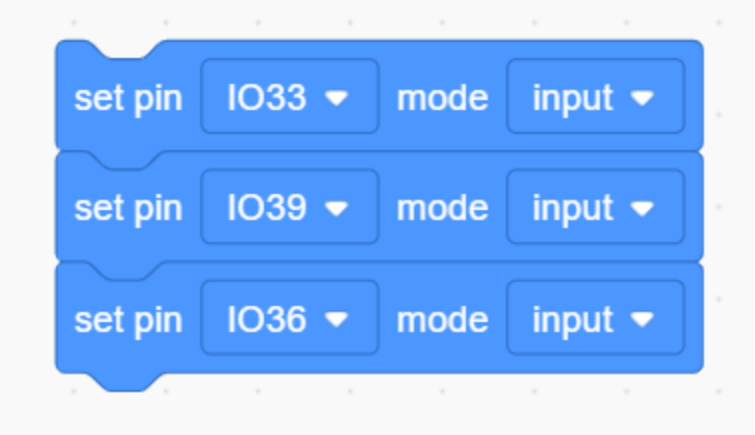
Change the WiFi name and password in the code to your own WiFi name and password, and connect to the WiFi hotspot.

Display the WiFi IP address on the OLED.

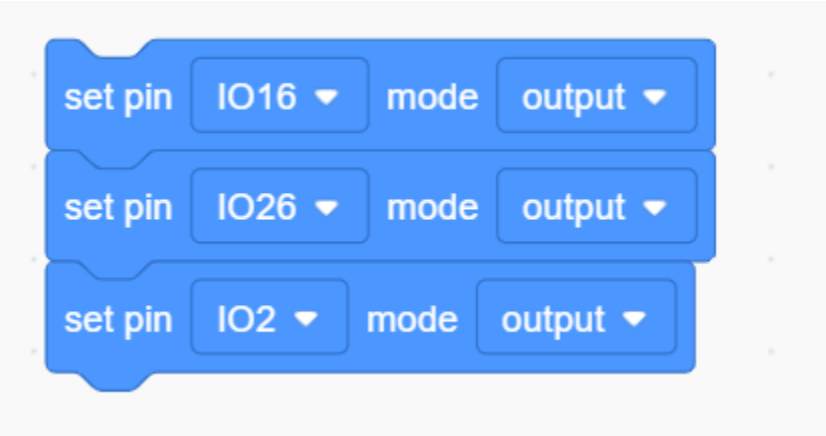
Initialize pin IO5 of the temperature and humidity sensor, and select the mode dht11.



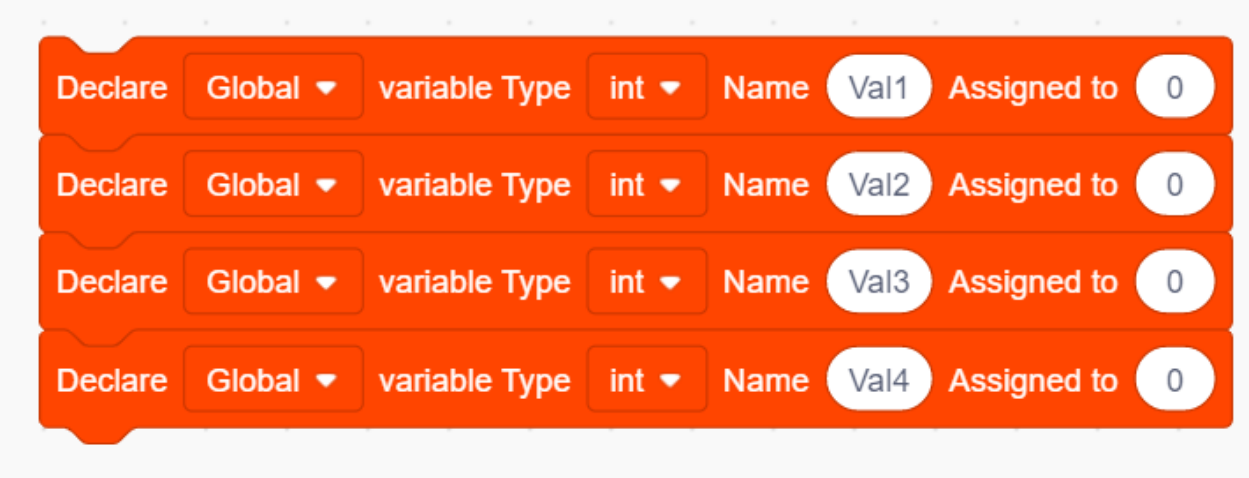
Initialize the pin IO33 of the soil moisture sensor, the pin IO39 of the water level sensor, and the pin IO36 of the photoresistor, and set them to "**Input**" mode.



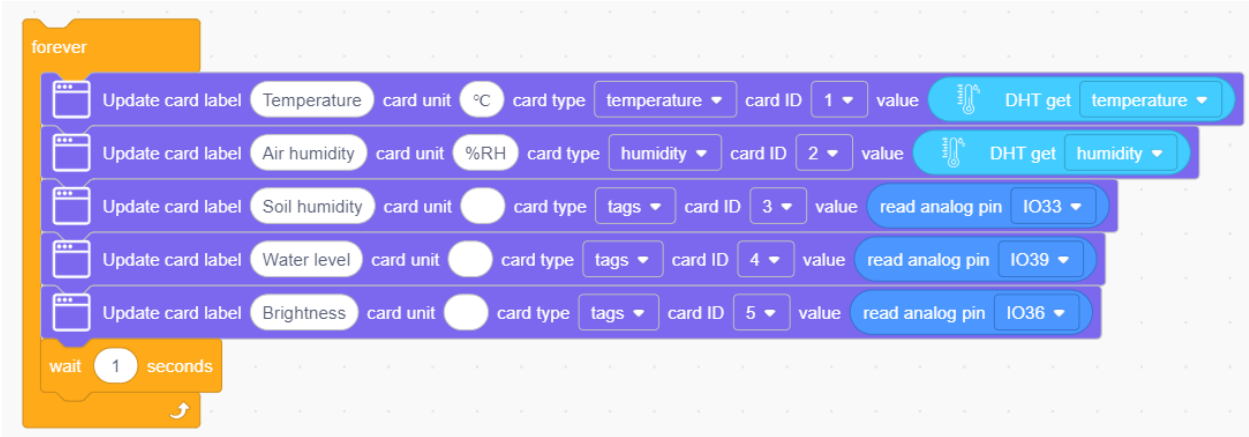
Initialize the pin IO16 of the LED module, the pin IO26 of the relay module and the pin IO2 of the passive buzzer, and set them to "**Output**" mode.



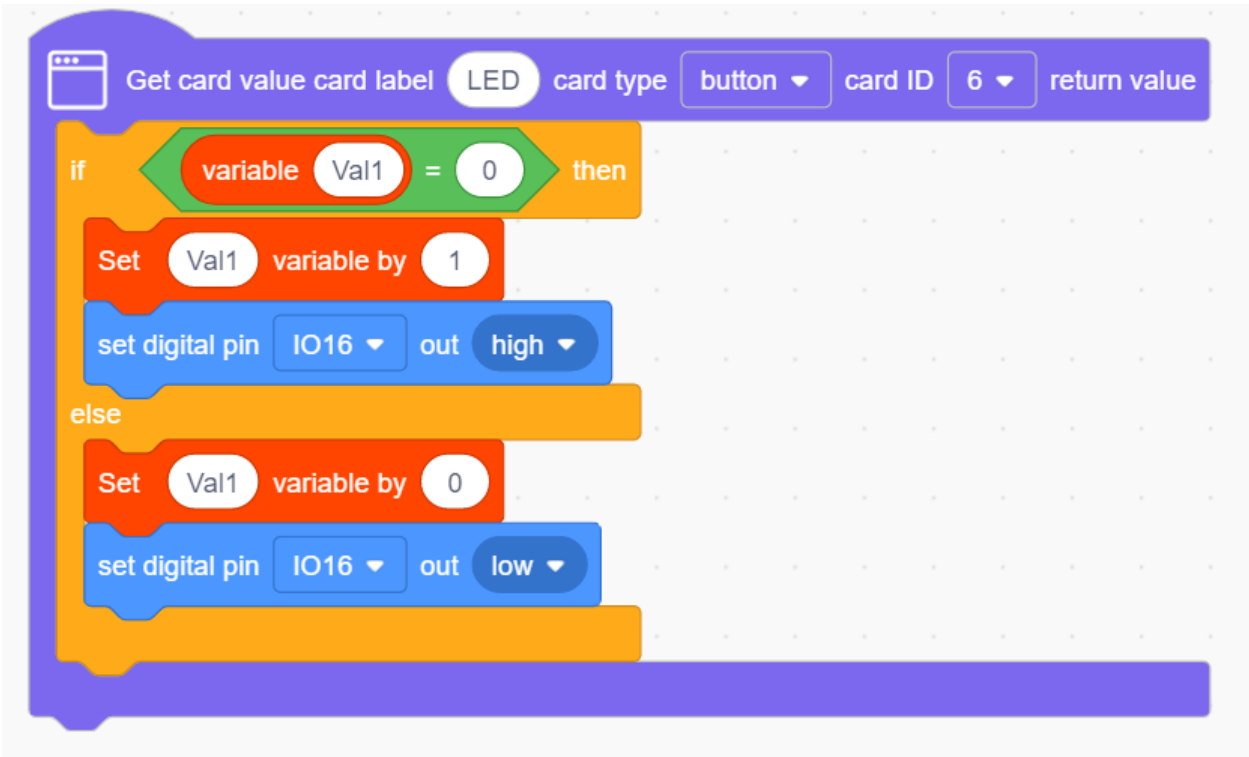
Define four global variables, namely "Val1", "Val2", "Val3" and "Val4".



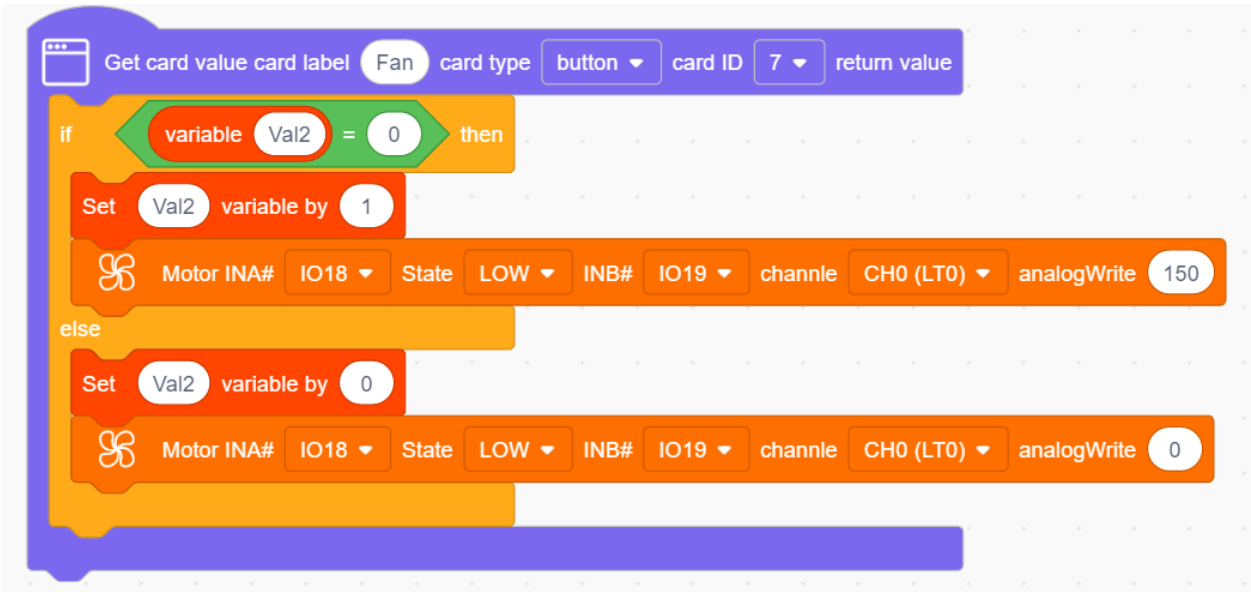
Set up multiple cards of the web page, namely TemperatureAir humiditySoil humidityWater level and Brightness.



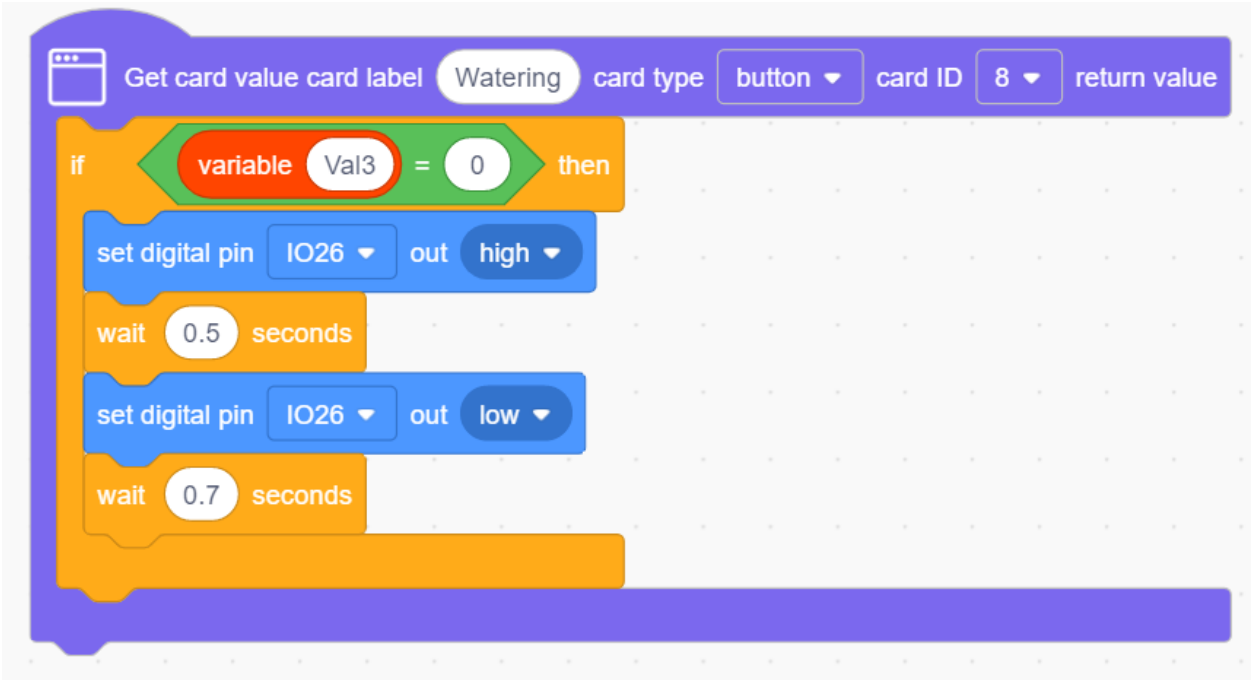
Set the button card to control LED on and off.



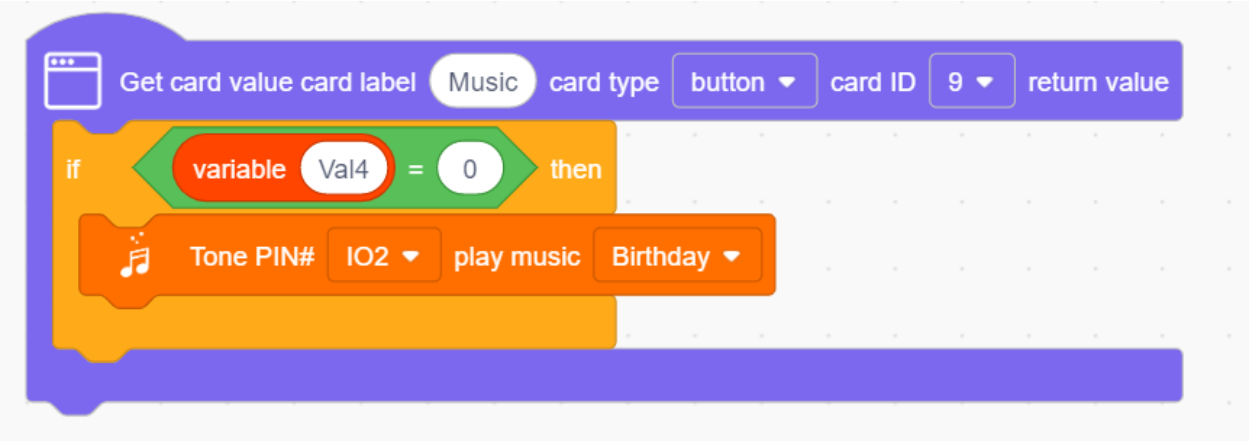
Set the button card to control the fan on and off.



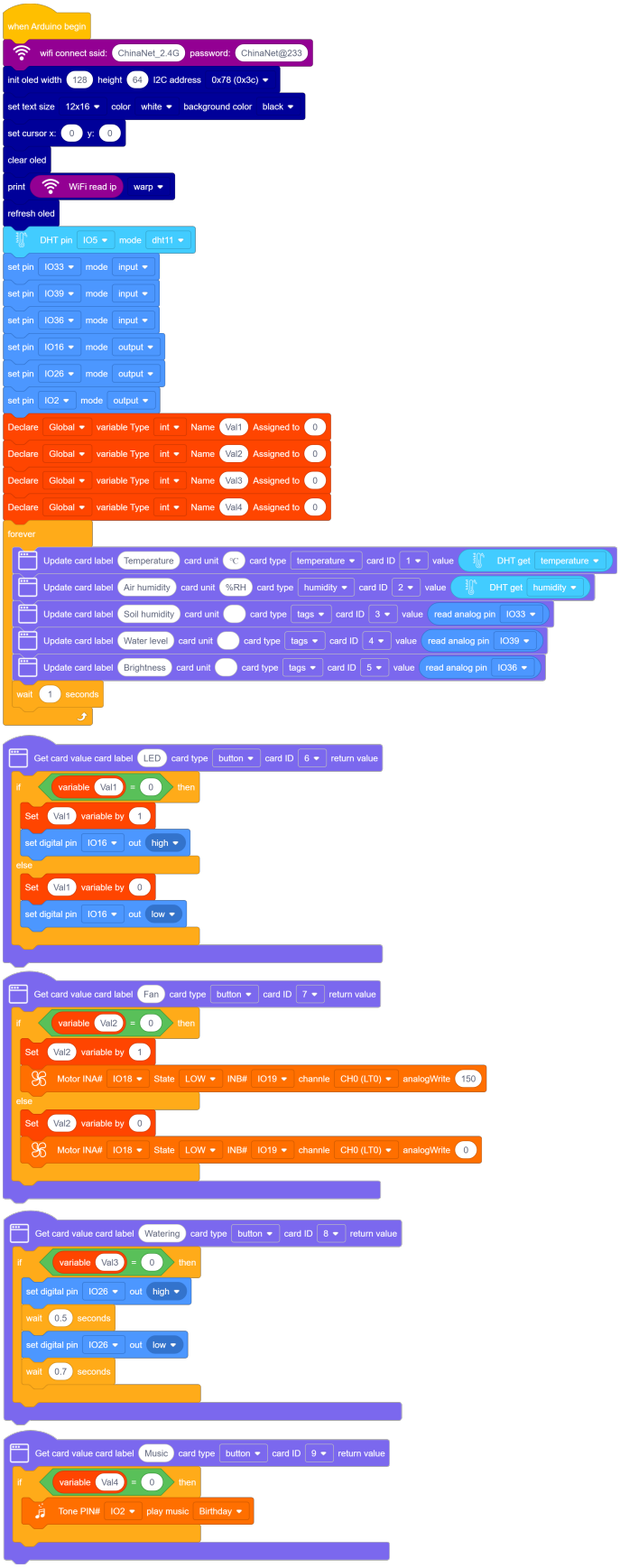
Set a button card to control the relay's on and off, thereby controlling whether the water pump pumps water.



Set the button card for playing music.



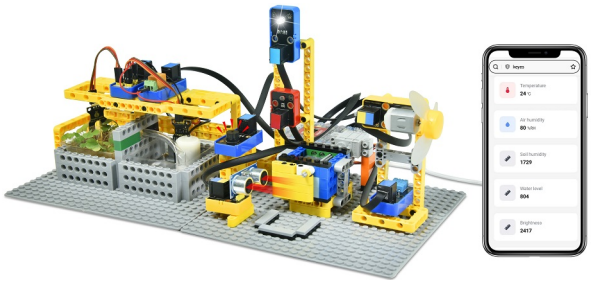
Complete Program


when Arduino begin
wifi connect ssid: ChinaNet_2.4G password: ChinaNet@233
init oled width 128 height 64 I2C address 0x78 (0x3c)
set text size 12x16 color white background color black
set cursor x: 0 y: 0
clear oled
print WiFi read ip warp
refresh oled
DHT pin IO5 mode dht11
set pin IO33 mode input
set pin IO39 mode input
set pin IO36 mode input
set pin IO16 mode output
set pin IO26 mode output
set pin IO2 mode output
Declare Global variable Type int Name Val1 Assigned to 0
Declare Global variable Type int Name Val2 Assigned to 0
Declare Global variable Type int Name Val3 Assigned to 0
Declare Global variable Type int Name Val4 Assigned to 0
forever
Update card label Temperature card unit °C card type temperature card ID 1 value DHT get temperature
Update card label Air humidity card unit %RH card type humidity card ID 2 value DHT get humidity
Update card label Soil humidity card unit card type tags card ID 3 value read analog pin IO33
Update card label Water level card unit card type tags card ID 4 value read analog pin IO39
Update card label Brightness card unit card type tags card ID 5 value read analog pin IO36
wait 1 seconds
Get card value card label LED card type button card ID 6 return value
if variable Val1 = 0 then
Set Val1 variable by 1
set digital pin IO16 out high
else
Set Val1 variable by 0
set digital pin IO16 out low
Get card value card label Fan card type button card ID 7 return value
if variable Val2 = 0 then
Set Val2 variable by 1
Motor INA# IO18 State LOW INB# IO19 channle CH0 (LT0) analogWrite 150
else
Set Val2 variable by 0
Motor INA# IO18 State LOW INB# IO19 channle CH0 (LT0) analogWrite 0
Get card value card label Watering card type button card ID 8 return value
if variable Val3 = 0 then
set digital pin IO26 out high
wait 0.5 seconds
set digital pin IO26 out low
wait 0.7 seconds
Get card value card label Music card type button card ID 9 return value
if variable Val4 = 0 then
Tone PIN# IO2 play music Birthday

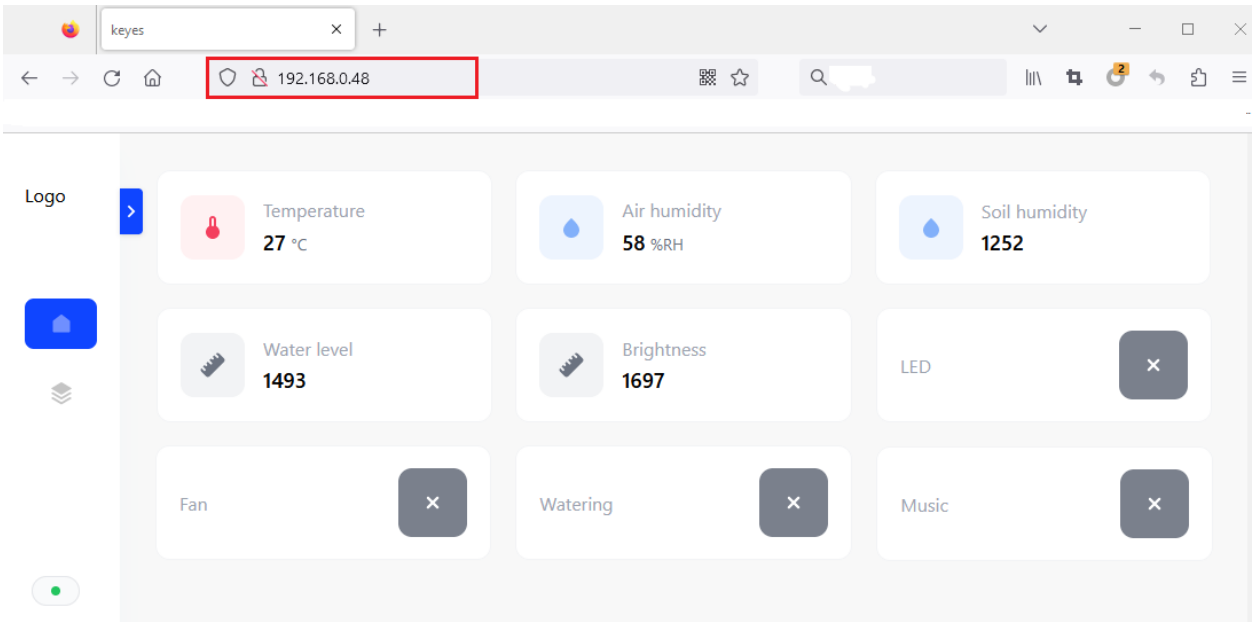
(2). Test Result

Click 

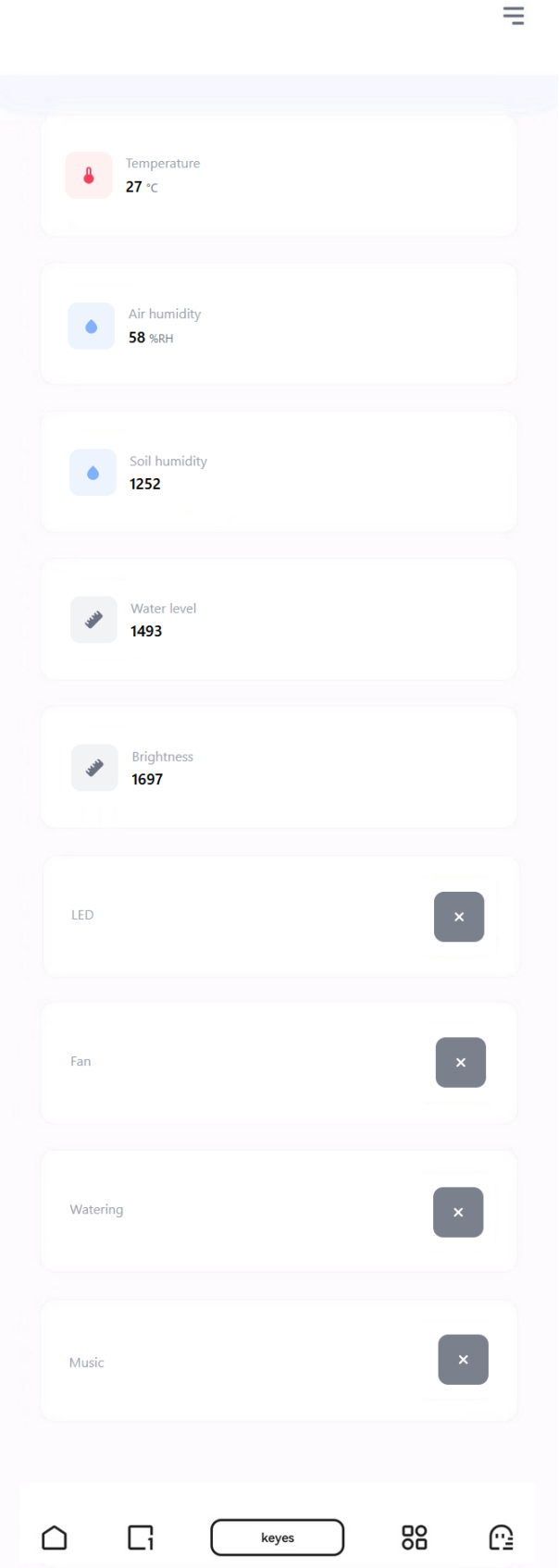
to upload the above complete code to the kidsIOT mainboard and power on. Once connected to WiFi, the OLED on the board will display the corresponding IP address. You can enter "**http://[IP address displayed on the OLED]**" in the browser to view the web page . The sensor data can be viewed on the web page, and LEDs and fans can also be controlled.



PC



Mobile phone


Temperature
27 °C
Air humidity
58 %RH
Soil humidity
1252
Water level
1493
Brightness
1697
LED
Fan
Watering
Music
keyes

Sensor data	Control
Ambient temperature(℃)	LED on/off
Ambient humidity(%RH)	Fan on/off
Soil moisture	Water pump on/off
water level of sink	Play music
Light(0~4095)	

7. Common Problems

Q1You can't connect the WiFi?

A: Please move the kidsIOT board to the vicinity of the router, press the reset button on the board to restart kidsIOT, and wait patiently for the connection. If you still can't connect, please check whether the WiFi name and password are filled in correctly.

Q2When remotely operating other sensors on the web page, the response is very slow?

A: Reasons:

- When multiple people are connecting it, the router's CPU resources are insufficient. Restart the router and reconnect to the network.
- The router system has been used for too long. Restart the router.
- When wireless interference occurs and the wireless signal is unstable, please do not use it through the wall.

For router related knowledge, please search on **google** yourself.