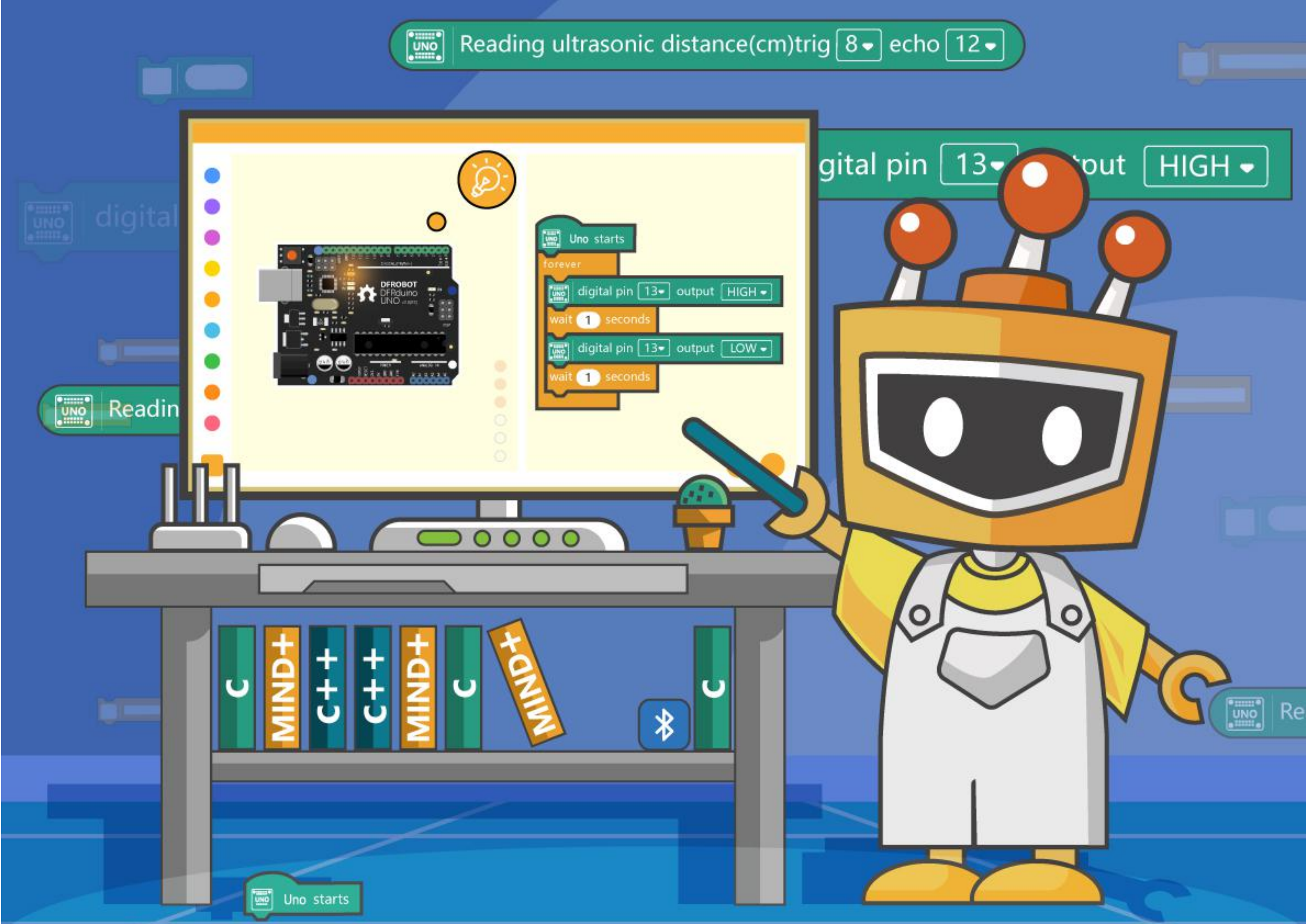
DFROBOT
DRIVE THE FUTURE

MindPlus Coding Kit for Arduino Started Tutorial



Reading ultrasonic distance(cm)trig 8 echo 12
Uno starts
forever
digital pin 13 output HIGH
wait 1 seconds
digital pin 13 output LOW
wait 1 seconds
C
MIND+
C++

www.DFRobot.com

Table of Contents

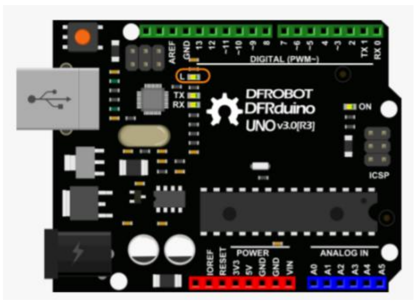
Introduction to Arduino

Chapter 1 Hello world

Lights illuminate our lives. Be it a candle, kerosene, incandescent, or energy-saving bulbs, lamps are a standard for every home, and our most common household appliances. Nowadays, there are a lot of methods on the market to control the on/off of a lamp, including switches, voice control, or other methods. How can we realize that with Arduino?
In this lesson, let's learn how to use Arduino to light up LED lights.

Project 1 Lighting up Onboard LED Light

First let’ s write one of the simplest codes to light up the onboard LED lights, which are the LED small lights labeled L on the UNO board (pin13). This step can help us learn how to download the program and test if our Arduino main-board works properly.
Our first task is to make this LED light blink.



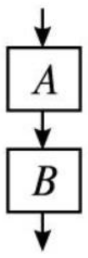
Task Navigation

1. Experience the programming steps of Mind+
2. Learn sequential structure and loop structure

Key Points Analysis

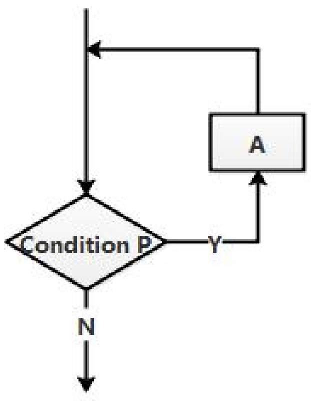
1. What is a sequential structure?

Designing a program of sequential structure is very easy. You only need to write the corresponding statements by the order of solution to the problem. It executes from top to bottom and executes sequentially. It will not execute B before A.

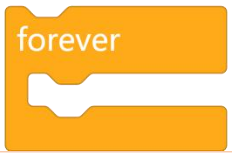
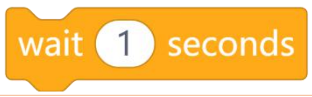
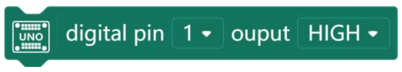


2. What is a loop structure?

A loop structure is designed for programs that need to repeatedly execute a certain function. There are three modules related to the loop structure in Mind+. They are infinite loop, finite loops, and until loop.



Command List

Module	Type	Description
	Infinite loop	An infinite loop is an infinitely repeated loop body. That is, the script contained in the module is repeatedly executed.
	Time Control	Wait for a certain duration before executing the next building block module.
	Pin settings	Set digital pin output to high / low. High level means the LED is on, low level means the LED is off

Hands-On

1. Mind+ Introduction

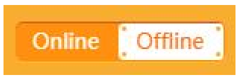
Mind+ is a graphical Programming platform that can support all kinds of open source hardware such as Arduino,micro: bit. It is not only suitable for primary and secondary school students, but also able to provide a great learning environment for makers who want to improve themselves by studying high-level Programming language such as Arduino, python, c, c++ etc. Drag and combine code blocks to make programs, easy to find the joy of creating. From beginners to experienced makers, Mind+ can satisfy all your needs: learning Programming, running programs without downloading, experiencing IoT(Internet of Things), sharing ideas with Mind+ community and so on. Come and play with Mind+, more surprises waiting for you!

2. Setup

1) Go to the Mind+ official website http://mindplus.cc/download-en.html, click download, and install. With a successful installation, we can see a Mind+ icon on the desktop.

Mind+
Home
Download
Discover
English
FREE
MIND+
The only thing and everything for kids to start hardware programming with creative mind.
Download
Code Online

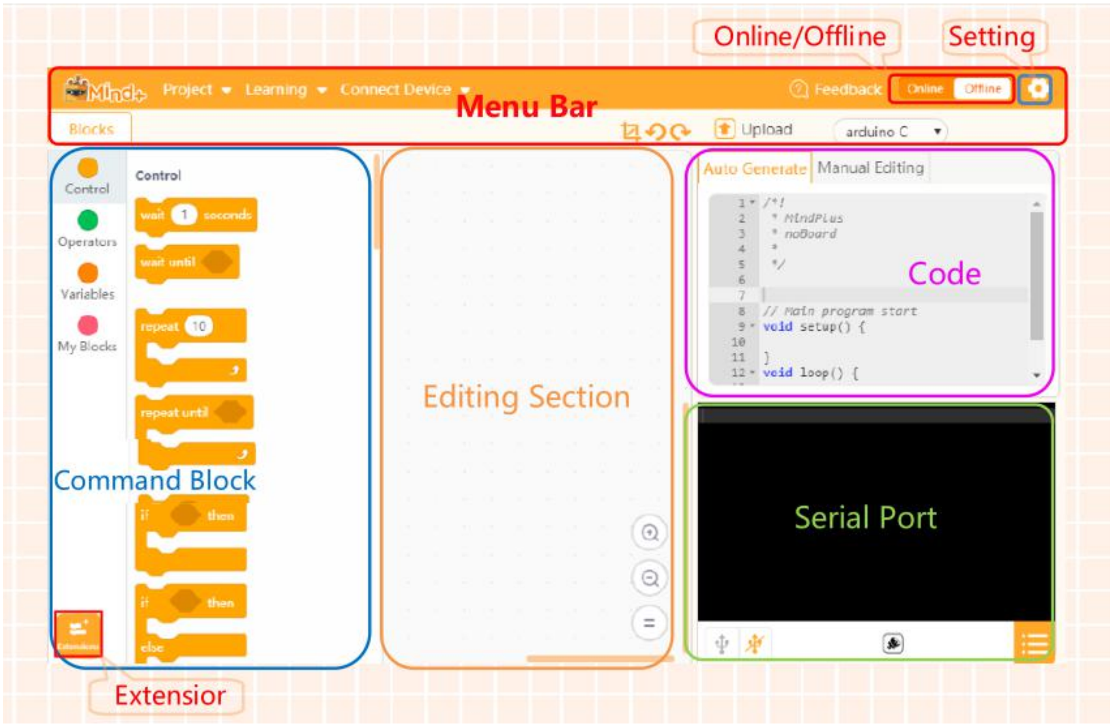
2) Double-click the icon on the desktop (as shown below) to open Mind+ software and switch the mode to "Offline Mode".



Double-click the icon to open the software.

Click to switch to "Offline Mode"

To convenient the operation, first we should be familiar with the basic functions of the soft-ware’ s interface. If we compare the Mind+ software as a stage, what are the functions of its different sections?

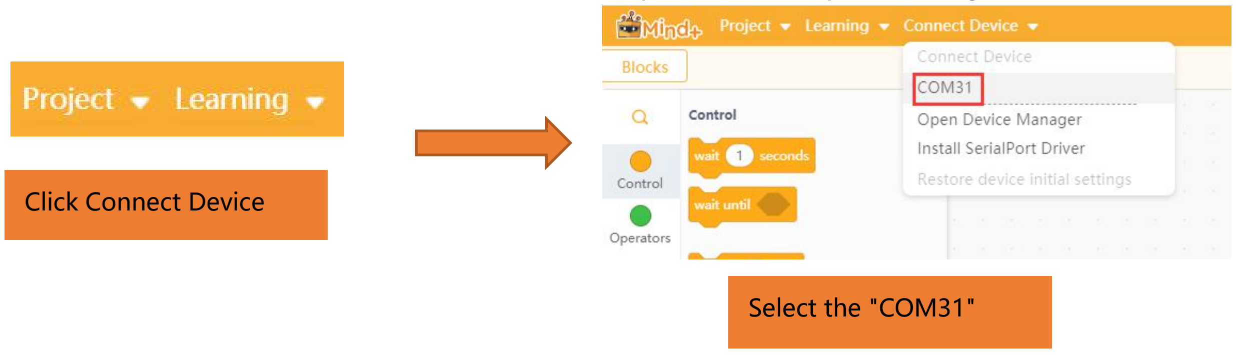


```
/*!
 * MindPlus
 * nobaord
 *
 */

// Main program start
void setup() {
}
void loop() {
```

Name	Function
Menu Bar	Project: create a new project, load a project, save a project, save a project as. Learning: where you can find lots of useful tutorials and examples programs. Connect Device: detect the connected device, and you can select to connect or disconnect the device.
Online/Offline	Online/Offline icon:switch mode to run program. Online: The executable program in the script area is executed in real time on the hardware and Mind + stage. Offline: upload the program to the hardware device for execution;
Setting	Setting: set the theme and language of the software; share and learn on our community or E-mail us your feedback.
Command Blocks	Blocks: This is the “tool section” of the stage. We need all kinds of tools to give an excellent performance on the stage.
Extensions	Expand icon: You can choose more additional props to support various hardware programming.
Editing Section	Programming Editor: Just drag the code blocks in the “Command” to here then you can program.
Code	Code: If you want to check the codes of the blocks in the “script” , this is the right place.
Serial Port	Serial port: open/close, scroll display open/close, clear output, baud setting, serialport input and output format control. Black area: can display the download status, for example, you can see the running status of the program, display serial communication data, etc.

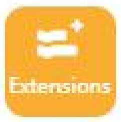
3) Connect device COMxx

Connect the Arduino main control board to computer’ s USB port through the USB cable.

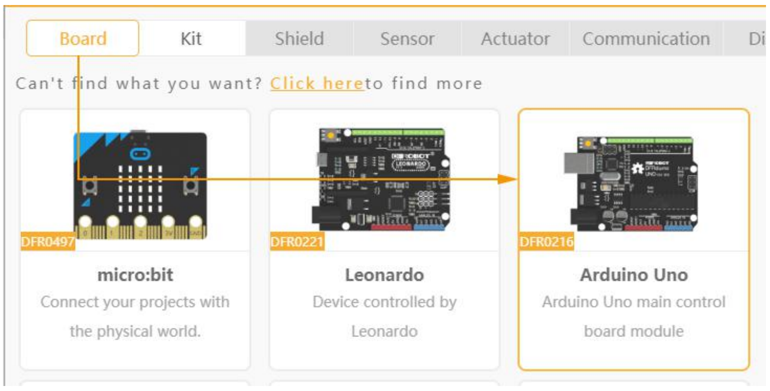


* If the port does not appear, please click “Install Serialport Driver” .

4) Select the main control board - Arduino Uno

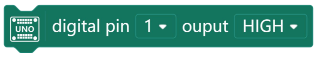
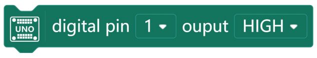


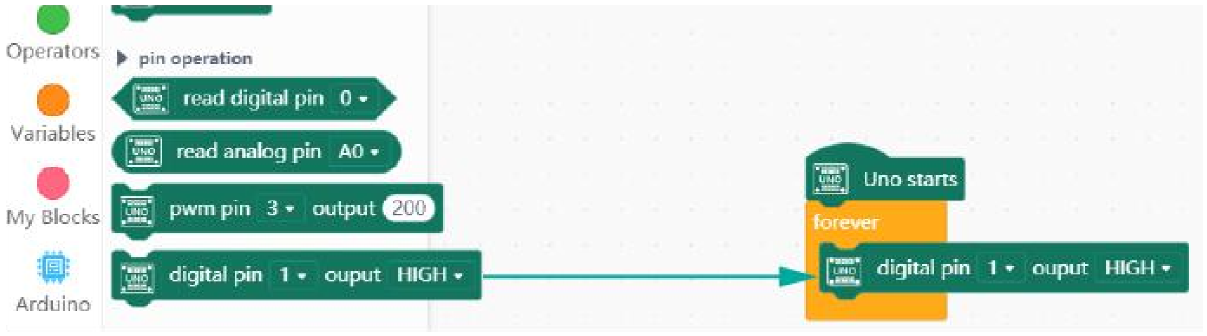
Click Extensions

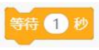
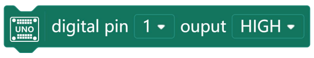

Board
Kit
Shield
Sensor
Actuator
Communication
Can't find what you want? Click hereto find more
DFR0497
micro:bit
Connect your projects with the physical world.
DFR0221
Leonardo
Device controlled by Leonardo
DFR0216
Arduino Uno
Arduino Uno main control board module

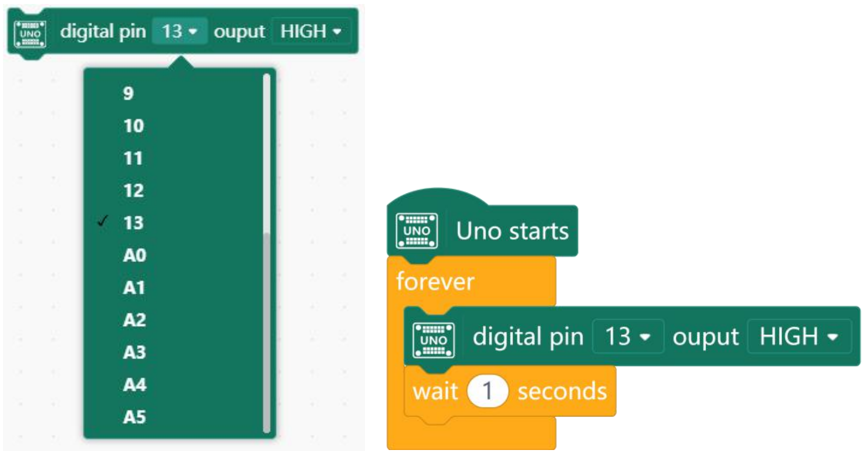
Select board - Arduino Uno

3. Write the program

1) Find the "  " command in the "Arduino" module and  drag it to the editing section on the right.

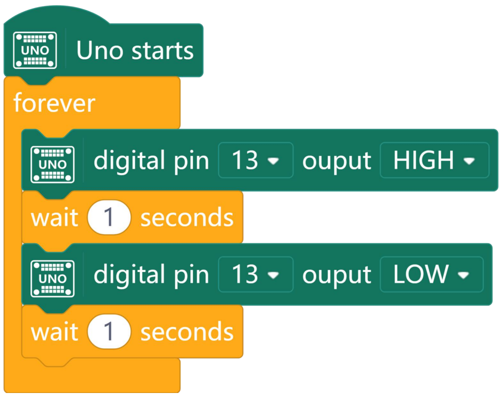


2) Find the "  " command in the "Control" module, drag and drop it to the editing section and attach to "  " command. Note that you need to switch the pin to "13", and the onboard LED it controls is number 13.



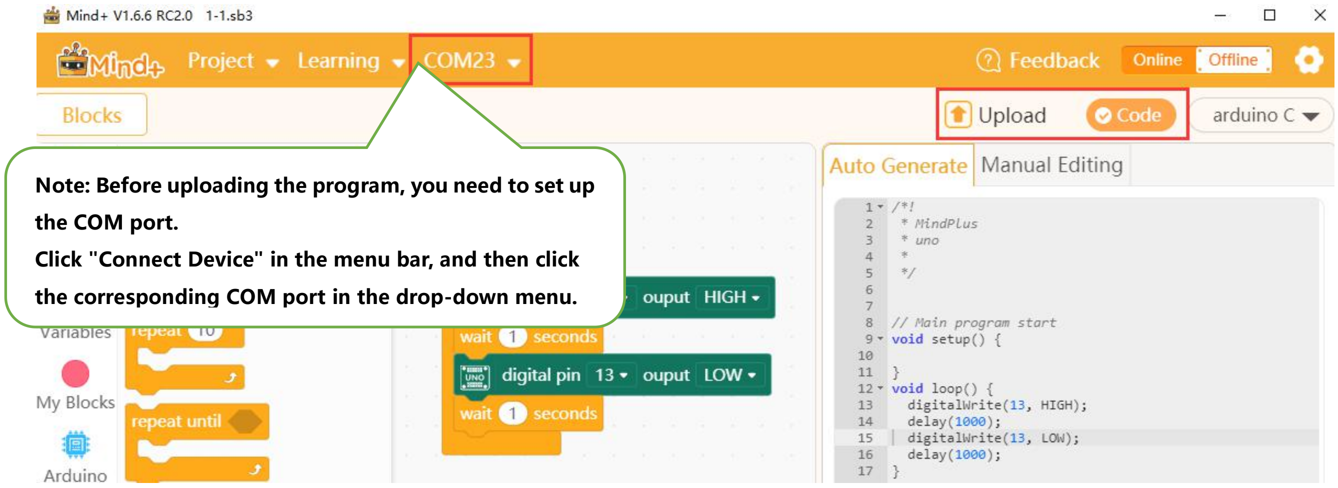
4. Program bulb to blink

The blink effect is achieved by setting the LED to on and off, so you need to make it lighten for some period of time and off as well too.



5. Upload program

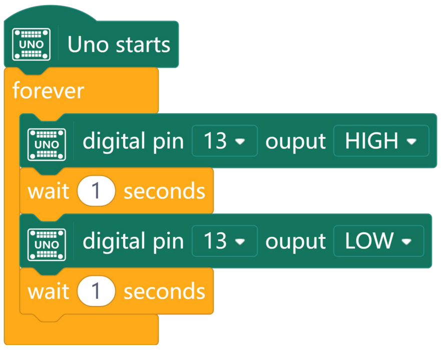
Upon completing the program, click “Upload to device” as shown below to upload the program to the Arduino mainboard.



Note: Before uploading the program, you need to set up the COM port.
Click "Connect Device" in the menu bar, and then click the corresponding COM port in the drop-down menu.

```
/*!
 * MindPlus
 * uno
 *
 */


// Main program start
void setup() {

}
void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

6. Reference program



7. Program effect

The onboard LED light L flickers every 1 second.

8. Program analysis

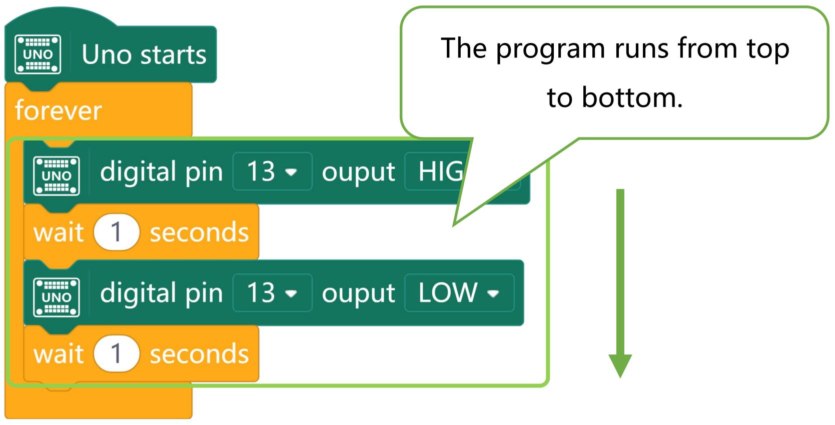
Arduino mainboard controls elements in digital output or analog output.

Here we are dealing with digital output first. It sends a digital signal to the output circuit, as it switches the circuit on and off by sending 0 or 1.

Its circuit state is as follows:

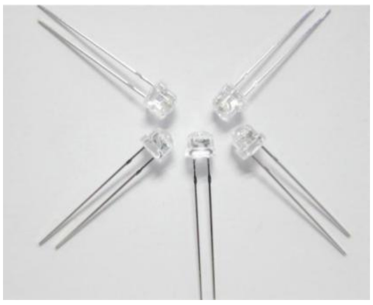
Digital signal	Output	Circuit state
0	Low level	Open
1	High level	Closed

In the reference program, the digital output of pin 13 is set to High level, as a result, the onboard LED(L) bulb connected to it is turned on. The program goes through 1 second delay then (**during the delay, the hardware will maintain the state before the delay**); at the end of the delay, set the digital output to low to turn off the LED and keep it off for 1 second.
Afterwards, we can see that the on-board LED light is turned on again after being off for 1 second, and then off again after 1 second, and repeats the pattern as it stays on for 1 second and off for 1 second. This is because the program is executed in an infinite loop from top to bottom.



Project 2 Turn on the external LED light module

In Project I, we learned how to light up the UNO onboard LED light, and then we will learn how to control an external LED light.



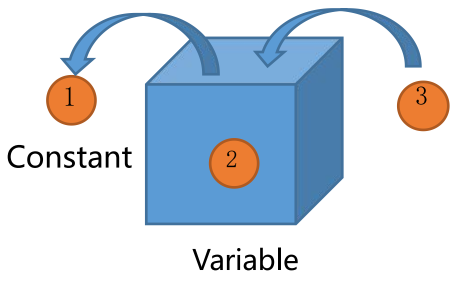
Task Navigation

1. Learn variables and constants
2. Make an LED light that flickers at an increasing rate.

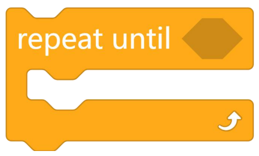
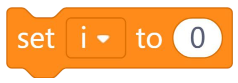
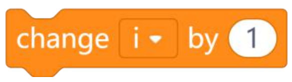
Key Points Analysis

What are variables and constants?

A variable holds a value that can change or vary in a changeable process.
A constant holds value that stays the same during a process of change.
Here, we can think the variable as a box for storing data, and it can store a variety of data temporarily. For example, we use it to store integers. After 1 is stored, 2 is stored; 1 will be replaced by 2 and we can only get 2 from the box. In the same way, if you store 3,4,5 ... the value will be replaced by latest data in order.
Constants can be understood as a certain data in the box. For example, the number in the lottery box. When you draw the number 10, this number is a constant.



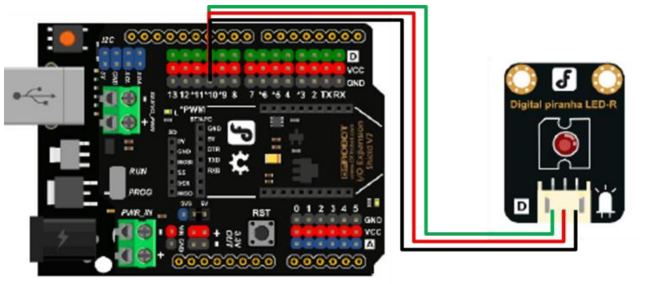
Command List

Module	Type	Description
	Conditional statement: repeatedly execute until	Keeps executing the subroutine until the conditions in ⬢ are met.
	Numeric type variables	Create a numeric variable
	Set variable value	Set the initial value of a variable
	Set variable value	Set the variable to increase or decrease by step of 1
	Operator	The “less than” operator is used to determine whether the left side is less than the right side.

Hands-On

1. Hardware connection

LED lights are solid-state semiconductors that convert electrical energy into visible light. When connecting ordinary LED lights and Arduino, a protection resistor in series connection is required. Since the LED light emitting module in the figure below integrates LED lights and protection resistors, and leads to a 3-pin interface, we can directly connect the 3-pin interface from the LED light module to the Digital Pin 10 of the Arduino mainboard.

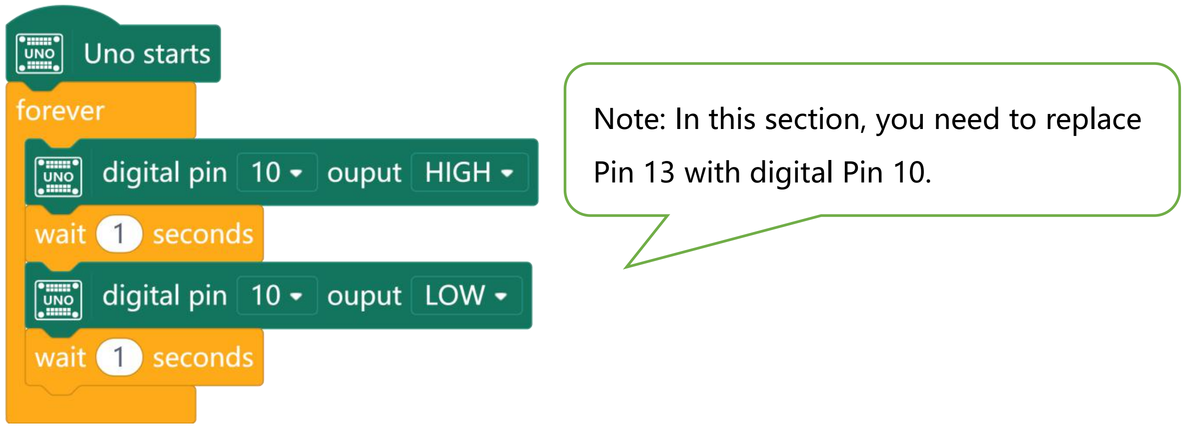


Hardware connection (LED -10)

Please match the colors when plugging

2. Programming

Modify the pins according to the reference program of the previous task.

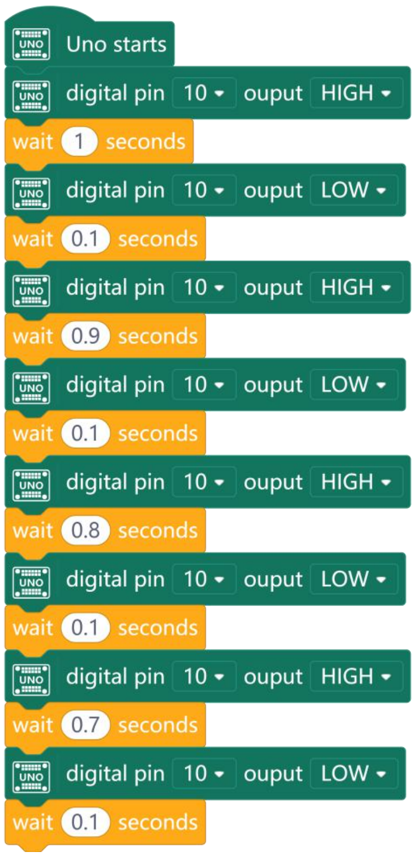


```
Uno starts
forever
    digital pin 10 ouput HIGH
    wait 1 seconds
    digital pin 10 ouput LOW
    wait 1 seconds
```

Note: In this section, you need to replace Pin 13 with digital Pin 10.

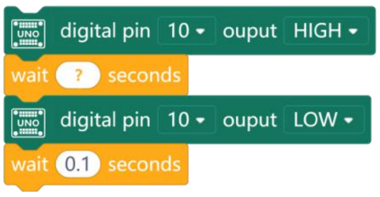
Upload the modified program to the Arduino main control board, and you can see the flickering LED lights. The current project now makes the LED light flicker every second. What if I want to make the LED light flicker faster and faster? How should it be achieved?

With two previous projects, we get that flickering effects are achieved by delay. To make the LED light flicker faster and faster, we need to gradually shorten the delay time for the small bulb to be on, thereby causing the flicker effect to go faster. Let’ s have a look at this program.



```
Uno starts
digital pin 10 ouput HIGH
wait 1 seconds
digital pin 10 ouput LOW
wait 0.1 seconds
digital pin 10 ouput HIGH
wait 0.9 seconds
digital pin 10 ouput LOW
wait 0.1 seconds
digital pin 10 ouput HIGH
wait 0.8 seconds
digital pin 10 ouput LOW
wait 0.1 seconds
digital pin 10 ouput HIGH
wait 0.7 seconds
digital pin 10 ouput LOW
wait 0.1 seconds
```

It’ s easy to see that this program consists of a repeated small piece of program. According to the principle of sequential execution described earlier, a set of modules will be executed from top to bottom. If we want to execute a lot of times, such as 100, are we really going to using 100 such blocks?



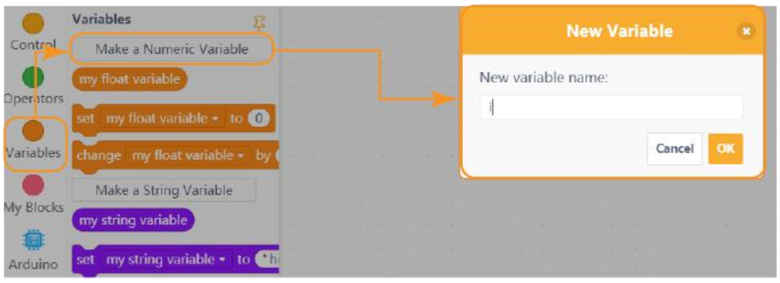
```
digital pin 10 ouput HIGH
wait ? seconds
digital pin 10 ouput LOW
wait 0.1 seconds
```

We know the program will be extremely lengthy just by thinking of it. Maybe you don’ t even remember how many times you’ ve dragged at the end. So, is there a function that can repeat the above process automatically? Yes, we are going to talk about it next.

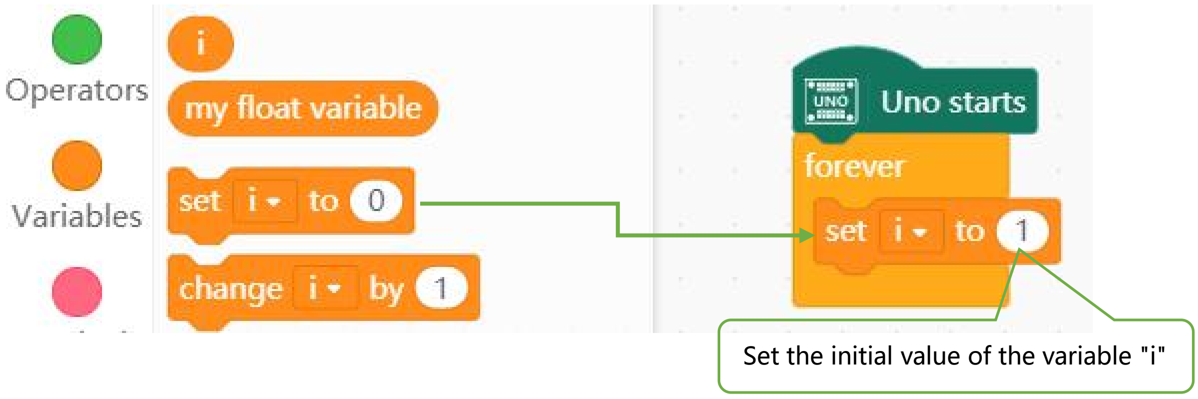
1) **Create variables**

Step 1. Find the variable module

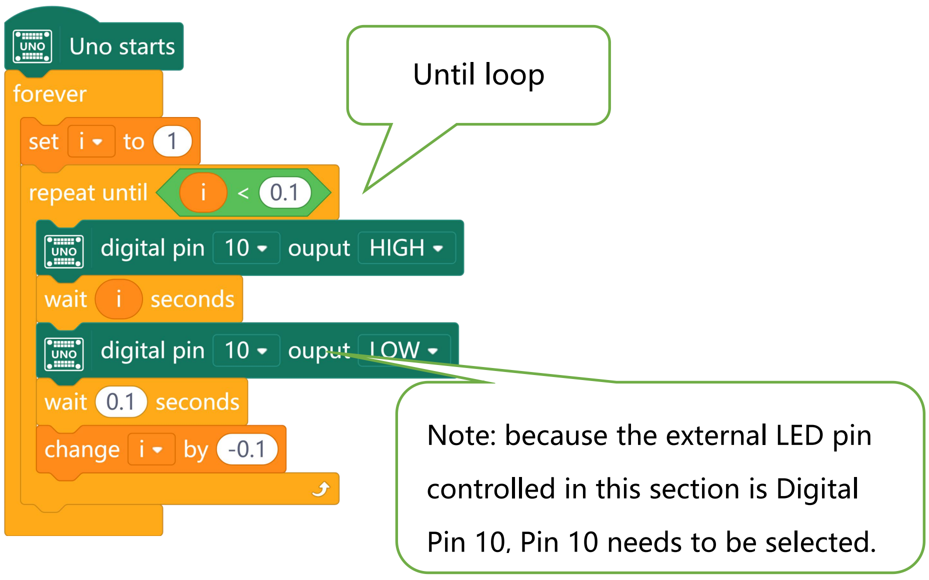
Click " Make a Numeric Variable " to create the variable "i"; the steps are as shown below.



Step 2. Find the " set i to 0 " block and drag it to the position as shown.



Step 3. After creating the variable, we can set up the number of the loops. This process is determined by "until loop" (conditional loop).

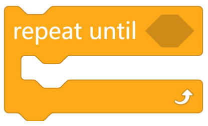

Uno starts
forever
set i to 1
repeat until i < 0.1
digital pin 10 ouput HIGH
wait i seconds
digital pin 10 ouput LOW
wait 0.1 seconds
change i by -0.1
Until loop
Note: because the external LED pin controlled in this section is Digital Pin 10, Pin 10 needs to be selected.

3. Program' s effect

The LED light will flicker faster and faster.

4. Program analysis

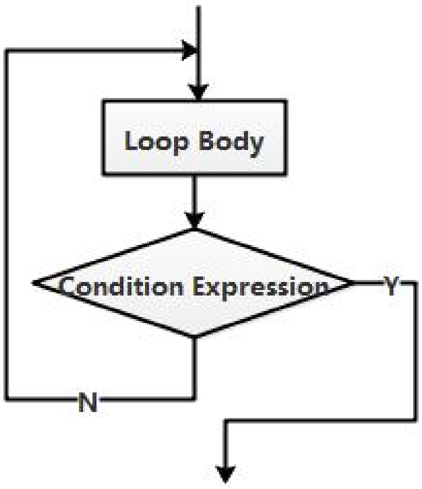
The above program implements the idea of "repetition". It is the commonly used "until loop" structure in programming.



Until loop structure in Mind+

In the first section, we introduced that until loop is a conditional loop. It means that after executing the loop body once, the condition is determined (the value of i is determine here), the execution is continued if the condition is not met, and the loop only ends when the condition is met.

The flowchart is as follows:

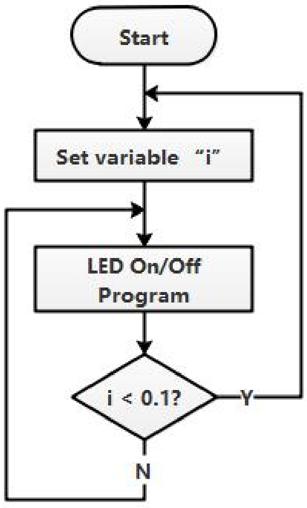


Until loop flowchart

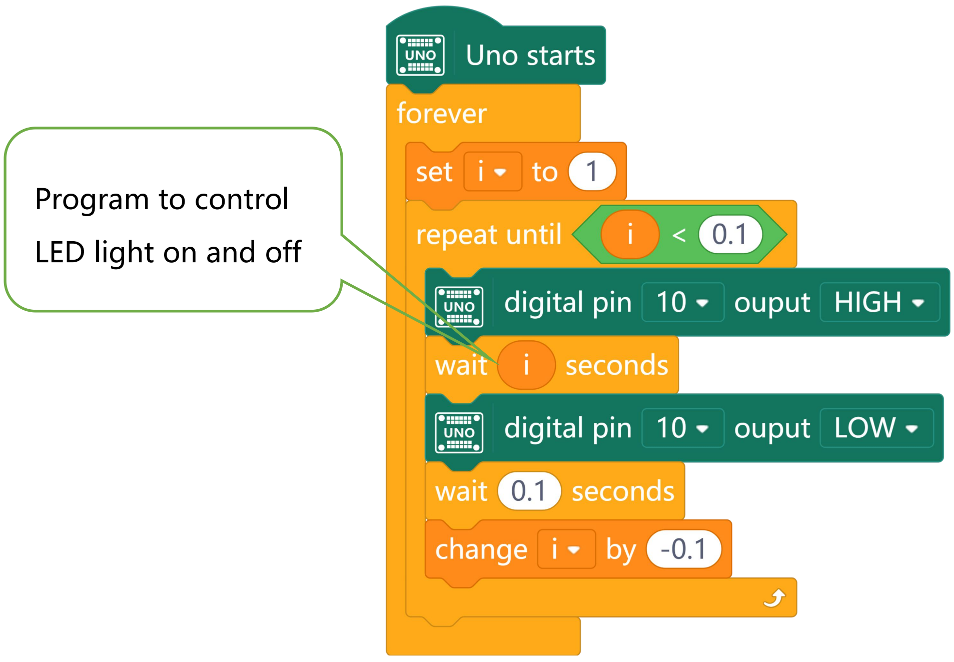
So from the program we can see that the number of loops is controlled by the value of the variable "i". During the execution, the variable "i" starts from 1 and is decremented by 0.1 step size each time. (See further reading: What is the step size?)

So at the second execution, i=1-0.1=0.9; the third i=0.9-0.1=0.8, the fourth, i=0.8-0.1=0.7...

In this way, it is decremented until "i" <0.1, and the loop ends. The overall program flowchart is as follows:



Note: each cycle will execute the program once, which controls the LED to turn on and off.



```
Uno starts
forever
    set i to 1
    repeat until i < 0.1
        digital pin 10 ouput HIGH
        wait i seconds
        digital pin 10 ouput LOW
        wait 0.1 seconds
        change i by -0.1
```

Further reading

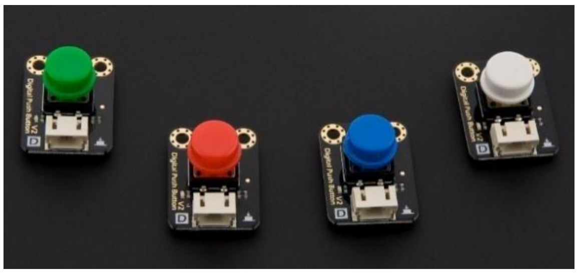
What is the step size?
The step size is the value of increment or decrement each time. If the step size is 1, the value is increased by 1 for each execution; if the step size is -1, the value is decreased by 1 for each execution. In the previous program, we set the initial value to 1. In order to make the LED light flicker faster and faster, we decreased it by 0.1.
Example: an operation is to be performed on the parameter N. Assuming that the step size is M, an operation is performed on N, then assign N + M to N. And then use the new N (N + M) value to perform the operation, and so on. As explained in the program above.

Chapter 2 Magic Buttons

In the three projects following up, we will learn how to perform simple controls by input module like a switch, and simulate some common life scenarios with buttons.

Project 3 Magic Button Switches

Button switches, also known as key switches, were called sensitive switches in the early days. They are widely applied in lights, socket master switches, doorbells, and car consoles, etc. Button switches add another protective layer to the usage of electricity. As they facilitate the controls of electrical appearance, they also further the protection of electrical components.
As we learn Arduino, we will be exposed to a variety of input devices. Among them, the button switch is the simplest and most widely used one. Here we will use Arduino to control the LED lights to make button pressing switch the light on and off every other time.



Task Navigation

1. Learn the basics of digital input and conditional statement structure
2. Turn on the LED with the button

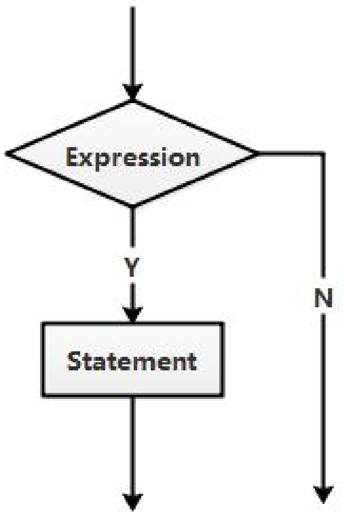
Key Points Analysis

1. What is a digital input?

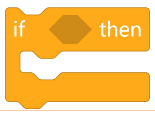
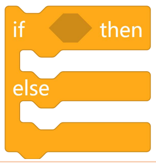
There are only two states, "high" or "low", which are called digital signals. Monitor the level "high" or "low" via digital pins.
Button pressed down means high, and button released it is low. At the same time, the button is a device that obtains external actions. We call it an input device.

2. What is a conditional statement structure?

The condition statement structure refers to a program structure that needs to decide if an operation to is be performed according to whether certain given conditions are met. Conditional statement is to check a condition and if that condition is met, execute.
Here we will learn the first structure of conditional judgment: if
Its semantics are: if the expression is true, then execute what follows; otherwise, the following statements are not executed. The process can be expressed as the following figure.

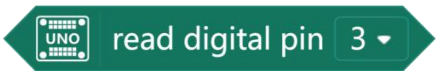
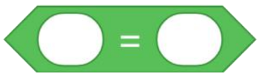


There are two types of modules related to conditional statement structure in Mind+: true conditional execution and true or false conditional execution.

Module	Type	Description
	True conditional execution	Perform a logical check on a single condition and execute the program.
	True or false conditional execution	Check a branch condition. If true, execute the first statement; if false, execute the second statement.

In this chapter we will use true conditional execution. We set a execution condition at , when the condition is met, following statements are executed.

Command List

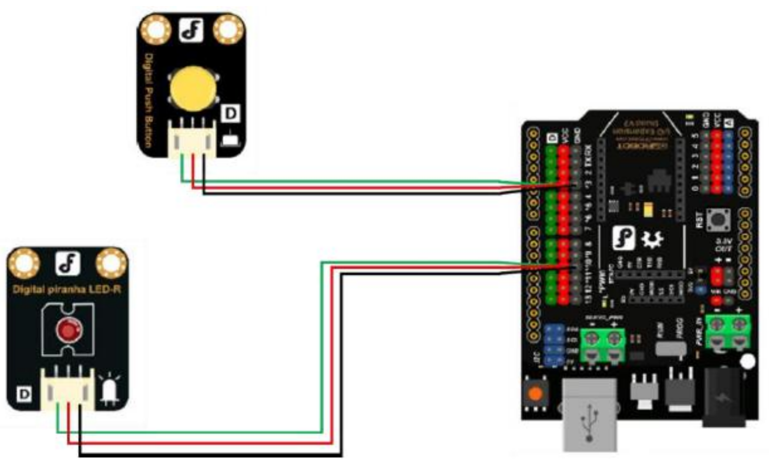
Module	Type	Description
	Read pin value	Read level at Pin 3 on UNO board to check 1 or 0 (high r low). In this project, the state of the button is read.
	Operator	The equal operator is used to determine if left and right sides are equal.

Hands-On

1. Hardware connection

The pins that support digital input in UNO are 2 to 13 and A0 to A5 (In fact, P0 and P1 also support digital input. However, they are often occupied by serial communication and thus not usually used.).

1. Connect button module: insert the 3-pin interface of the button module directly to the Digital Pin 3 of Arduino Uno. You can also choose other digital pins, but do remember changing the digital pins in the code to the corresponding pin numbers.
2. Connect the LED module: plug the 3-pin interface of the LED module directly to the Digital Pin 10 of Arduino Uno.

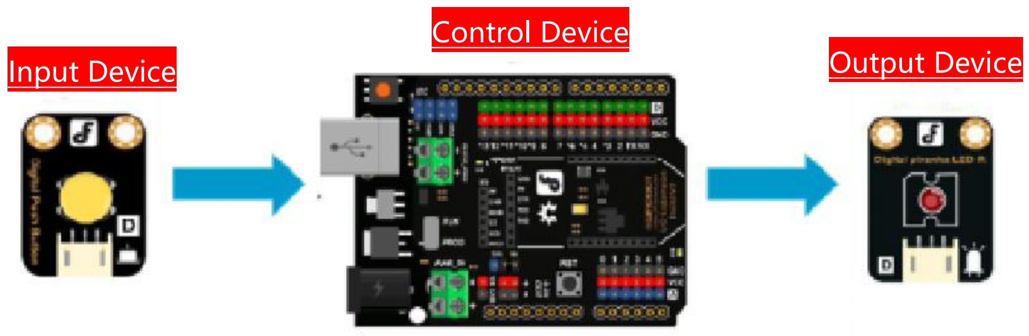


Hardware connection (LED-10, button-3)

Please match the colors when plugging

2. Programming

On the basis of controlling LEDs directly with UNO, we introduce a button to control the LED, forming a relationship of “input-control-output”, as shown below:



In theory, the input device sends a signal to the control device; the control device receives the signal and processes it, and then controls the output device to perform the corresponding output work.

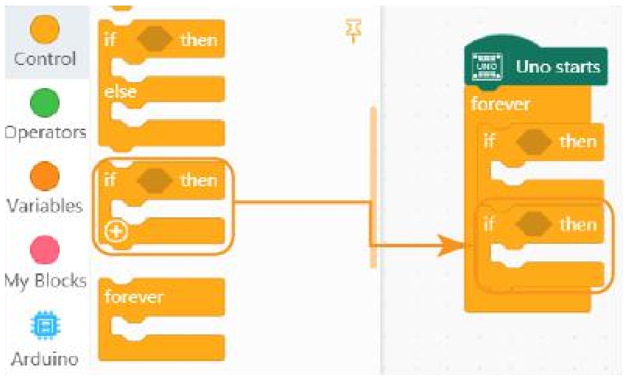
In button-controlled LED, the button module sends a signal (high / low level) to UNO; UNO receives the signal and processes the signal according to the command set by the code, and finally controls the LED to turn on / off. At this moment, we need to introduce conditional statement structure as a bridge between the button and LED control.

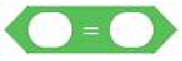
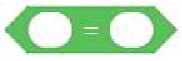
1. Write the program

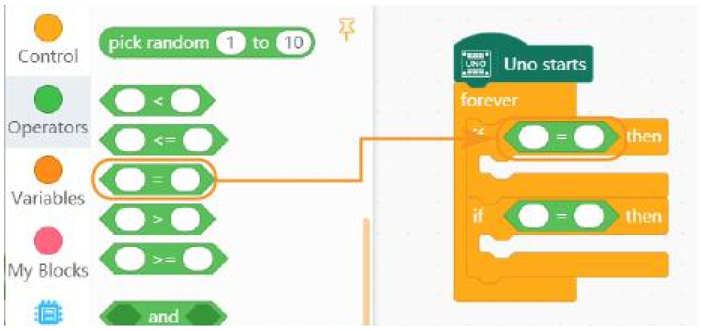
1) Find the "

" command in the "Control" module and

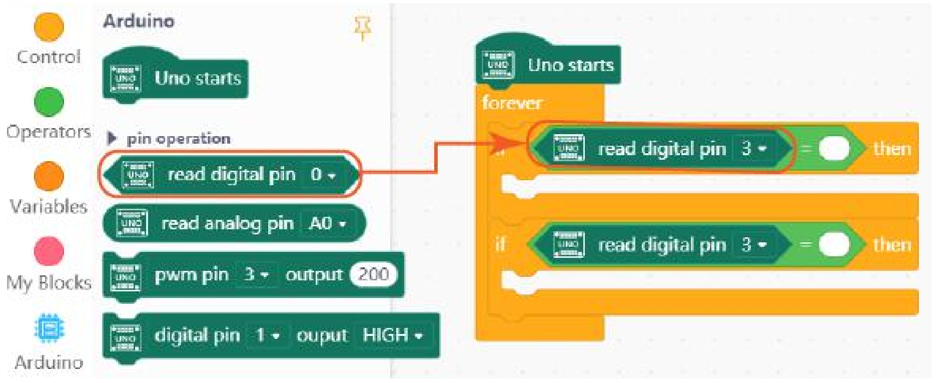
drag it to the editing pane on the right.



2) Find the "  " command in the "Operator" module and  drag it to the editing pane on the right.

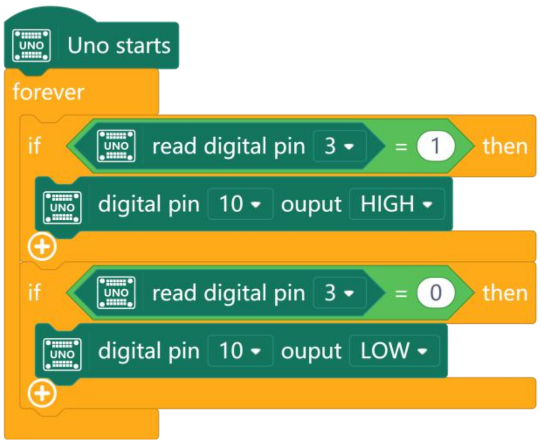


3) Find the " " command in the "Arduino" module , drag it to the editing pane on the right and combine it with the operator "=" (you can also combine it first and then nest it into the conditional statement structure).



Pay attention to changing “0” to the corresponding pin number. In the example, the button is connected to Pin 3.

2. Reference program



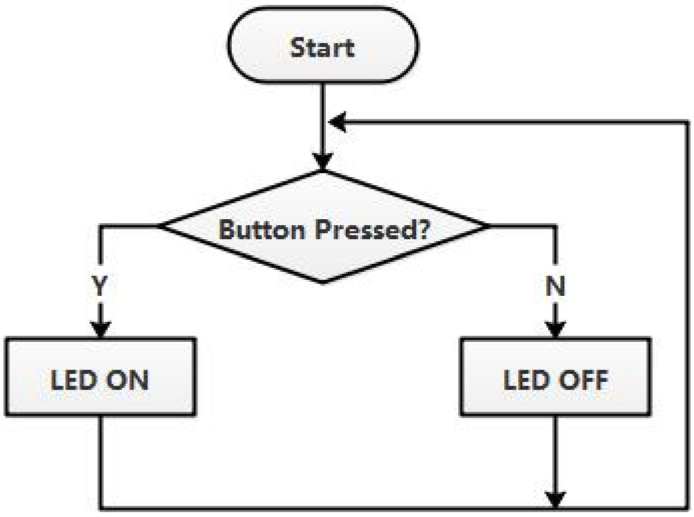
```
Uno starts
forever
  if read digital pin 3 = 1 then
    digital pin 10 ouput HIGH
  if read digital pin 3 = 0 then
    digital pin 10 ouput LOW
```

3. Program effect

Button pressed down, the LED is on; button released, the LED is off.

4. Program analysis

Conditional statement structure needs to check all input signals from input devices. We do we only have to set two conditions? The reason is that button input has only two cases, 1 and 0.



When the button is pressed, Pin 3 inputs high level and Pin 10 outputs high level accordingly; When the key is released, Pin 3 inputs low level, and Pin 10 outputs low level at this time.

Press Input-> Master-> Output Analysis

Button (input): button pressed down, Pin 3 connected to the button inputs high level; button released, Pin 3 inputs the low level.

UNO (Control): Digital Pin 3 reads high or low level, and Digital Pin 10 outputs high or low level accordingly.

LED (output): receives the high or low levels of UNO output; high means on and low means off.

Further Exercise

Now we know how to control LED with button switch, can we control a small speaker in the same way? What about a small fan?

Project 4 Simple Timer Light

What does a timer light do? Imagine this: it’ s getting late and you are going to bed. However, the lamp is some distance away from your bed. After turning it off, you have to feel your way in utter darkness to your bed. Students who sleep on the top beds in the dormitory have the same trouble.
In the previous project, we learned how to use buttons to control LED lights. Next, we will further apply the conditional statement structure to learn how to make simple timer lights.

Task Navigation

1. What is a conditional statement structure?
2. Make a timer LED light.

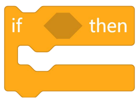
Hands-On

Hardware connection

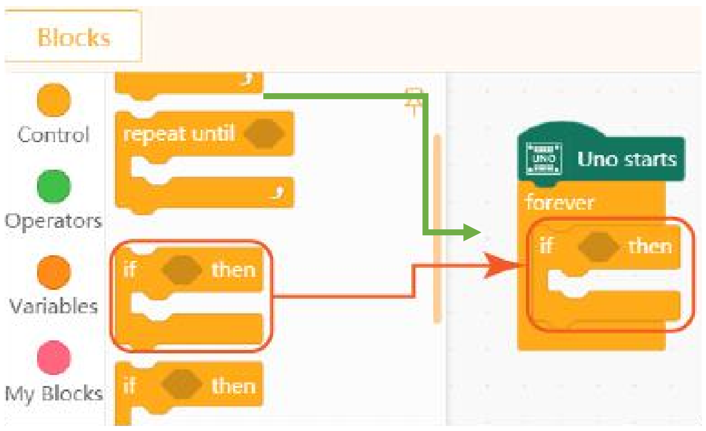
Refer to “Magic Button Switches” for circuit connection.

Programming

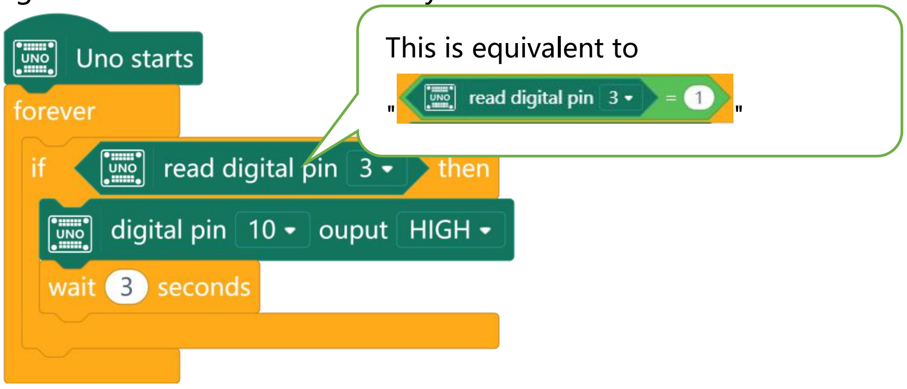
1. Write the program

1) First find the " 

" instruction in the "Control" module and drag it to the editing area on the right.



Similarly, first set up a conditional to read button switch status, and find " read digital pin 0 " in the "Arduino" module and embed it in the conditional structure. To keep LED on during the delay, first set the output to high level and then add the delay command.



```
Uno starts
forever
  if read digital pin 3 then
    digital pin 10 ouput HIGH
    wait 3 seconds
```

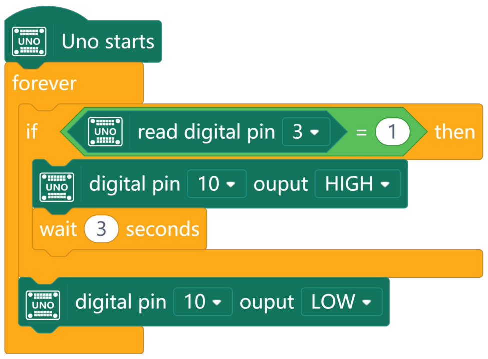
This is equivalent to
" read digital pin 3 = 1 "

2) Why we can use “to replace the previous” here? This is because the digital pin output only has two values of "0" and "1", and in C, "0" stands for "False" and "1" stands for "True".

For the " " conditional control module, if the condition is met, the value is True, which is equivalent to "1".

3) Because turning off the lights is the final result of the program execution and does not belong to the category of conditional, we place " digital pin 10 ouput LOW " outside the conditional structure.

2. Reference program



```
Uno starts
forever
  if read digital pin 3 = 1 then
    digital pin 10 ouput HIGH
    wait 3 seconds
  digital pin 10 ouput LOW
```

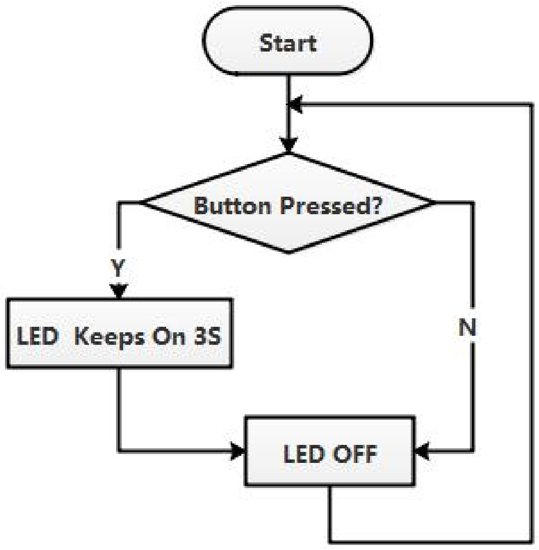
3. Program effect

After the button press, the LED turns on for a few seconds before going off.

4. Program analysis

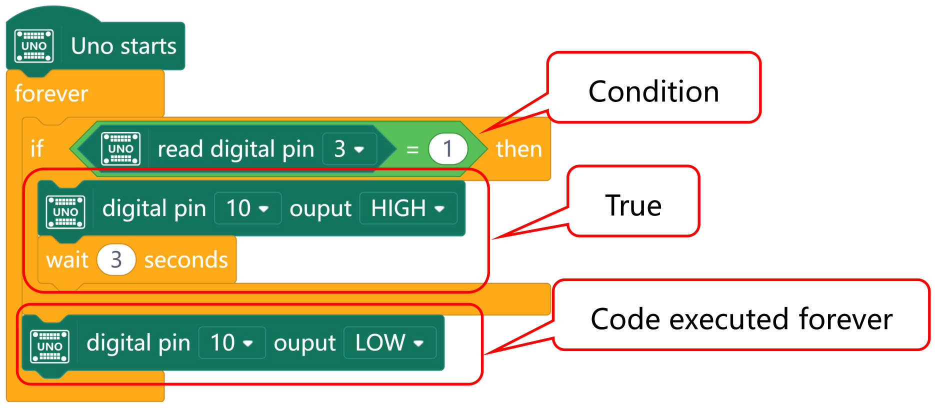
Just like using button to control LED in the previous chapter, it also deals with the relationship of "button-UNO master control-LED". The difference here is that it adds a delay before turning off and simplifies the switch function.

The logical relationship analysis is as follows:



Here is a simple application of the conditional structure, so how does it go?

If the current status meets the requirements of the conditional, the program in the conditional module (inside the green box) will be executed. If the condition is not met, the conditional module is skipped and the program inside the blue box is executed.



Further reading

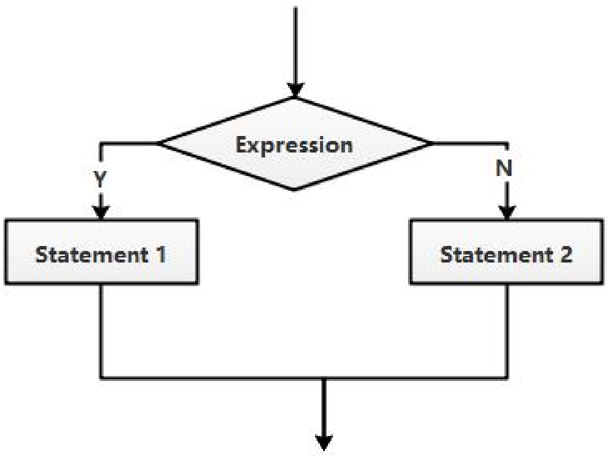
The other two forms of conditional structure can be referred to the code pane on the right side of Mind+.

1. The second form is: if-else

```
if (expression)
    Statement 1;
```

```
else
    Statement 2;
```

The semantics are: if the value of the expression is true, then execute statement 1, otherwise execute statement 2. The execution process can be expressed as the following figure.

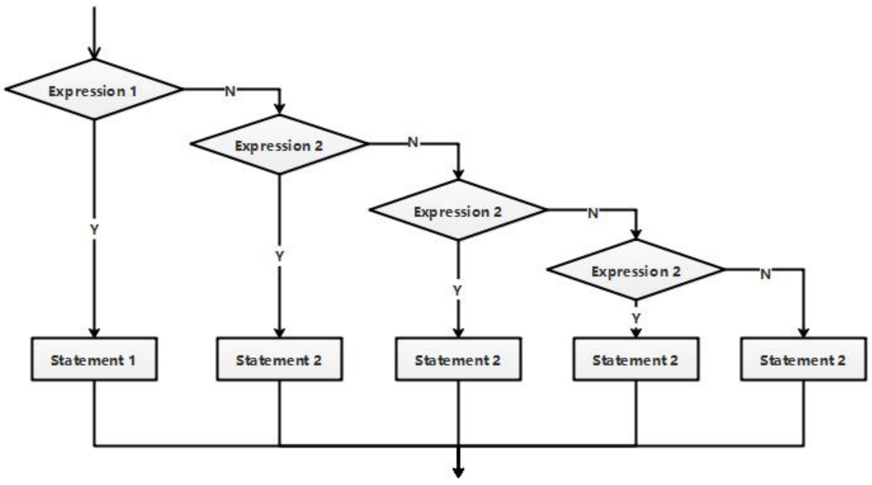


2. The third form is the if-else if.

The first two forms of if statements are generally used when there are two branches. When there are multiple branches, an if-else-if statement can be used, and its general form is:

```
 if (expression 1)
      statement 1;
else if (expression 2)
      statement 2;
else if (expression 3)
      statement 3;
      …
else if (expression m)
      statement m;
else
      statement n;
```

Its semantics are: check the values of the expressions in order; if a value is True, execute the corresponding statement. Then it exits the if statement scope and continue to execute. If all expressions are False, then the statement n is executed. After that, continue with the subsequent program. The execution process of the if-else-if statement is shown in the following figure.



Further Exercise

Now that you have mastered timer light, try making a small timer microphone.

Project 5 Make a Button Switch

In the first project, we utilized a button to make LED on when the button is pressed and off when button released. However, in reality, the first button press switches on a light and the subsequent press turns it off -- this is how light switches work in our daily life. If you want to keep LED on without keeping the button pressed, then you should check out this chapter, which turns the button into a real light switch.

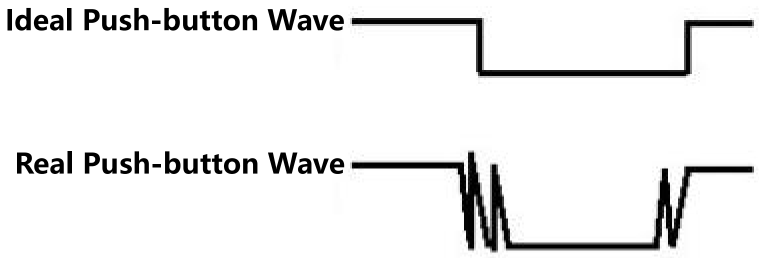
Task Navigation

1. How to eliminate button jittery?
2. Make a button switch

Key Points Analysis

What is button jittery?
We would imagine that the switch circuit works like "when button pressed down, the circuit closes" and "button pressed again, the circuit breaks". However, that is actually not the case.

Buttons are usually made of mechanical elastic switches. When mechanical contacts are broken or closed (usually lasts for around 10ms), a series of vibrations will occur due to elasticity, which causes button switches to not immediately close the circuit upon closing, and also not immediately breaks the circuit upon opening.

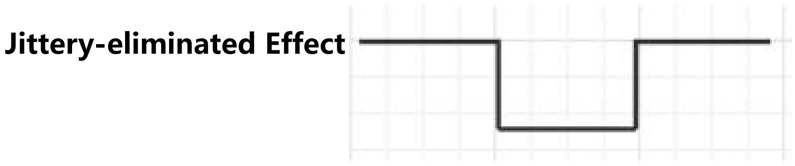


Button jittery

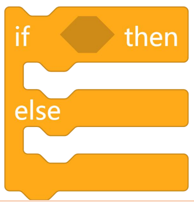
So how to eliminate button jittery?

There are two common solutions: by software and by hardware. Here we' ll focus on the simpler one, software solution.

We already know that button jittery cased by elastic inertia lasts for around 10ms. So we can use a delay command to postpone the execution, as well as a conditional, in order to eliminate button jittery.



Command List

Module	Type	Description
	True or false conditional execution	Set conditions at ⬢; if the condition is met, then the value True, and execute the first statement; if the conditions are not satisfied, then it is False, execute the second statement.

Hands-On

Hardware connection

Refer to "Magic Button Switches" for circuit connection.

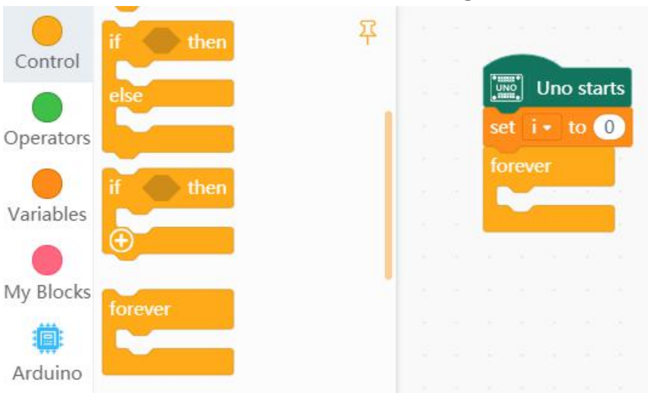
Programming

1. **Write the program**

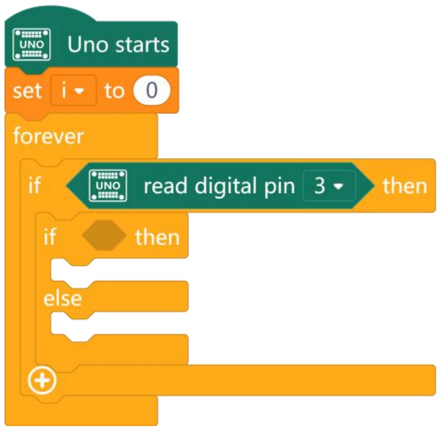
What to do when we want two different results when "press" the button all the same?

In order to continuously switch (no pun intended) between "turning on/ turning off", we need to apply what we learnt in last chapter, variable, which will be used as an intermediate.

1) First set a variable (*if you have questions about how to create a new variable, go back to Project II in Chapter I*) and initialize the variable to 0. Again, this is a looped event.

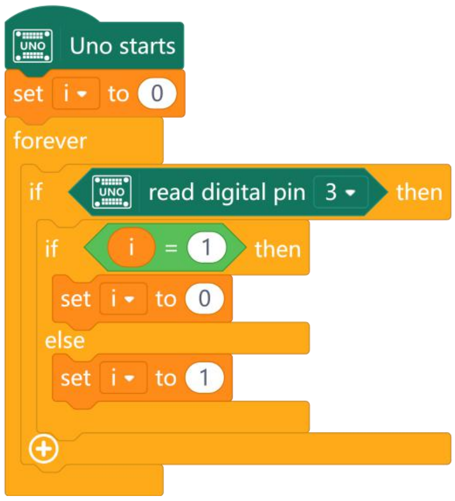


2) Next, set up the True / False conditional when button is pressed.



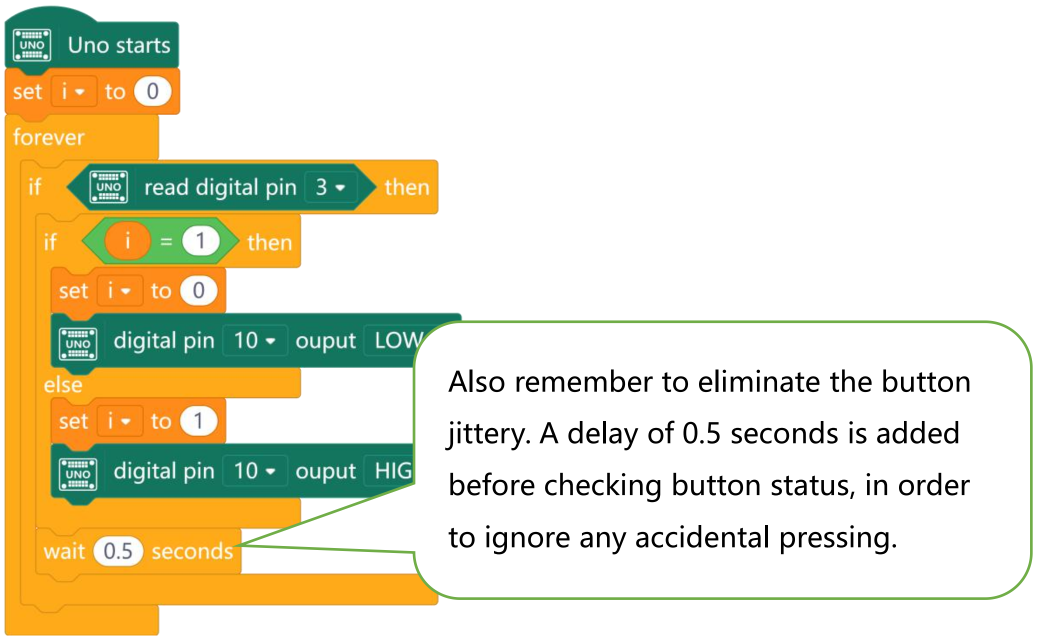
3) Now comes the hardest part. We need to find a way to utilize the variable to realize "turning on on first press and turning off on the subsequent press" .

In the conditional, initial value of i is 0, and True / False correspond to 1 and 0. So we set two situations when i is 1 and 0:



4) Then control the output. Output low level for 0, high level for 1, plus the delay command.

2. Reference program



```
Uno starts
set i to 0
forever
    if read digital pin 3 then
        if i = 1 then
            set i to 0
            digital pin 10 ouput LOW
        else
            set i to 1
            digital pin 10 ouput HIG
        wait 0.5 seconds
```

Also remember to eliminate the button jittery. A delay of 0.5 seconds is added before checking button status, in order to ignore any accidental pressing.

3. Program effect

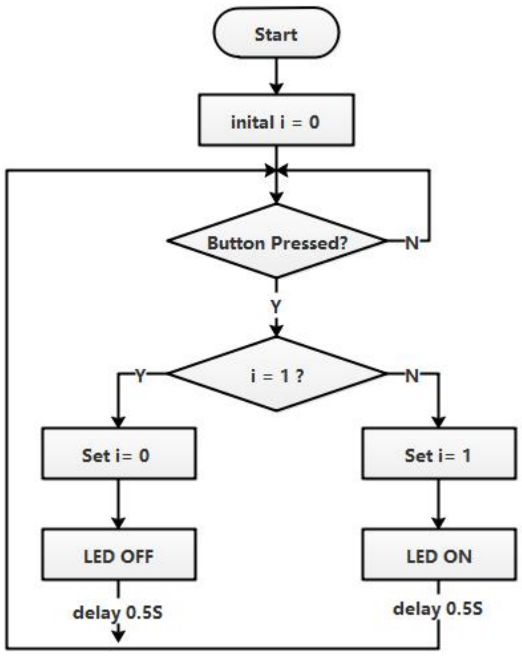
When the button is pressed, LED turns on; a subsequent press turns it off.

4. Program analysis

You may be confused that why use the variable i?

Let's think about what happens without the variable i. Then we go back to the case where we control the LED simply using the button. When the button is pressed, LED turns on. However the LED only turns on when button pressed down. It doesn’ t turn off on a subsequent press.

Therefore, we must introduce the variable i as an intermediate variable in order to realize the function by checking the True / False values.



Further reading

About Two-Way switch

Two-way switch means two sets of switch can control a light without affecting each other. You can turn on or off the light with either of them. If fact, every light can be wired on a circuit of multiple switches. Two-way, three-way or even multiple-way switches are all possible. Two-way switch means two switches controlling one light, Three-way switch means three switches controlling one light, and so on. We only need to check the conditions of multiple switches.

Common two-way switches include staircase switch, bedroom bed switches, and etc.

Further Exercise

Now that you have mastered one-way switch, try making a two-way switch.

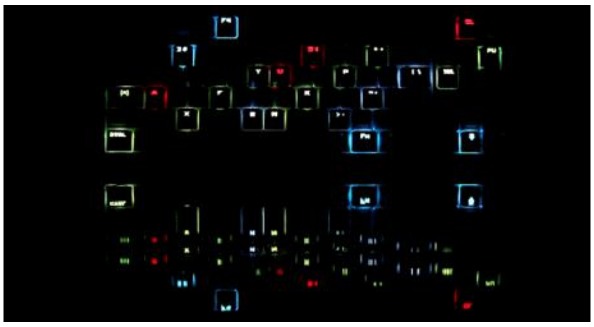
Chapter 3 Controllable Lights

RGD LED is actually just one category of LED lights. What makes it unique is that while common LED emits a single shade of white lights, RGB LED is capable to displaying all colors via its three bulbs of red, green, and blue colors.

Now let’ s try making all kinds of controllable LED lights!

Project 6 Breathing Lights

As its names suggests, breathing lights consist of of lights that go from dim to bright and back to dim by the rhythm of breathing. Breathing lights have been applied in many areas, including missed calls / SMS notifications on mobile phones, mouse breathing lights, keyboard backlights, external batteries charging lights, and cosmetic lightings on sound systems, etc.



Task Navigation

1. What is a function?
2. Make a breathing light

Key Points Analysis

What is a *function*?

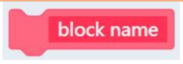
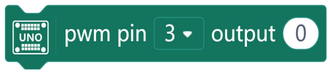
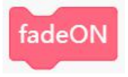
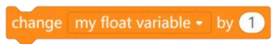
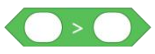
In essence, a *function*' s role is to perform a certain function. Every *function* is made for a particular feature, while the name of the *function* suggests what it is for.

We have functions in mathematics too. y=f(x) is a general form of functions. In mathematics, x will go through the calculations defined between () in f(x), and return as y. It' s the same in Mind+: when y=f(x) is executed, x is defined as the variable, and y is returned after being processed by function f.

Variables can also be defined within a function, which are valid only when the function is executed. When the function execution is over, these variables are no longer valid.

A "function call" means an execution of a function. A function can call other functions as well as itself.

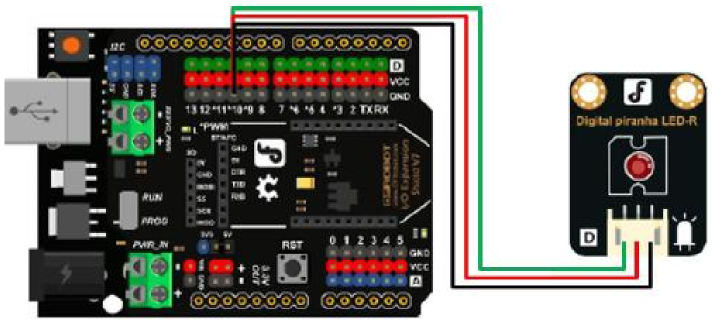
Command List

Module	Type	Description
	Custom function	Define and name the target function
	Set PWM pin output value	PWM means Pulse Width Modulation. Brightness can be controlled via PWM signals.
	child function call	call custom function
	control variable	change the value of a numeric variable
	Operator	The "greater than" operator to check if left side is greater than right side

Hands-On

Hardware connection

Connect the red LED to digital pin 10 on the UNO board.



Hardware connection (LED-R-10)

Please match the colors when plugging

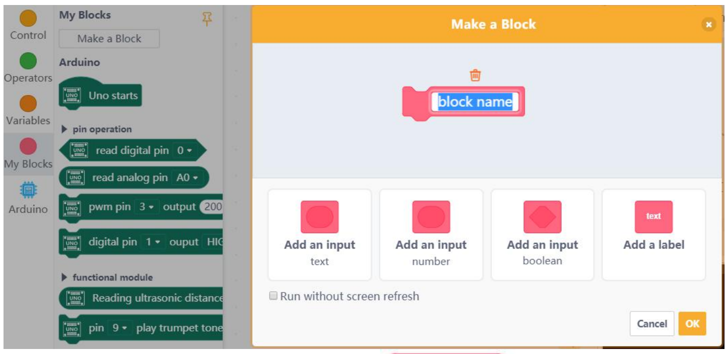
Programming

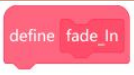
To make breathing light fade-in and fade-out, we' ll break it down to three steps. We' ll design the overall structure first, then achieve fade-in and fade-out effects.

1. **Create variables**

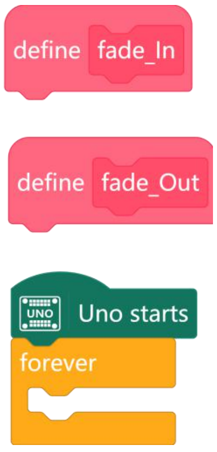
Find the function module first.

Click " " to create function "fade-in" and "fade-out".



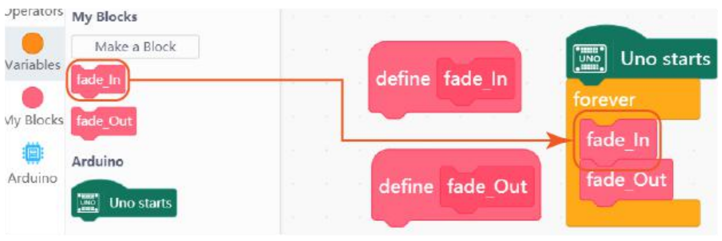
When function is defined, a function module "" will be generated automatically.

To delete a created module, select the function and drag it away from edit pane:



2. **Write the program**

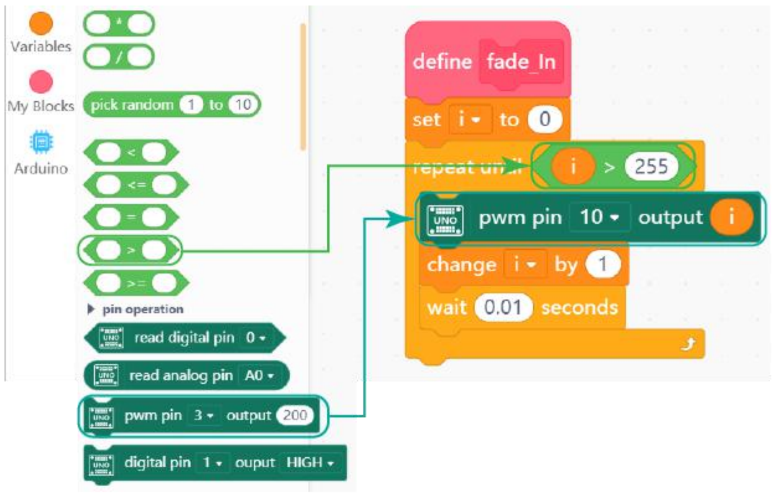
1) Design overall structure



```
define fade_In

define fade_Out

Uno starts
forever
    fade_In
    fade_Out
```

2) Achieve child function feature "fade-in"

To make lights go from off to gradually brighter, we can set a variable' s initial value to 0, and make brightness value increase by 1 every 0.01 second until maximum brightness.

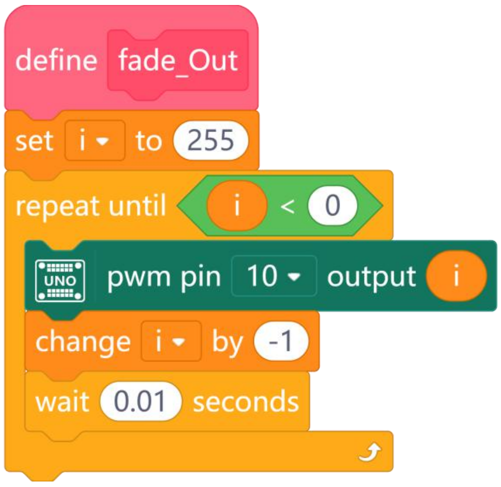


```
define fade_In
set i to 0
repeat until i > 255
    pwm pin 10 output i
    change i by 1
    wait 0.01 seconds
```

3) Achieve child function feature "fade-out"

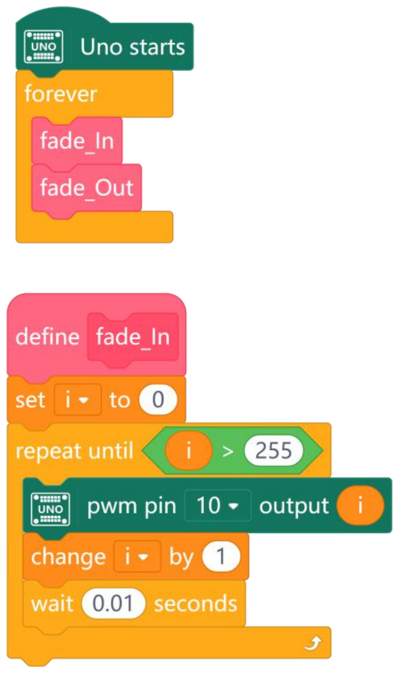
The same for lights to go from maximum brightness to gradually dimmer: we can set a variable' s initial value to the maximum value 255

(See Further Reading for why it' s 255), and make brightness value decrease by 1 every 0.01 second, until it reaches 0.



```
define fade_Out
set i to 255
repeat until i < 0
    pwm pin 10 output i
    change i by -1
    wait 0.01 seconds
```

Reference program



```
Uno starts
forever
    fade_In
    fade_Out

define fade_In
set i to 0
repeat until i > 255
    pwm pin 10 output i
    change i by 1
    wait 0.01 seconds

define fade_Out
set i to 255
repeat until i < 0
    pwm pin 10 output i
    change i by -1
    wait 0.01 seconds
```

The main body of the program is the main program on the left calling functions on the right, which are two functions defined.

3. Program’ s effect

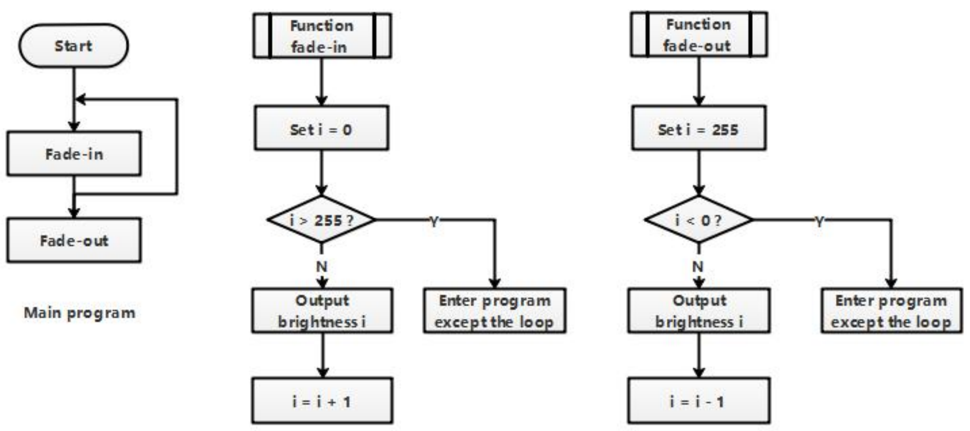
LED lights go gradually brighter and then gradually dimmer, demonstrating breathing effects.

4. Program analysis

In the main program, we set the variable i to 0 as an initial value. Although the default system value is already 0, but we still did it just to develop a good habit when we code.

After i is increased to 255 (maximum value) and the function is executed for a second time, i enters the next statements block, where it is decreased by 1 every time, i.e. increased by -1.

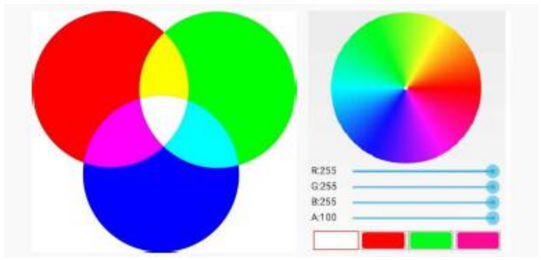
This program achieves “breathing lights” effects by controlling LED’ s fade-in and fade-out using Arduino’ s analog output. Breathing lights “take breathes” , as LED’ s brightness increases gradually to maximum after an interval, before gradually dims to completely off and repeats the cycle. Changes in brightness are achieved by analog output module.



Further reading

RGB color model uses RGB model to assign an intensity value, which is within 0-255, for the RGB component of every pixel in the image. Various colors are constituted through different intensities in three primary colors, red, green, and blue.

Each of the red, green, and blue color channels has 256 levels of brightness. At 0, the "light" has lowest brightness — light is off; while at 255, the "light" is brightest. When three colors have the same value of greyscale, they produce grey hues of different shades. That is to say, when their greyscales are all 0, they make the darkest black hue; when their greyscales are 255, they make the brightest white hue.



All colors on a computer screen are constituted by combinations of red, green, and blue lights of different proportions. A set of red, green, and blue lights is the minimal display unit. Any color on the screen can be expressed in a set of RGB values.

Further Exercise

We achieved breathing light effects by function call. Now can we achieve "fade-in and fade-out" directly, without function at all? How?

Project 7 3-step Adjustable Light

In the making of breathing light in the previous section, we achieved the effect of fade-in and fade-off through function calls. In this section, we will use buttons to control the LED module to achieve multi-step brightness feature. Multi-step control is often used in car's air-conditioning keys, desk lamp, stage lighting, sound system, etc.

Task Navigation

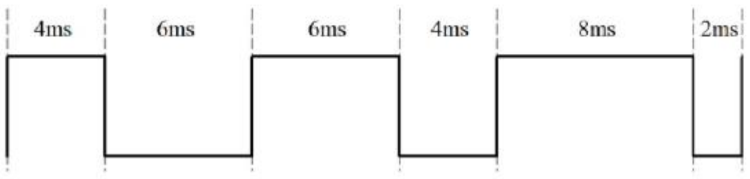
1. What is a PWM output?
2. Make a 3-step adjustable light

Key Points Analysis

Do you feel familiar with PWM? In the making of the breathing light in the previous section, we have used the PWM output " pwm pin 3 ▾ output 0 ".

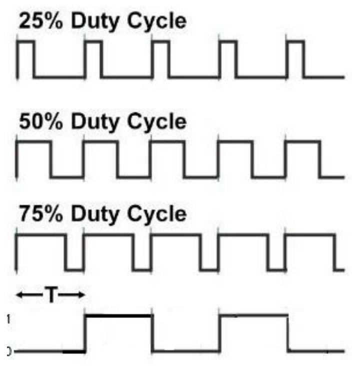
PWM is the abbreviation of "Pulse Width Modulation". It **can convert the digital output waveform into an equivalent analog waveform**.

PWM mechanism: by adjusting the **duty** cycle of the pulse square wave in each cycle, without changing the pulse square wave period, in order to achieve an equivalent analog voltage output.



PWM Wave

Duty Cycle: refers to the time ratio of high level in a cycle.



Digital signal waveform and PWM modulation

Right now, you only need to have a general understanding of PWM. You don't need to dig into it yet, and will learn more in the future.

Command List

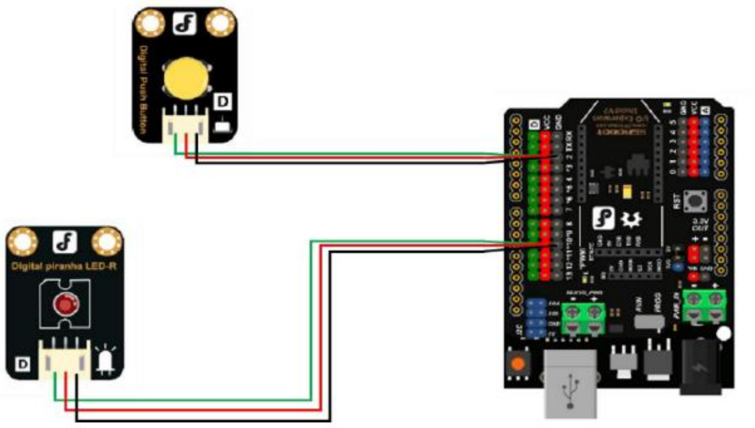
Module	Type	Description
pwm pin 3 ▾ output 0	Set PWM pin output value	PWM is Pulse Width Modulation, and the brightness is controlled by the PWM signal
( ) mod ( )	Remainder operation	The remainder refers to the integer that is not divided by the dividend in integer

		division; value of the remainder is an integer between 0 and the divisor (excluding the divisor).
	Operator	Multiplication
	Operator	Division

Hands-On

Hardware connection

Connect the red LED to digital pin 10 on the UNO board. The button is connected to Digital Pin 2.



Hardware connection (LED-R-10, button-2)

Please match the colors when plugging

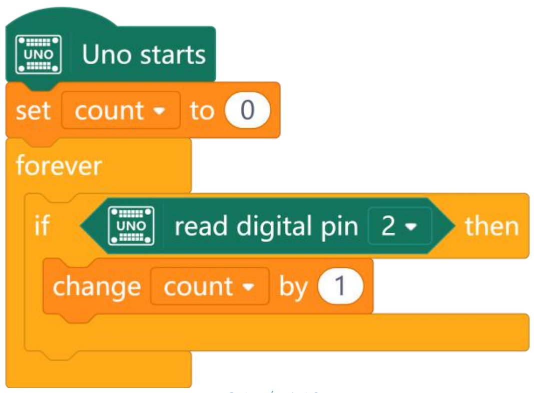
Programming

1. Write the program

First imagine the use scenario: the button is not pressed, the LED is off- > the first time the button is pressed-the LED is slightly brighter- > the second time the button is pressed - the LED is further brighter- > the third time the button is pressed-the LED is on highest brightness; press the button for the fourth time-the LED goes out ...

First, create a variable Count with an initial value of 0 (representing 0 step, off)

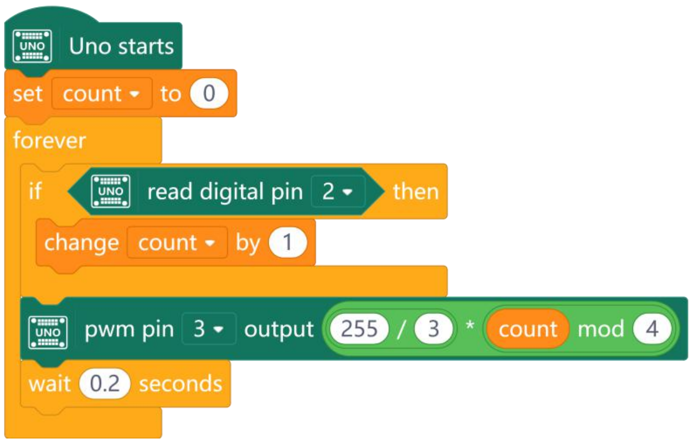
Have a look at the process, and we know we can use “if ... else”, the first form of a conditional.



```
Uno starts
set count to 0
forever
    if read digital pin 2 then
        change count by 1
```

Given condition is that the LED brightness range is 0 ~ 255 when we make a 3-step adjustable light by PWM. How to do it?

2. Reference program

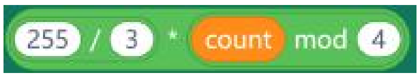


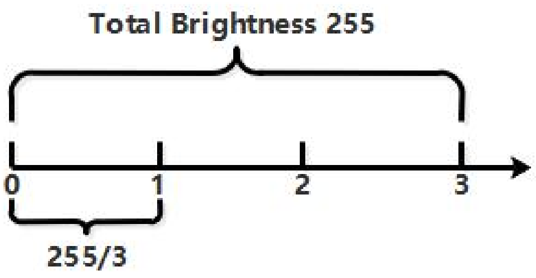
```
Uno starts
set count to 0
forever
    if read digital pin 2 then
        change count by 1
    pwm pin 3 output 255 / 3 * count mod 4
    wait 0.2 seconds
```

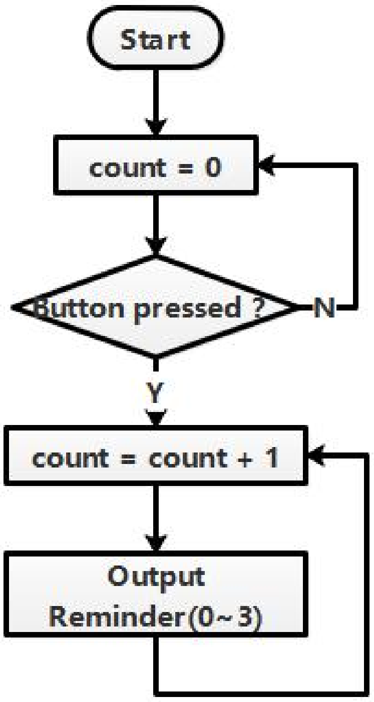
3. Program effect

Press the button for the first time to turn on the LED, but the brightness is low; press it again to increase the brightness; press it for the third time, the LED will be the brightest. Another press will turn off the LED. Further presses will cycle those effects.

4. Program analysis

The difficulty here is why  `255 / 3 * count mod 4` is divided by 4 not divided by 3? Although it is a 3-step adjustable light, this light actually has 4 steps, which is (0,1,2,3).





Imagine what would be if we neglect 0 step and set it to 3?

Further reading

Essentially, computers don’ t exist without mathematics. Von Neumann, crowned as “father of computers”, is himself one of the most prominent mathematicians in 20th century. At the very beginning, computers were made only to assist humans in complicated and time-consuming calculations. You might think that modern day computers all present User Interface, so they have nothing to do with mathematics. That is not the case. Computers are still executing instructions in the CPU. Every instruction is stored and parsed by binary numbers, like 0 and 1. This is the most basic knowledge in mathematics. Therefore, computers and mathematics are inseparable.
All this is the relationship between computers and mathematics, but what about computer science? The relationship between computer science and mathematics might be a little odd. Around twenty to thirty years ago, computer science was still only a branch of mathematics. Now, computer science has a wide range of research fields and a huge number of researchers, which in turn promotes the development of mathematics in many ways. In a sense, it can be said that the child has grown taller than his mother. But mother’ s blood can still be found in child’ s vein. This blood is the mathematical underpinning of computer science.
Another inter-discipline between modern computational science and mathematics is computational science or numerical analysis. However, majors of computer science are studying programming. What does programming have to do with mathematics? Let's talk about the potential impact that mathematics may have on future computer science majors.
1. Database data.
Computers are inseparable from data. Mathematics goes with data hand in hand. The background programs of an Internet service, database storage, high concurrency, and big data, all are related to many theorems and formulas in mathematics.
2. The data structure is largely the study of algorithm
The core technology of a company is often algorithms. Companies might be happy to opensource many other technologies they have, but core algorithms are always trade secrets. In job interviews, there are often tests on algorithms, many of which are data structures. Topics are often optimal path, binary tree, etc.
3. Geometry, Linear Algebra
3D video games and Photoshops are all based on space geometry. Even making a Sniper model move around in Dota game is related to mathematics.
It might seem that mathematics doesn’ t play an important role in your path to a successful programmer. But it is vital when you crave for technological innovations. Mathematics isn’ t necessary while learning programming, but knowledge of mathematics will only help you in writing better code.

Further Exercise

Try making a 5-step adjustable light!

Project 8 Knob Adjustable Light

In this section, we introduce a new sensor to simulate a knob-adjustable light to achieve a stepless brightness adjustments.

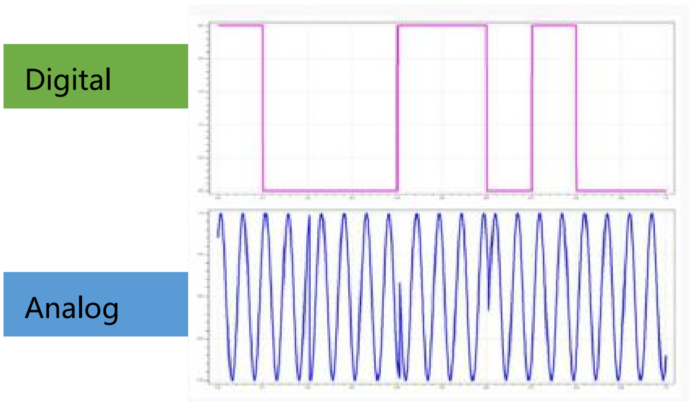
Task Navigation

1. What is an analog input sensor?
2. What is an analog signal?
3. What is mapping?
4. Make a knob adjustable light.

Key Points Analysis

1. Analog and digital signals

Both digital output and digital input are digital signals. They are either 0 or 1, corresponding to low and high levels, respectively.
In the world of Arduino, there is also analog signals besides digital signals.
Arduino’ s main control panel controls elements in two methods, digital output and analog

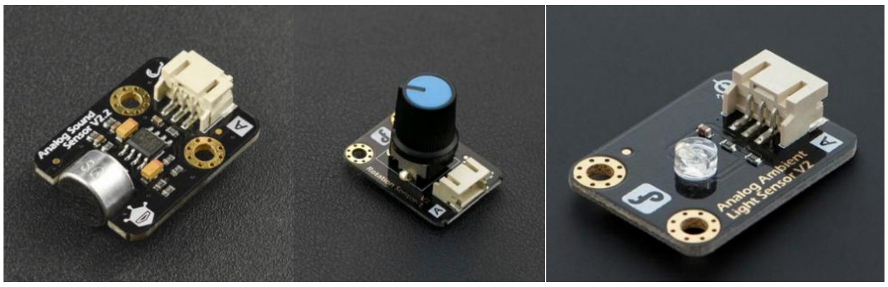


output. As for digital output we learnt before, it sends digital signals to the output circuit -- 0 or 1 to control the circuit switch.

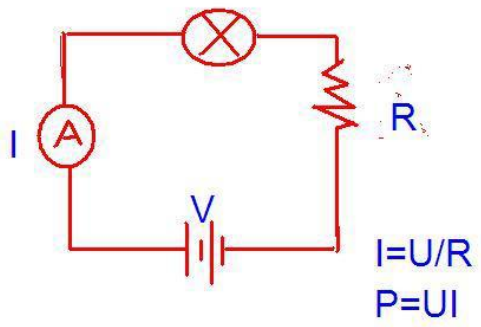
Why is an analog signal called so? What exactly does it simulate?
In the process of analog signal transmission, sensors are used to convert various continuous signals in natural world into almost identical electrical signals. For example, the sound of speech is originally vibrations of the vocal cords. Microphone collects them, and converts acoustic signals to electrical signals, whose waveforms are identical. It is only being represented and transmitted in another physical quantity (using electrical signal to simulate the original vibration signals of the sound).
In addition, the pressure sensor also converts the magnitude of the pressure into electrical signals through conversion. Therefore, the **analog signal using electrical signals to simulate physical quantities in the natural world.**

2. What is an analog input sensor?

The following are all analog input sensors: sound sensor, analog angle sensor and light sensor. Any input signals that can be detected are also analog signals.

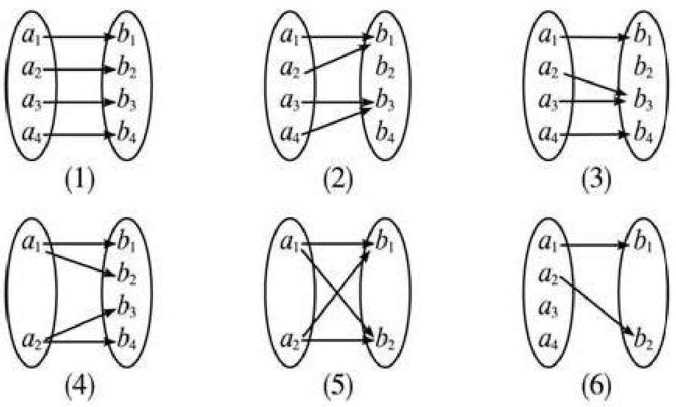


Compared with digital sensors, there is a larger variety in analog sensors, which can obtain more external information. We can regard all **analog input sensors** as **adjustable resistors.** The rotation module that will be used here is one of them.
As Ohm's Law defines: V = I x R; voltage = current x resistance. The introduction of analog sensors in the circuit is equivalent to the introduction of adjustable resistors; thus, changes in resistance will also cause voltage changes.



3. What is mapping?

In mathematics, mapping is a functional relationship. It is a "corresponding" relationship between elements of two different sets.



We can understand it as changing a variable's original range to a range we need. In Mind+ 's mapping module map 0 from[0 , 1023] to[0 , 255] , the "[ ]" in [0 , 1023] represents the original value range; and the "[ ]" in [0 , 255] represents the target range.

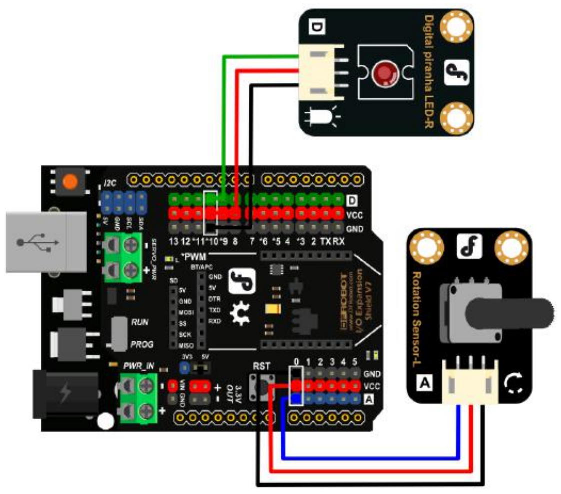
Command List

Module	Type	Description
read analog pin A0	Analog input value	Read the input value of an analog input sensor
map 0 from[0 , 1023] to[0 , 255]	Mapping	Decrease or increase the value in the first "[]" proportionally to the value in the second "[]". Here, the value in [0,1023] is scaled down and converted to the value in [0,255].

Hands-On

Hardware connection

Connect the red LED to Digital Pin 10 and the analog angle potentiometer to analog Pin A0.



Hardware connection (LED-R-10, analog angle potentiometer-A0)

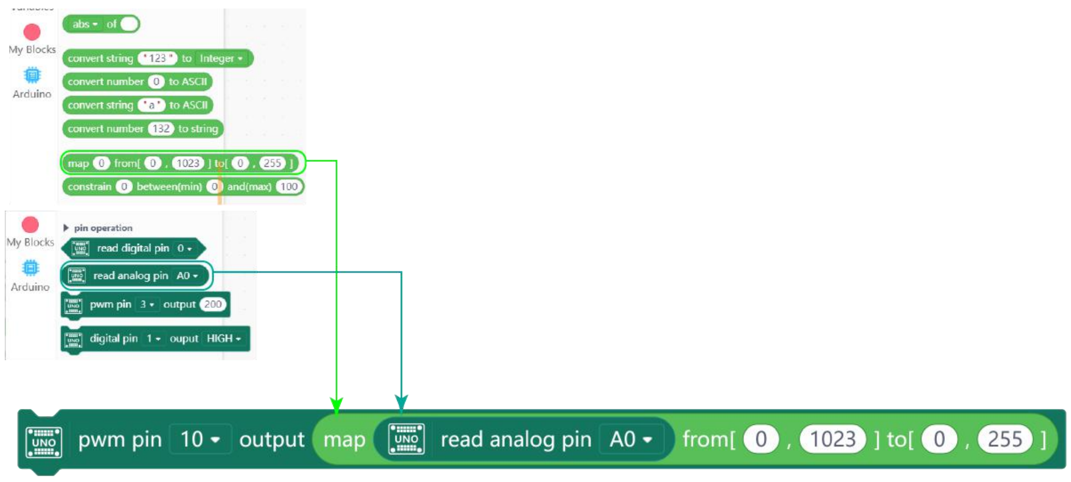
Please match the colors when plugging

Programming

1. Write the program

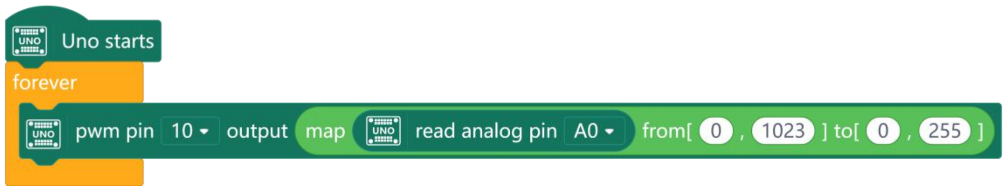
As an analog sensor, the analog angle module is actually an adjustable resistor. Knob rotation can trigger the rotation of the internal potentiometer, thereby changing the resistance value. It corresponds to the strength of the analog input signal of the variable.

Arduino can control the brightness of the LED by reading the strength of analog input signals and converting it by mapping.


My Blocks
Arduino
abs of
convert string "123" to Integer
convert number 0 to ASCII
convert string "a" to ASCII
convert number 132 to string
map 0 from[0 , 1023] to[0 , 255]
constrain 0 between(min) 0 and(max) 100
pin operation
read digital pin 0
read analog pin A0
pwm pin 3 output 200
digital pin 1 ouput HIGH

```
pwm pin 10 output map read analog pin A0 from[ 0 , 1023 ] to[ 0 , 255 ]
```

2. Reference program



```
Uno starts
forever
    pwm pin 10 output map read analog pin A0 from[ 0 , 1023 ] to[ 0 , 255 ]
```

3. Program’ s effect

Rotating the knob changes the brightness of the LED. The greater the rotation angle, the higher the brightness.

4. Program analysis

The program here is relatively simple. We have introduced the concept of mapping, which is a change to the range. But why the original analog signal range is [0, 1023]? Also, what does range [0, 255] mean?

The knob controls the brightness. [0,255] is the brightness range, and [0,1023] is the range of the analog input signal.

Further reading

Analog signal range

Arduino has 8-bit analog value range (0 - 255) and 10-bit analog value range (0 - 1023).

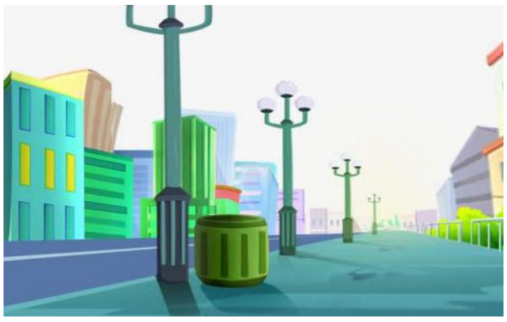
What is "bit"? Most of mathematics calculations are done in decimal, where one means ten units of the next smaller digit. However in computers, we use binary, where one means two units of the next smaller digit.
The analog output signal is 8 bits, that is, $2^8 = 256$; and the initial value is 0, so the range is 255;
The analog input signal is 10 bits, that is, $2^{10} = 1024$; and the initial value is 0, so the range is 1023.

Further Exercise

Now that you have learned how to control the brightness with the knob, what else can the knob control? Use your imagination.

Chapter 4 Smart Light

In the previous chapter, we learned how to control the brightness of LED lights. In this chapter, we will learn how to let the environment decide when to switch lights on and off. The corridor lights "listen" for footsteps and turns on; the street lights "see" the dusk fall and go on, and off when it is dawn. These are smart lights that turn on or off according to conditions without human actions.



Project 9 Sound-Activated Light

To make a sound-activated light, we need "ears" first. We are using a new element here -- analog sound sensor.
Analog sound sensors can recognize sound and convert the decibels of sound into analog signals. Like analog angle sensor introduced previously, the input signal range of the analog sound sensor is also 0 ~ 1023.

Task Navigation

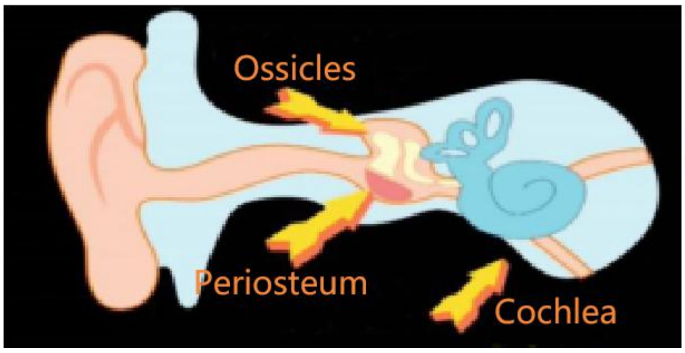
1. Learn analog sound sensors
2. Learn the serial monitor
3. Make a sound-activated light

Key Points Analysis

Learn analog sound sensors

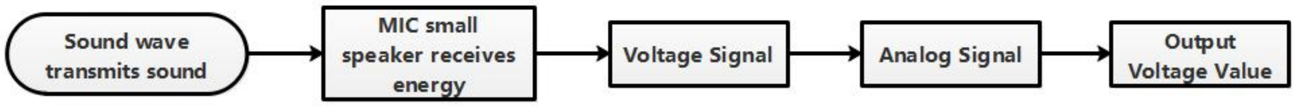
We learn on our physics class that sound is produced by vibrations and propagates in sound waves. A scream in snowy mountains can cause avalanches. This is the proof of the energy of sound.

How does human hear sounds? First, the sound reaches eardrum through the ear canal, can causes eardrum to vibrate. The vibration is then transmitted to ossicles, which then vibrate and send the signals to the brain through cochlea. Now, we hear sounds.



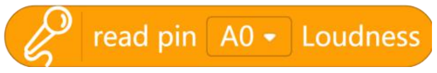
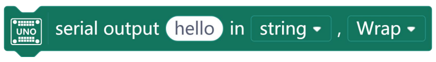
The analog sound sensor is the "electronic ear" of the device and can determine the strength of the sound. How does it do it?

Look at the small metal cylindrical element on the analog sound sensor in the upper right picture. This is a MIC speaker that can output different voltage signals according to the strength of the sound. When external sound is transmitted to the MIC speaker through sound waves, the MIC speaker can receive the "energy" of the sound wave with high sensitivity and convert the sound wave energy into different voltage signals. Working together with other circuit elements, MIC speaker converts the voltage signals into analog signals, and outputs voltage value to express the detected sound strength. This is how analog sound sensor "hears".



Tip: One of the features of analog sound sensors is that they can use air flow instead of sound, which can effectively avoid noise interference.

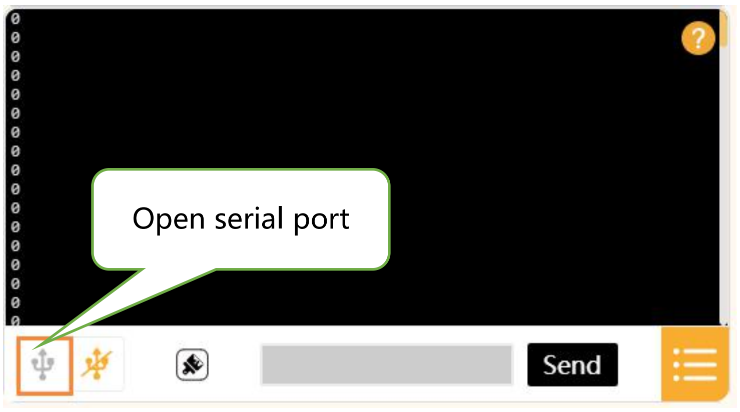
Command List

Module	Type	Description
 read pin A0 Loudness	Sound input	Read analog sound sensor’ s input signal
 serial output hello in string , Wrap	Serial output	Display the current data through the serial port via serial port monitor

The serial output here is on the bottom right of Mind+.

Arduino’ s analog input pin reads the analog input, sends it to Arduino main control panel. Then “serial output” can display the data that is currently going through the serial. The displayed position is the above serial monitor area. Click the icon in the lower left to open the serial port, and you can see the data being uploaded.

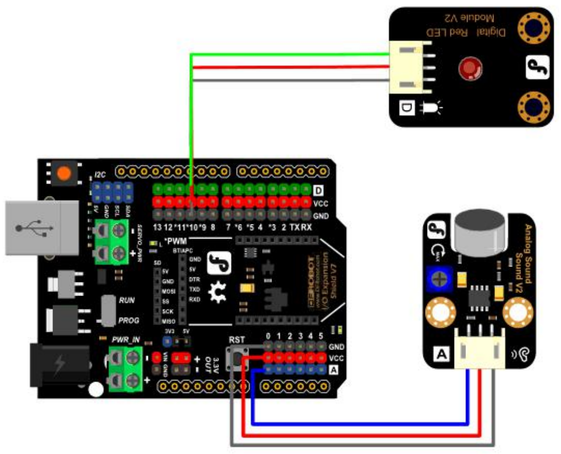
USB cable must always be connected to the Arduino and the computer USB port when serial reading.



Hands-On

Hardware connection

Connect the red LED to Digital Pin 10, and the analog sound sensor to Analog Pin A0.



Hardware connection (LED-R-10, analog sound sensor-A0)

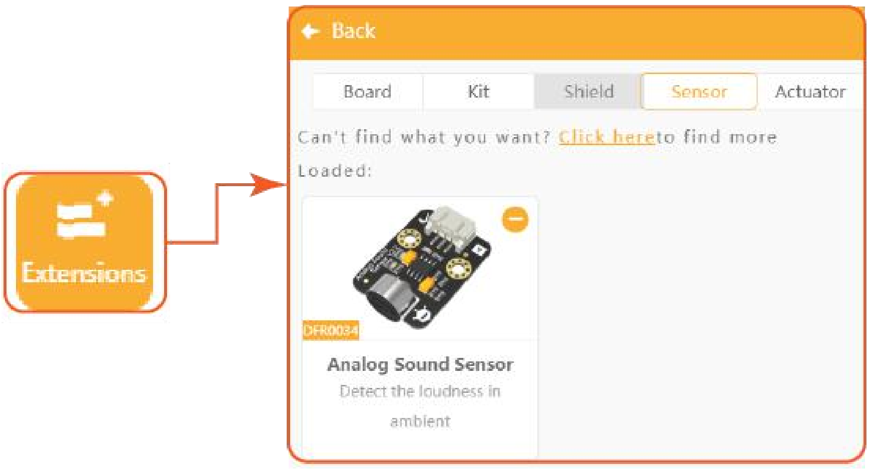
Please match the colors when plugging

Programming

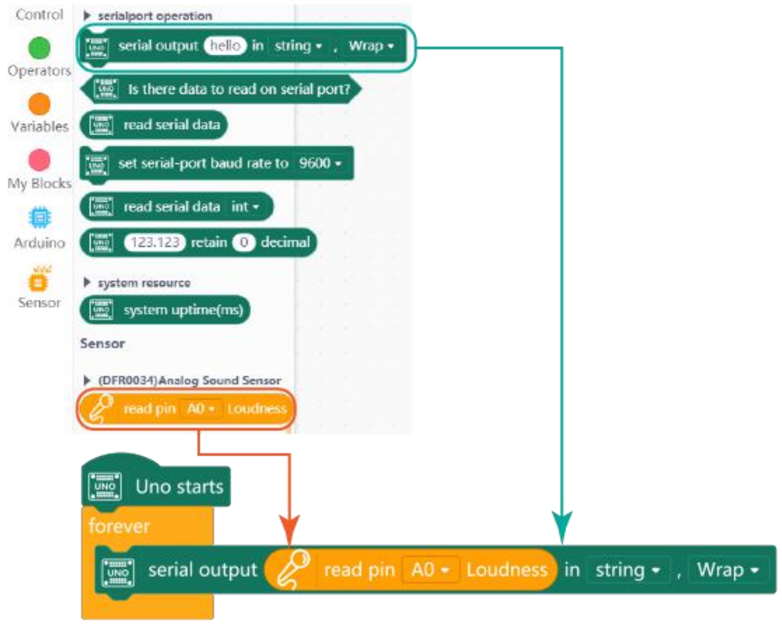
1. Write the program

1) Click on the extension to add " Analog Sound Sensor " to the sensor :

2) After the addition, you can see it in the module Sensor . If you want to delete the added extension, click "Extension" to return to "2" and click " " to delete it.

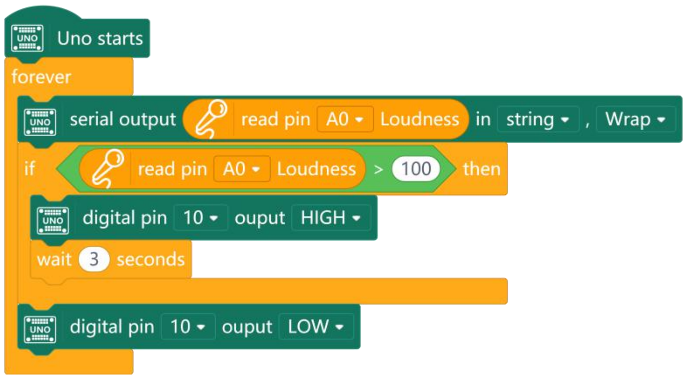

Extensions
Back
Board
Kit
Shield
Sensor
Actuator
Can't find what you want? Click hereto find more
Loaded:
DFR0034
Analog Sound Sensor
Detect the loudness in ambient

3) Use serial port to read sound strength



Remember to click the icon on the lower left to open the serial port to see the data being uploaded!

2. Reference program



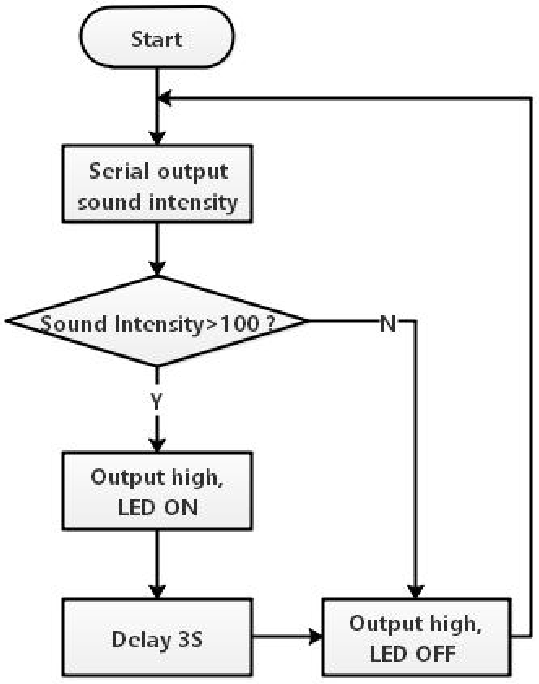
```
Uno starts
forever
    serial output  read pin A0 Loudness  in string , Wrap
    if  read pin A0 Loudness > 100  then
        digital pin 10 ouput HIGH
        wait 3 seconds
    digital pin 10 ouput LOW
```

3. Program effect

When the Mic is rubbed or there is sound input, the LED light goes on and the serial port reading is high; when it is quiet, the LED light is off and the serial port display reading is low.

4. Program analysis

After we understand reading the signal of the analog sound sensor using the serial port, we need to determine whether there is sound and turns on or off accordingly, so we need to introduce a conditional structure: if ... then ... else.



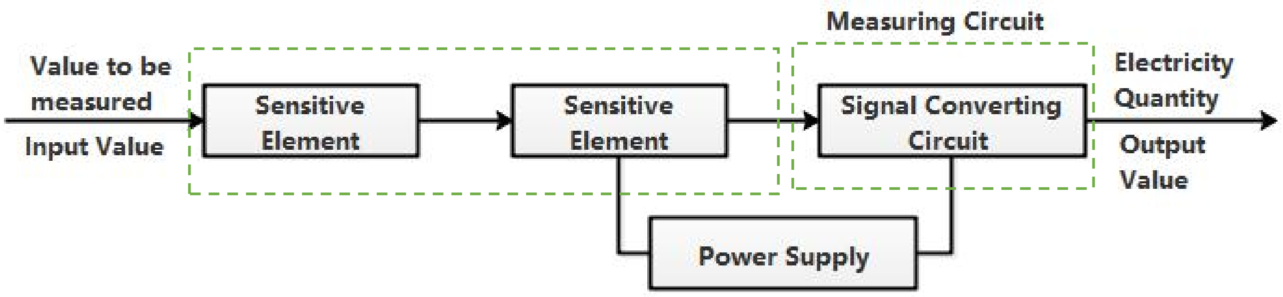
Note that when somebody passes, the sound-activated light needs to be kept on for a while, so don’ t forget to wait for a few seconds.

Further reading

What is a sensor?

The sensor is simply a measuring device. The input quantity can be physical value, chemical value, biological value, etc.; the output value is mainly electrophysical value, it can also be aerophysical or photophysical values.

The sensor consists of a sensitive element and a converting element. **The sensitive element** can directly sense the object being measured, and output other values of elements in a certain relationship with it. **The converting element is** also called a converter. Generally, it does not directly sense the value, but converts the output of the sensitive element into an electrical value output.



The basic composition and working mechanism of the sensor

Further Exercise

Now that we have learned how to use sound-activated lights, can you now use an analog sound sensor to make an alarm light?

Project 10 Stairs Light

There are relatively few places in the life that use only sound-activated lights. If lights are activated solely by sound, they will turn on during the day. This is not desirable. After all, everyone needs to use electricity responsively.

The two-way control light in the stairs combines an analog light sensor and analog sound sensor, so it only turns when it' s dark and there is sound. In the previous section we learnt "electronic ears" , the analog sound sensor. Now let' s make a new friend -- "electronic eyes" , the analog light sensor, also known as photosensitive sensor.

Task Navigation

1. Meet the photosensitive sensor
2. Make a stairs light

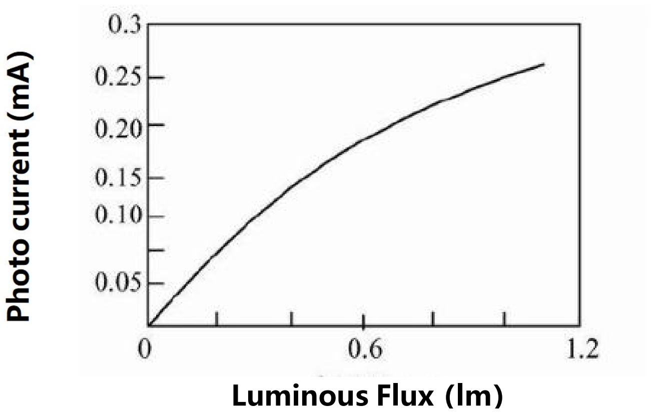
Knowledge point analysis

Meet the photosensitive sensor

The analog sound sensor can convert the surrounding sound into analog signal ands input it to the Arduino motherboard. The photosensitive sensor works the same way. It can convert ambient light into analog signals and input it to the Arduino motherboard.

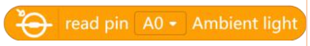
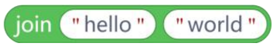
First look at the core elements of this module , the photoresistor located in the center area of the module. By the look of it, it' s much the same with LED we used previously.

Ordinary resistors have fixed resistance values, while photoresistors have lower resistance under strong light conditions, and higher resistance under low light conditions and complete darkness. As the light increases, the resistance of the photoresistor becomes lower; when light decreases, the resistance of the photoresistor becomes higher.



Relationship between photocurrent and luminous flux of the photoresistor

Command List

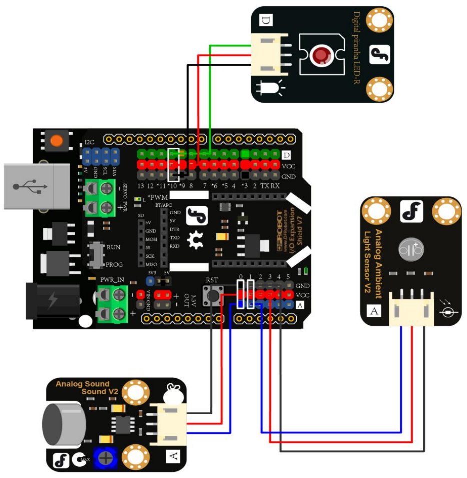
Module	Type	Description
 read pin A0 Ambient light	Analog signal input	Read the ambient light intensity and convert it analog signals
 join "hello" "world"	Operator	Combine strings or numbers and output strings: For example, combining "1" and "2" will output the string "12"; Combine "Hello" and "World" to output the string

		"Hello World"
and	Operator	And operation, the symbol is &, which is true only when both the left and right conditions are met.

Hands-On

Hardware connection

Connect the red LED light to Digital Pin 10, the analog sound sensor to Analog Pin A0, and the analog light sensor to Analog Pin A1.



Hardware connection (LED-R-10, analog sound sensor-A0, analog light sensor-A1)

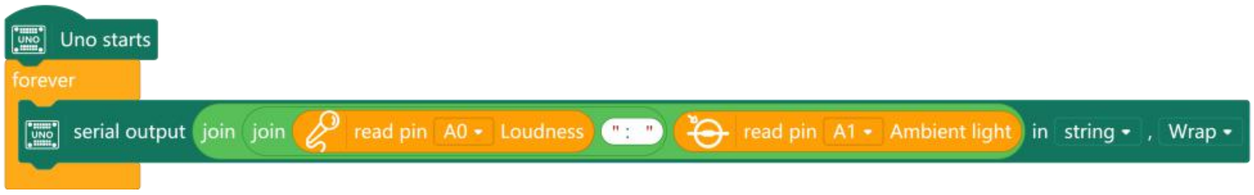
Please match the colors when plugging

Programming

1. Write the program

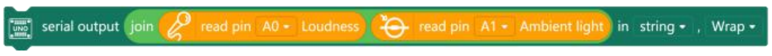
1) Click on the extension to add "Analog Sound Sensor" and "Analog Ambient Light Sensor" to the sensors

2) Serial port reads sound intensity and ambient light intensity

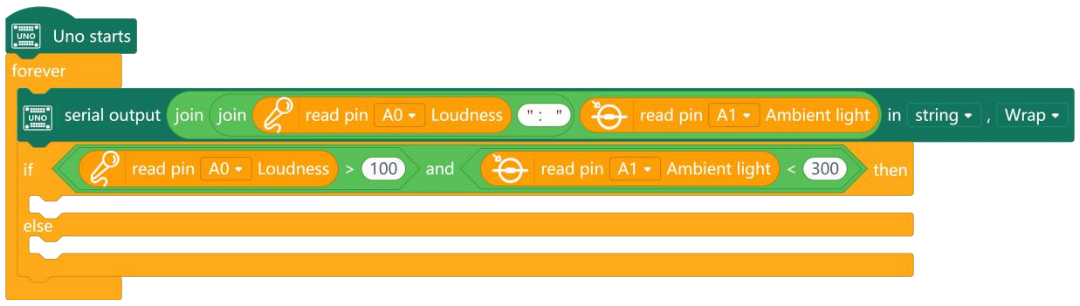


```
Uno starts
forever
    serial output join join read pin A0 Loudness ":" read pin A1 Ambient light in string , Wrap
```

You may be wondering why you should add a ":" in the middle. This is a break for a better view on the serial print results.

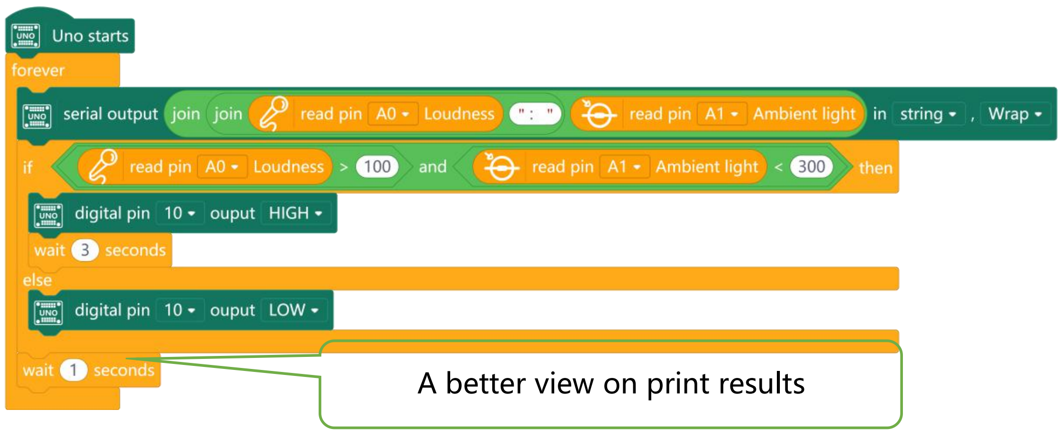
You can also try entering  serial output join read pin A0 Loudness read pin A1 Ambient light in string , Wrap directly without “:”
and see what you will get.

3) Set up conditions



```
Uno starts
forever
  serial output join join read pin A0 Loudness ":" read pin A1 Ambient light in string , Wrap
  if read pin A0 Loudness > 100 and read pin A1 Ambient light < 300 then
  else
```

2. Reference program



```
Uno starts
forever
  serial output join join read pin A0 Loudness ":" read pin A1 Ambient light in string , Wrap
  if read pin A0 Loudness > 100 and read pin A1 Ambient light < 300 then
    digital pin 10 ouput HIGH
    wait 3 seconds
  else
    digital pin 10 ouput LOW
  wait 1 seconds
```

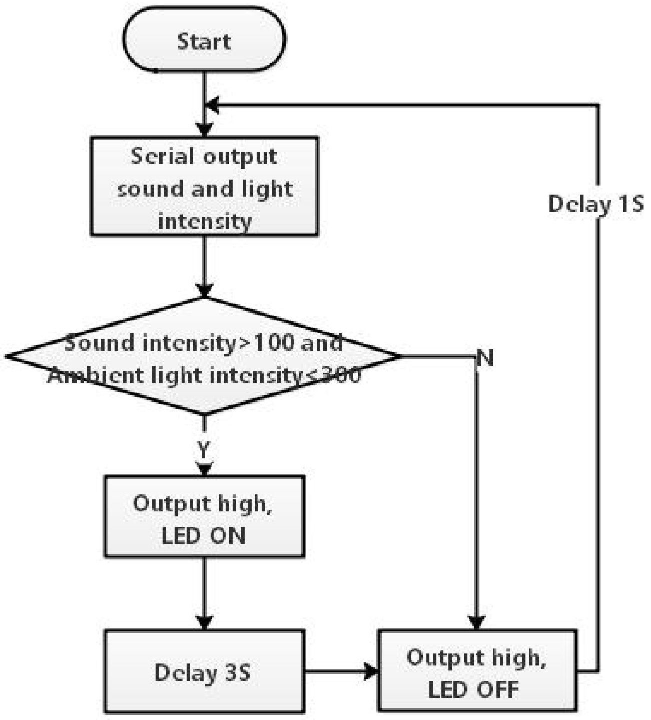
3. Program results

When the light is strong, the LED will not turn on whether it is noisy or quiet; when the light is low, rub the MIC or speak loudly, the LED turns on.

4. Program analysis

When writing our programs, we should always follow the procedure of input device-> control device-> output device. There are two input devices here: analog sound sensor and analog light sensor. In the previous section, we already know that the analog signals are passed into Arduino through the Digital Pin D and A of Arduino UNO, and are printed with serial monitor.

We already know that the stairs light is a two-way switch light, controlled by sound and light. The light only turns on when both conditions are met: sound is loud enough and light is low enough; otherwise, it does not turn on. The AND operation and provides a solution to the combination of multiple conditions. And it is only true when all conditions are met.


Start
Serial output sound and light intensity
Delay 1S
Sound intensity>100 and Ambient light intensity<300
N
Y
Output high, LED ON
Delay 3S
Output high, LED OFF

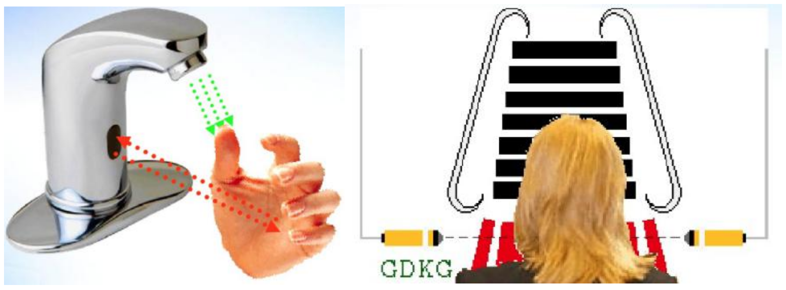
Further Reading

About photosensitive sensor

Photosensitive sensor is a kind of non-contact sensor that features fast response and high reliability.

Photosensitive sensors are applied in four forms in industrial applications: proximity-sensing, absorption-sensing, beam arrangement, and retro-reflective.

Photosensitive devices have played an important role in automatic flushing equipment in the bathrooms and automatic-active escalators.



Further Exercise

Besides stairs light, is there anywhere else that photosensitive sensor can be applied?

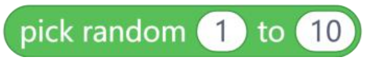
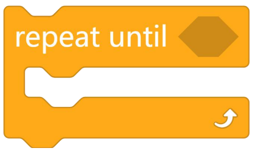
Project 11 Electronic Candle

The most anticipated moment for each birthday is when you make a wish with a candle on the birthday cake. Turn off the light, light up the candle. Make a wish with hands kept together. At last, take a deep breath and blow out the candles. It is a heartwarming moment when your family’ s good wishes and smiles surround you, amid the shimmering candle light.
However, when we eat the cake, drops of candle wax always spoil the taste and ruin the moment. Is there a way to do away with wax?
Let's try to make some electronic candles together.

Task Navigation

1. Understanding random numbers
2. Make an electronic candle

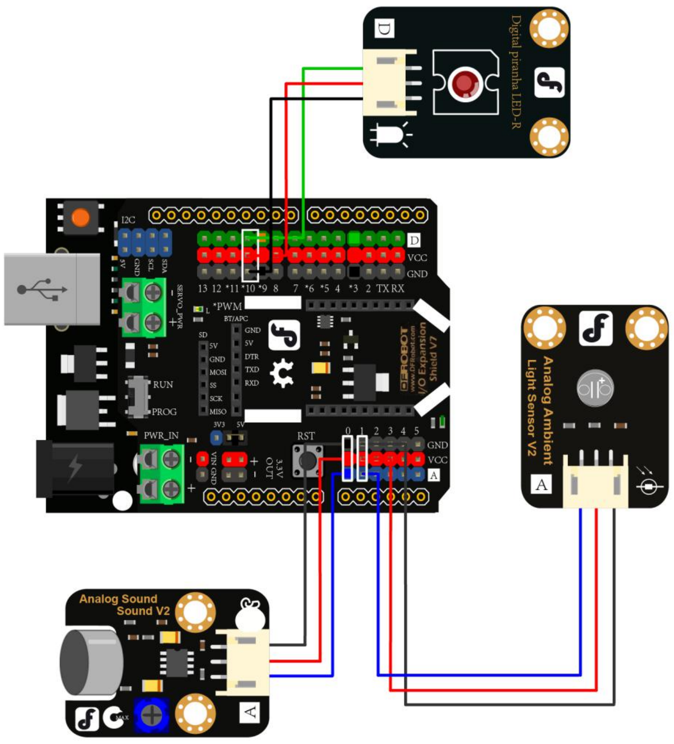
Command List

Module	Type	Description
 pick random 1 to 10	Operator: get a random number (random)	Get any integer in a certain range
 repeat until	Conditional statement: repeatedly execute until	Keeps executing the subroutine until the conditions in ⬢.

Hands-On

Hardware connection

Connect the red LED light to Digital Pin 10, the analog sound sensor to Analog Pin A0, and the analog light sensor to Analog Pin A1.



Hardware connection (LED-R-10, analog sound sensor-A0, analog light sensor-A1)

Please match the colors when plugging

If you paid attention to details, you might have realized that the hardware connections are identical to the ones we used in stairs light section. The reason is that we are still controlling the LED light module by sound and light. The only difference is the effect we are after. The previous one is turning on / off, while the current one is the shimmering light effect of candles.

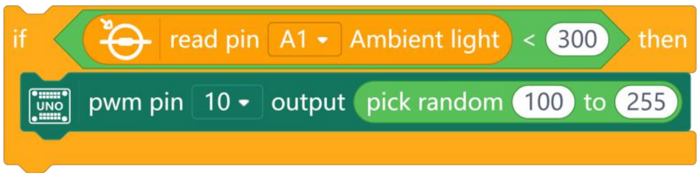
Programming

1. Write the program

1) Click on the extension to add "Analog Sound Sensor" and "Analog Ambient Light Sensor" to the sensors

2) Serial port reads sound intensity and ambient light intensity

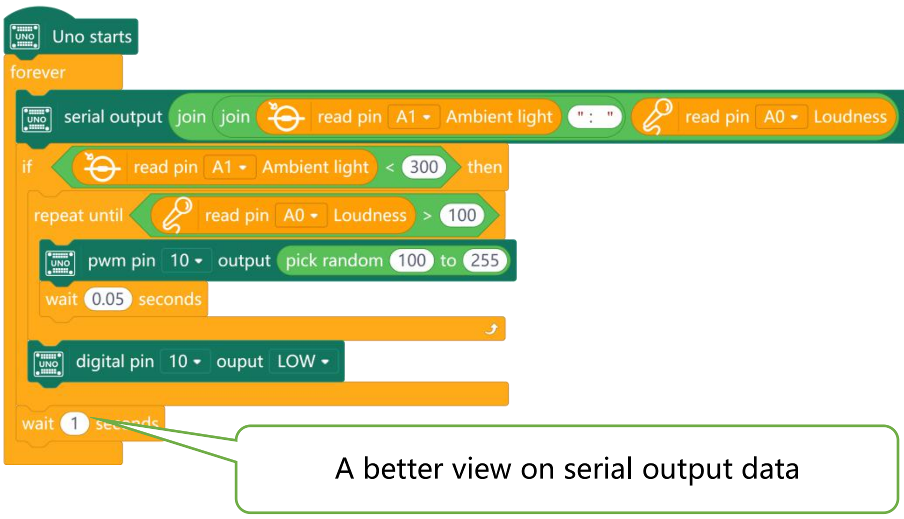
3) Setting conditions: the candle keeps flickering in the dark environment to achieve the candle effect.



```
if read pin A1 Ambient light < 300 then
    pwm pin 10 output pick random 100 to 255
```

4) Then, sound intensity brings the candle “blown out” effect.

2. Reference program



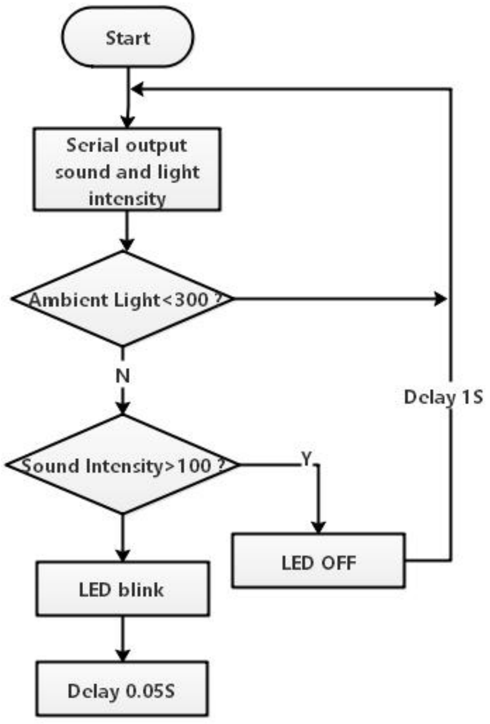
```
Uno starts
forever
    serial output  join  join  read pin A1 Ambient light  " : "  read pin A0 Loudness  in string , Wrap
    if  read pin A1 Ambient light < 300  then
        repeat until  read pin A0 Loudness > 100
            pwm pin 10 output  pick random 100 to 255
            wait 0.05 seconds
        digital pin 10 ouput LOW
    wait 1 seconds
```

3. Program results

When indoor lighting is off, LED turns on and flickers; when air is blown into MIC, LED turns off.

4. Program analysis

For flicker effects, we introduced a random number module. PMW controls the brightness and outputs brightness determined by random number. When Arduino obtains sound signals (when we blow out the candle), candle goes out.



Further Reading

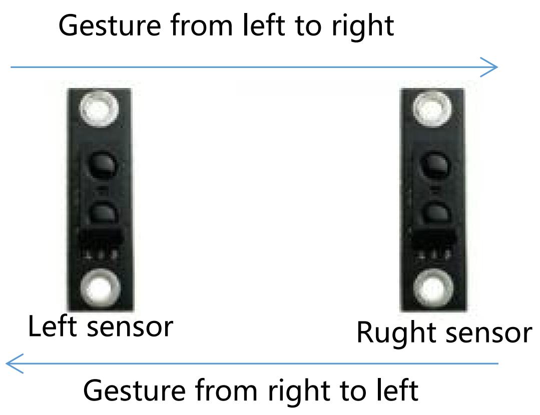
Gesture-controlled LED

With the development of science and technology, human-computer interaction has become an important part of people's daily life. The advances of visual computing, especially, enable

computers to "see" the user's actions. Among them, gesture recognition is a key technology that is indispensable for ushering a new generation of human-computer interaction.
Gesture can have various directions, such as left to right, right to left, top to bottom, bottom to top, and so on. For Arduino beginners, it' s certainly difficult to make gesture recognition work. For these reasons, we design a simple gesture recognition pattern here, which is using two sensors to determine the chronological order of gestures to make out the direction. There are many scenarios when it comes to gesture-controlled LED. We have designed three solutions here:

1) Solution 1: By the gesture direction detected by the sensors, a left to right gesture would turn on LEDs while right to left gesture would turn off.
2) Solution 2: By the gesture direction detected by the sensors, a left to right gesture turn on LEDs from left to right, while a right to left gesture would turn off LEDs from right to left.
3) Solution 3: By the gesture direction detected by the sensors, a left to right gesture turns on red LEDs while a right to left gesture turns on green LEDs. If a gesture is present for 3 seconds or longer, turn off LEDs.

Here is a brief introduction to solution 1: "By the gesture direction detected by the sensors, a left to right gesture would turn on LEDs while right to left gesture would turn off."



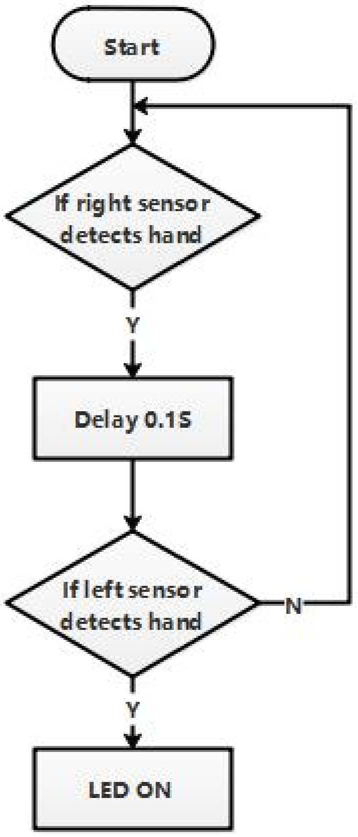
Required hardwares

LED x 1

DFRobot digital anti-fall sensor x 1

Arduino UNO x1 (or other main control board)

Design concept



Further Exercise

With concepts provided by “Further Reading”, can you make a gesture-controlled LED?

Chapter 5 Master Sound Devices

In the previous chapter, we learned how to control the light and brightness. In this chapter, we will learn the alarm device and ultrasonic module.

Project 12 Analog Sound Device

Alarm timer device? Wow, it looks so fancy. In fact, it’ s not at all. Alarm timers are already everywhere in our life. The beeps when washing machine finishes procedure, and when memory bars or GPU isn’ t properly plugged in, are made the buzzer.

Task Navigation

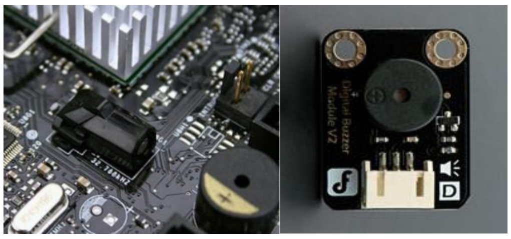
1. Learn the buzzer
2. Making an analog sound device

Key Points Analysis

Learn the buzzer

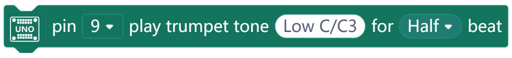
We should be very familiar with speakers. Common headphones are two small speakers, as well as radios, MP3, MP4 players. The sound elements of TV and sound systems are all speakers. Speaker is also called loudspeaker. During the electro-acoustic conversion, it converts analog electrical signals into sound signals. Speaker is a wide-frequency sound device.

The buzzer is an integrated electronic siren, which can emit monotonous sound under different driving waveforms, and is a narrow-frequency sound device. We can change the volume of the buzzer by changing the frequency.



Ostensibly, the biggest difference between speaker and buzzer is that speaker is capable of a variety of sounds while buzzer can only make a few. For their sound mechanisms, buzzer uses piezoelectric ceramics to convert electrical signals into mechanical vibration signals, while speakers rely on an electromagnet to do the same conversion.

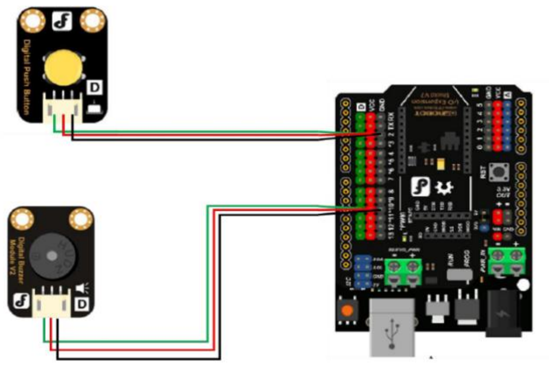
Command List

Module	Type	Description
 pin 9 play trumpet tone Low C/C3 for Half beat	Music control	Sound is produced by adjusting frequency by PWM. It also controls the duration of the sound, which is beats.

Hands-On

Hardware connection

Connect button to Digital Pin 3 and buzzer to Digital Pin 8



Hardware connection (button-3, buzzer-8)

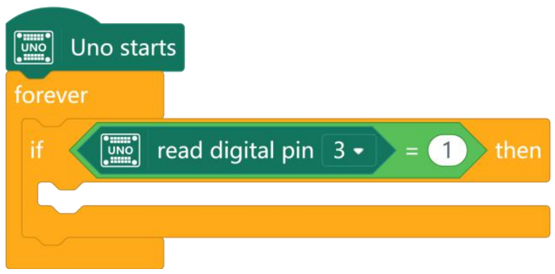
Please match the colors when plugging

Programming

Take washing machine as an example. We simulate a simplified alarm timer on washing machine. How do we use the washing machine? First, we turn on the washing machine, and select a duration (8 minutes spin or 15 express). We press start. When the procedure is done, buzzer goes off and washing machine turns off automatically.

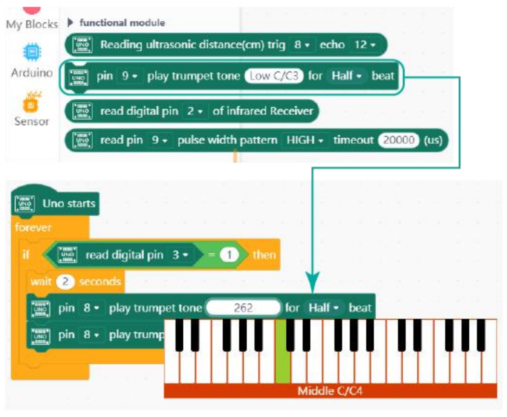
Write the program

We first set up the button actions to lay an overall structure:



```
Uno starts
forever
  if read digital pin 3 = 1 then
```

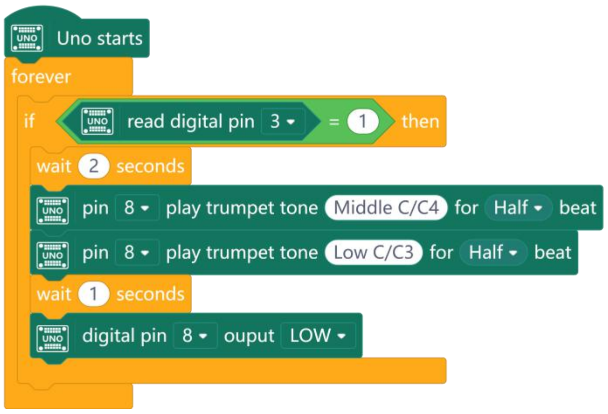
Then set the pitch:



```
My Blocks
Arduino
Sensor
functional module
Reading ultrasonic distance(cm) trig 8 echo 12
pin 9 play trumpet tone Low C/C3 for Half beat
read digital pin 2 of infrared Receiver
read pin 9 pulse width pattern HIGH timeout 20000 (us)

Uno starts
forever
  if read digital pin 3 = 1 then
    wait 2 seconds
    pin 8 play trumpet tone 262 for Half beat
    pin 8 play trump
```

Finally, the turning off beep sound.

1. Reference program

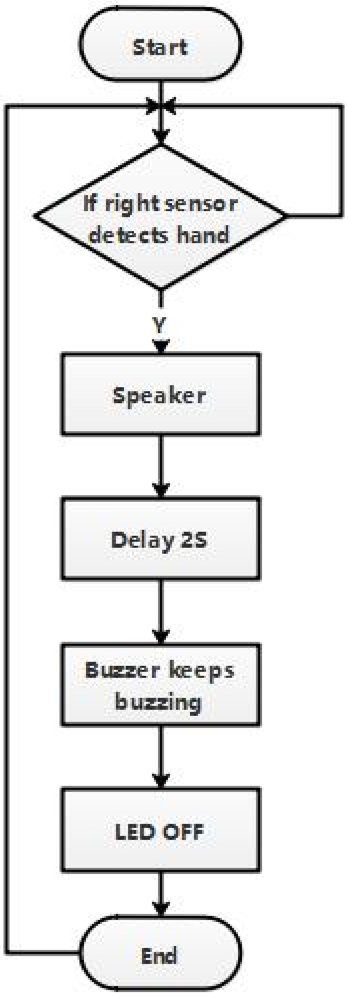


```
Uno starts
forever
  if read digital pin 3 = 1 then
    wait 2 seconds
    pin 8 play trumpet tone Middle C/C4 for Half beat
    pin 8 play trumpet tone Low C/C3 for Half beat
    wait 1 seconds
    digital pin 8 ouput LOW
```

2. **Program results**

When the button is pressed, the buzzer will wait a few seconds, go off, and then stop.

3. **Program analysis**

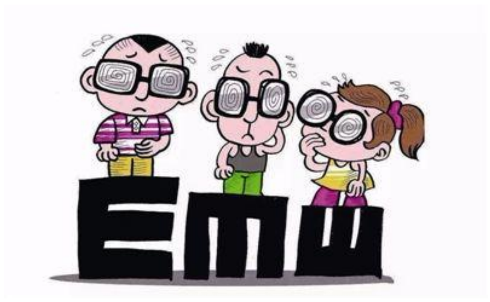


Further Exercise

Now you know how to use a buzzer, can you add LED flashing effects on top of the sound effects?

Project 13 Nearsightness Alert

Nearsightness is epidemic. Aside from genetic factors, it is mainly caused by bad habits. We know we shouldn’t read a book with a bow-back or humpback, but we just can’t help it when we sit in front of the desk.
For our health and eye protection, let’s make a simple nearsightedness alert device using buzzer and ultrasonic sensor.



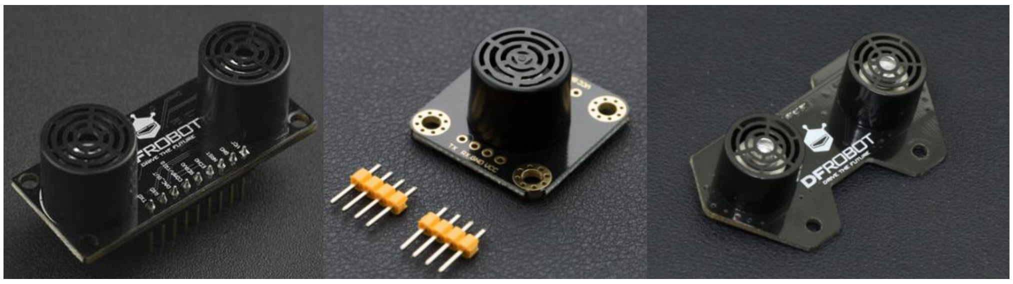
Task Navigation

1. Learn about ultrasound
2. Making a nearsightness alert device

Key Points Analysis

1. Understanding Ultrasonic Sensors

The current mainstream ranging sensors are ultrasonic ranging sensors, infrared ranging sensors, laser ranging sensors and radar sensors. Among them, the ultrasonic sensor is used for long range stationary plane ranging. The ranging range of ordinary ultrasonic sensors is about 2cm - 450cm. We can clearly see that there are 4 corners on the physical ultrasonic sensor: VCC -- 5V power pin, Trig -- trigger end, Echo -- receiving end, and GND -- ground.

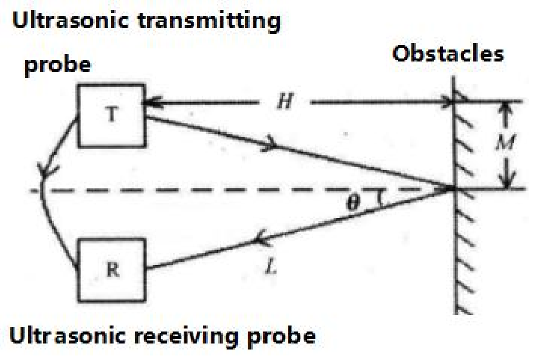


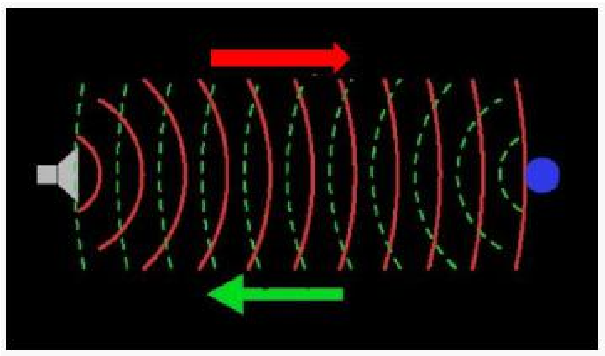
Among the dual-probe sensors in the picture, one is used to send ultrasound and the other is used to receive. The single-probe ultrasonic sensor in the middle can do both by its own. This sensor is the first four-pin sensor we deal with, and the wiring is relatively special. We need to pay special attention when we connect the hardware later.

2. Ultrasound

We already know in the sound-activate light section that sound is generated by the vibration of objects. The number of vibrations per second is the frequency of the sound. The unit is Hertz. Human can produce sounds ranging from 20Hz to 8kHz, while human ears can hear sounds between 20 to 2000 Hz. The sound below 20 Hz is called low frequency sound, and the sound above 20 kHz is called ultrasound. Ultrasonic waves can propagate through any gases, liquids, and solids, at various speeds. The propagation speed in the air is C = 340m / s.

3. Ultrasonic ranging mechanism



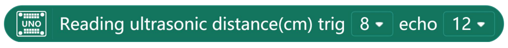


The mechanism of ultrasonic ranging is using the propagation speed of ultrasonic wave in the air as a given condition, and measure the time difference between sending ultrasonic wave and receiving the reflection from the obstacle, which is used calculate the actual distance between the sending point and obstacle.

The process starts when ultrasonic wave transmitter sends ultrasonic waves in a certain direction and starts timing; ultrasonic waves travel in the air and returns upon meeting any obstacles; ultrasonic receiver receives the returning ultrasonic waves and stops timing. Distance L = C * T / 2 (L: distance, C: propagation speed in the air, T: total time). The value of C is related to temperature.

Temperature	-30	-20	-10	0	10	20	30	100
sound velocity	313	319	325	323	339	344	349	386

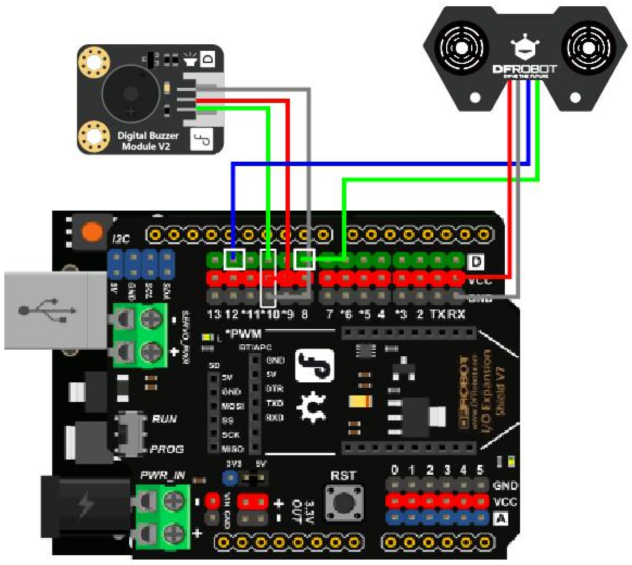
Command List

Module	Type	Description
 Reading ultrasonic distance(cm) trig 8 echo 12	Ultrasonic ranging	Read the distance measured by the ultrasonic sensor (in cm)

Hands-On

Hardware connection

Connect the trigger end of the ultrasonic ranging sensor Trig to digital pin 8 pin, Echo to digital pin 12 pin, GND to GND, and + 5V to VCC.



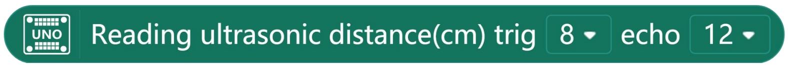
Connect UNO and ultrasonic sensors

(Note the wire matching and relative positions)

Programming

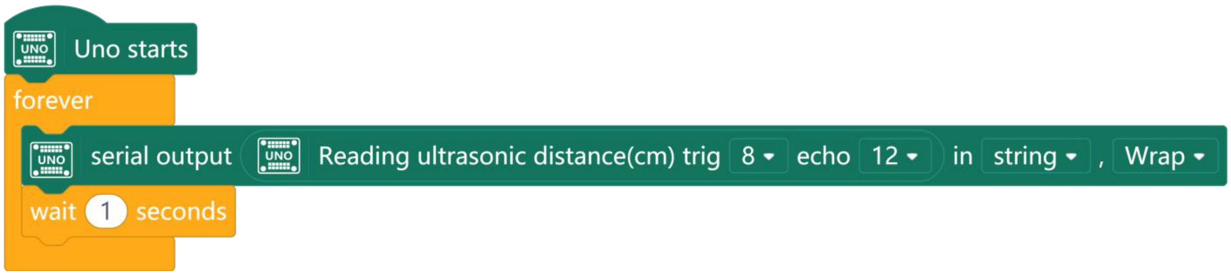
1. Write the program

1) First, find the ultrasonic command in the Arduino function module.



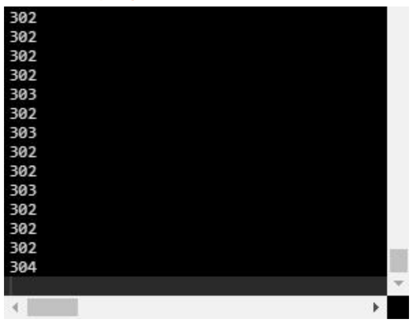
```
Reading ultrasonic distance(cm) trig 8 echo 12
```

2) Maintain a healthy sitting posture, then use the serial port to print the data read by the ultrasonic sensor, in order to determine the correct distance when sitting posture is healthy.

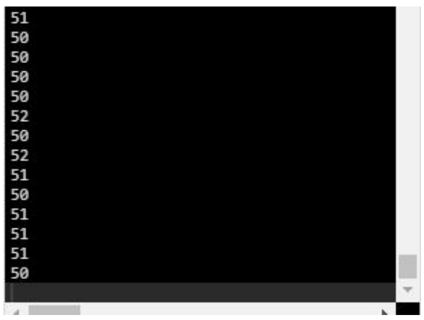


```
Uno starts
forever
    serial output Reading ultrasonic distance(cm) trig 8 echo 12 in string , Wrap
    wait 1 seconds
```

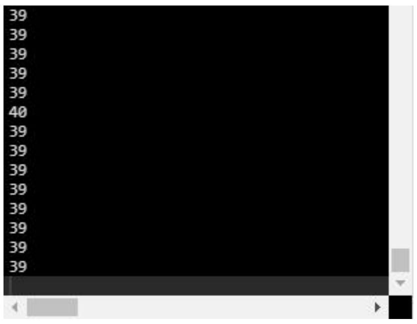
Please pay attention to the ultrasonic sensor' s test direction (probe direction) is unobstructed.



```
302
302
302
302
303
302
303
302
302
303
302
302
302
304
```



```
51
50
50
50
50
52
50
52
51
50
51
51
51
50
```

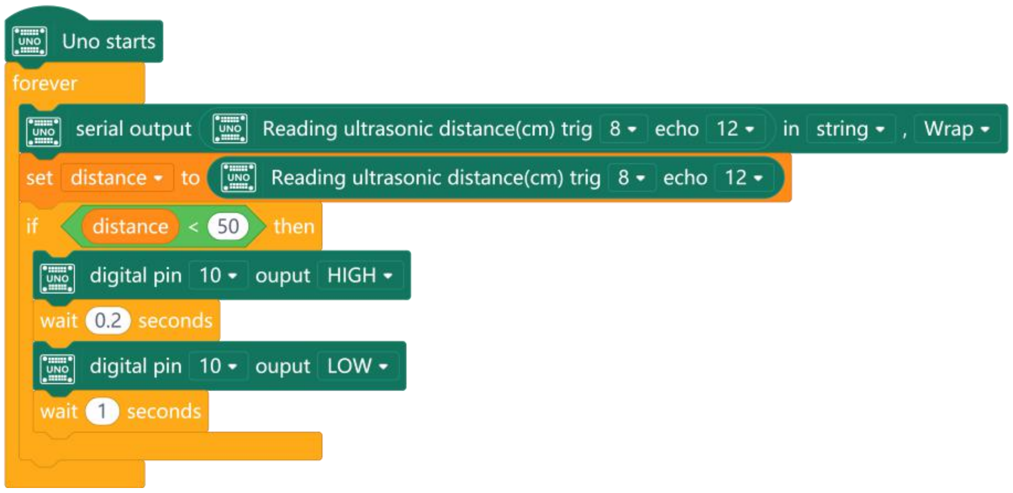


```
39
39
39
39
39
40
39
39
39
39
39
39
39
39
```

Sit tightly　　Head downwards slightly　　Head downward

With initial experimental data, we can set up alert conditions.

2. Reference program



```
Uno starts
forever
    serial output  Reading ultrasonic distance(cm) trig 8 echo 12  in string , Wrap
    set distance to  Reading ultrasonic distance(cm) trig 8 echo 12
    if  distance < 50  then
        digital pin 10 ouput HIGH
        wait 0.2 seconds
        digital pin 10 ouput LOW
        wait 1 seconds
```

3. Program results

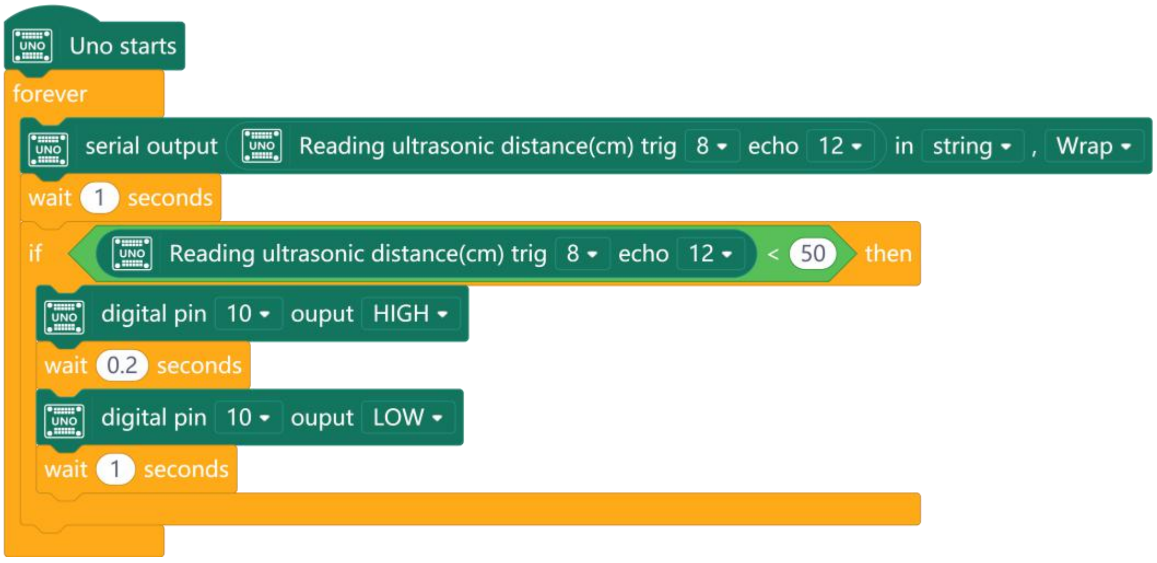
When the head is buried, the buzzer goes off for a second and stops. The buzzer stops upon posture change.

4. Program analysis

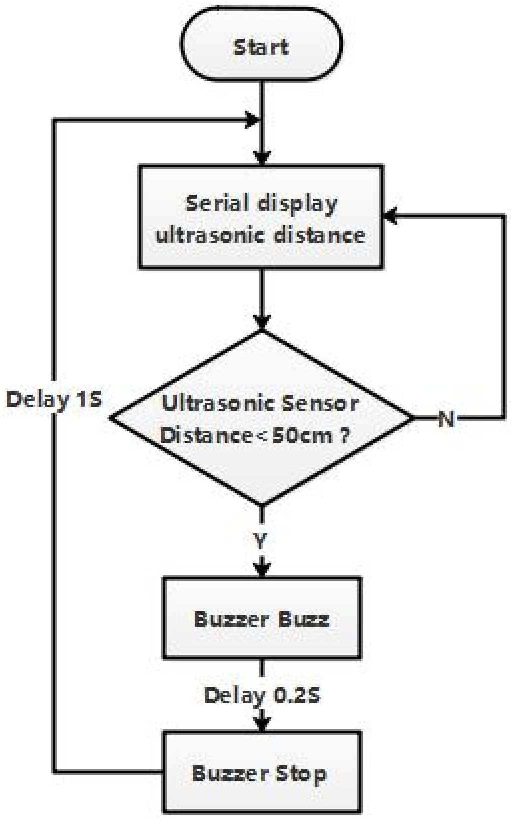
According to the test results of the serial port, with the position of ultrasonic sensor fixed, the distance measured when sitting healthily is more than 300cm, and the distance of an unhealthy is a little more than 50cm. So we set 50cm as the threshold value.

The buzzer sound is a little loud and we need no more than an alert, thus a 1s buzz duration.

Pay attention that we utilized variables properly to store the detected distance value. This is to battle the problem of repeated detection causing insensitivity in the conditional. Try comparing the effects of the following error programs.



```
Uno starts
forever
    serial output  Reading ultrasonic distance(cm) trig 8 echo 12  in string , Wrap
    wait 1 seconds
    if  Reading ultrasonic distance(cm) trig 8 echo 12 < 50  then
        digital pin 10 ouput HIGH
        wait 0.2 seconds
        digital pin 10 ouput LOW
        wait 1 seconds
```


Start
Serial display ultrasonic distance
Ultrasonic Sensor Distance＜50cm ?
N
Y
Delay 1S
Buzzer Buzz
Delay 0.2S
Buzzer Stop

Further Reading

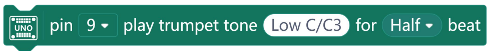
Application of Ultrasonic Ranging

As a typical non-contact measurement method, sonic ranging has been widely used in many occasions, such as industrial automatic control, construction engineering measurement and robot vision recognition.

Compared to to other ranging methods, sonic wave ranging has far less time measurement accuracy than laser ranging or microwave ranging, as sonic wave travels much slower then light and radio waves in the air. Therefore, the ultrasonic ranging system is easy to implement with a simple structure and low cost. Plus ultrasonic wave is not affected by smoke, air visibility and other factors during the propagation process, and is widely used in various occasions.

However, ultrasonic ranging has many limitations in practical applications, which all affect the accuracy of ultrasonic ranging. First, the ultrasonic wave is greatly attenuated in the air. Due to the difference in measurement distance, echo signal would fluctuate, causing high errors in the measurement of arrival time. Second, ultrasonic pulse echoes are greatly widened in the receiving process, which affects the resolution of the ranging, especially when applied to close range measurement. Ambient temperature, wind speed, and other factors will also affect the measurement accuracy to a certain extent. These factors all limit the application of ultrasonic ranging in some situations that require high measurement accuracy. It is of great practical significance in providing solutions to improve the ultrasonic ranging’ s measurement accuracy.

Further Exercise

Combining what you learned of module “ 
pin 9 play trumpet tone Low C/C3 for Half beat
” , can you control the buzzer to make sounds of different frequencies for different distances?

Chapter 6 Real-Time Test Device

Project 14 Ultrasonic Rangefinder

We already learnt how ultrasonic ranging works in the making of nearsightness alert In this section, we will learn to use an electronic screen to display the ultrasonic ranging results more intuitively.

Task Navigation

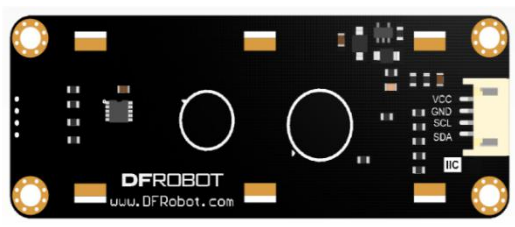
1. Learn about IIC LCD
2. Making an ultrasonic rangefinder

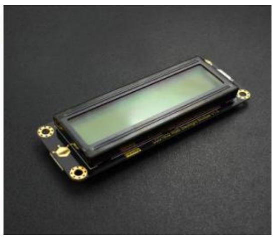
Key Points Analysis

IIC-1602 LCD Display

1.Up to 16 characters per line, two lines in total.

2.Clear the screen before output in case of garbled texts.





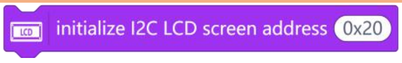
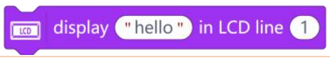
Mo.	Name	Description
1	VCC	Power+
2	GND	Power-
3	SCL	I2C Clock Line
4	SDL	I2C Data Line

We deal with VCC a GND everyday, but what are SCL and SDA?

First of all, we must know that IIC is a bus structure. Generally, there are two signal lines, one is the clock line SCL, and the other is the bidirectional data line SDA. SCL is called the clock line and SDA is called data line. Among all buses, IIC bus uses the least signal lines and has the most functions. Therefore, IIC bus is very convenient and flexible when used to design a computer system. It is

also small and can be widely applied. That’s what we need to know for now. We will learn more as we explore the programming world.
To use this module, you need to select the corresponding IIC address for initialization.

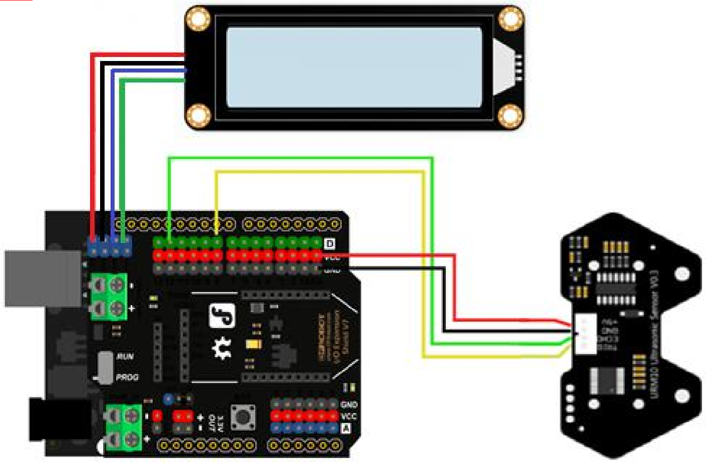
Command List

Module	Type	Description
 initialize I2C LCD screen address 0x20	initialization	Screen IIC address initialization settings
 display "hello" in LCD line 1	IIC display settings	Set what displays on each line on IIC

Hands-On

Hardware connection

Connect ultrasonic ranging sensor’s Trig to digital pin 8 pin, Echo to digital pin 12 pin, GND to GND, and + 5V to VCC. Connect the OLED screen to the I2C interface on the UNO.

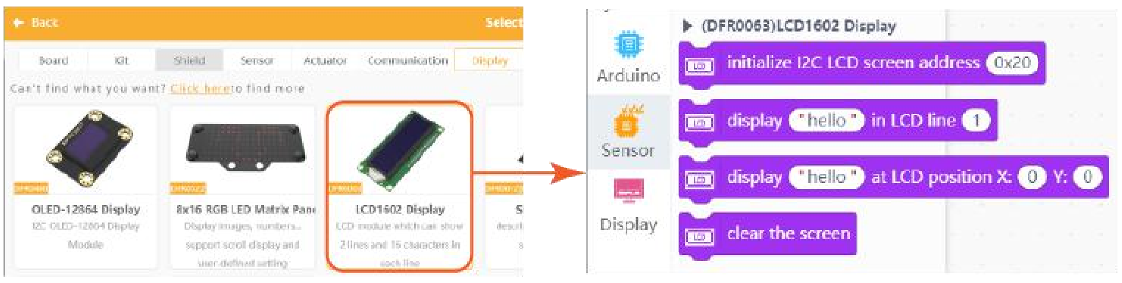


trig ——D8
echo——D12
GND——GND
+5V——VCC
(Note the wire matching and relative positions)

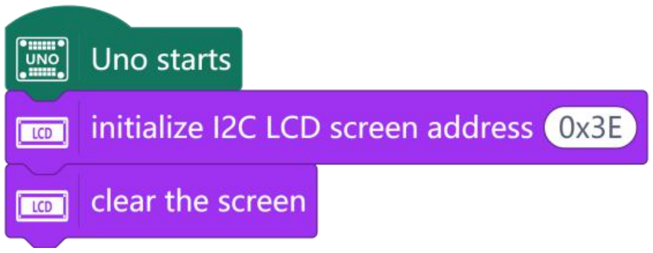
Programming

1. **Write the program**

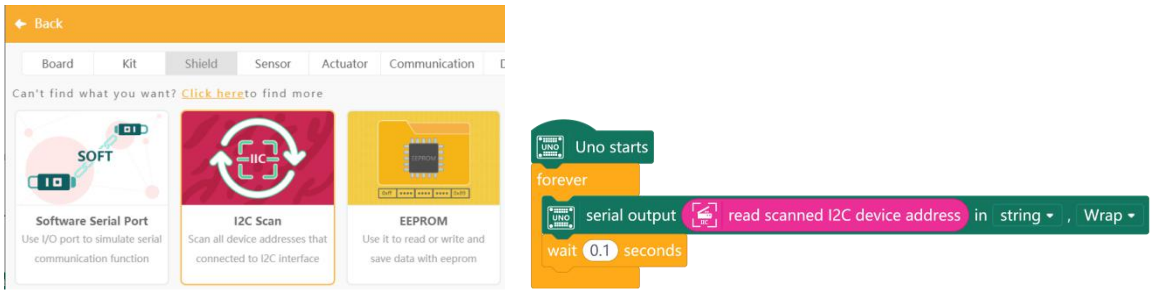
1) Click "Extensions" in Mind+ and select LCD1602 in "Monitors":


Back
Select
Board
Kit
Shield
Sensor
Actuator
Communication
Display
Can't find what you want? Click here to find more
OLED-12864 Display
8x16 RGB LED Matrix Panel
LCD1602 Display
Arduino
Sensor
Display
(DFR0063)LCD1602 Display
initialize I2C LCD screen address 0x20
display "hello" in LCD line 1
display "hello" at LCD position X: 0 Y: 0
clear the screen

2) Initialize IIC address:

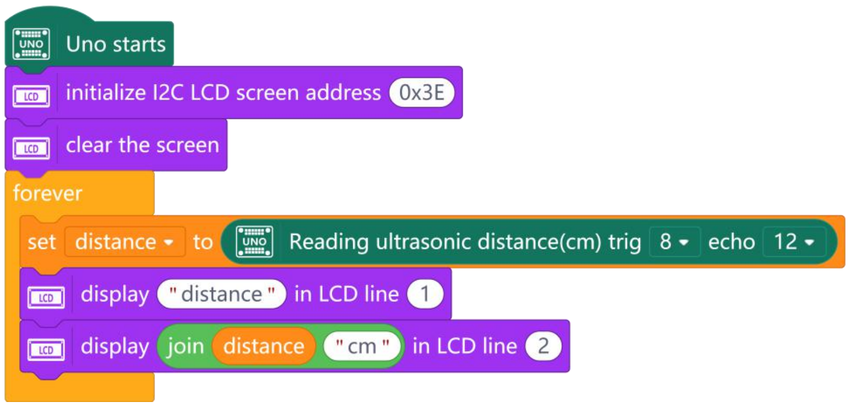

```
Uno starts
initialize I2C LCD screen address 0x3E
clear the screen
```

3) Generally, the IIC address of this screen is 0x3E. For any special cases, we can click "Extensions" to select "IIC functional address scan" in "Function modules" to load the building block read scanned I2C device address to scan the current IIC device device address.


Back
Board
Kit
Shield
Sensor
Actuator
Communication
Can't find what you want? Click here to find more
Software Serial Port
Use I/O port to simulate serial communication function
I2C Scan
Scan all device addresses that connected to I2C interface
EEPROM
Use it to read or write and save data with eeprom
```
Uno starts
forever
    serial output read scanned I2C device address in string , Wrap
    wait 0.1 seconds
```

When more than one IIC address is scanned, it means that a device with IIC is embedded in the device, then we need to try one by one until the screen is lit up.

2. Reference program

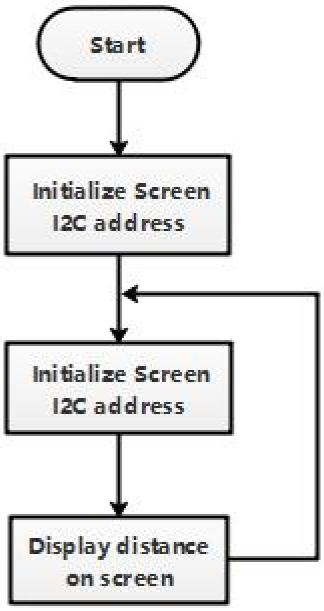

```
Uno starts
initialize I2C LCD screen address 0x3E
clear the screen
forever
    set distance to Reading ultrasonic distance(cm) trig 8 echo 12
    display "distance" in LCD line 1
    display join distance "cm" in LCD line 2
```

3. Program results

The screen lights up and the ranging results are output in real time.

P.S. Currently it only supports English display.

4. **Program analysis**


Start
Initialize Screen I2C address
Initialize Screen I2C address
Display distance on screen

Further Reading

Display screens are used in all aspects of our lives, such as mobile phones, home televisions, home appliances, calculators, computers, and other devices we use every day. There are many types of screens. And even LCD has many sub-types. Also, we have the new OLED flexible displays, miniLED displays, and microLED displays. So, what are the types of displays and how are they defined?

1) Seven-Segment Displays

Seven-Segment Display was invented by Japanese and was introduced into China in 1980s. It was used to replace LED digital tubes (consisting 7 segments for displaying numbers 0-9) in calculators and clocks. They can only display numbers and are simple in structure.

There are many alternative names too: segment LCD, small-sized LCD, "8" display, pattern LCD and others.

2) Matrix Displays

Matrix displays are divided into LCD matrices and LED matrices. In simple terms, matrix displays are many dots arranged together, which are called pixels. LCD matrix displays consist of dots arranged in an array. For example, 12864 display consist of 128 dots horizontally and 64 vertically.

3) TFT color display

TFT is a type of LCD, and is the prototype of all current improved LCD displays. They were installed on old mobile phones. It is also a matrix display with pixels, where the color is the concept of color levels. Color level is an index standard reflecting the brightness of the LCD screen, commonly referred to as the color index. There are generally three color qualities commonly seen on the market today: 256 colors, 4096 colors, and 64K (i.e., 65536) colors and even higher 260k colors. Display quality varies with different color qualities. They can display three kinds of content: general text, simple images (like cartoons, which are mainly used for menu items and standby screens), and photo images. As for users with higher requirements for photo quality, 64K color is certainly a better choice.

4) LED Displays

LED display is a simple concept. It is made of many LED lights. The shop signs we see everywhere are LED displays.

5) OLED Displays

OLED displays display images by self-emitting pixels. From this perspective, the technology of OLED screens is more advanced than that of LCD screens. In addition, the OLED screen can be made thinner, beneficial to the dimension utilization of the device.

Generally speaking, most displays are either LCD or OLED. There is an essential difference in how the light is emitted by two kinds of displays. One uses external light source while the other one is self-luminous. From a current perspective, two camps will continue to divide as different users have different needs for color performance.

6) miniLED Displays

miniLED, also known as "sub-millimeter light emitting diode", is an LED with a size of about 100 microns first proposed by Epistar. miniLED is a transitional technology between traditional LED and micro LED. It is an improved version of traditional LED backlight.

7) MicroLED Displays

The microLED screen is a new generation display technology with a matrix of miniaturized LEDs. In short, LED backlights are thinner and more compact, and LED units are smaller than 100 microns. Like OLEDs, each pixel is individually addressed and driven to emit light (self-emitting).

Further Exercise

Besides simple numbers display, we are also seeing coordinates commands in the building blocks loaded by the display. Think what else we can do with IIC LCD?

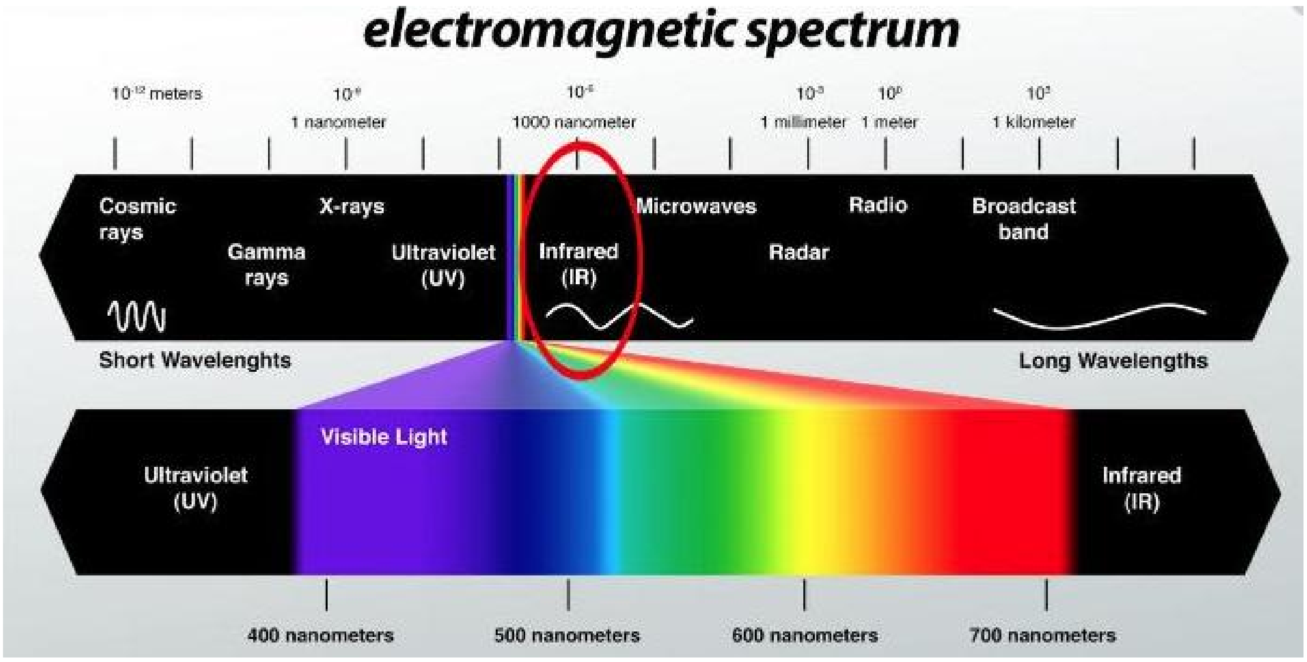
Project 15 Intrusion Detector

In the film *Entrapment*, there is a breathtaking scene when Virginia steals the famous painting. And what exactly are the red lines we see in the film? Is it visible to the human eyes? Let's find out.

Task Navigation

1. Learn about infrared ranging
2. Making an infrared intrusion detector

Key Points Analysis



1. **Infrared Rays**

Infrared is an electromagnetic wave with a wavelength between microwave and visible light, which is between 1mm and 760 nanometers (nm). It’ s an invisible light that has longer wavelength than red light.

Anything above absolute zero (-273.15 ° C) can generate infrared rays. Modern physics calls it heat rays. Medical infrared can be divided into two categories: near infrared and far infrared. Infrared rays contain heat and that’ s most of the sun’ s heat is transmitted to earth.

As shown in the figure, we call radiation outside red light infrared rays (and radiation outside purple light is called ultraviolet), which is invisible to the human eyes.

2. **Infrared sensor**

Infrared Obstacle Avoidance Switch

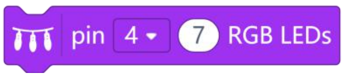
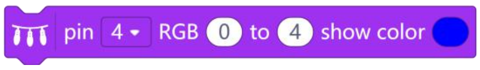
The button module used in Chapter II is a very typical digital input sensor. When we press the button, the signal port of the module will maintain high level; when the button is released, the signal port will maintain low level.

Are there any digital sensors that output signals without contatct?

Infrared obstacle avoidance switch. When an object approaches it, the yellow signal line will output a low level and the red LED on the sensor will light up; when the object is removed, it will output a high level and the red LED will go out.

There is another version with slightly different pins, brown -- + 5V; blue -- ground; black -- signal.

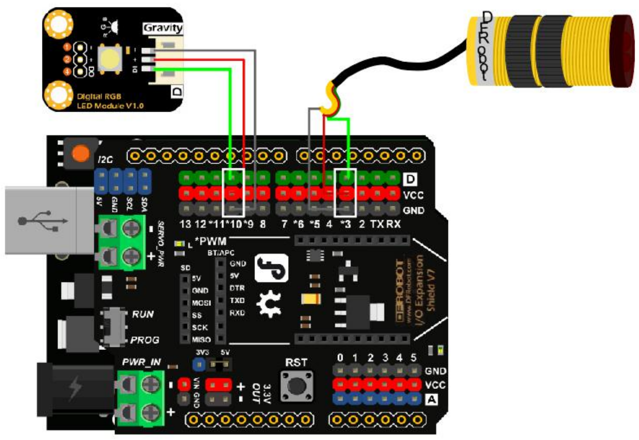
Command List

Module	Type	Description
 pin 4 ▾ 7 RGB LEDs	control brightness	Control the display brightness and number of RGB lights that are lit
 pin 4 ▾ RGB 0 to 4 show color 🔵	Control colors	Control the display colors of RGB lights

Hands-on

Hardware connection

Connect the infrared obstacle avoidance switch to Digital Pin 3, and the RGB light module to Digital Pin 10.



Hardware connection (IR obstacle avoidance switch --3, RGB light module --10)

Please match the colors when plugging

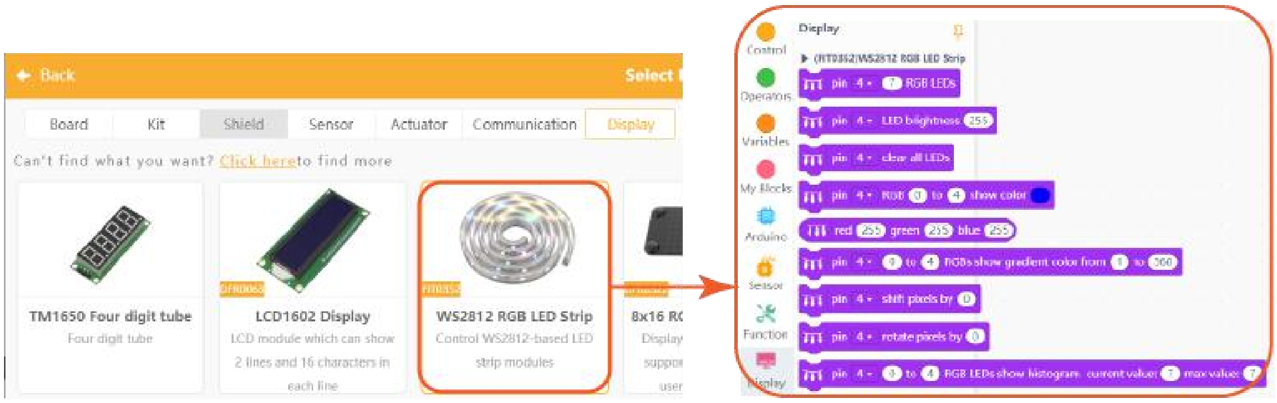
Programming

In the Adjustable Light chapter we learned how to control the LED brightness, with an primitive understanding of the mechanism of the RGB lights : RGB color mode uses the the RGB model give a intensity value between 0 and 255 to every RGB component in every pixel in the image. Various colors are constituted through different intensities in three primary colors, red, green, and blue.

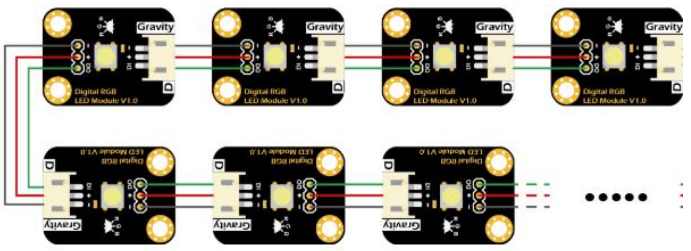
In fact, controlling RGB colors is not easy. Fortunately, the software has ready-made modules to reduce programming difficulty.

1. Write the program

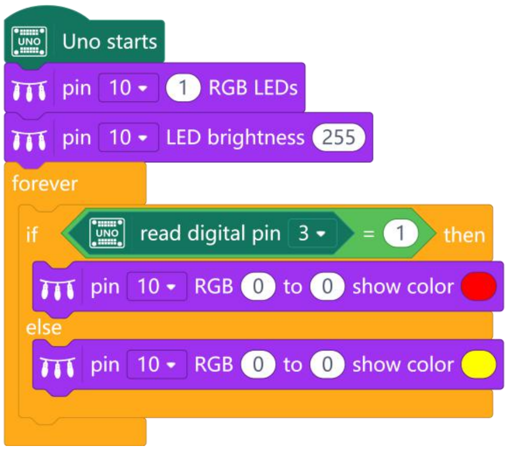
Open "Extensions" and select "RGB Lights" in "Monitor":


Back
Select
Board
Kit
Shield
Sensor
Actuator
Communication
Display
Can't find what you want? Click here to find more
TM1650 Four digit tube
Four digit tube
LCD1602 Display
LCD module which can show 2 lines and 16 characters in each line
WS2812 RGB LED Strip
Control WS2812-based LED strip modules
8x16 R
Control
Operators
Variables
My Blocks
Arduino
Sensor
Function
Display
(RT0352)WS2812 RGB LED Strip
pin 4 7 RGB LEDs
pin 4 LED brightness 255
pin 4 clear all LEDs
pin 4 RGB 0 to 4 show color
red 255 green 255 blue 255
pin 4 0 to 4 RGBs show gradient color from 1 to 360
pin 4 shift pixels by 0
pin 4 rotate pixels by 0
pin 4 0 to 4 RGB LEDs show histogram current value 7 max value 7

You might ask questions that why it’ s a strap of lights? The RGB light module we actually use is only one light bulb of the them. This module is also applicable to single RGB bulb, which can be cascaded by wires.



2. Reference program



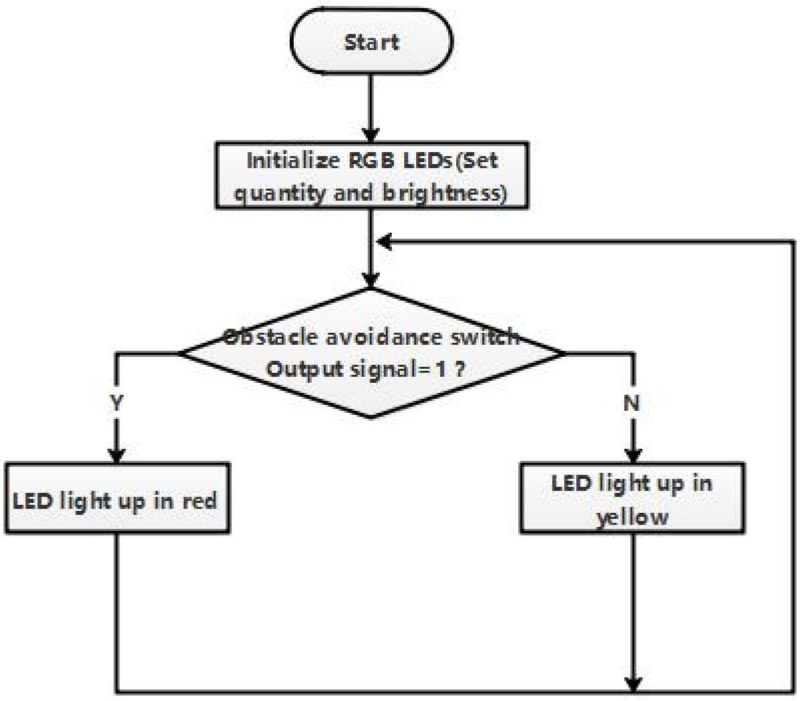
```
Uno starts
pin 10 ▾ 1 RGB LEDs
pin 10 ▾ LED brightness 255
forever
  if read digital pin 3 ▾ = 1 then
    pin 10 ▾ RGB 0 to 0 show color 🟥
  else
    pin 10 ▾ RGB 0 to 0 show color 🟨
```

3. Program results

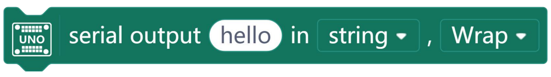
When the device is placed in an unoccupied room, RGB displays yellow; when someone enters the room and the device detects him/her, RGB displays red.

4. Program analysis

Why set "0 to 0 display"? We are accustomed to counting from 1, while computers count from 0. From "0 to 0" means only one RGB light is turned on. It starts with the first LED with index 0 and ends with the first LED at index 0.


Start
Initialize RGB LEDs(Set quantity and brightness)
Obstacle avoidance switch Output signal=1 ?
Y
N
LED light up in red
LED light up in yellow

If you want to see the data output in real time, remember to add building blocks "



serial output hello in string ▾ , Wrap ▾

"

Further reading

How did humans discover infrared?
How can a dark place be "hotter" than a bright place? The story begins at two centuries ago.
Before 1800s, people now the “white” light of sunlight can be dispersed into red, orange, yellow, green, blue, indigo, and violet lights. This experiment was first successfully carried by the famous Newton in 1666. 100 years later, nobody bothered to think what else exists besides these seven colors.
However, William Herschel (1738-1822), a British physician and astronomer who was born in Germany, came up with a whimsy. What exists “beyond” these seven visible lights in those invisible areas? So, he did the following experiment in 1800.
He made the sunlight pass through a prism and refract it to the white paper screen on the back. Of course, he also got seven colors like Newton. The difference is that this time he also put nine identical thermometers in each color area. Seven were put in every color area, while the last two were put “outside” red and purple. With the illumination of colorful light refracted by the sunlight, the temperature of the thermometers in the seven visible light areas have increased. For example, the temperatures of red, green, and purple areas have each increased by 5 °C, 3 °C, and 2 °C; but the temperature outside the purple light has not increased. He also found that the temperature outside the red-light area not only increased, but also rose even higher than the red-light area, by 7 °C! This was a surprise -- there was no light there!
Would the temperature be higher in the area farther from the red? Then he moved the thermometer to an area farther from the red area, but the temperature did not increase at this time, dropped to room temperature instead. After repeated experiments and researches, he finally determined that "infrared" or "infrared radiation" exists in the area near the red light. He also demonstrated with experiments that, infrared rays, no matter from earth or sun or other sources, also obeys the laws of refraction and refraction just like visible lights. But it is more easily absorbed by air than visible light. Because it is "invisible", it was called "invisible radiation" when it was first discovered.

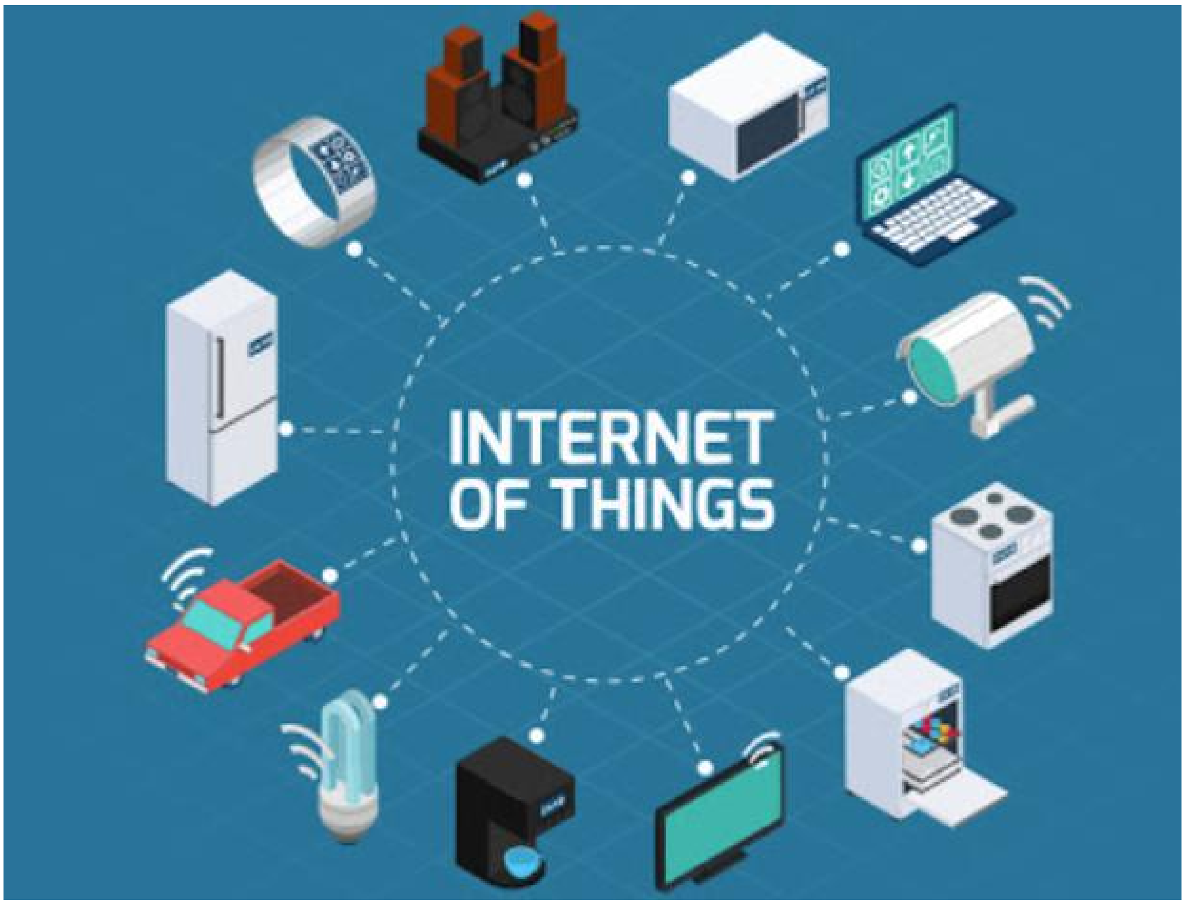
Further Exercise

Now that you have learned to make a light warning device, review the previous chapter and try adding a buzzer warning function to it!

Arduino and IOT

Chapter 7 Getting to Know the IoT (Internet of Things)

Over the past few centuries, mankind has experienced a series of technological revolutions, and each revolution is guided by a certain mainstream technology.For example: the industrial revolution of the 18th century opened the mechanical age; the second industrial revolution of the 19th century ushered in the electrical age; the 20th century celebrated information age, when Internet and computers became the symbol of the era;In 21st century, as senors, embedded



The Internet of Things means that everything is connected to the Internet

systems and internet popularized, Internet of Things is called as the third wave of world' s information industry revolution after the advents of computers and internet. As the backbone of current smart home development and urban construction, the Internet of Things will be applied to various fields and lead people into an era with increased intelligence.

Project 16 Learn about the Internet of Things

Task Navigation

1. Learn about the Internet of Things

2. Understanding the IoT infrastructure
3. Learn how the Internet of Things works

Learn about the Internet of Things

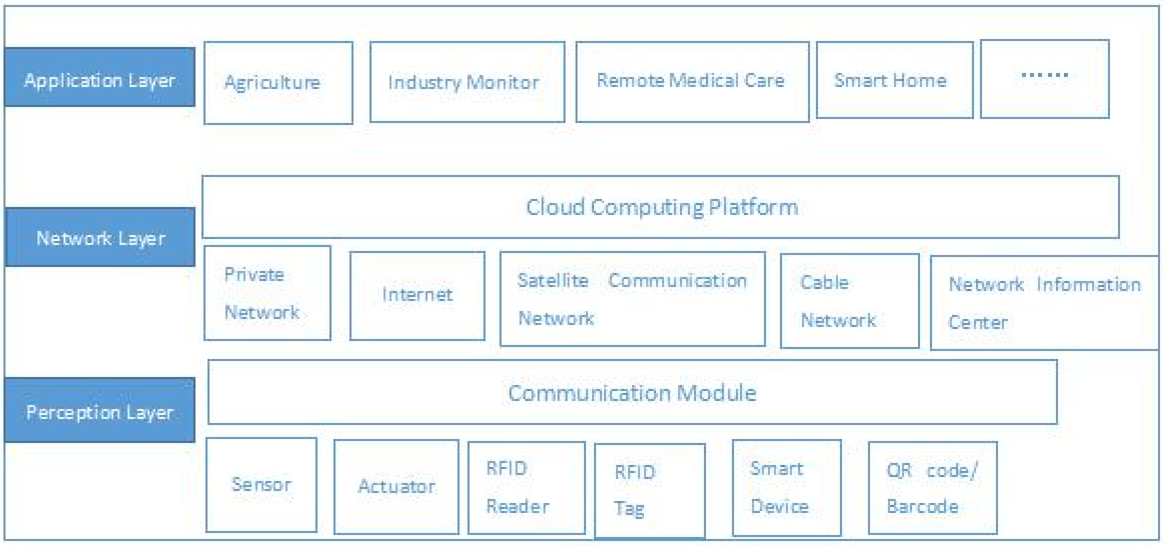
One morning, you wake up from your dream and find that it was 7:40. The 7 o'clock alarm you set last night did not go off! You are about late for work. You get out of bed in a hurry and run into the bathroom to brush your teeth and wash your face. No time for your messy hair. After you get dressed, you run into the kitchen, open the refrigerator, and find nothing inside as you forgot to buy new food yesterday. There is no way but to go to work without a breakfast. You grab the key, take the folder and come to the underground parking. When you drive the car out of the apartments block, you find that the road to the company is jammed due to rain. You can do nothing to traffic in front of you... Have you even been in these situations in your life?
Next, we imagine another day. At 7 in the morning, the sun shines through the curtains. The curtains in the room open automatically. The sun shines in. The weather forecast automatically broadcasts the weather and transmits the data to the wardrobe, which presents a selection of dress and shoes for you. While you are in the bathroom, the robotic arm in the kitchen picks up the toasted bread, and a cup of coffee, and places it on the dining table. After breakfast, you drive out. At this time, you take out the mobile phone and enter the starting point. The mobile phone has already told you the fastest route. When you arrive at the company, the mobile phone has also pushed the best parking location.
At noon, your mobile phone alerts you that there may be a biological invasion in your home. You immediately turn on the real-time monitoring and find that it' s a cat is lingering at your door. Then you find that the sockets were not turned off and windows and doors are not closed properly. Then you tap the phone, and the sockets and doors and windows of the home are automatically dealt with. The moment the door closed, the cat was startled and feet as if he was shocked by your "smart home".

In the afternoon, the grandfather calls from hometown and complains that vegetables grown at home withered in the greenhouse in the afternoon for no reason. Now, you use your phone and find that the soil humidity is too low and the temperature in the greenhouse is too high. You tap the phone and the sprinklers start to spray, and the A/C systems start running.

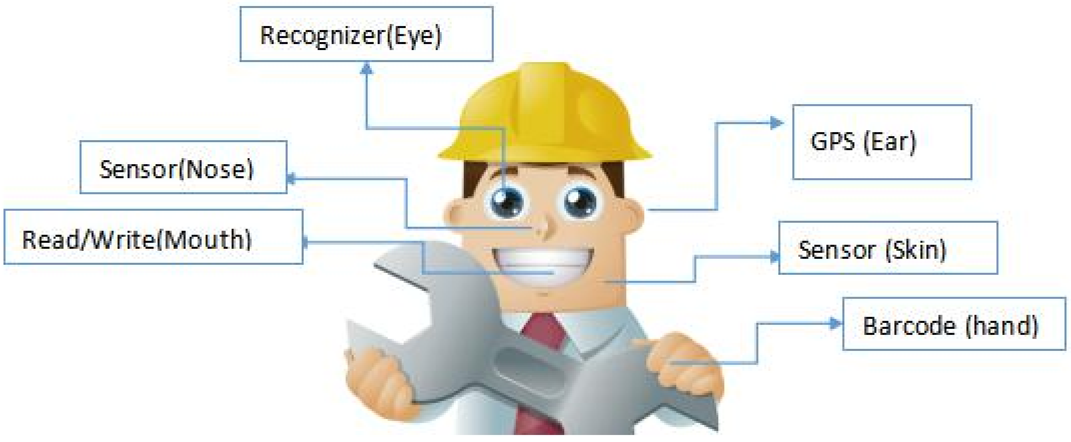
At the end of the day, you just want to lie down comfortably when you are home. So, on your way to home you start your home environment with one tap. At this moment, the air conditioner automatically turns on, the rice cooker starts to work, and the cleaning robot starts cleaning. When you get home, everything you think of a smart home is done for you.
Now you would think, these are the scenes in sci-fi films! Correct! In the future, these will become reality by the Internet of Things. So what is the Internet of Things?
The "things" in the Internet of Things (IoT) are all the items around you. Common things in daily life are: washing machines, refrigerators, and other home appliances; cars, houses and other fixed assets; even tables and chairs are included. Items are not so everyday including: thermometers, speedometers, etc. Therefore, Internet of Things is a world where everything is connected. In the Internet of Things, all items in your life can be connected to the Internet with electronic tags, and then they exchange information and communicate with each other to achieve smart identification, positioning, tracking, monitoring and management.

Understanding the IoT Infrastructure

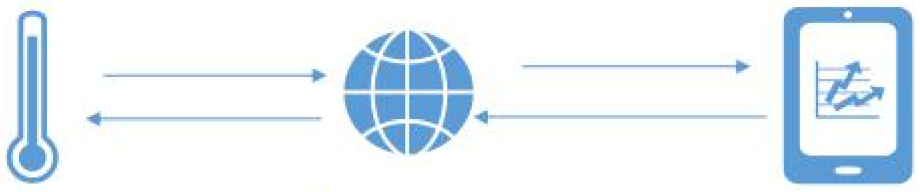
The Internet of Things, which enables each item to have the same "thinking capability" and "executive capability" as humans. Three basic architectures together make this possible: sensor layer, network layer, and application layer. As shown below.



The sensor layer is the bottom layer of the Internet of Things. It consists of various sensors and sensor gateways, including carbon dioxide concentration sensors, temperature sensors, humidity sensors, QR code tags, bar codes, RFID tags and readers, cameras, GPS and other sensor-terminals. The sensor layer works like human' s eyes, ears, mouth, nose, and skin, functioning to identify objects and collect information.



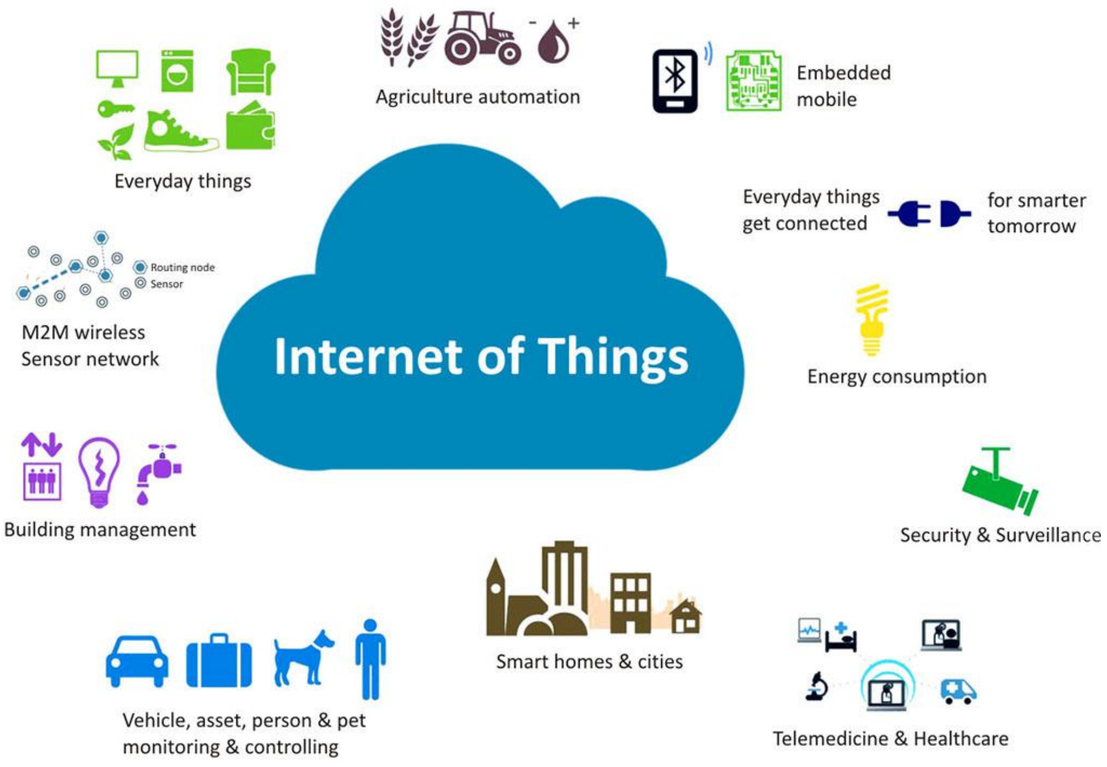
The network layer is the intermediate layer of the Internet of Things. It consists of various private networks, the Internet, wired networks, wireless networks, network management systems, and cloud computing platforms. Working like human’ s nerve center and brain, it is mainly responsible for transmitting and processing the information obtained by the sensor layer. The internet, together with other networks, transmits all information obtained by sensing layer, and facilitate the transparent data transmission between two end systems. Specific functions include addressing, routing, and the establishment, maintenance, and termination of connections.



Data Transmission on the Internet

The application layer is the interface between the Internet of Things and users (people, organizations, other systems). It is combined with industry needs to implement intelligent applications of IoT.

At present, the industrial characteristics of the Internet of Things are mainly reflected in its application fields. Currently, a number of industries such as green agriculture, industrial monitoring, public security, urban management, telemedicine, smart home, intelligent transportation, and natural resources monitoring, are enjoying the applications of IoT.



IoT Applications

Learn How the Internet of Things works

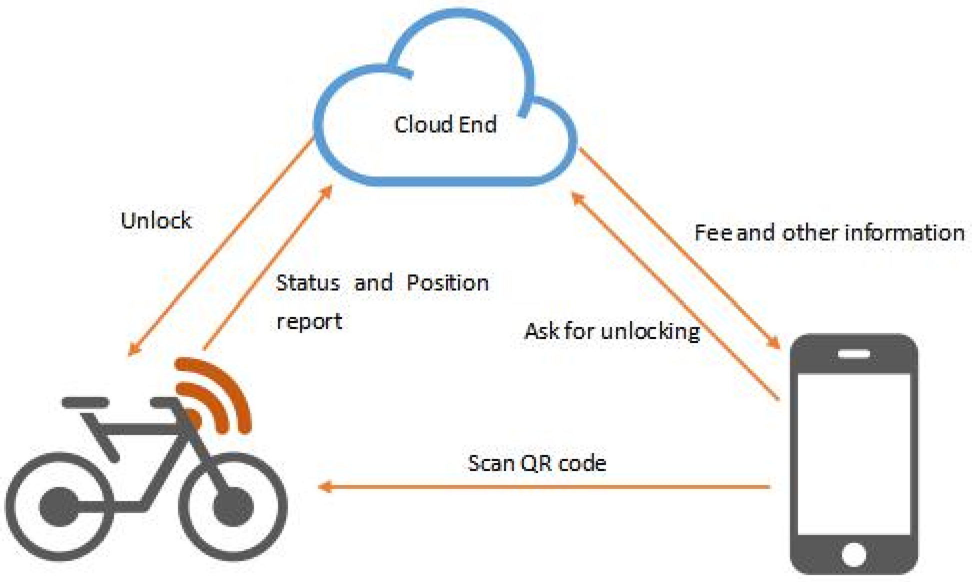
The Internet of Things is changing our lives at an increase pace. Driving, shopping, and even home heating is improved by IoT. With previous introduction to the basic infrastructure of IoT, do you have a clear understanding of how it works?

We implant the surrounding devices with sophisticated sensors and chips, which will continuously collect and send valuable data. These data will in turn give us a better understanding of how these devices are working together. But how should these devices share data at such a large quantity? Also, how should we utilize these data? Whether it is to improve the production efficiency of factories, update the best parking spots for urban residents in real time, or detect personal health conditions, IoT platform has a role to play in integrating information and provide a universal language for communication between **equipment** and **application software**. The entire process starts with the device itself. They communicate securely with the Internet of Things platform; the platform integrates and analyzes all the data, extracts the most valuable part, and sends it to applications that satisfy specific industrial needs.

Let's take a simple example: bicycle-sharing. Sharable bicycle is very convenient to use, and their most prominent feature is that scanning the QR code can unlock the bicycle. So the IoT aspect of sharable bicycle mainly uses the "phone-cloud-bike" architecture.

We can use the mobile app to find nearby bicycles, top up, reserve unlocks and more.

The cloud is the server side and the console of the entire bicycle-sharing system. It can communicate with all bicycles, collect information instructions, and respond to user and administrator commands.
The bicycle end is the end that collects information and executes commands, such as GPS positioning, unlocking, and etc.



Further Reading

What do you know about practical applications of the Internet of Things?
Fields of application of IoT are everywhere. Besides applications in bicycle-sharing, industry, agriculture, environment, transportation, logistics, security, there are other areas in our life where IoT is doing its job. Let' s have a closer look.

- Convenience Brought by Internet of Things

Almost every community block has some smart package storage lockers. The self-service pick-up and drop system are largely regarded as a new trend in logistics industry.
The smart package lockers can be left unattended. Once courier drops the package in the locker, the recipient will receive a message with a pickup code. The recipient can open the locker with the pickup code.

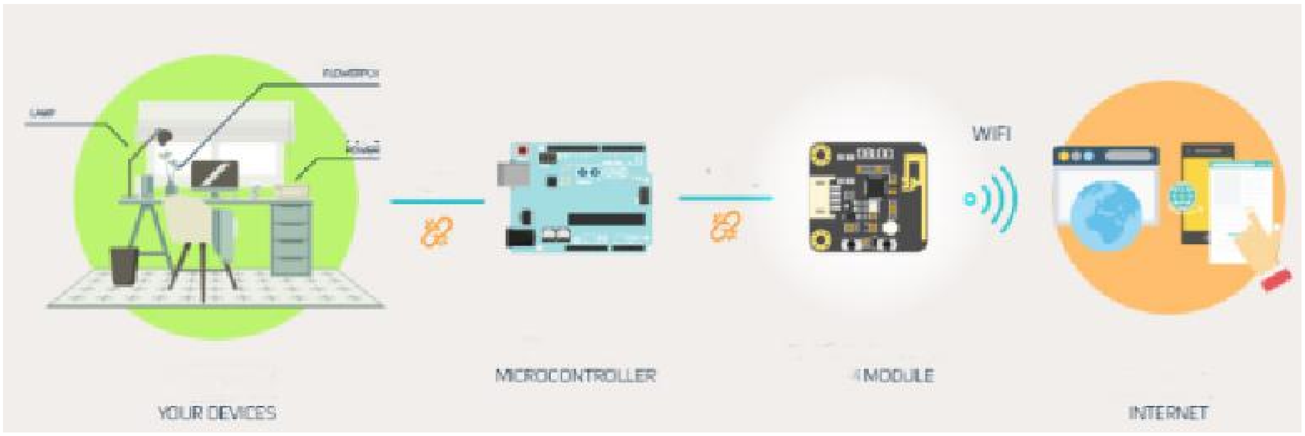
- Clothing Industry with the Internet of Things

In recent years, RFID integrated system vendors are exploring the overall application of RFID system with merchants. Besides warehouse management, RFID is also applied at front-end store sales process, in order to understand the hot-selling styles in time before the sales data is reported. So merchants can adjust production plan on the fly, and avoid inventory backlog, which is particularly important for fashion and fast-selling clothing.

Every piece of clothing is attached with a unique RFID, so manual check and counting is a thing of past -- from factory to shelf. Because the clothes will "tell" the shop handler of their identity information. At the store, they themselves will report, in real time, how many customers take them off the shelves, or how many take them to the fitting rooms.
Right now, HLA, Decathlon, H&M, ZARA and other brands have all begun to deploy RFID systems. Clothes that can "speak" bring more intelligent production and save a lot of manpower.

- Healthcare with the Internet of Things

Healthcare is surely a major concern of our lives. In before, we need to register when we go to hospital, and we often need to wait for our examination results. Now, we can do registration online, and we can check our examination results on our phones. In addition, we can talk to doctors online, so that resources of the health system are fully utilized.
In our life, applications go far beyond in these three areas. IoT is also contributing tremendously in education, finance, catering, agriculture, film and television, etc. With the development of the Internet of Things, those near-magical scenes we see in sci-fi movies are increasingly likely to happen in real life.
In the previous chapters, we have already learnt the Mind+ programming tool and Arduino UNO mainboard. Other essential tools for making IoT projects are WiFi IoT IoT module and Easy IoT Platform. Let us learn how to communicate with the Easy IoT platform through the WiFi IoT IoT Module and upload local information to the cloud.



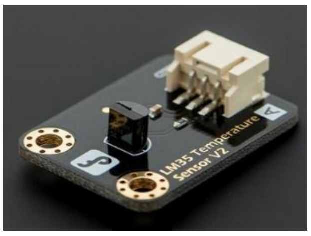
Project 17 IoT Temperature Detectors

Task Navigation

1. Learn to use WiFi IoT
2. Easy IoT platform
3. Mind+ IoT communication settings
4. Learn about temperature sensors
5. Send data to Easy-IoT

Key Points Analysis

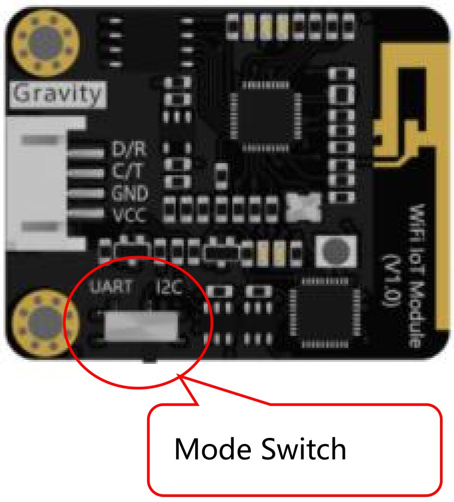
1. Learn about Temperature Sensor



This is a LM35-based semiconductor temperature sensor that can be used to detect ambient temperature.
Commonly used sensors for temperature measurement include thermocouples, platinum resistors, thermistors, and semiconductor thermometry chip. Among them, thermocouples are often used for high temperature measurement, platinum resistances are used for medium temperature measurement (to about 800 degrees Celsius), and thermistor and semiconductor temperature work within 100-200 degrees Celsius. Among them, semiconductor temperature sensor can be applied with little effort, and demonstrates good linearity and high sensitivity.
LM35 semiconductor temperature sensor is a linear temperature sensor produced by National Semiconductor. The temperature measurement range is -40 ° C to 150 ° C, with sensitivity at 10mV / ° C, and the output voltage is directly proportional to the temperature. The LM35 linear temperature sensor is used in combination with the Arduino dedicated sensor expansion board to facilitate interactions about ambient temperature sensing.

2. Learn to use WiFi IoT

Besides that, the module is designed with easy-to-use Gravity interface and employs UART and I2C communication protocols. You can use it to build IoT applications with other mainboards like micro:bit, Arduino, STM32, etBesides that, the module is designed with easy-to-use Gravity interface and employs UART and I2C communication protocols. You can use it to build IoT applications with other mainboards like micro:bit, Arduino, STM32, etc.
To avoid conflict with other UART devices, I2C mode is recommended.



No.	Name	Function
1	D/R	Data Line(I2C)/Receiver(UART)
2	C/T	Clock Line(I2C)/Transmitter(UART)
3	GND	-
4	VCC	+

First, Get to know the WiFi IoT module interface, indicator light, and mode switch function:

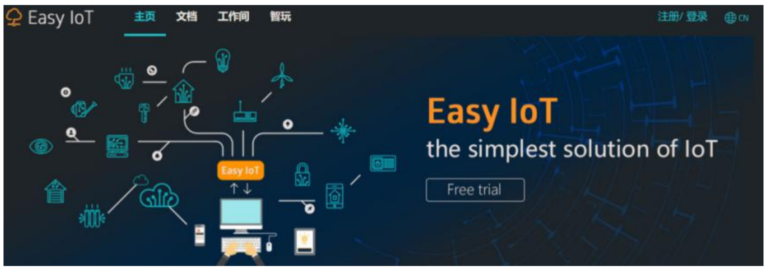
- Interface: The WiFi IoT module needs to communicate with the UNO main control board through the two wires of D/R and C/T, that is, the receiving and sending ends.
- Mode Switch: dial to I2C, I2C communication; dial to UART, UART communication
- Status indicator: When Wi-Fi is connected, it is difficult to troubleshoot network problems.

You need to use the indicator’ s color to determine the problem.

Color	**Indication status**
Red	Not functioning properly
Blue	Connecting to Wi-Fi
Green	Working properly
Purple	MQTT connection is down

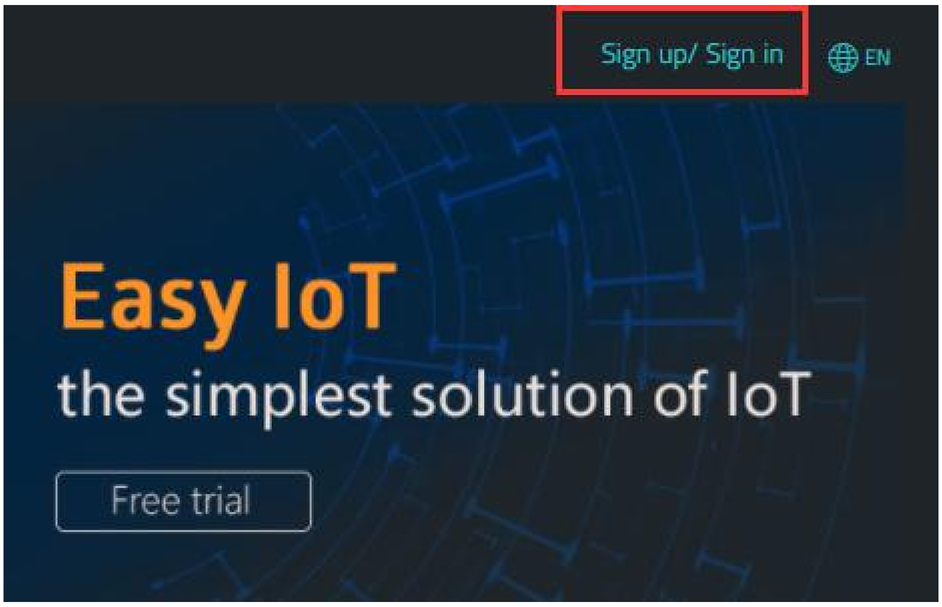
3. Easy IoT platform

Click IoT.dfrobot.com.cn to enter the EasyIOT platform, register for an account and log in.



- Register and log in

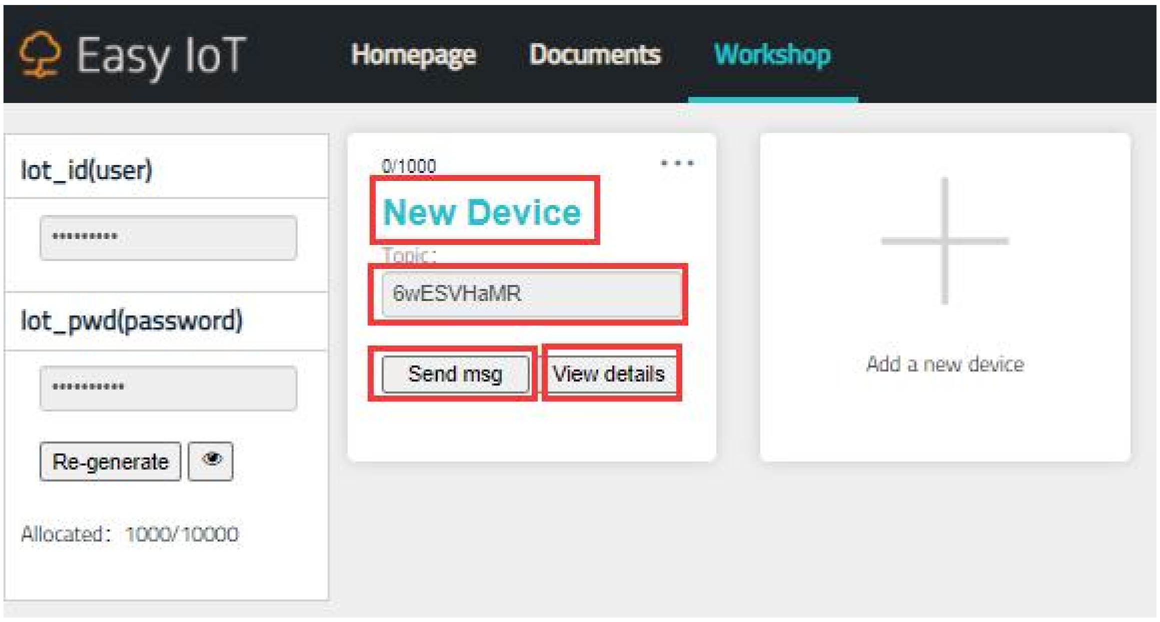
1) Before entering the terminal webpage of Easy IoT, you need to register a personal console monitoring account. Click " Register / Sign In " in the upper right corner.



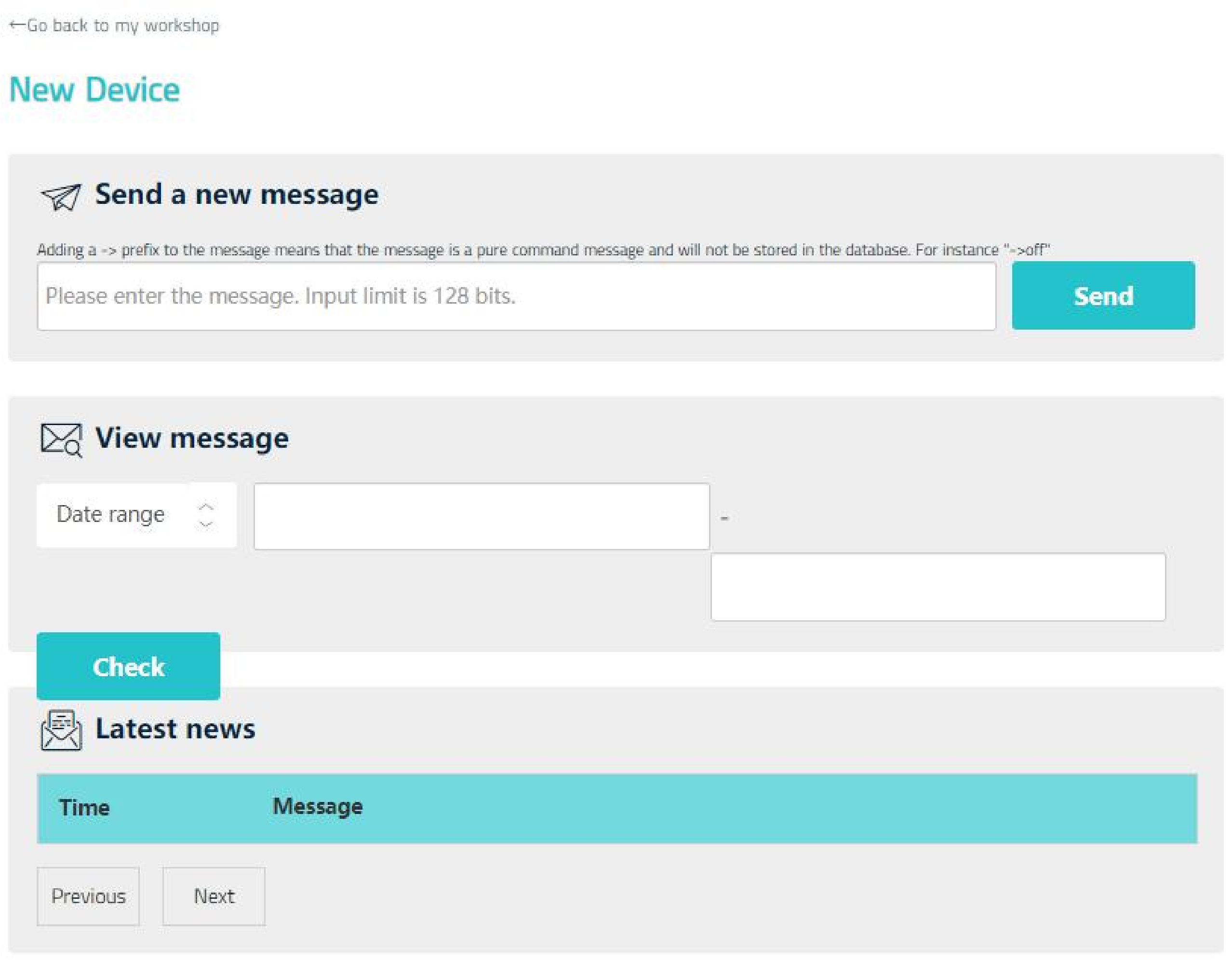
2) Fill in the personal information, you can register through your personal mobile phone

or email.

3) After the registration is successful, log in to your account to enter the workshop, as shown in the figure below. It is a data interface that can communicate with the local device. Click Add New Device to communicate with the device.



- Easy IoT and device communication settings
  1) Hover your mouse over New Device and you will see " ✎ ". You can change the device name by clicking on it.
  2) After adding a new device, a random topic name for the device will be generated. You can click on it and rename the Topic.
  3) Click " Send Message " to enter the send message details page


←Go back to my workshop

New Device

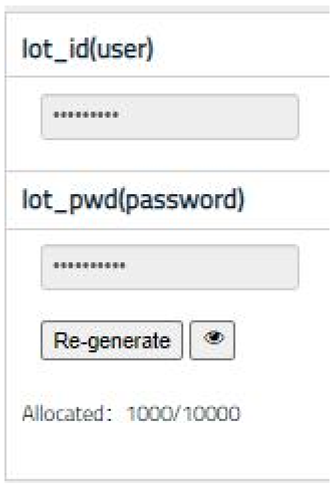
Send a new message

Adding a -> prefix to the message means that the message is a pure command message and will not be stored in the database. For instance "->off"

Please enter the message. Input limit is 128 bits.

4) Click " View Details " to see the generated icons.

- Others



5) IoT_id(user): ID, which is the user name. Since the ID randomly generated by Easy-IoT are unique, users can communicate with the device based on the IDs.

6) IoT_pwd (password): each username has its own password. Enhance account security protection.

7) 👁: Click on small eyes to view or hide the current account name and password.

8) Regenerate: Click to generate new IoT_id and IoT_pwd.

9) 已分配：2000/10000 : Each account name enjoys 10,000 free information storage.

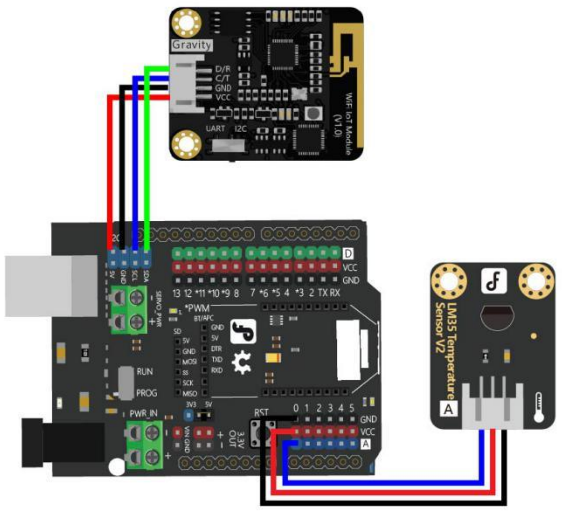
Command List

Module	Type	Description
Obloq send message "hello" to cloud platform Topic_0	Communication command	Send data to EasyIOT platform
read pin A0 LM35 temprature(℃)	Sensor command	Read temperature sensor data
Obloq mqtt initial parameter interface I2C	WiFi IoT initialization	IoT function initialization settings

Hands-On

Hardware connection

Connect the LM35 sensor to the analog pin A0, D/T of WiFi IoT to SDA/RX on the UNO board, C/R to SCL/TX on the UNO board, GND to GND, and VCC to VCC.



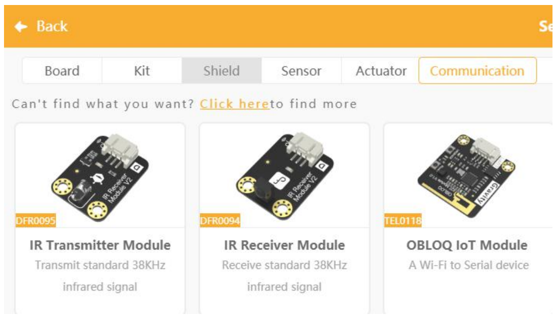
Please match the colors when plugging

Programming

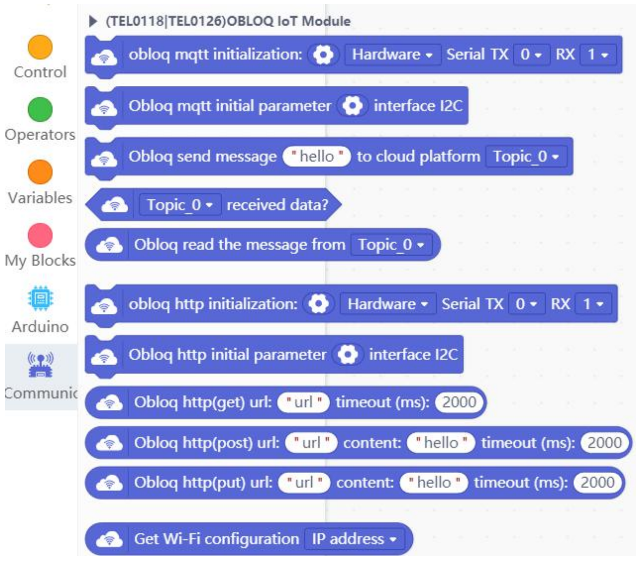
1. Write the program

The function we want to achieve is to make the Arduino read the LM35 temperature sensor data and send the temperature data to the IoT device through the WiFi IoT module.

After selecting the master as UNO, click on the Extensions and select the WiFi IoT module at the communication module.



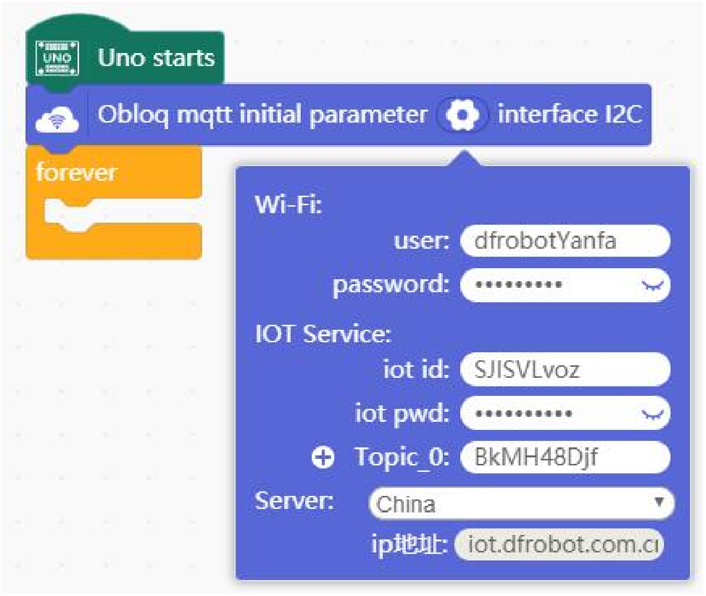
Then we can view the corresponding communication module:



(TEL0118|TEL0126)OBLOQ IoT Module
obloq mqtt initialization: Hardware Serial TX 0 RX 1
Obloq mqtt initial parameter interface I2C
Obloq send message "hello" to cloud platform Topic_0
Topic_0 received data?
Obloq read the message from Topic_0
obloq http initialization: Hardware Serial TX 0 RX 1
Obloq http initial parameter interface I2C
Obloq http(get) url: "url" timeout (ms): 2000
Obloq http(post) url: "url" content: "hello" timeout (ms): 2000
Obloq http(put) url: "url" content: "hello" timeout (ms): 2000
Get Wi-Fi configuration IP address

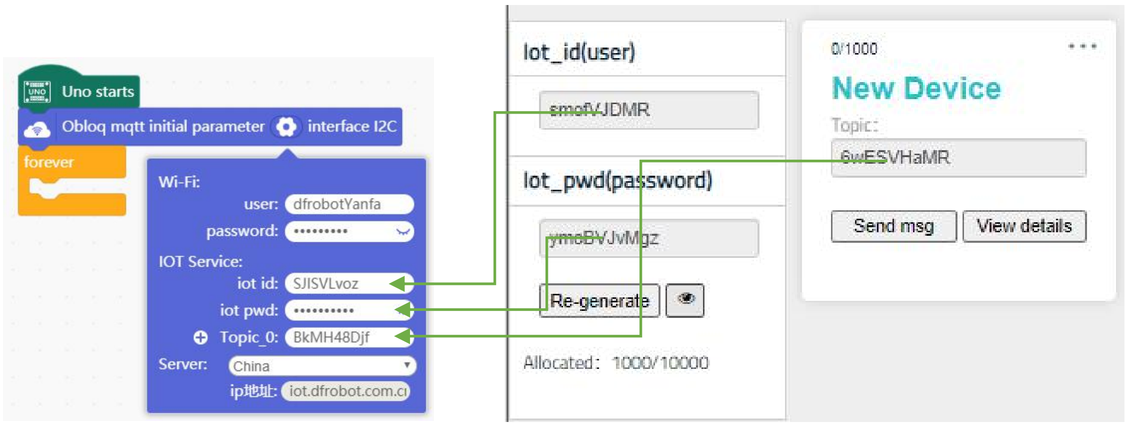
- Initial settings:

Click building blocks “Obloq mqtt initial parameter interface I2C” setting button “ ” to start initialization settings.



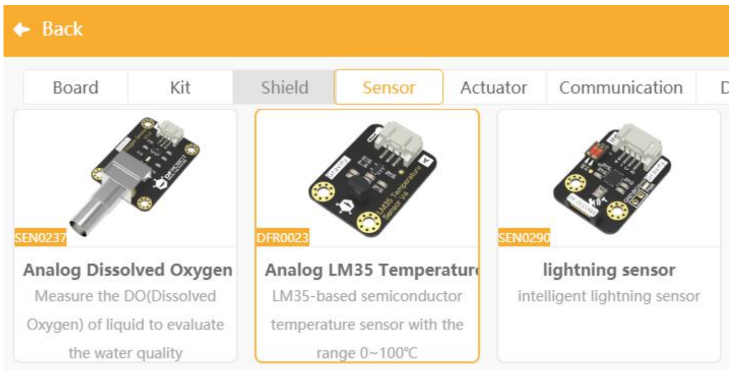
Wi-Fi:
user: dfrobotYanfa
password: ••••••••••
IOT Service:
iot id: SJISVLvoz
iot pwd: ••••••••••
Topic_0: BkMH48Djf
Server: China
ip地址: iot.dfrobot.com.cı

Here, the IoT platform parameters (IoT_id, lot_pwd, Topic) need to exactly match the parameters on the Easy IoT platform. Set the Wi-Fi account and password and connect to the network.

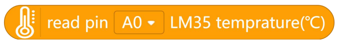


Here, select “hard port” option. Arduino UNO serial communication divides into hard serial and soft serial ports. What do they mean? Hard serial port sets Pin0 for RX and Pin1 for TX by default. However, you can define which Pin is for TX and which pin is for RX -- this is soft serial port. In this project, we use hard serial ports, which are 0 and 1.

Click on the extension and find the relevant function module L35 in the sensor:

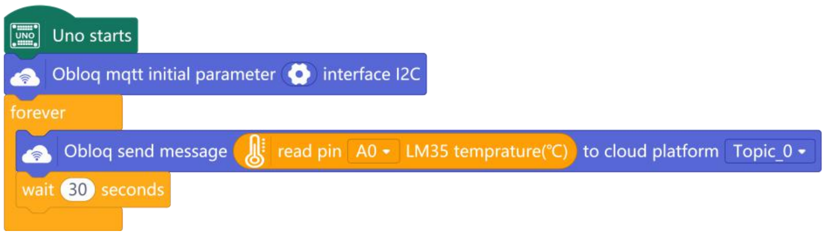

Back
Board
Kit
Shield
Sensor
Actuator
Communication
SEN0237
Analog Dissolved Oxygen
Measure the DO(Dissolved Oxygen) of liquid to evaluate the water quality
DFR0023
Analog LM35 Temperatur
LM35-based semiconductor temperature sensor with the range 0~100℃
SEN0290
lightning sensor
intelligent lightning sensor

After loading is complete, we can directly call and read the temperature sensor module "



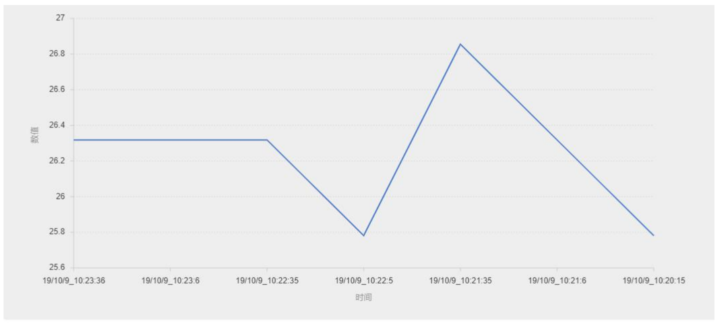
".

2. Reference program


Uno starts
Obloq mqtt initial parameter
interface I2C
forever
Obloq send message
read pin A0 LM35 temprature(℃)
to cloud platform Topic_0
wait 30 seconds

3. Program results

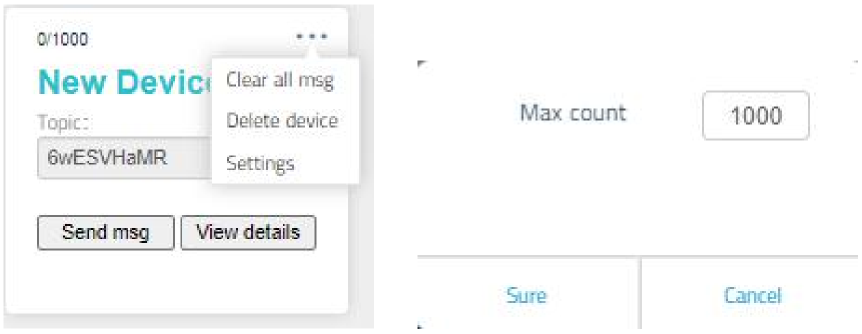
Click EasyIoT to see the details. You can see intuitive charts, the data of which can also be exported in the form of Excel sheets.



4. Troubleshooting

At the beginning, I forgot adding wait time. And temperature sensor also failed to properly read data. EasyIoT details page was flooded by “hello” messages, making temperature data that were properly read later very hard to find. Is the WiFi IoT transmission disconnected? But WiFi IoT’s light has always been green, and everything was OK.

After some efforts, I found that you can select “Clear all messages” at workshop page, and you can set message storage limit at “settings”. With the settings changes, the display problem of temperature data is solved.



Further Reading

Gateways and IoT gateways

What is a gateway?

To walk from one room to another, you have to pass through a door. Similarly, sending information from one network to another must also pass through a "door", which is the gateway. The gateway is also called a network connector and a protocol converter.

What is an IoT gateway?

In the architecture of the IoT, an intermediate device is needed between sensing layer and the network layer, and that is the "Internet of Things Gateway".

The IoT gateway can be used for both WAN and LAN interconnection. In addition, the IoT gateway also needs to have device management functions. Through the IoT gateway device, network operators can manage the underlying sensing nodes, understand the relevant information of each node, and implement remote control.

Further Exercise

Now that you have learned to send local information to the cloud, can you try sending information to your local device?

Chapter 8 Have fun with IoT

After learning to use WiFi IoT to upload local data to the IoT, we can try some projects. In this chapter, we will use the IoT to solve some pesky problems in our life, such as checking whether the delivery package has been carefully handled, and automatically control the laundry rack to take back clothes when it rains, and when the baby cries, alert parents automatically.

Project 18 Package Handling Monitoring

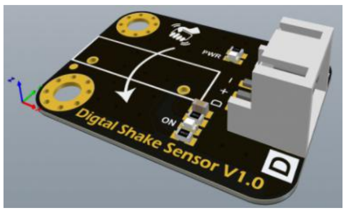
In the last few years, e-commerce has developed rapidly. And more and more people are shopping online. Occasionally, we receive damaged packages. What if it contains valuables insides?

Can we make a package handling sampling tester to check how many times the package is carelessly handled, and see how much damage it receives during the transportation?

Task Navigation

1. Learn how digital shake sensors work.
2. Test package delivery handling using digital shake sensor.

Knowledge point analysis



Learn how digital shake sensors work.

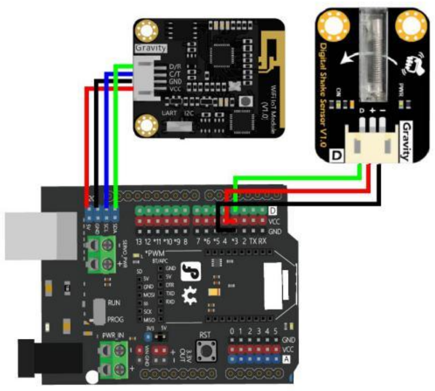
The digital shake sensor is a digital sensor that is only sensitive to one-way movement caused by human hand. It uses a spring-style vibration switch, and outputs high level while stationary. When the user shakes the sensor with power towards a predefined direction, the module outputs a low-level pulse and the onboard indicator light

flashes. Thanks to the unidirectional sensitivity of the vibration switch and the corresponding filter circuit, a medium-intensity collision or vibration (such as: drop, impact) will not accidentally trigger an interrupt signal. Therefore, the sensor is highly impact-proof and interference-proof.
When the user holds the module as shown in the figure on the right, and shakes it in the direction of the arrow (from + Y to Y-), a valid low-level interrupt pulse can be generated on the "D" Pin, and ON will produce bright flash. Shake, impact or vibration in X or Z direction will not generate effective interrupt pulse.
In the process of package delivery, the sensor can be placed inside the package, where it can detect the number of vibrations or whether it is upside down. Also, multiple sensors can work together to draw a picture of how the package is handled through the trip.

Hands-On

Hardware connection

Connect the digital shake sensor to digital pin 3, D/T of WiFi IoT to SDA on the UNO board, C/R to SCL on the UNO board, GND to GND, and VCC to VCC.



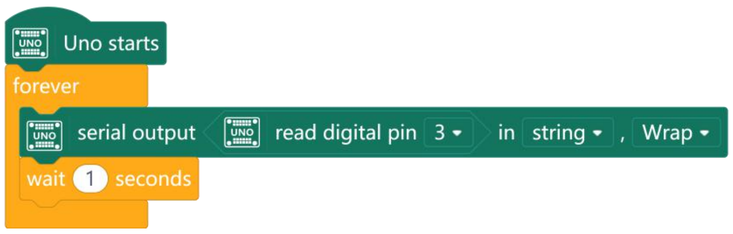
Hardware connection (digital shake sensor -- D3, WiFi IoT: GND-GND, VCC-VCC, D/T-SDA, C/R-SCL)

Please match the colors when plugging

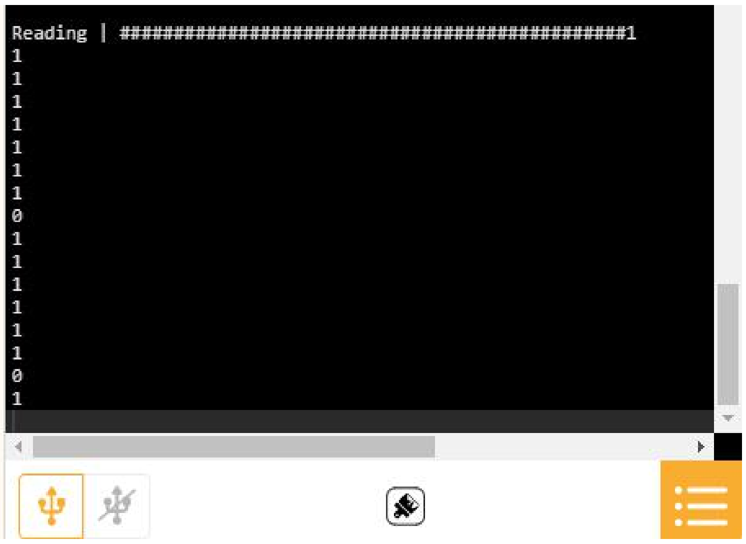
Programming

1. Write the program

The function we want to achieve is to send information to the cloud as soon as there is a vibration. Can we print out digital output information using a serial port for real-time monitoring? And can we find a suitable angular position to install the sensor?



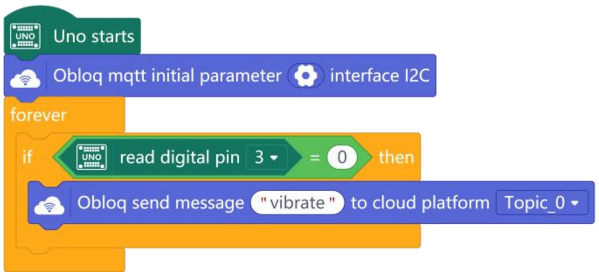
```
Uno starts
forever
    serial output  read digital pin 3  in string , Wrap
    wait 1 seconds
```



```
Reading | ##################################################1
1
1
1
1
1
1
1
0
1
1
1
1
1
1
0
1
```

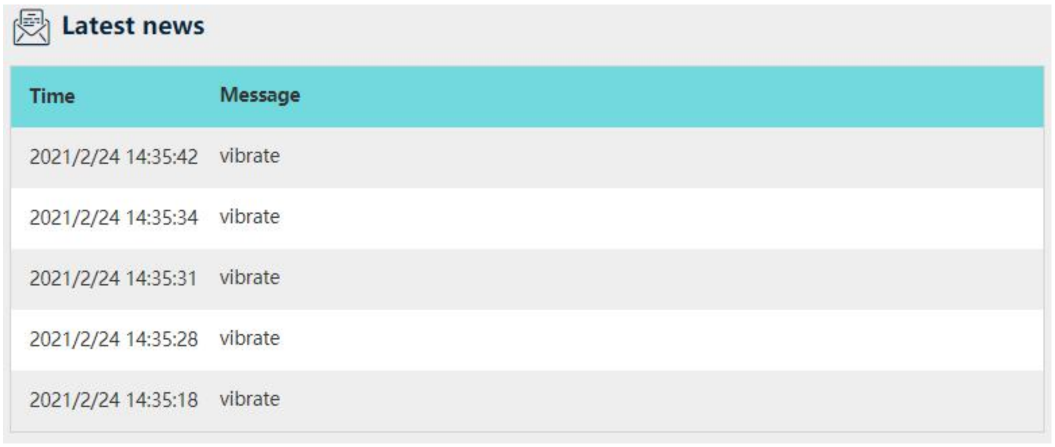
Very good. The reading is normal.

2. **Reference program**



```
Uno starts
Obloq mqtt initial parameter [⚙] interface I2C
forever
  if read digital pin 3 = 0 then
    Obloq send message "vibrate" to cloud platform Topic_0
```

3. **Program results**



Latest news

Time	Message
2021/2/24 14:35:42	vibrate
2021/2/24 14:35:34	vibrate
2021/2/24 14:35:31	vibrate
2021/2/24 14:35:28	vibrate
2021/2/24 14:35:18	vibrate

EasyIoT receives vibration information in real time, and achieves vibration monitoring.

4. **Program analysis**

Further Reading

Basic of Serial Communication

Arduino’ s Hardware Serial and Software Serial

Taking Arduino UNO as an example, there is only one set of serial ports on the panel, namely Pins 0 (RX) and 1 (TX). These two ports facilitate the communication between computer and Arduino. USB port is connected to two serial port pins via a conversion chip (often ATmega16 u2).

Ostensibly, the computer is not connected to these two pins via external wire, but it works the same way. When the Arduino controller is connected to the computer with a USB cable, a serial connection is established between the two. By this connection, the Arduino controller can communicate with the computer.

Usually, a serial port can only connect one device for communication.

Further Exercise

In Chapter VI Ultrasonic Rangefinders, we have learned how to use an LCD display. Can you modify the reference program to display the information on the electronic screen?

Project 19 Smart Baby Crib

"Baby baby go to sleep…" . The mother hums the lullaby by the crib, and gently rocks the baby to the dreamland. The mother pads out of the room for a glass of water and housework, while the closed door silences the baby's heartbreaking cry. When the mother comes back, the baby was already crying out of breath. The mother can only hold the baby and say sorry, with regret and reproach all over her heart. Wouldn' t it be better if there is a smart crib that reports any situations of the baby, and soothes the baby when the mother goes back?

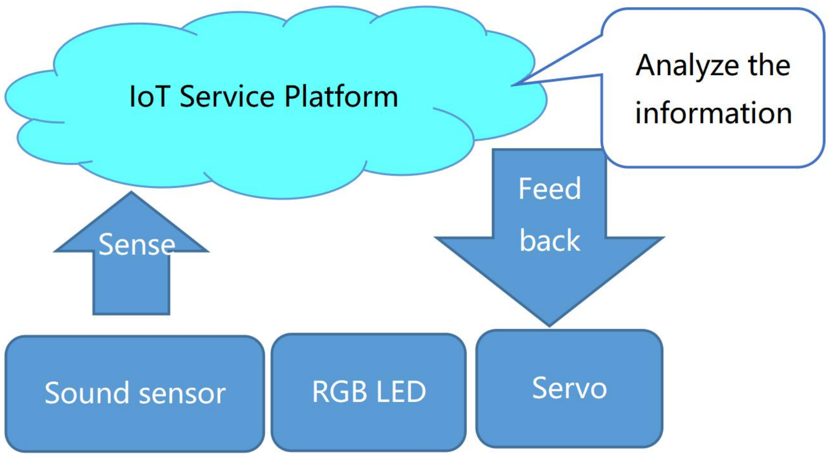
Let's try to make a smart crib now!

Task Navigation

1. Making a Smart Baby Crib

Knowledge Point Analysis

Make a smart baby cradle and connect a sound sensor to transmit the sound intensity value of the current environment to the IoT service platform. When value is received, check whether the baby is crying. Then send information to the WiFi IoT module, and send control command to the RGB light module, which starts to change colors. At the same time, it can also control the rotation angle of the servo to to make the crib swing.



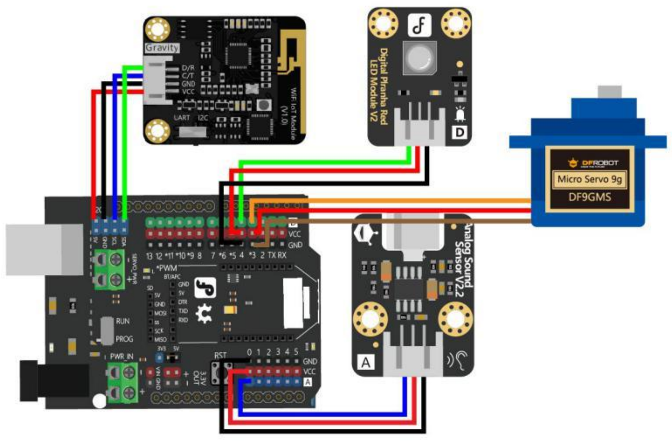
Command List

Module	Type	Description
	Operator	larger than or equal to

Hands-On

Hardware connection

Connect the analog sound sensor to the Analog Pin A0, servo to Digital Pin 3, the red LED light Digital Pin 4, D/T of WiFi IoT to SDA on the UNO board, C/R to SCL on the UNO board, GND to GND, and VCC to VCC.



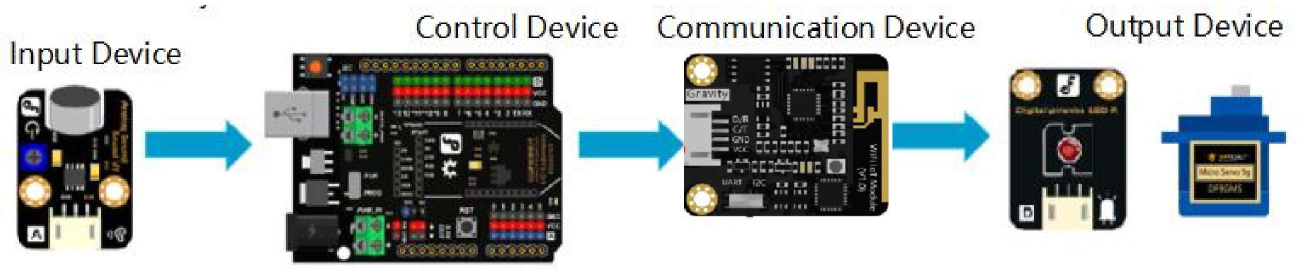
Hardware connection (analog sound sensor -- A0, servo -- D3,
LED-R -- D4，WiFi IoT: GND -- GND, VCC -- VCC，D/T-SDA, C/R-SCL)

Please match the colors when plugging

Programming

1. Write the program

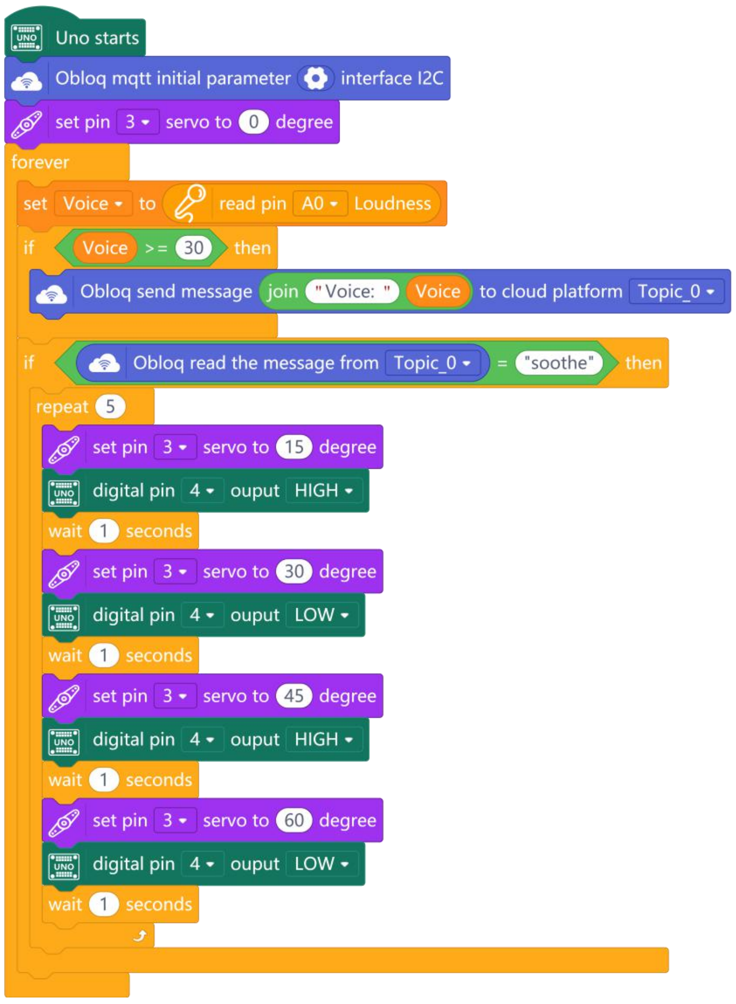
That’ s an awful lot of hardware at first glance. But just relax, everything except communication is what we already used before.



Still, we need to open “Extensions” in Mind+ to load corresponding modules.

The idea here is similar to the previous chapter. The only differences are temperature and humidity sensor is replaced by analog sound sensor, and output devices also include LED lights. The overall structure of the program remains largely the same.

2. Reference program



```
Uno starts
Obloq mqtt initial parameter interface I2C
set pin 3 servo to 0 degree
forever
    set Voice to read pin A0 Loudness
    if Voice >= 30 then
        Obloq send message join "Voice: " Voice to cloud platform Topic_0
    if Obloq read the message from Topic_0 = "soothe" then
        repeat 5
            set pin 3 servo to 15 degree
            digital pin 4 ouput HIGH
            wait 1 seconds
            set pin 3 servo to 30 degree
            digital pin 4 ouput LOW
            wait 1 seconds
            set pin 3 servo to 45 degree
            digital pin 4 ouput HIGH
            wait 1 seconds
            set pin 3 servo to 60 degree
            digital pin 4 ouput LOW
            wait 1 seconds
```

3. Program results

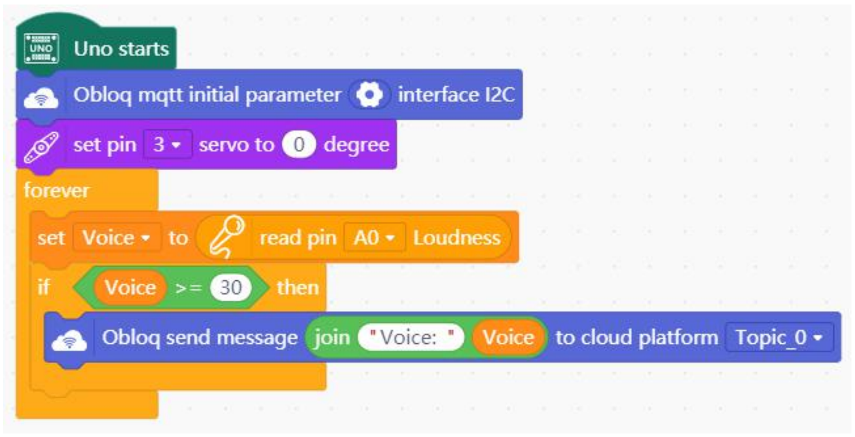
When the sound intensity is larger than 30, it sends a message to the cloud. When we receive the message, we send "soothe" command to WiFi IoT module. Upon receiving the command, lights turn on and servo rotates to soothe the baby, until sound intensity reduces.

4. Program analysis

the crib has to control the light flashing in addition to swing and stop actions to soothe the baby. Considering the actual application scenario, the swing has to be gentle, so the servo rotation angles shall be small and intervals shall not be too short.

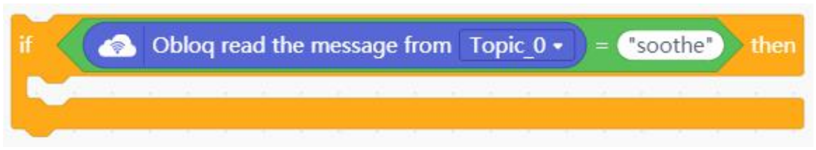
Next, we analyze the functions in two parts, upload and execution.

Initialize the servo and WiFi IoT



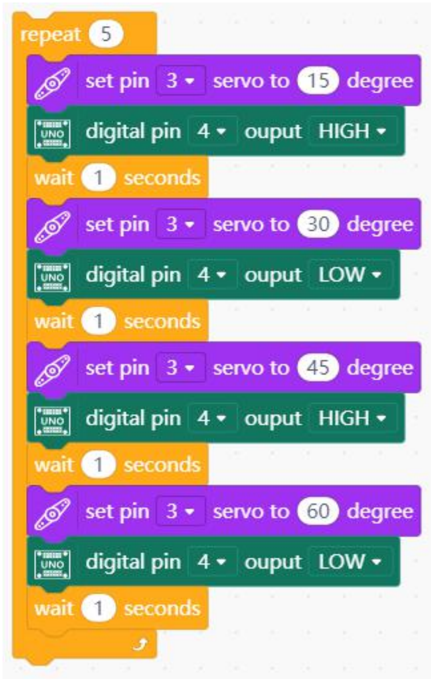
```
Uno starts
Obloq mqtt initial parameter ⚙ interface I2C
set pin 3 ▾ servo to 0 degree
forever
    set Voice ▾ to read pin A0 ▾ Loudness
    if Voice >= 30 then
        Obloq send message join "Voice: " Voice to cloud platform Topic_0 ▾
```

Upon receiving the "soothe" command, execute:



```
if Obloq read the message from Topic_0 ▾ = "soothe" then
```

Fill in what functions to perform:



```
repeat 5
    set pin 3 ▾ servo to 15 degree
    digital pin 4 ▾ ouput HIGH ▾
    wait 1 seconds
    set pin 3 ▾ servo to 30 degree
    digital pin 4 ▾ ouput LOW ▾
    wait 1 seconds
    set pin 3 ▾ servo to 45 degree
    digital pin 4 ▾ ouput HIGH ▾
    wait 1 seconds
    set pin 3 ▾ servo to 60 degree
    digital pin 4 ▾ ouput LOW ▾
    wait 1 seconds
```

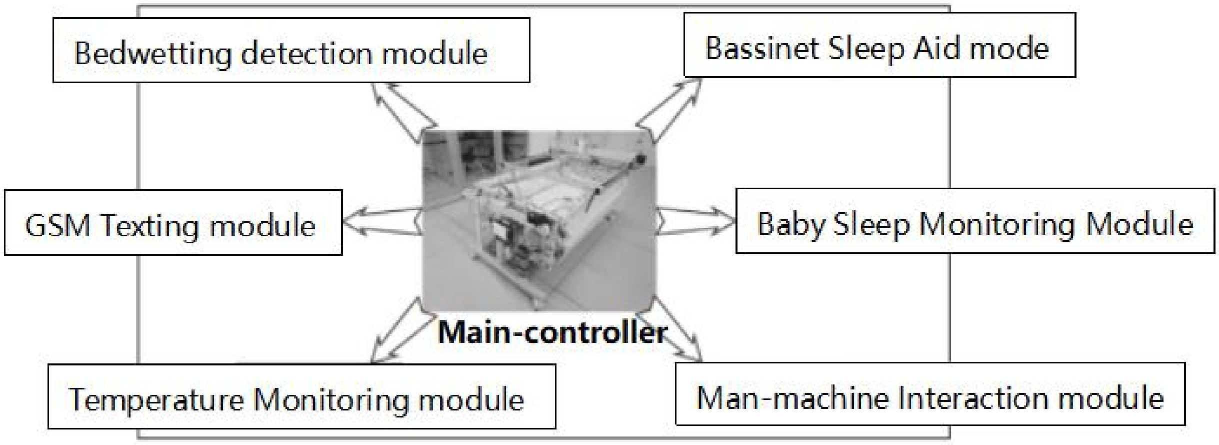
Further Reading

Smart Baby Swing Crib Design

With the rapid progress of society and the continuous improvement of life quality, the requirement of baby products is getting higher. As one of the most commonly used baby products, cribs have received the attention of many parents. Modern life’ s fast pace also greatly exhausts new parents when they have to do babysitting and maintain day-to-day work. Traditional crib adopts a fixed-legs or wheels design, as a result they can only be used as a bed. It cannot notify parents when baby wets the bed or soothe the baby when he/she is crying, a huge headache at night for parents. This article presents a design of a smart baby swing crib to solve the above problems.

Overall Design

The overall design of the baby swing crib in this article is shown in Figure 1. When the baby wakes up, the parents are first reminded by sound detection sensors to know the baby is no longer asleep. At the same time, the smart crib uses a motor connected to a swing arm to drive the automatic slow swing, accompanied by a recorded lullaby or mother’ s voice, in order to help the baby go to sleep again. For the baby’ s poo and pee, they are detected in real time through the installed temperature and humidity sensor; a text message will be sent to the parents or caregiver through the GSM module as an alert. And the temperature sensor detects the quilt temperature of the crib in real time, and can check whether the quilt is covered properly. In addition, a settings module can provide custom settings for every baby and parents’ vairous needs.


Bedwetting detection module
Bassinet Sleep Aid mode
GSM Texting module
Baby Sleep Monitoring Module
Main-controller
Temperature Monitoring module
Man-machine Interaction module

Further Exercise

With what’ s introduced in Further Reading, can you add new features to the crib?

Arduino Bluetooth communication

Chapter 9 Getting to Know Bluetooth

Bluetooth is closely related to our lives as a wireless communication technology. Bluetooth headsets and sound systems are all based on this technology. Bluetooth is a radio technology that supports short-range (usually within 10m) device communication. Its biggest advantage is that it

transmits data highly efficiently with very low power consumption.
In this lesson, we will learn how to use the Bluetooth module to achieve data communication between the mobile phone and the Arduino main control board.

Project 20 Bluetooth configuration

First, we need to configure the Bluetooth module. It mainly needs to configure the Bluetooth master and slave mode and naming of Bluetooth devices. This step will guide us to understand how to configure Bluetooth modules, and check if Bluetooth module is working properly.

Task Navigation

1. Understanding Bluetooth AT commands
2. Configure the Bluetooth module

Knowledge point analysis

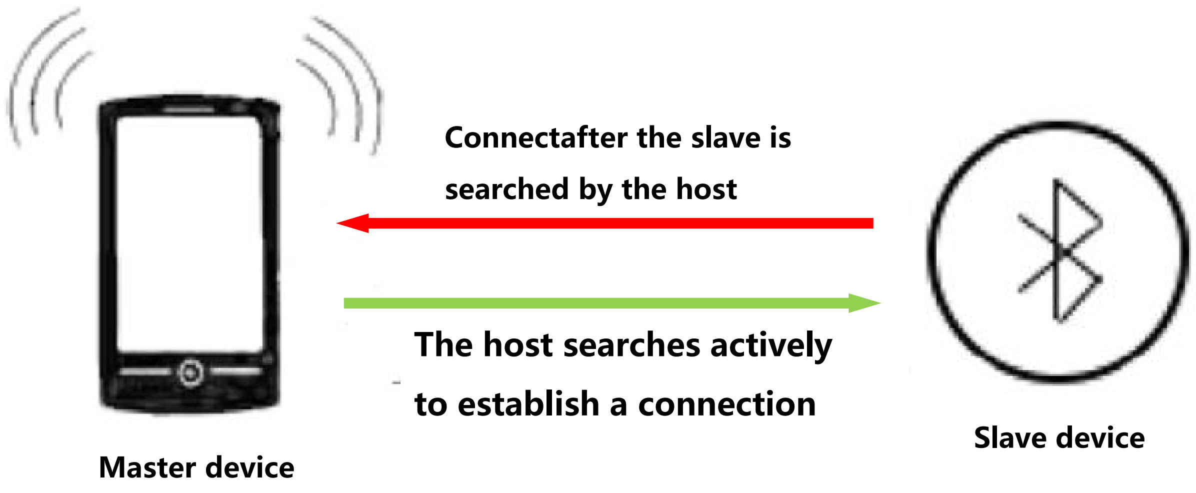
1. What is an AT instruction?

AT is the abbreviation of Attention. AT instruction is an instruction for connections and communications between Bluetooth devices. In this section we use AT commands to change the working status of Bluetooth. The list of commands can be referred to from BLE Bluetooth module AT command list.
In this section we mainly use the AT command to control the Bluetooth module to modify the name, set master-slave mode, restart, etc.

AT Command List	**Description**

AT+SETTING=DEFAULT	Restore initial settings
AT+SETTING=DEFPERIPHERAL	Return to the initial setting of the slave mode
AT+SETTING=DEFCENTRAL	Return to the initial setting of the master mode
AT+SETTING=?	Ask for initial settings (default: DEFPERIPHERAL).
AT+NAME=DF_Bluno	Set the module name to "DF_Bluno" (the name in quotation marks can be customized)
AT+NAME=?	Query the current module name. The default value is "Bluno" .
AT+RESTART	Restart the Bluetooth module

2. What is the master-slave mode of Bluetooth ?


Connectafter the slave is searched by the host
The host searches actively to establish a connection
Master device
Slave device

Master device mode: working as the master and can connect to a slave device. In this mode, you can search for surrounding devices and select devices that you want to connect to. In this section our master device is a mobile phone.

Slave device mode: The Bluetooth module working in slave mode can only be searched by the host, and cannot start a search. When connected to master device, slave device can send to and receive data from master device. In this section, we need to set the Bluetooth module to slave mode to be searched from master device.

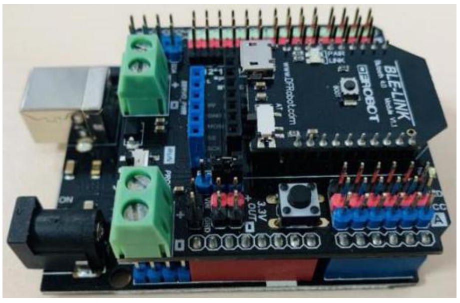
3. What is the difference between master mode and slave mode?

The master device is capable of searching other devices and initializing connection, while slave device cannot initialize a connection and has to wait for other devices to connect to it.

Hands-On

Hardware connection

First connect the Bluetooth module to the IO sensor extension board, and then connect the extension board to the Arduino UNO, as shown in the figure.

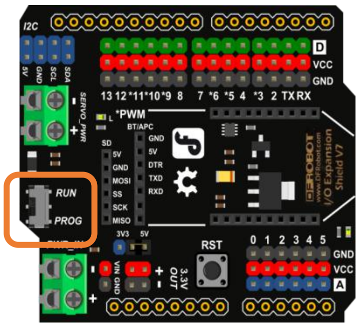


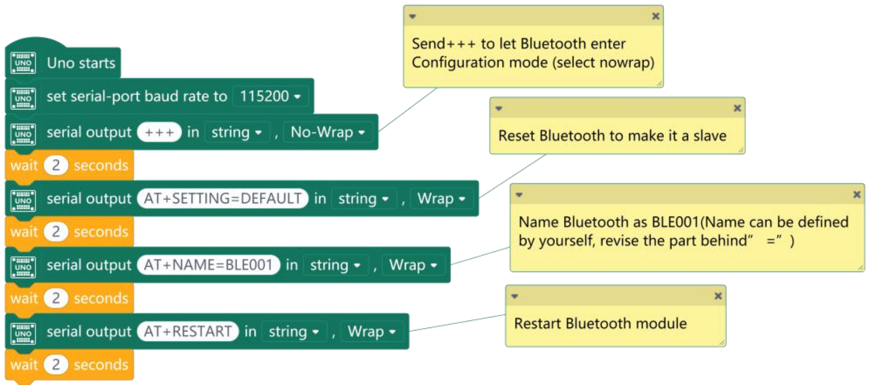
Programming

1. Reference program

The naming of the Bluetooth module: it is recommended to define a name for yourself. When multiple people use the BLE Bluetooth module at the same time, it is recommended that everyone set a unique name for Bluetooth so that it is easy to distinguish which module is yours. Keep the name in your mind after defining it, because it will be used in later cases.

When you upload the program, the dial on the extension board needs to be on PROG (upload mode). After uploading, switch it back to RUN to run the program.





```
Uno starts
set serial-port baud rate to 115200
serial output +++ in string , No-Wrap
wait 2 seconds
serial output AT+SETTING=DEFAULT in string , Wrap
wait 2 seconds
serial output AT+NAME=BLE001 in string , Wrap
wait 2 seconds
serial output AT+RESTART in string , Wrap
wait 2 seconds
```

Send+++ to let Bluetooth enter Configuration mode (select nowrap)

Reset Bluetooth to make it a slave

Name Bluetooth as BLE001(Name can be defined by yourself, revise the part behind" =")

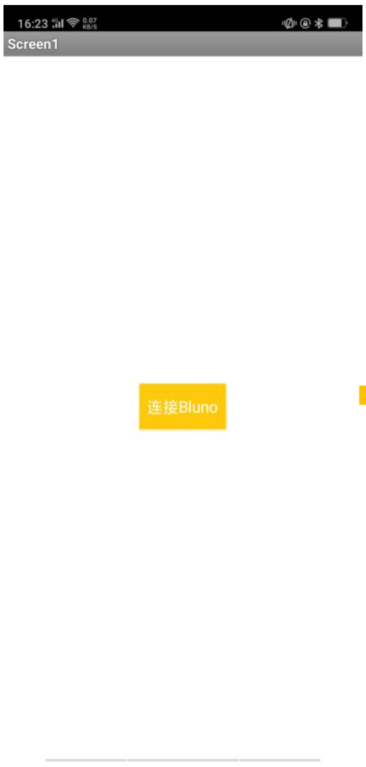
Restart Bluetooth module

2. Program results

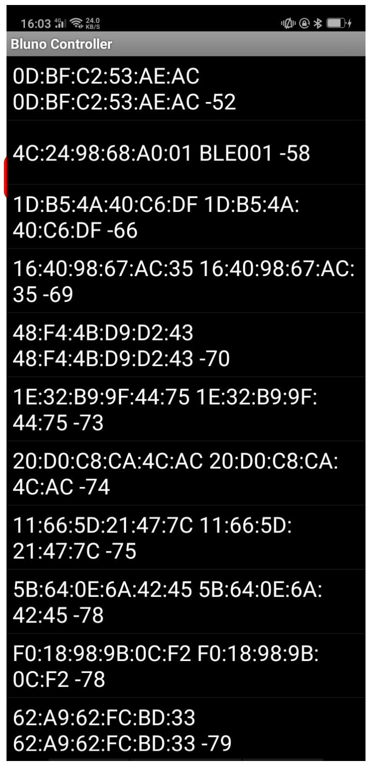
1. Connect Bluetooth with your phone

After completing the above configurations, we can use the provided debugging app (note that currently the Bluetooth control app can only be used on Android system phones) and search for the Bluetooth named "BLE001" .

(This chapter aims to help you get to know Bluetooth. First, you need to connect to Bluetooth using a mobile App. Download the "BLEtest.apk" , then install it on your mobile phone.)

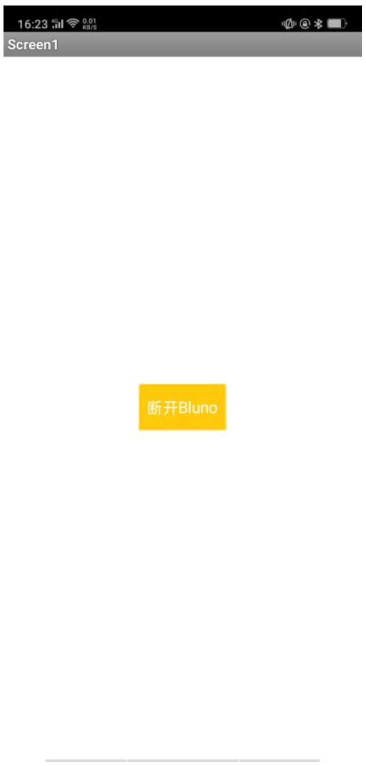


Click to connect Bluno



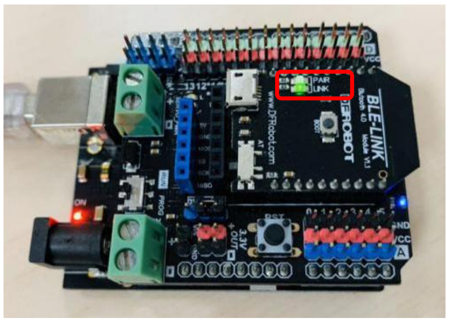
```
0D:BF:C2:53:AE:AC
0D:BF:C2:53:AE:AC -52
4C:24:98:68:A0:01 BLE001 -58
1D:B5:4A:40:C6:DF 1D:B5:4A:
40:C6:DF -66
16:40:98:67:AC:35 16:40:98:67:AC:
35 -69
48:F4:4B:D9:D2:43
48:F4:4B:D9:D2:43 -70
1E:32:B9:9F:44:75 1E:32:B9:9F:
44:75 -73
20:D0:C8:CA:4C:AC 20:D0:C8:CA:
4C:AC -74
11:66:5D:21:47:7C 11:66:5D:
21:47:7C -75
5B:64:0E:6A:42:45 5B:64:0E:6A:
42:45 -78
F0:18:98:9B:0C:F2 F0:18:98:9B:
0C:F2 -78
62:A9:62:FC:BD:33
62:A9:62:FC:BD:33 -79
```

Find BLE001 (Bluetooth Name) in the pop-up list



When the text on the button becomes "disconnected", it indicates a successful connection by Bluno.

Use the mobile app to connect the BLE Bluetooth module. After the connection is successful, the LINK indicator light will be on.



Project 21 Make A Phone App

Everyone we use a variety of apps on our phones. These apps are developed by various companies and individuals for us to use immediately. Therefore, all the features and interfaces are designed by other people. Sometimes we would want some custom interfaces and features, but as it requires a lot to develop an app, we often struggle to implement what we have in mid.

Now we can use **APP Inventor** to implement our ideas. In the first section, we learned the configuration method of the BLE Bluetooth module. Next, we will learn how to design and make a mobile app to work with the BLE Bluetooth module.

Task Navigation

1. Learn how to use APP Inventor

Key Points Analysis

What is app Inventor?
App Inventor is a mobile app development and programming software launched by Google. You can use graphical programming to design and make mobile apps. App Inventor is an Android programming environment developed entirely online. It has no complicated codes but features building blocks for Android program development.

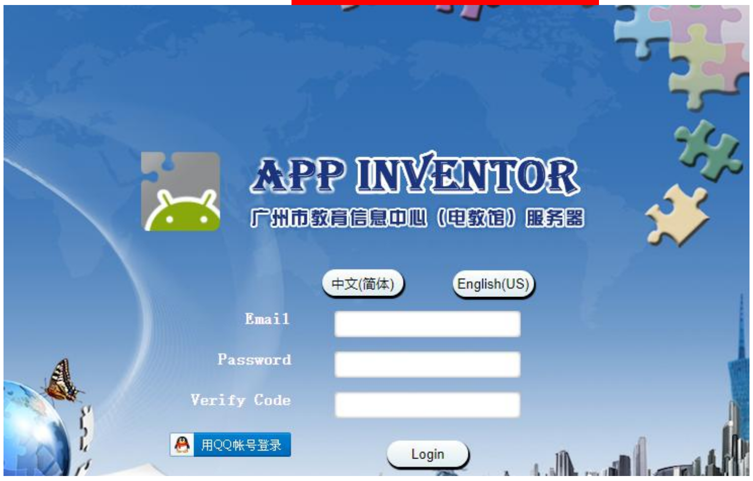
Hands-On

App production

1. Interface design

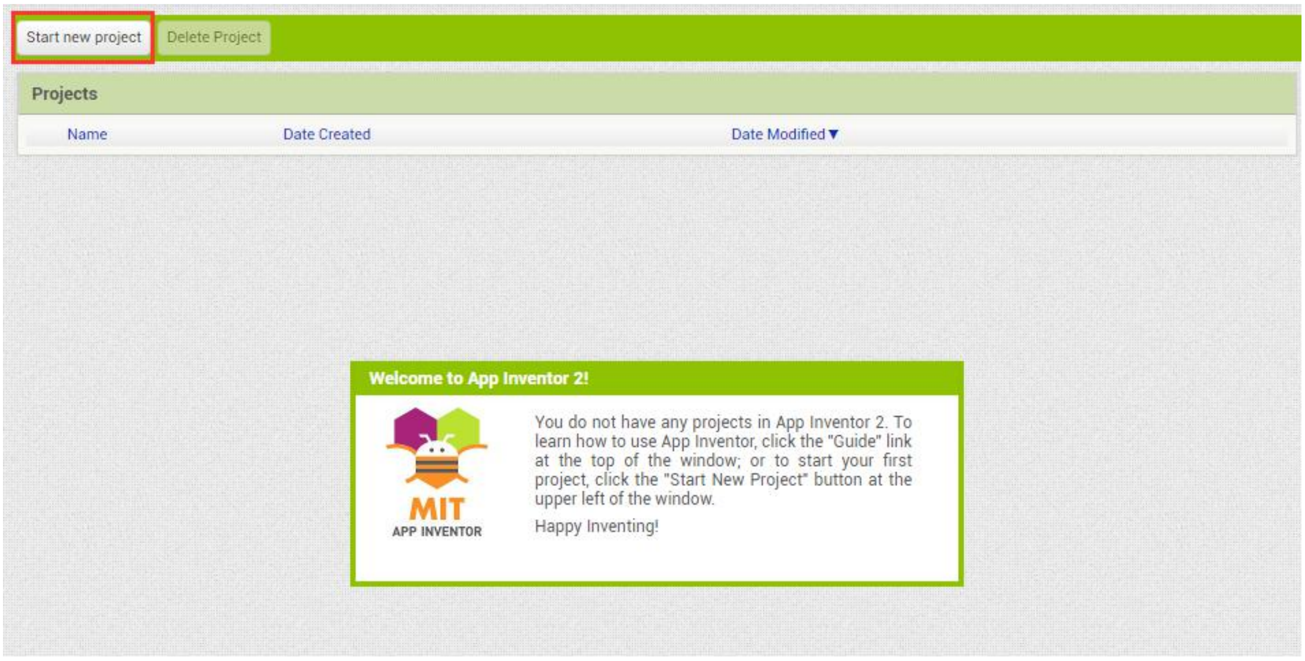
1) First open the appinventor web page: (http://app.gzjkw.net/login/)

You can register your own account or log in directly with QQ.



2) On the first log in, you will have to agree to Terms of Service. Read through the Terms of Service and click accept.

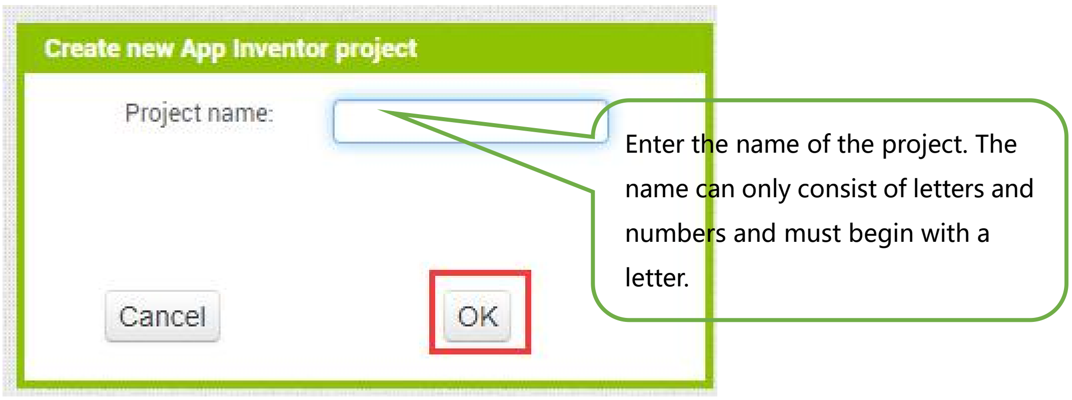
3) Click " **New Project** " on the interface after login.



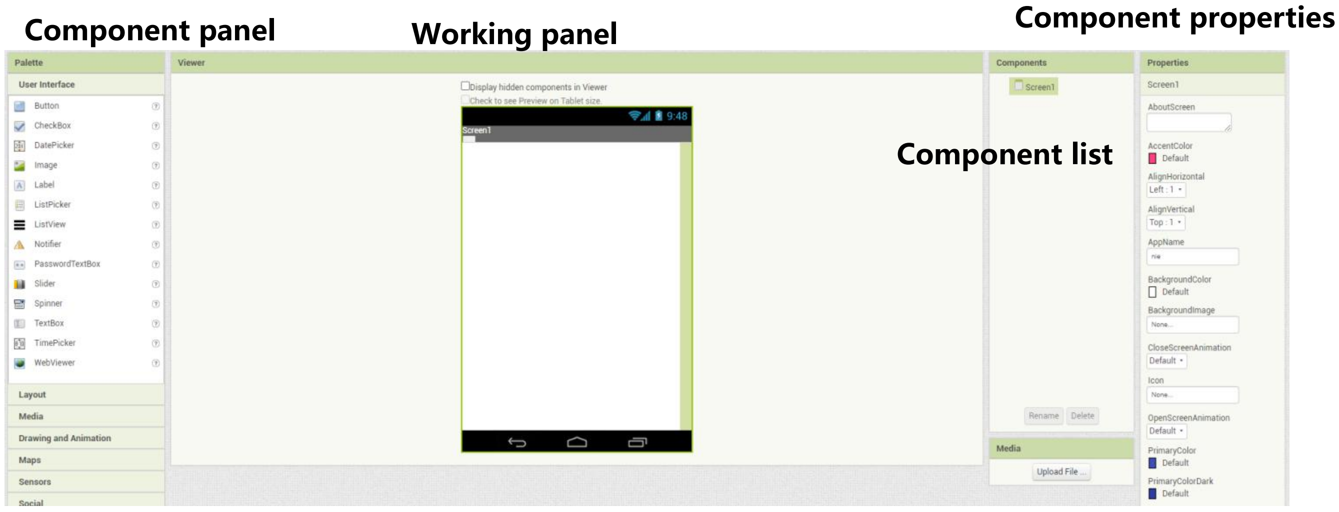
You do not have any projects in App Inventor 2. To learn how to use App Inventor, click the "Guide" link at the top of the window; or to start your first project, click the "Start New Project" button at the upper left of the window.

Happy Inventing!

4) Fill in the project name in the pop-up window and click confirm.

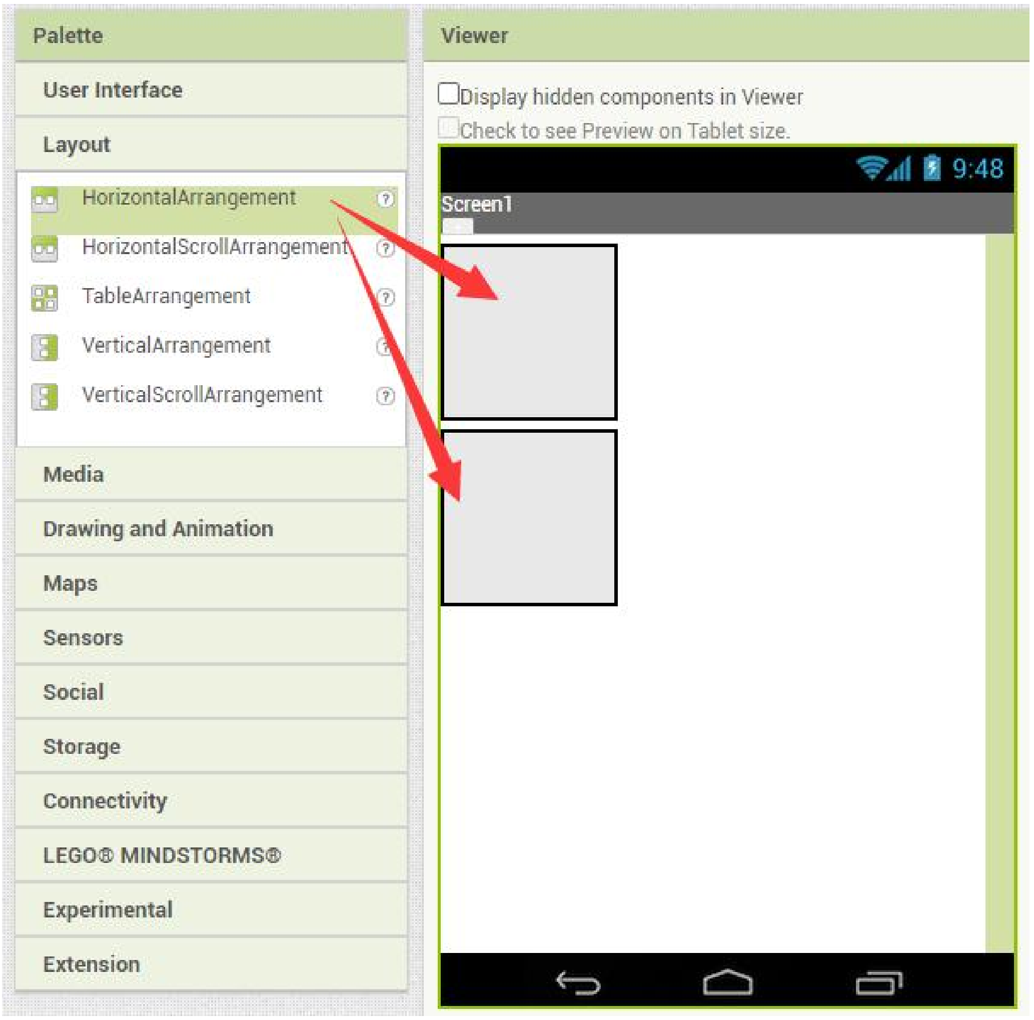

Create new App Inventor project
Project name:
Cancel
OK
Enter the name of the project. The name can only consist of letters and numbers and must begin with a letter.

5) The newly created project interface is shown in the figure.

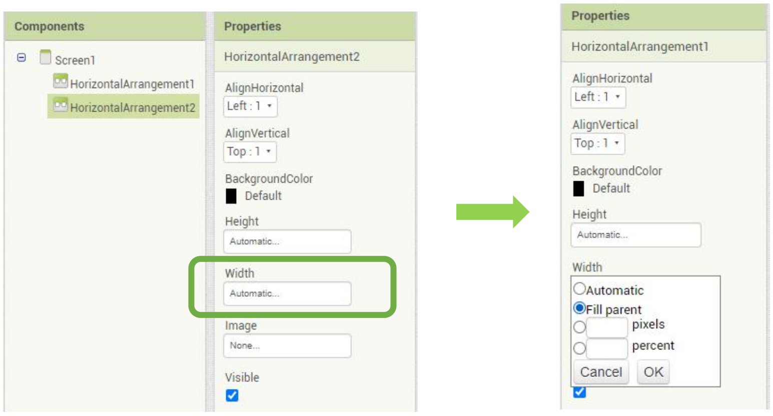


- **Component panel**: contains a lot of functional components, which can be dragged into the working panel when needed. A component is display, interaction, or actions on the interface. It also plays an important role in the subsequent logic design.
- **Working panel** : simulates app display interface and adds components. You can arrange different components in different layout and decorations according to your needs.
- **Component list** : shows the **components** we have placed in the working panel. By clicking it you can edit, rename, and delete the component in the corresponding component properties.
- **Component properties** : the properties of the component, such as color, shape, text, image, function, etc. You can edit the corresponding component properties for the components placed in the work panel. The editable parts of different components may vary.

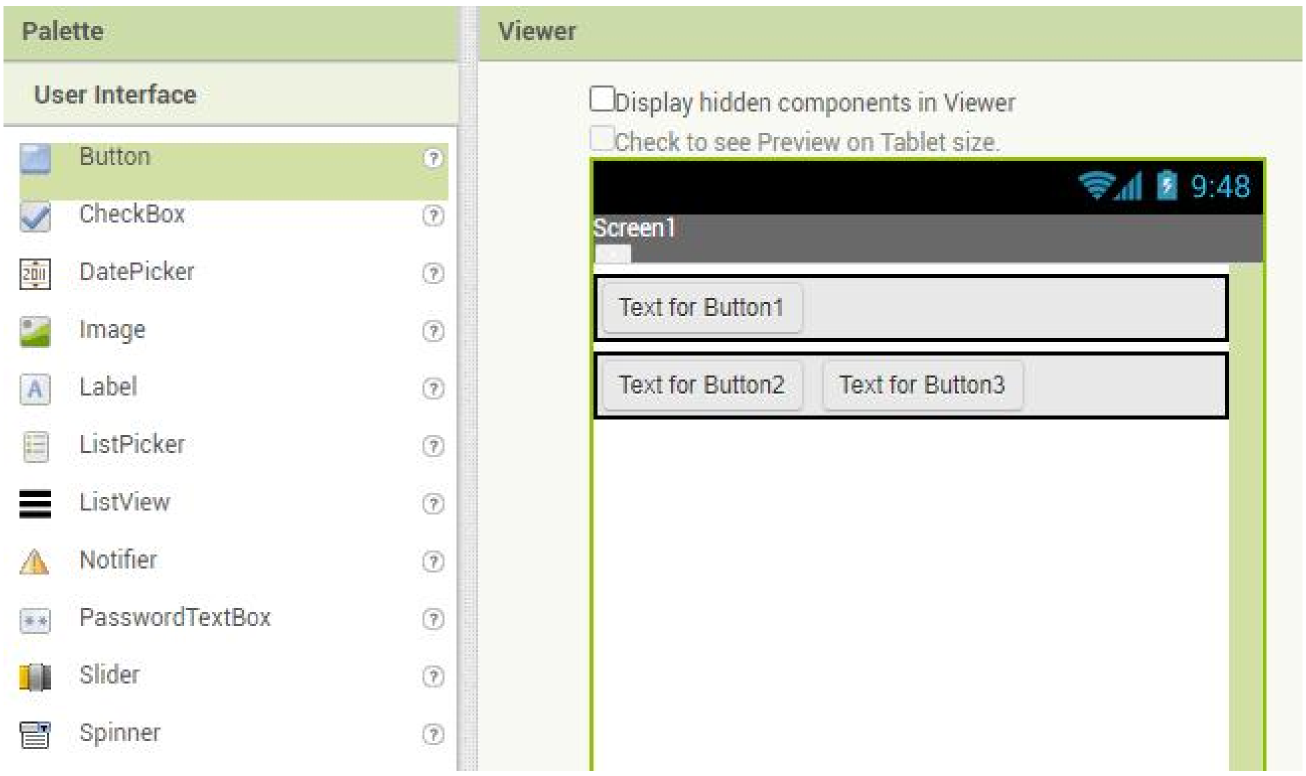
6) Find the interface layout component. The layout component can divide display function zones. Drag the two **Horizontal Layout** components into the working panel one after the other.



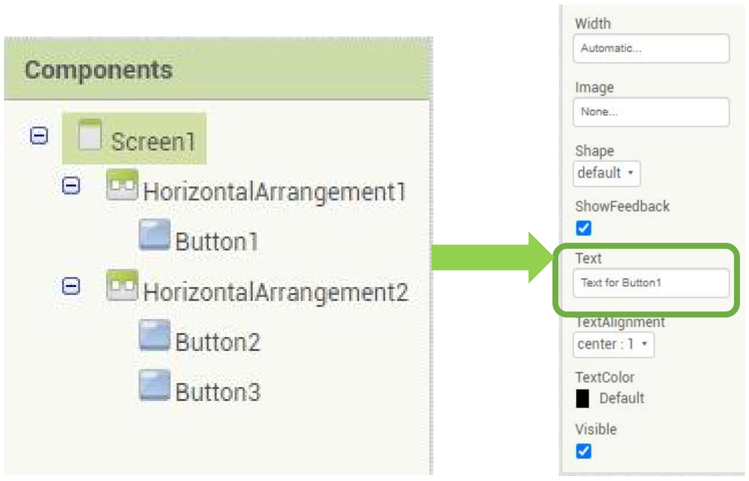
7) Change the width properties of both **Horizontal Layout** components to **Fill**.



8) Drag three " **button** " components to the work panel.

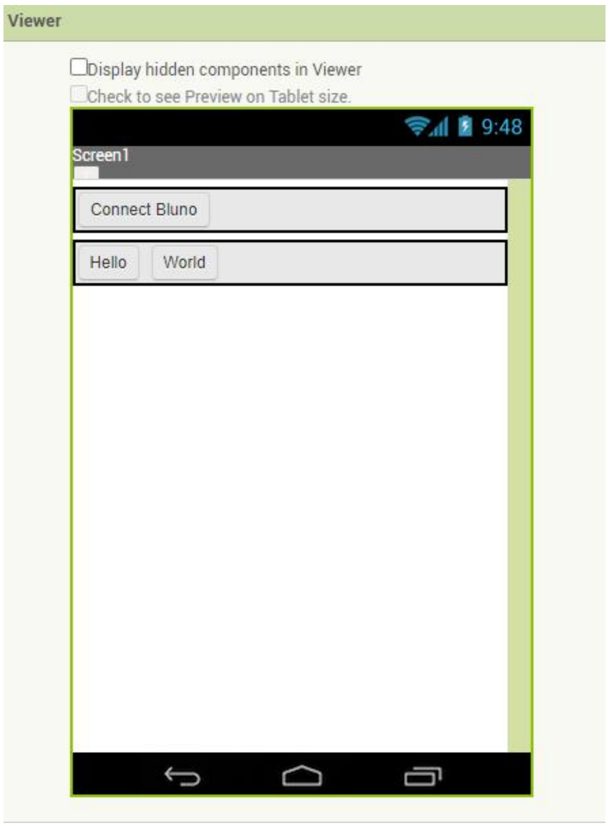


9) Click " **Button 1** " in the component list. In the component properties, you can modify the text content displayed on the button in the interface (at the bottom of the page).

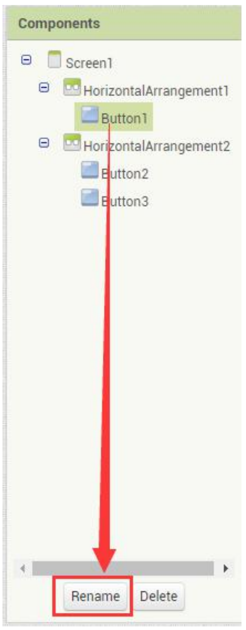


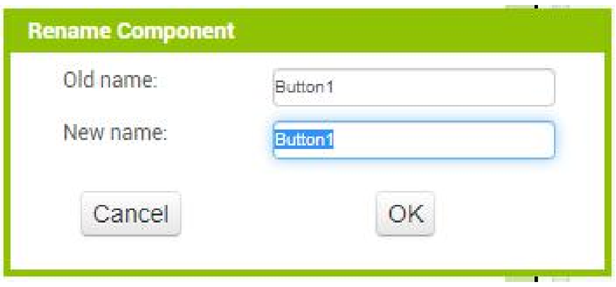
Width
Automatic...
Image
None...
Shape
default
ShowFeedback ☑
Text
Text for Button1
TextAlignment
center : 1
TextColor
■ Default
Visible ☑

10) **Change** the display text of " **Button 1/2/3** " to "Connect Bluno", "Hello", and "World", as shown in the figure.

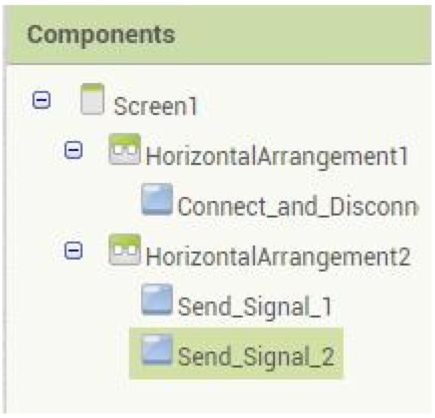

Viewer
Display hidden components in Viewer
Check to see Preview on Tablet size.
9:48
Screen1
Connect Bluno
Hello
World

11) Rename " **Button 1/2/3** " in the component list. Select " **Button 1** " and click " **Rename** ". In the pop-up window, type in "Connect and Disconnect Bluetooth" and click Confirm.
(The reason we rename buttons here is to make it clear what each button does when we design logic later. If we leave them as Button 1/2/3, once the project becomes complicated and more components are added, it would be difficult to keep track of functions of the buttons.)

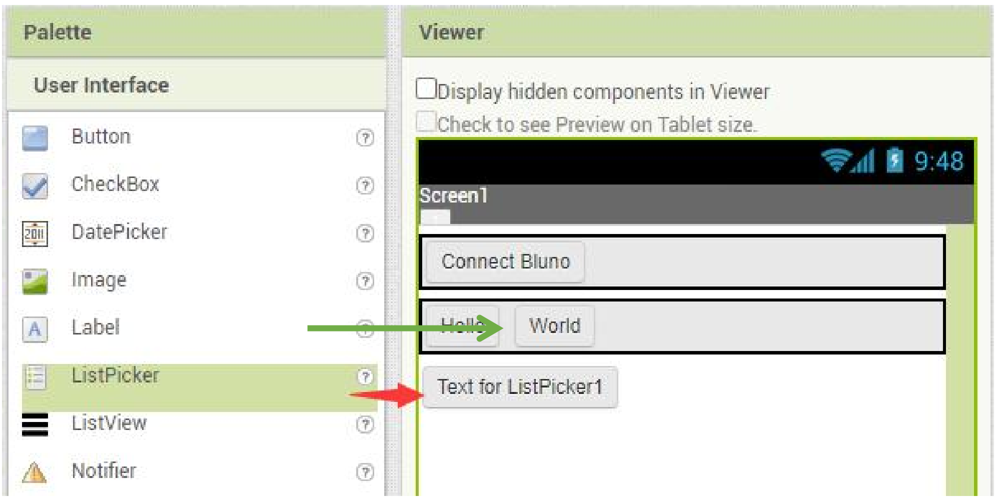




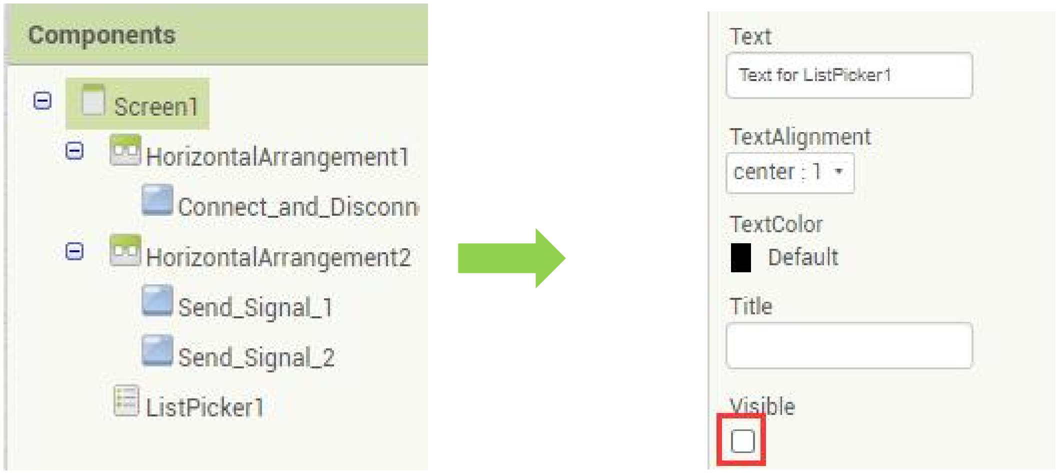
12) **Change** the names of **Button 2** and **Button 3** to "Send Signal 1" and "Send Signal 2".



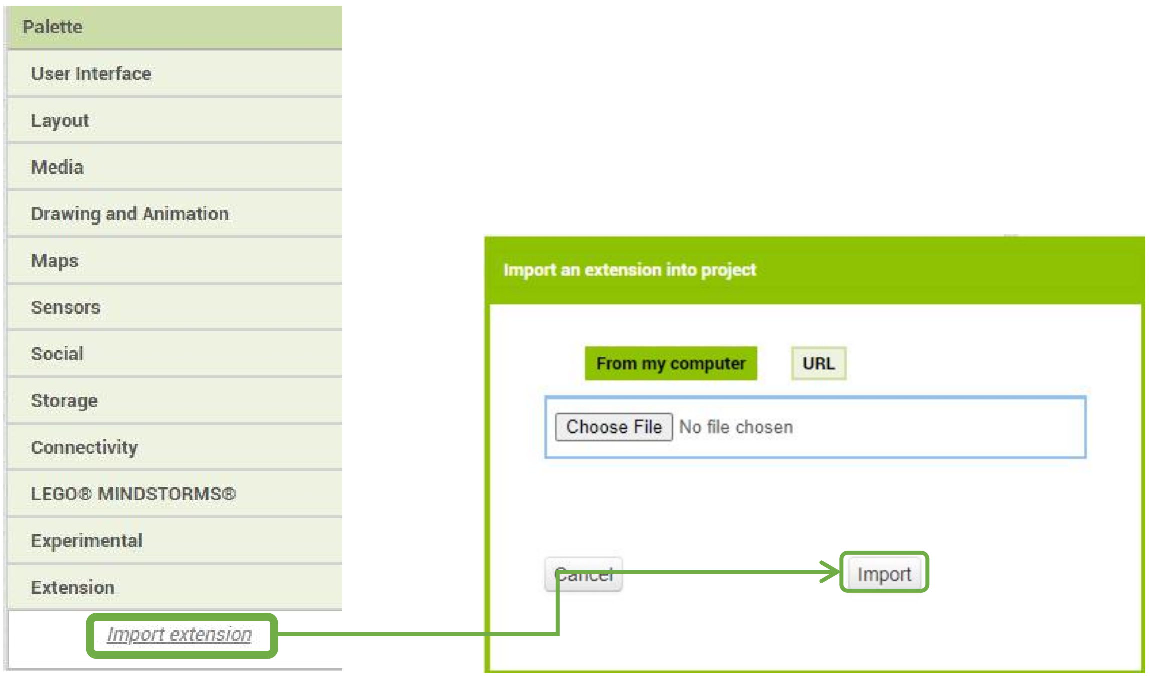
13) Add a **"list selection box"** , by dragging it into the working panel that lists Bluetooth devices available for connection currently during a search.



14) Modify the properties of the list selection box and uncheck the "Visibility" of the list selection box (at the bottom of the properties).



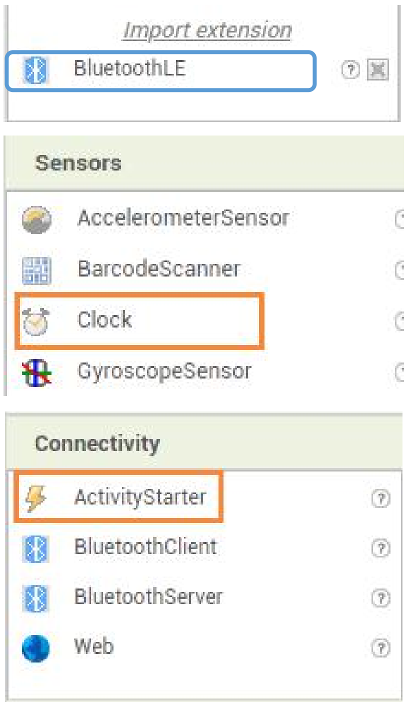
15) Add the BLE Bluetooth plug-in (please search for " Getting to Know Bluetooth -- Bluetooth Configurations " in DFRobot Community www.dfrobot.com, and download edu.mit.appinventor.ble.BluetoothLE.aix). Click Import to confirm loading BLE Bluetooth Plugin.

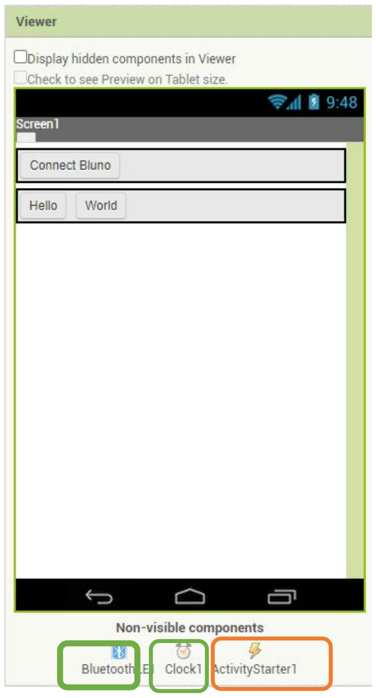


Select Add New Component

Upload the specified aix format file

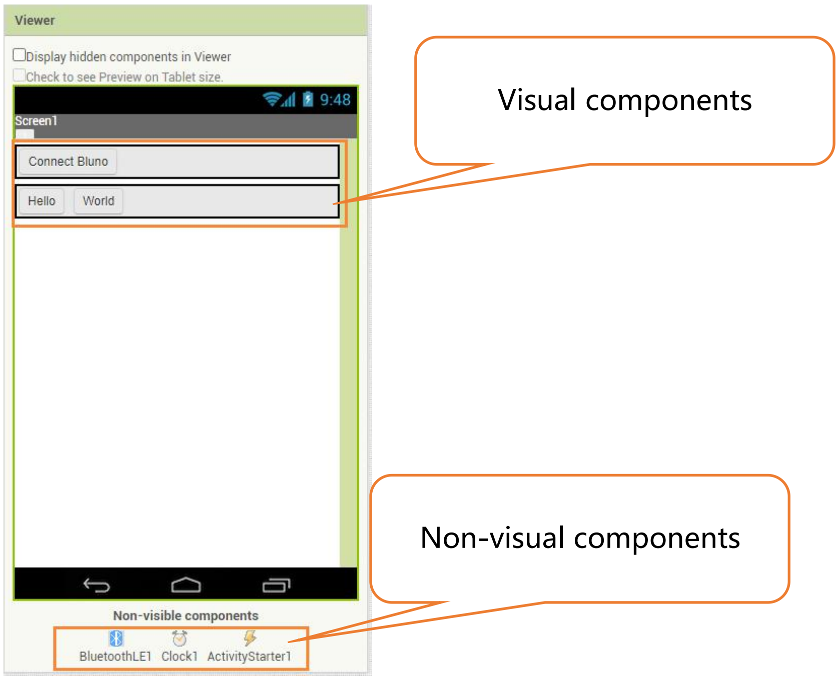
16) Drag the **BLE Bluetooth component (BluetoothLE)**, **timer component**, **Activity launcher** to the working panel.



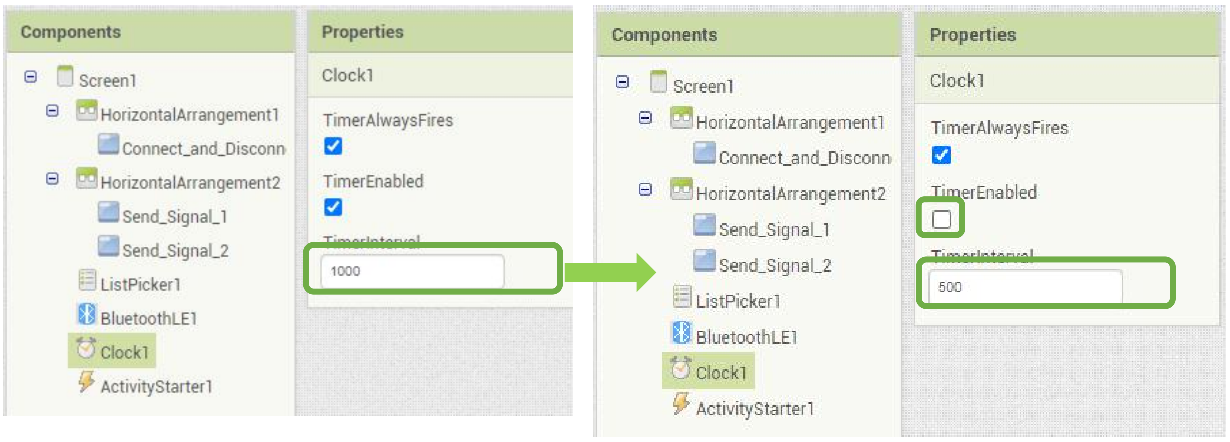


We can now see that after adding the module, " **non-visual components**" appears on the interface. And three modules we added this time are all under the non-visual components. Let's have a look at what the "visual components" and "non-visual components" are.

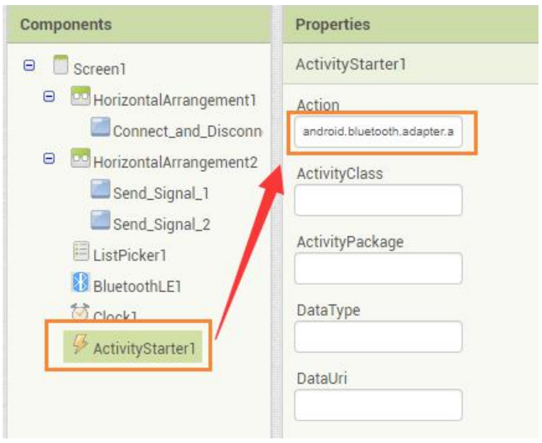
Visual components are the components that we can see after the application is launched, such as buttons, text, and labels. These components together make the user interface of our application. Non-visual components are not visible after the application is launched, so they are not part of the user interface. They are usually used to access the built-in functions of the device. For example: Bluetooth components are used to connect to Bluetooth to send and receive data; timers are used to record specified durations, and the proximity sensor is used to determine the position of the device, etc. Non-visual components provide functional building blocks for app' s logic design.



17) Change the timer interval of " **Timer 1** " to "500ms" and uncheck the "Enable timing" option.

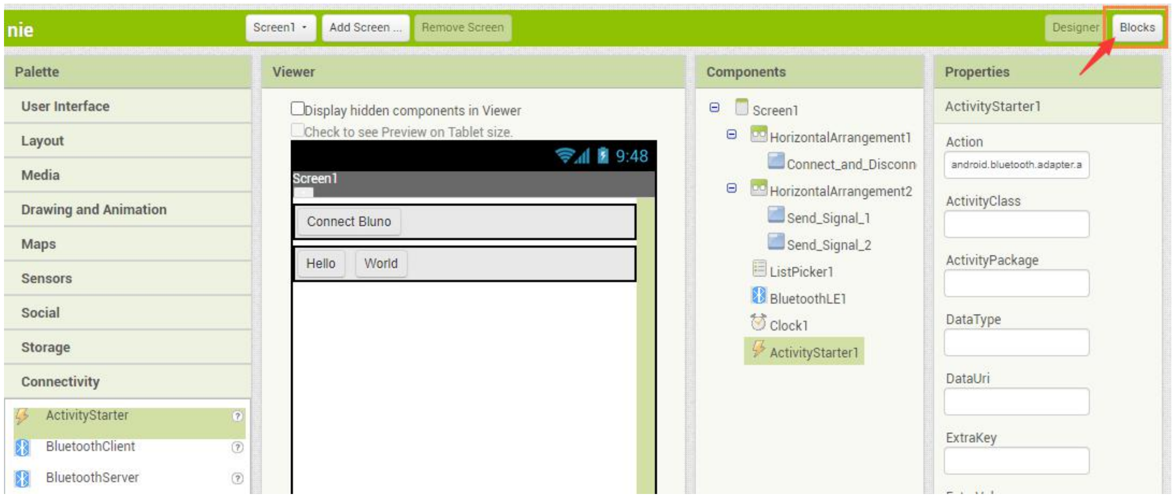


18) Set the component properties of the **Activity launcher** and add: "android.bluetooth.adapter.action.REQUEST_ENABLE" in the Action to turn on the phone's Bluetooth

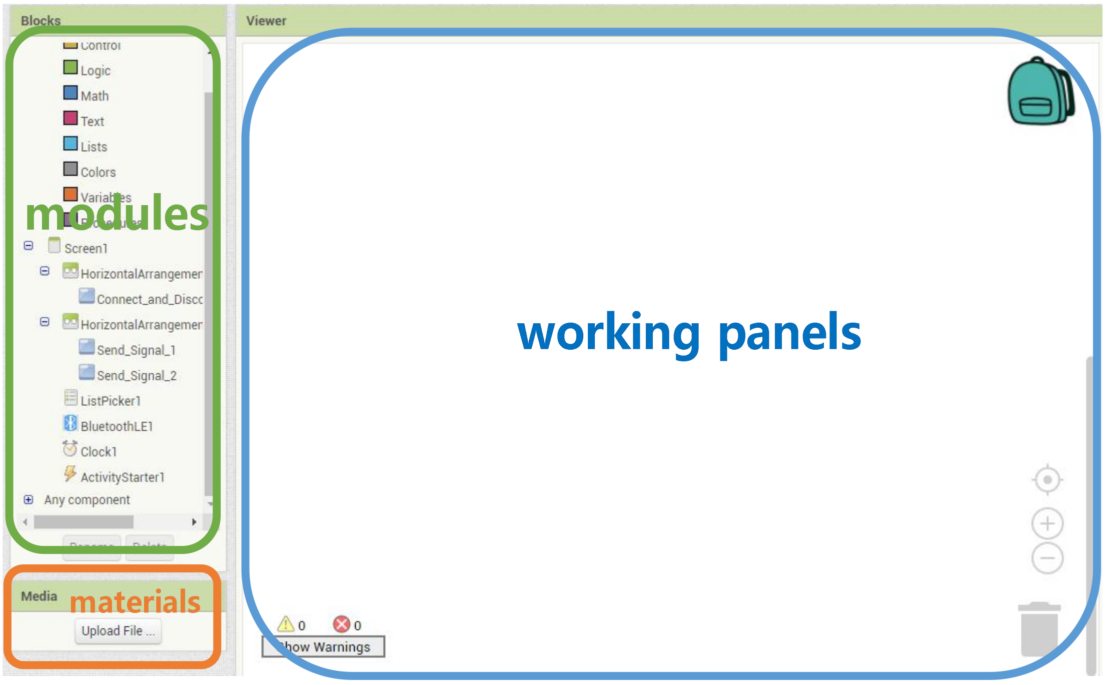


2. Logic design

Upon finishing with interface of the app, we need to design modules and internal logics of the app. Click the **logic design** in the upper right corner of the interface to switch the interface.



1) Switch to the logical design interface. The logical design interface consists of three parts: **modules**, **working panels,** and **materials.**



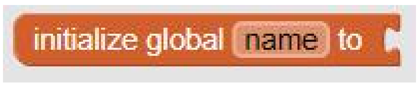
Modules: The modules contain built-in blocks (general modules, similar to the module when no extension is added in Mind+) and the components we added during component design. You can edit the specific function on which you would like to control.

Working panel: you can drag and drop functional modules from the modules list into the working panel for combination and editing, etc.

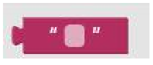
Materials: add image materials can be used according to the content of the design. (This feature is not used in this section)

Click on the module and we will find that App Inventor' s modules are similar to the ones in Mind+. They are also used in the similar way. However, the biggest difference is that: in Mind+, we usually have only one program at the same time. It goes linear and from top to bottom. However, in App Inventor, there will be many groups programs. There will be a separate group for defining different functions and commands.

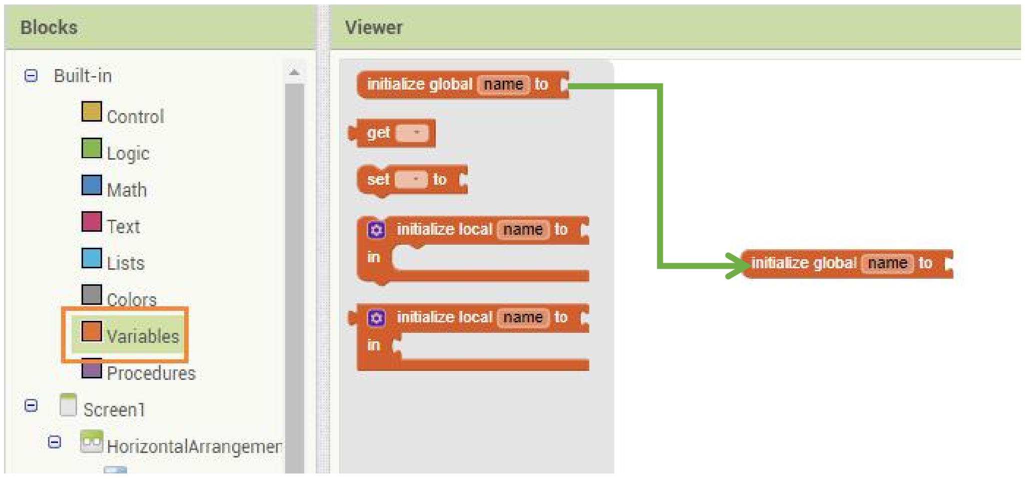
2) Since the communication with the BLE Bluetooth module requires an address, two global text variables need to be initialized:

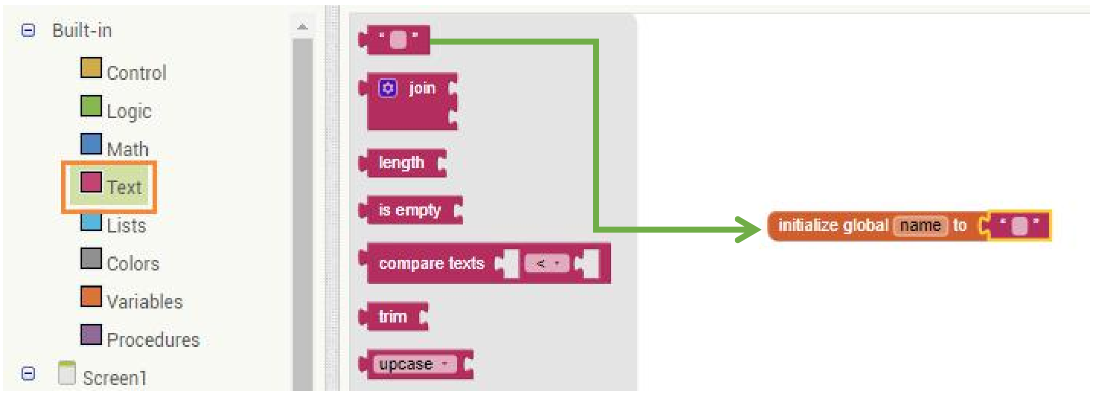
Find 

in the **variables** module and drag it to the working panel;

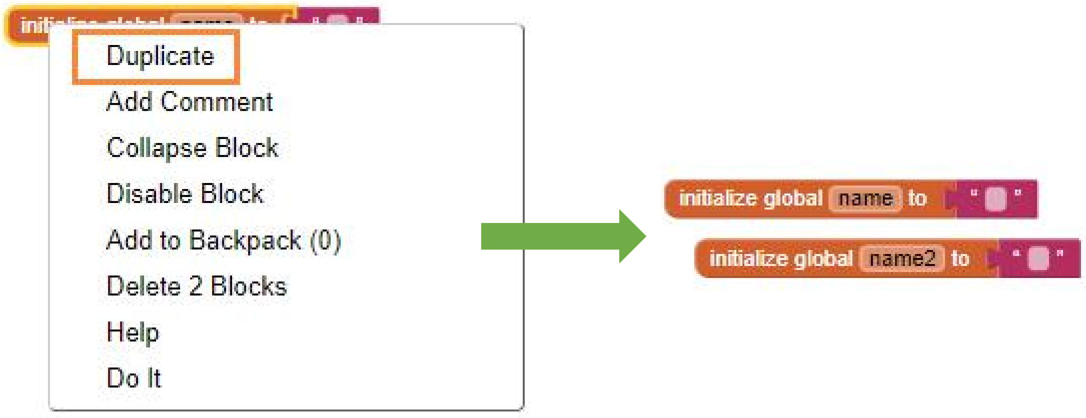
Find 

in the **texts** module and drag it to working panel, before combining them.

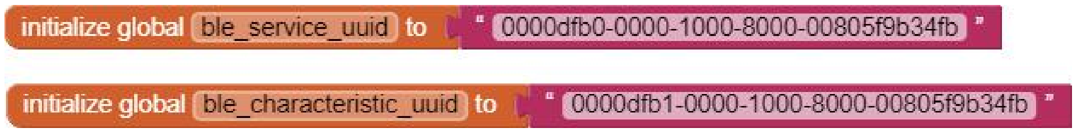




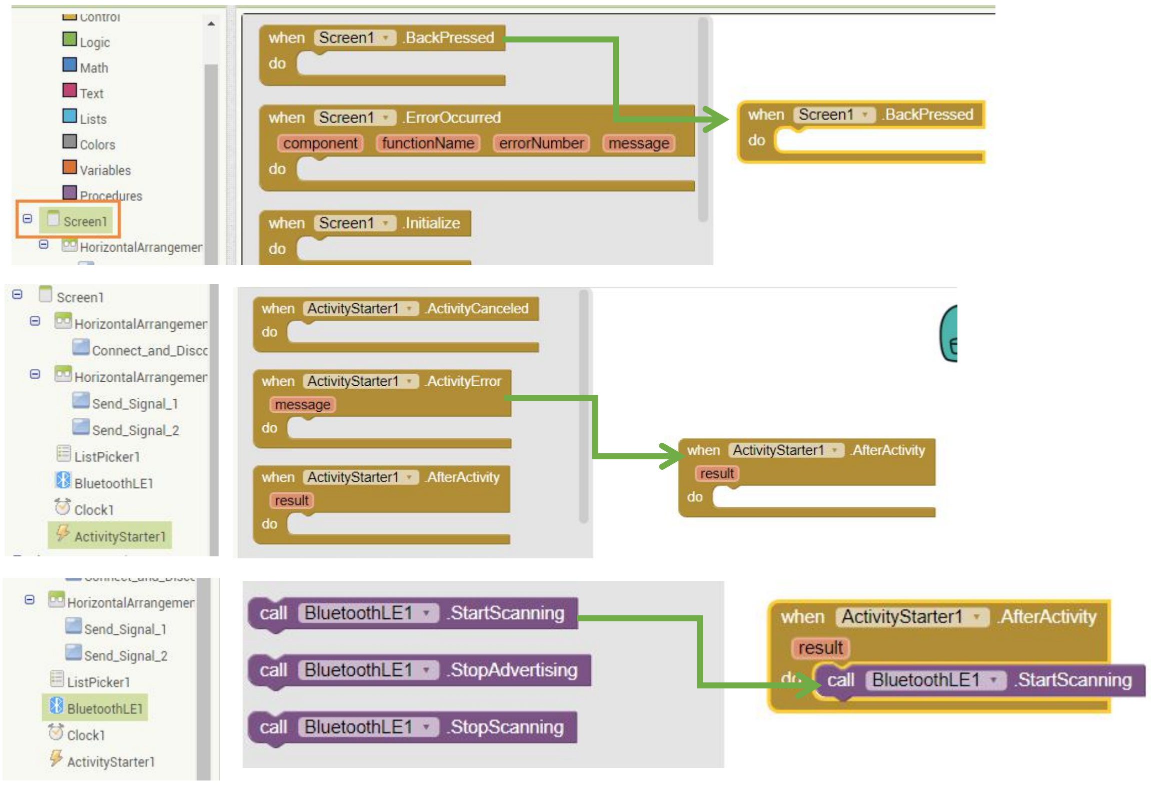
Right-click -- copy the code block to have one extra copy of the above combined modules.



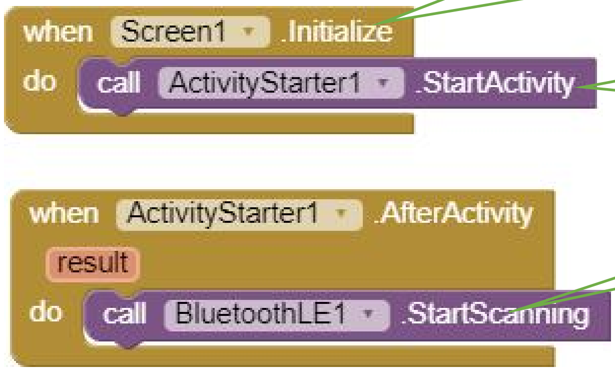
3) Next, define the variable, and change the "variable name" to "ble_service_uuid " as the address of the Bluetooth service, which is defined by the BLE Bluetooth module firmware and cannot be modified. Lock the text to " 0000dfb0-0000-1000-8000-00805f9b34fb "
Change "Variable name 2" to "ble_characteristic_uuid" as the device communication address, which is defined by the BLE Bluetooth module firmware and cannot be modified. Lock the text to "0000dfb1-0000-1000-8000-00805f9b34fb "


initialize global ble_service_uuid to " 0000dfb0-0000-1000-8000-00805f9b34fb "
initialize global ble_characteristic_uuid to " 0000dfb1-0000-1000-8000-00805f9b34fb "

4) Next, at the beginning of the program, use the launcher to turn on Bluetooth to obtain the permission to scan for surrounding Bluetooth devices.


Control
Logic
Math
Text
Lists
Colors
Variables
Procedures
Screen1
HorizontalArrangemen
when Screen1 .BackPressed do
when Screen1 .ErrorOccurred component functionName errorNumber message do
when Screen1 .Initialize do
Connect_and_Disc
Send_Signal_1
Send_Signal_2
ListPicker1
BluetoothLE1
Clock1
ActivityStarter1
when ActivityStarter1 .ActivityCanceled do
when ActivityStarter1 .ActivityError message do
when ActivityStarter1 .AfterActivity result do
call BluetoothLE1 .StartScanning
call BluetoothLE1 .StopAdvertising
call BluetoothLE1 .StopScanning
when ActivityStarter1 .AfterActivity result do call BluetoothLE1 .StartScanning

Complete program example:



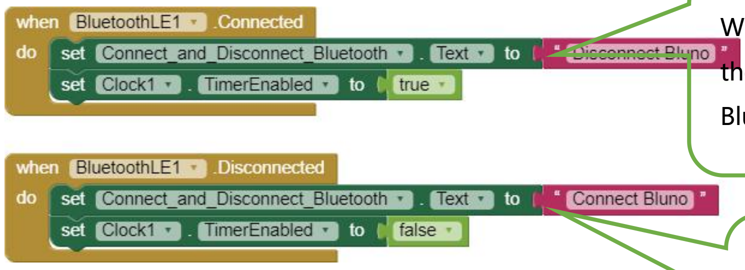
```
when Screen1 .Initialize
do  call ActivityStarter1 .StartActivity

when ActivityStarter1 .AfterActivity
  result
do  call BluetoothLE1 .StartScanning
```

The specified function module can be found under the component with corresponding name. For example, Screen1 can be found in Screen1 on modules panel on the left side.

Activate launcher when Screen1 (the display) is initialized

Launcher turns on Bluetooth to search for surrounding Bluetooth devices.

5) Check whether Bluetooth is connected. Define "connect/disconnect Bluetooth" button' s status change. Why start the timer? Checking for Bluetooth data every 500ms will prevent the program check too frequently and cause phone to be not responding.

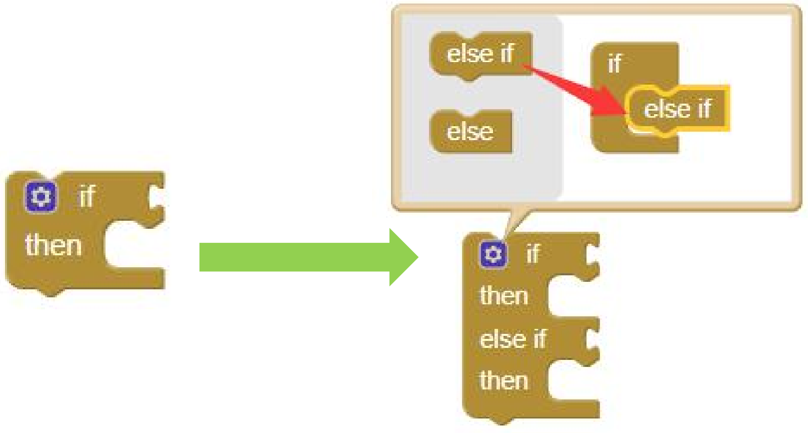


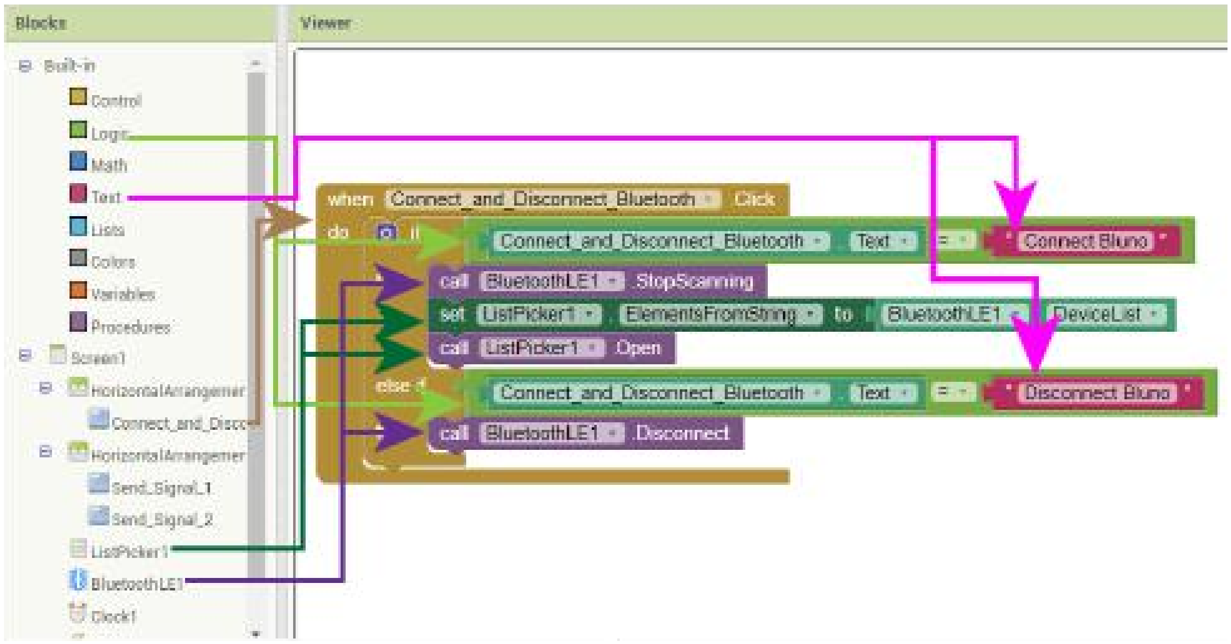
```
when BluetoothLE1 .Connected
do  set Connect_and_Disconnect_Bluetooth . Text to " Disconnect Bluno "
    set Clock1 . TimerEnabled to true

when BluetoothLE1 .Disconnected
do  set Connect_and_Disconnect_Bluetooth . Text to " Connect Bluno "
    set Clock1 . TimerEnabled to false
```

When the Bluetooth connection is successful, the text on the button reads "Disconnect Bluno", and then starts the timer.

When Bluetooth is disconnected, the text on the button reads "Connect Bluno" and then the timer stops.

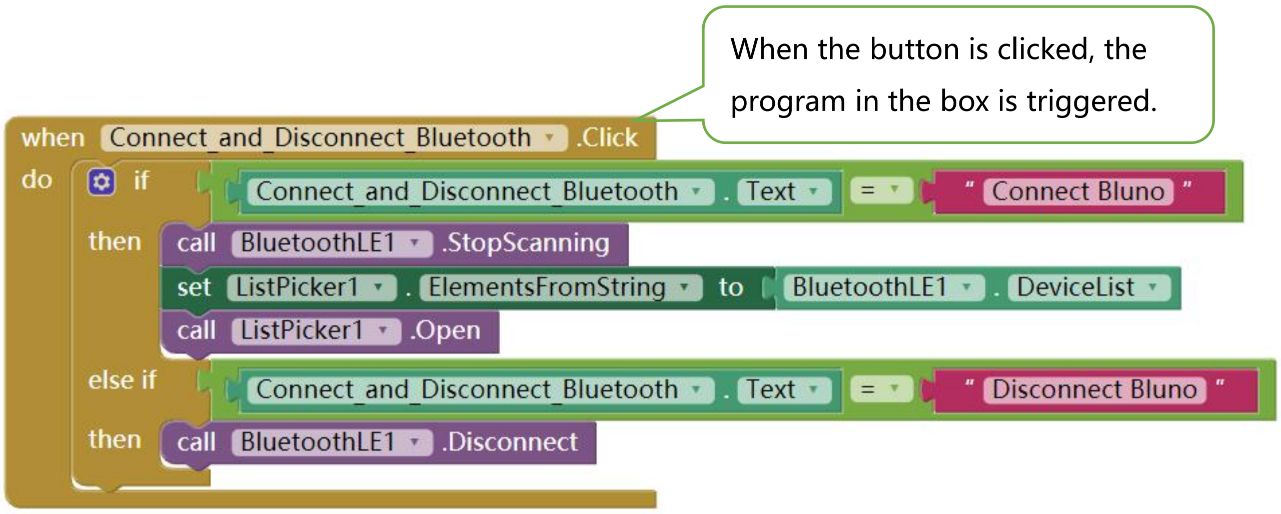
6) After Bluetooth is turned on, you need to open the device list for the user' s selection. First, find **if-then** module in **control** options. Add **else, if** in **if-then** module.
Click on the gear icon on top left of **if-then** module, and drag **else, if** into **if** box on the right side.





```
when Connect_and_Disconnect_Bluetooth .Click
do if Connect_and_Disconnect_Bluetooth . Text = " Connect Bluno "
   call BluetoothLE1 .StopScanning
   set ListPicker1 . ElementsFromString to BluetoothLE1 . DeviceList
   call ListPicker1 .Open
   else if Connect_and_Disconnect_Bluetooth . Text = " Disconnect Bluno "
   call BluetoothLE1 .Disconnect
```

Complete program example:


When the button is clicked, the program in the box is triggered.

```
when Connect_and_Disconnect_Bluetooth .Click
do  if    Connect_and_Disconnect_Bluetooth . Text = " Connect Bluno "
    then  call BluetoothLE1 .StopScanning
          set ListPicker1 . ElementsFromString to BluetoothLE1 . DeviceList
          call ListPicker1 .Open
    else if Connect_and_Disconnect_Bluetooth . Text = " Disconnect Bluno "
    then  call BluetoothLE1 .Disconnect
```

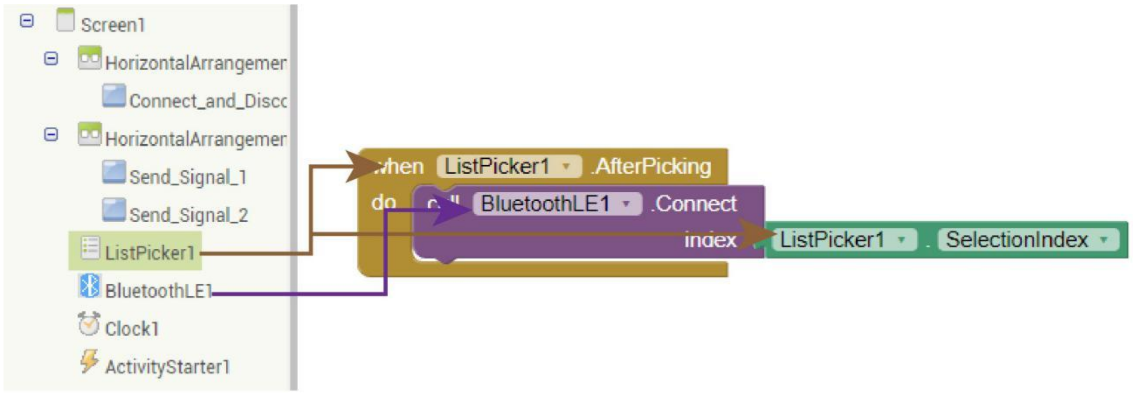
When the button is clicked, check text inside the button. If the text in the button is "Connect Bluno", execute:

1. Stop searching for Bluetooth devices (In the previous procedure, the search for Bluetooth devices started after the screen was initialized. Therefor, in this step, stop searching for Bluetooth first, so that the list can show the Bluetooth devices that were found when the button was pressed.);
2. Match the contents of the list with the list of devices searched by Bluetooth;
3. Open the list and show contents. (At this time, you can select the Bluetooth device to be connected. The selected function is described in the next step.)

If the text in the button is "Disconnect Bluno", execute:

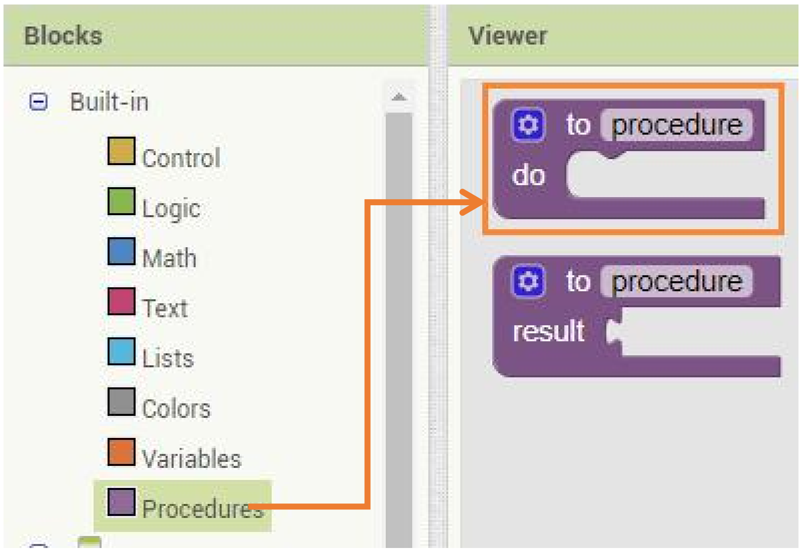
Disconnect the connected Bluetooth. (If the text shows "Disconnect Bluno", it means that the mobile phone and Bluetooth are connected at this time. The text conversion process will be reflected in the next few steps.)

7) When the user clicks on the list and selects a device, we should connect the device. Connect the selected Bluetooth device, which is the selected option.

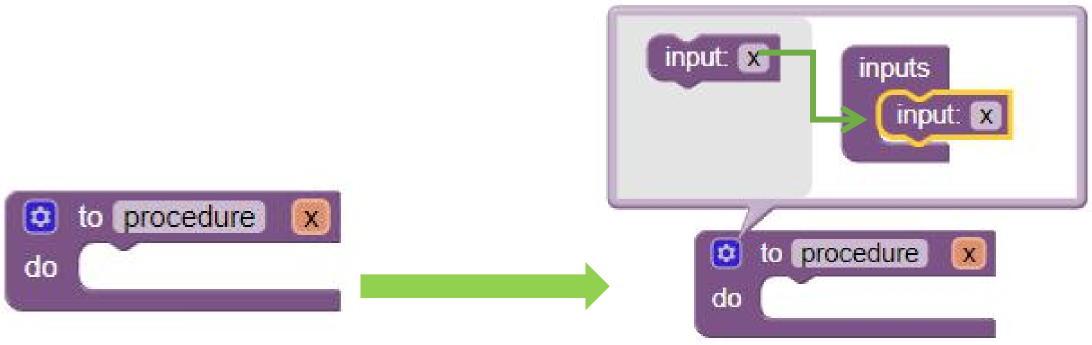


8) After establishing the Bluetooth connection, we need to start the data transmission between phone and BLE Bluetooth module. In this section, we are mainly sending data to BLE Bluetooth module via a phone. All the data sent can be checked at serial port data in Mind+.

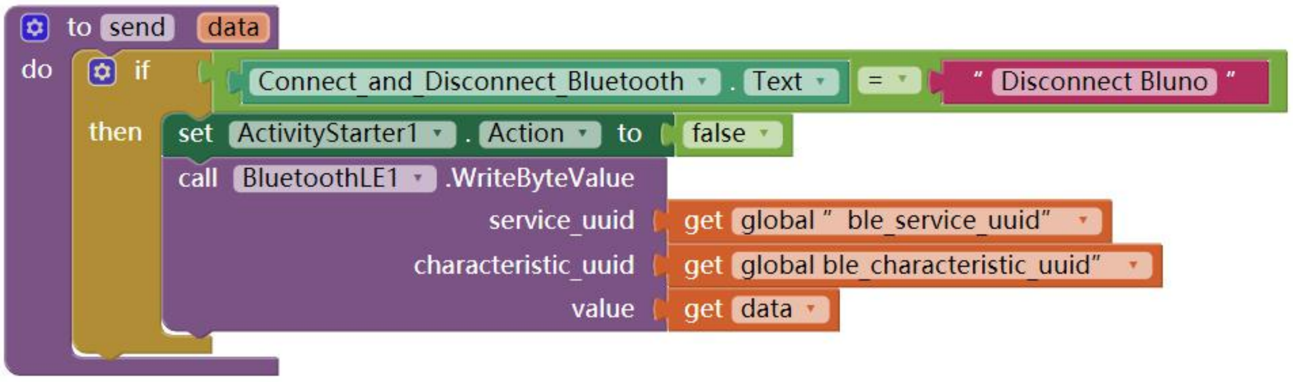
Set up a **definition process** to send data.



Set the process name as "**send**", click the gear button in the upper left corner of the "**define process**" block, add "**input: X**" and change the input to "**input: data** "

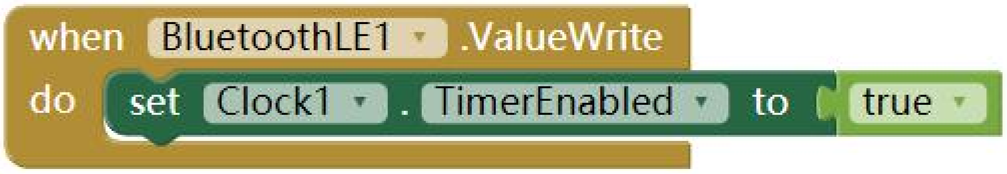


When a connected Bluetooth is found, stop the timer first then send the data.



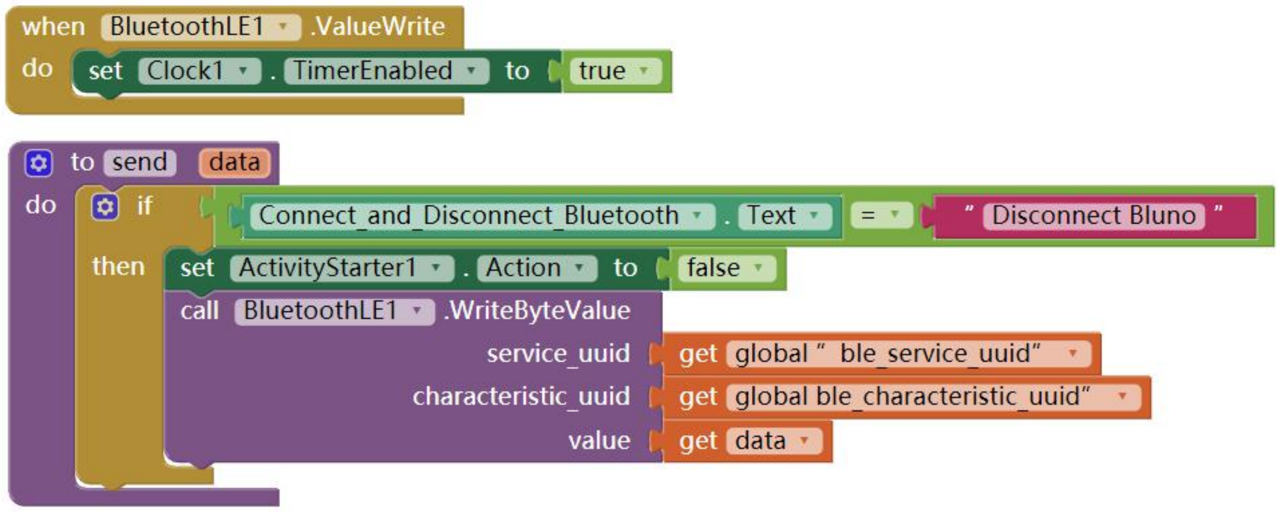
```
to send data
do  if Connect_and_Disconnect_Bluetooth . Text = " Disconnect Bluno "
    then set ActivityStarter1 . Action to false
         call BluetoothLE1 .WriteByteValue
                    service_uuid  get global " ble_service_uuid"
             characteristic_uuid  get global ble_characteristic_uuid"
                           value  get data
```

After the data is sent, the timer is re-enabled.



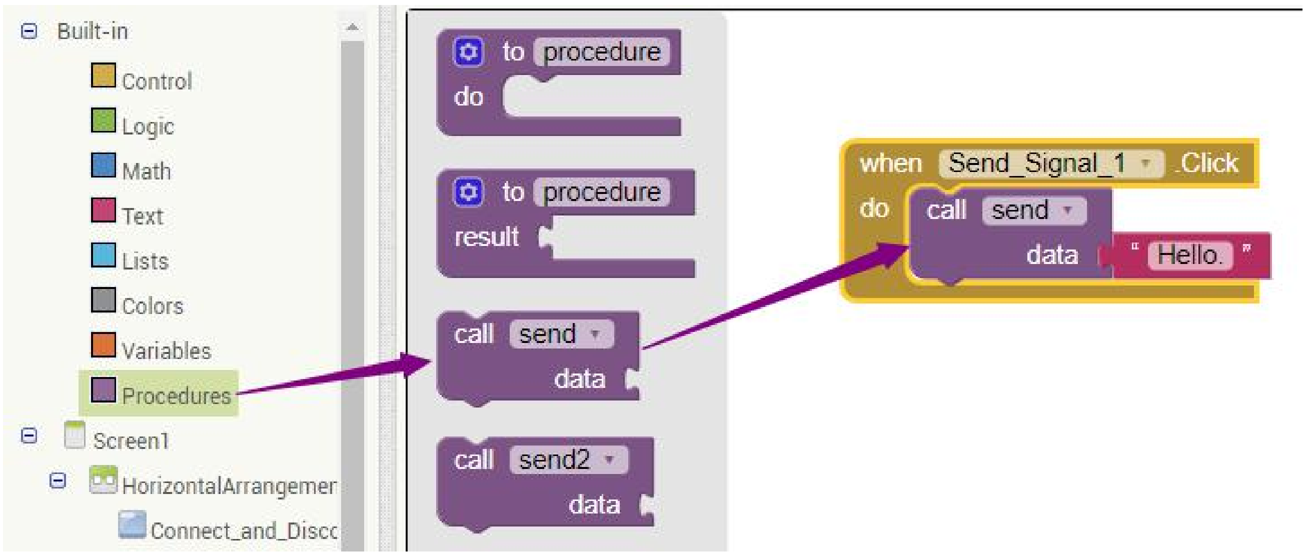
```
when BluetoothLE1 .ValueWrite
do  set Clock1 . TimerEnabled to true
```

Complete program example:

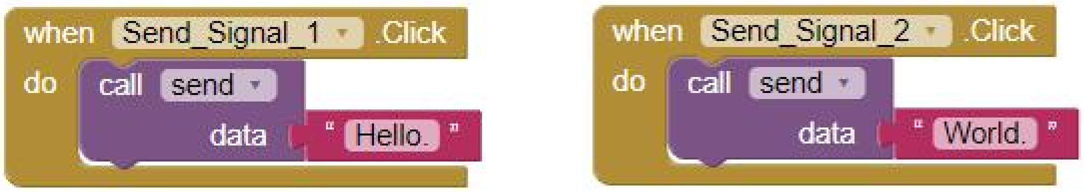


```
when BluetoothLE1 .ValueWrite
do  set Clock1 . TimerEnabled to true

to send data
do  if Connect_and_Disconnect_Bluetooth . Text = " Disconnect Bluno "
    then set ActivityStarter1 . Action to false
         call BluetoothLE1 .WriteByteValue
                    service_uuid  get global " ble_service_uuid"
             characteristic_uuid  get global ble_characteristic_uuid"
                           value  get data
```

9) When the "**Send Signal 1**" button is clicked, send "Hello." And when the "**Send Signal 2**" button is clicked, Send "World. ". A comma is used to mark the end of the sent data for the BLE Bluetooth module to parse the data. (The content of the data sent can be modified as needed.)

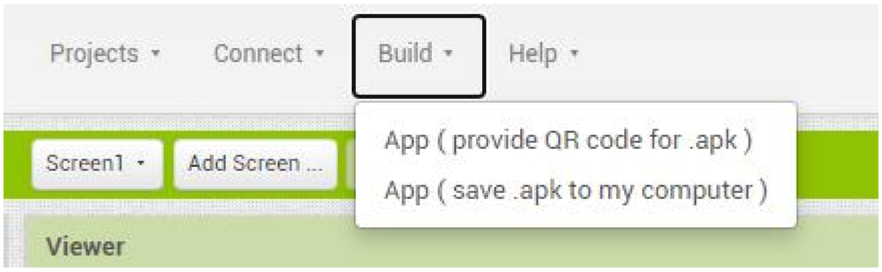


Complete program example:

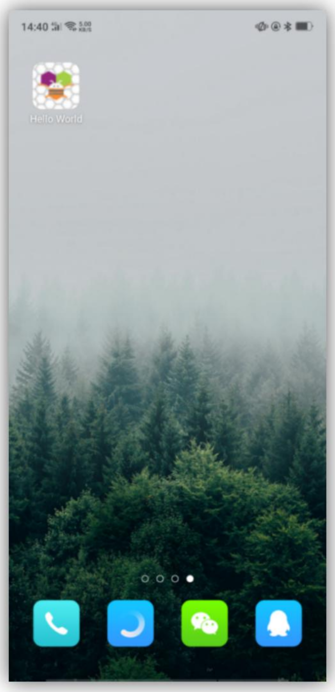


10) After the design of the program is complete, we can export the .apk format file and install it on an Android phone.

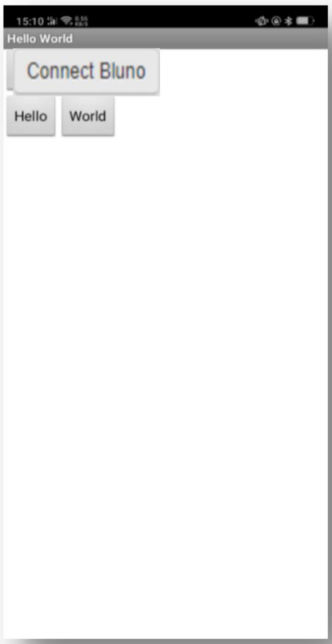
At the top of the page, there is a "**package apk**" option, in which there are two methods: you can scan the QR code by phone to download directly to the phone for installation; or you can download the apk file to the computer, and install it by sending the app from computer to the phone via a cable.


Projects
Connect
Build
Help
Screen1
Add Screen ...
App (provide QR code for .apk)
App (save .apk to my computer)
Viewer

11) After the installation is complete, we can see the app we designed on the mobile interface.



Phone interface



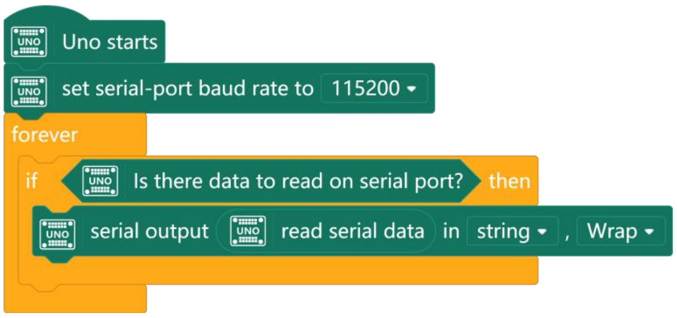
Open the app interface

3. Program operation:

Connect phone to the computer with a data cable and start Mind+.

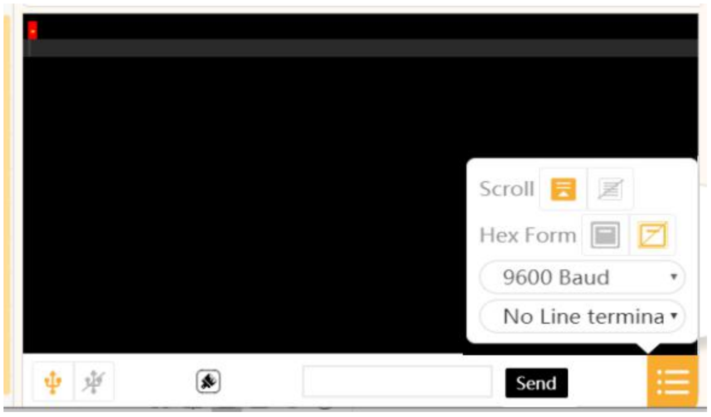
Upload a program that reads the serial port data to the main control board. Pay attention the dial has to be on "PROG" option when uploading. After the upload is complete, return the dial to "RUN".

Example program:

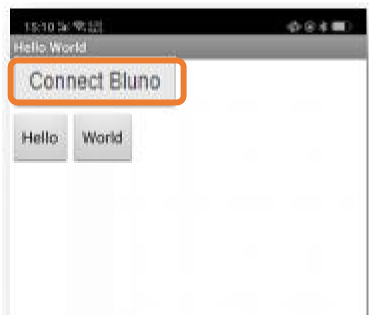


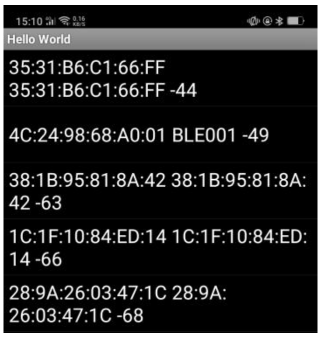
```
Uno starts
set serial-port baud rate to 115200
forever
    if Is there data to read on serial port? then
        serial output read serial data in string , Wrap
```

Open the serial port and set the baudrate to 115200.



Open the mobile app and select "Connect Bluno", find the BLE Bluetooth module (BLE001, the name you set in the previous section, customizable), wait for the button text to display "Disconnect Bluno" (means the connection is successful), click the buttons "Hello" and "World".




Hello World

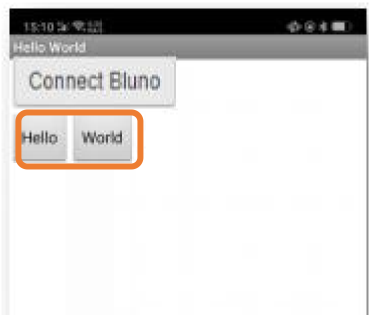
35:31:B6:C1:66:FF
35:31:B6:C1:66:FF -44

4C:24:98:68:A0:01 BLE001 -49

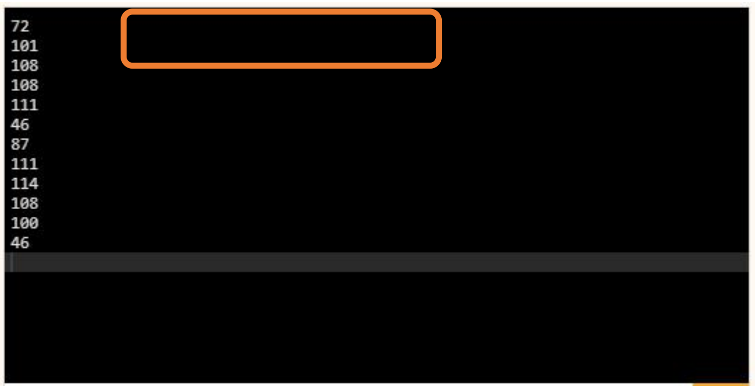
38:1B:95:81:8A:42 38:1B:95:81:8A:
42 -63

1C:1F:10:84:ED:14 1C:1F:10:84:ED:
14 -66

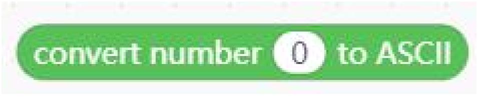
28:9A:26:03:47:1C 28:9A:
26:03:47:1C -68



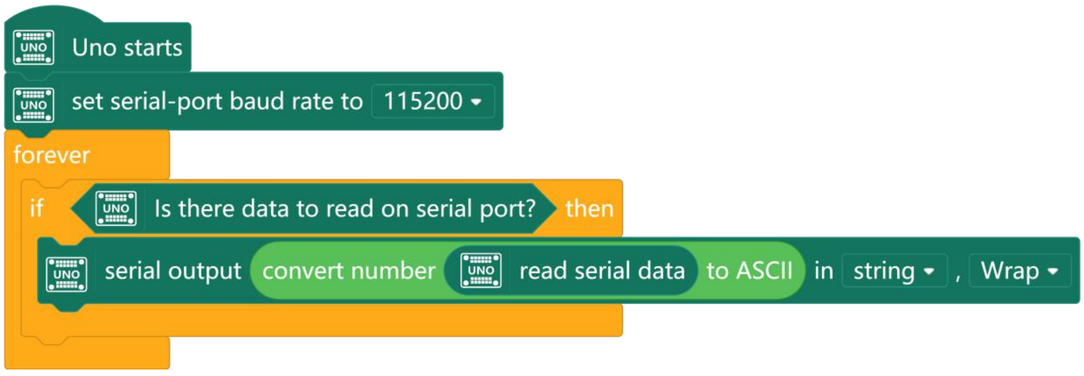
Mind+ serial interface:



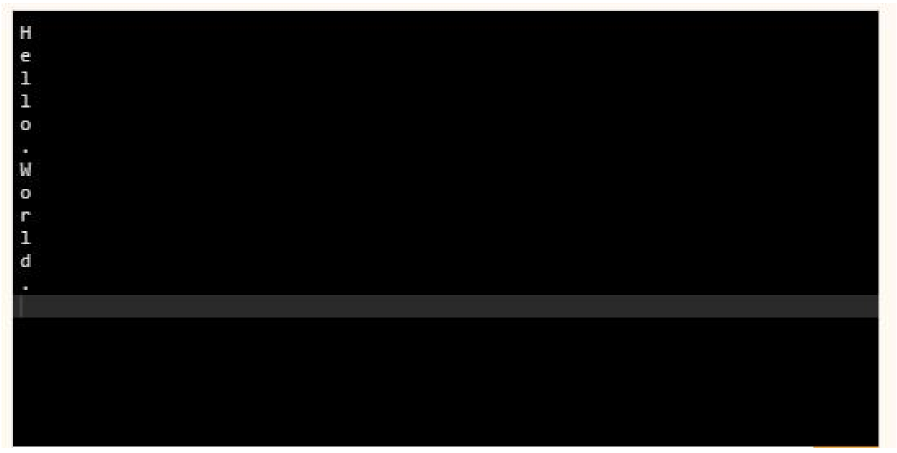
```
72
101
108
108
111
46
87
111
114
108
100
46
```

We found that the content sent from the app is a string of letters, but what is received is a string of digits. Why? We add an operator module  to the program and read the serial port after converting the data.

Example program:


Uno starts
set serial-port baud rate to 115200
forever
if Is there data to read on serial port? then
serial output convert number read serial data to ASCII in string , Wrap

Mind+ serial interface:



```
H
e
l
l
o
.
W
o
r
l
d
.
```

4. **Program’ s effect:**

Click the Hello button, the string "Hello." is received in the serial data in Mind+, and click the World button, and the string "World." is received in the serial data.

Further Reading

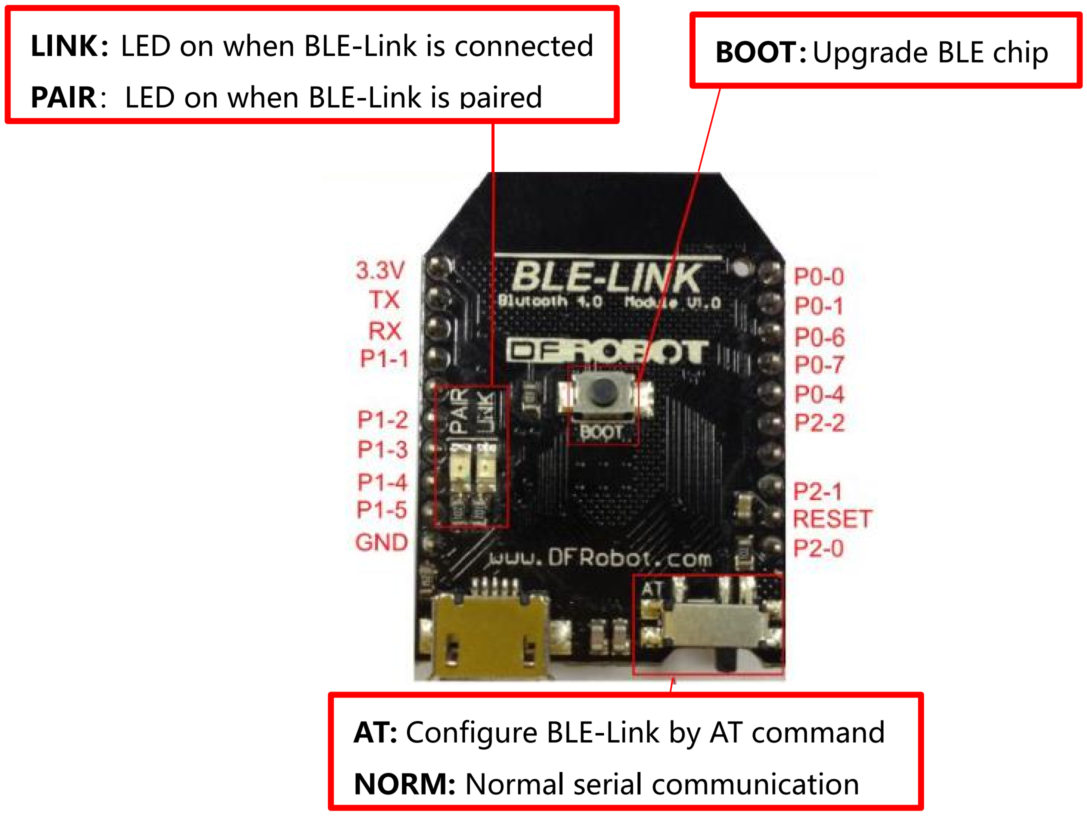
The role of the timer

As we know that the Bluetooth module is not limited to receiving data, but can capable of sending data (detailed in the next section). The timer is mainly used in the program to scan whether Bluetooth data is received. If yes, read the data and process it (we just have to display them in the app). When the timer goes off, check Bluetooth connection status and receives Bluetooth data if connected. We set the timer to 500ms. Therefore, every 500ms there is a check on if any data is received. If the data is too small, it will cause packet loss in the transmitted data, thus a 500ms timer is set.

BLE Bluetooth module pins

The most important modules in Bluetooth usasge are RX and TX, besides 3.3V and GND.
TX means “transport” , and RX means “receive” . It can also be understood as upload (TX) and download (RX). They are the pins used when BLE Bluetooth module receives and sends data. In this example, our BLE Bluetooth module can be directly plugged onto the pins of the extension board, so we didn’ t have to find the pins. If we don’ t have an extension board or are using other kinds of Bluetooth module, we will have to wire by ourselves.
When the Bluetooth module is connected to the main control, we need to connect the TX (transmit) of Bluetooth to the RX (receive) on the main control board, and connect the RX (receive) of Bluetooth to the TX (send) of the main control board.

This is the reason why we don’ t usually use P0 (RX) and P1 (TX) pins on Arduino. They are multiplexed pins, which can be general pins or hard serial port pins and will affect program’ s uploading and downloading, as well as reading of the ports.


LINK: LED on when BLE-Link is connected
PAIR: LED on when BLE-Link is paired
BOOT: Upgrade BLE chip
3.3V
TX
RX
P1-1
P1-2
P1-3
P1-4
P1-5
GND
P0-0
P0-1
P0-6
P0-7
P0-4
P2-2
P2-1
RESET
P2-0
BLE-LINK
Blutooth 4.0 Module V1.0
DFROBOT
PAIR
LINK
BOOT
www.DFRobot.com
AT
AT: Configure BLE-Link by AT command
NORM: Normal serial communication

Chapter 10 Bluetooth Remote Control

Project 22 Bluetooth Controlled LED

In the previous section, we learned how to use a mobile phone to send data to Arduino via Bluetooth. Now, how to use these data to control the LEDs? In this section, we will use Bluetooth to turn on and off of LEDs.

Task Navigation

1. What is the baudrate?
2. What are ASCII characters?

3. Use your phone to light up the LEDs.

Key Points Analysis

1. baudrate

The official definition of the baudrate is the number of code elements transmitted per second. The communication between two modules only goes when the same baudrate is set on both sides. So at the beginning of the program we will set the baudrate. which is 115200 by default for BLE Bluetooth modules.

2. What are ASCII characters?

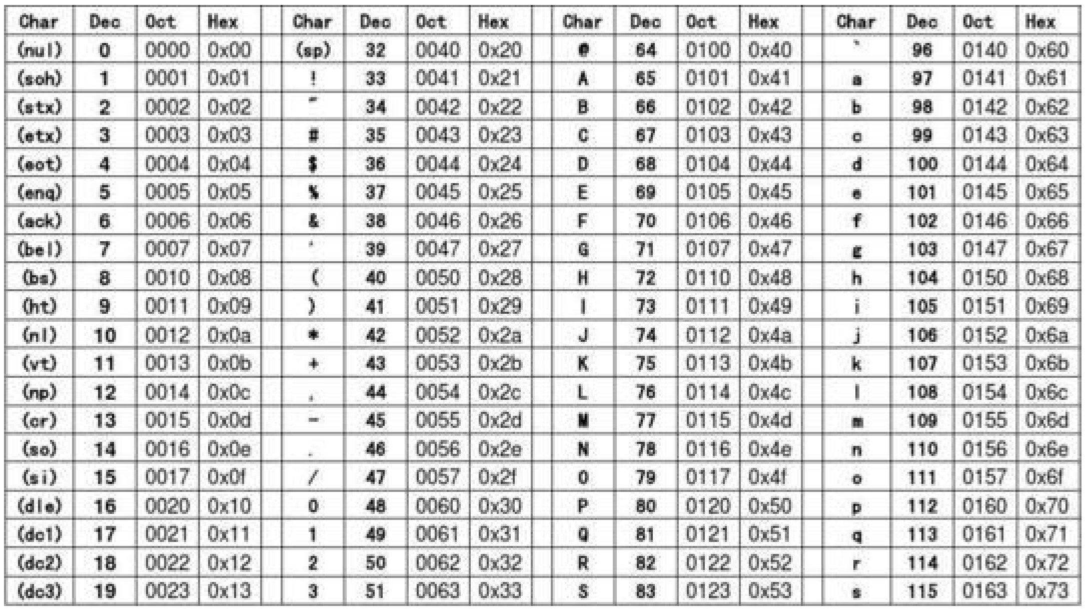
What is ASCII? ASCII (American Standard Code for Information Interchange) is a computer coding system based on the Latin alphabet. It is mainly used to display modern English and other Western European languages.

It is the most common information exchange standard and is equivalent to the international standard ISO / IEC 646. ASCII was first published as a standard in 1967, and was last updated in 1986, with a total of 128 characters defined so far.

Why makes ASCII? In computer, all data must be represented in binary during storage and computing (because computers use high level for 1 and low level for 0). For example, 52 letters such as a, b, c, d... (including uppercase), numbers such as 0, 1... and some commonly used symbols (such as *, #, @, etc.) are also represented in binary numbers when stored in the computer. However, everyone can have their standard on what binary number represents a certain symbol. And that standard is called an encoding. If we want to communicate with each other without confusion, we have to use the same encoding rules. Therefore, the relevant standardization organizations in the United States introduced the ASCII code, which specifies which binary numbers are used for the above-mentioned common symbols.

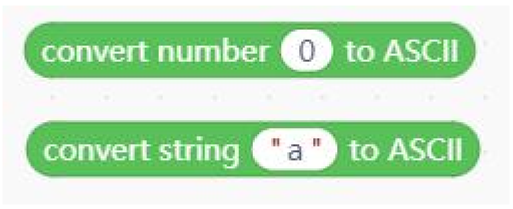
If we look at the data read by Bluetooth, we will see that we sent out letters but received a string of numbers. Why?

Because the original data received by Bluetooth transmission is ASCII values. They are numbers if not converted upon receiving. We can view the data of the corresponding character by looking at the ASCII conversion table.



Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex
(nul)	0	0000	0x00	(sp)	32	0040	0x20	@	64	0100	0x40	`	96	0140	0x60
(soh)	1	0001	0x01	!	33	0041	0x21	A	65	0101	0x41	a	97	0141	0x61
(stx)	2	0002	0x02	"	34	0042	0x22	B	66	0102	0x42	b	98	0142	0x62
(etx)	3	0003	0x03	#	35	0043	0x23	C	67	0103	0x43	c	99	0143	0x63
(eot)	4	0004	0x04	$	36	0044	0x24	D	68	0104	0x44	d	100	0144	0x64
(enq)	5	0005	0x05	%	37	0045	0x25	E	69	0105	0x45	e	101	0145	0x65
(ack)	6	0006	0x06	&	38	0046	0x26	F	70	0106	0x46	f	102	0146	0x66
(bel)	7	0007	0x07	'	39	0047	0x27	G	71	0107	0x47	g	103	0147	0x67
(bs)	8	0010	0x08	(	40	0050	0x28	H	72	0110	0x48	h	104	0150	0x68
(ht)	9	0011	0x09	)	41	0051	0x29	I	73	0111	0x49	i	105	0151	0x69
(nl)	10	0012	0x0a	*	42	0052	0x2a	J	74	0112	0x4a	j	106	0152	0x6a
(vt)	11	0013	0x0b	+	43	0053	0x2b	K	75	0113	0x4b	k	107	0153	0x6b
(np)	12	0014	0x0c	,	44	0054	0x2c	L	76	0114	0x4c	l	108	0154	0x6c
(cr)	13	0015	0x0d	-	45	0055	0x2d	M	77	0115	0x4d	m	109	0155	0x6d
(so)	14	0016	0x0e	.	46	0056	0x2e	N	78	0116	0x4e	n	110	0156	0x6e
(si)	15	0017	0x0f	/	47	0057	0x2f	O	79	0117	0x4f	o	111	0157	0x6f
(dle)	16	0020	0x10	0	48	0060	0x30	P	80	0120	0x50	p	112	0160	0x70
(dc1)	17	0021	0x11	1	49	0061	0x31	Q	81	0121	0x51	q	113	0161	0x71
(dc2)	18	0022	0x12	2	50	0062	0x32	R	82	0122	0x52	r	114	0162	0x72
(dc3)	19	0023	0x13	3	51	0063	0x33	S	83	0123	0x53	s	115	0163	0x73

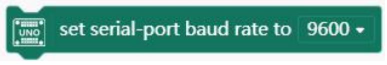
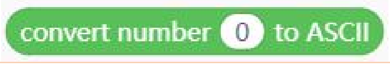
If we want to see characters, we can use the operator in Mind+ to convert them to characters.



convert number 0 to ASCII

convert string "a" to ASCII

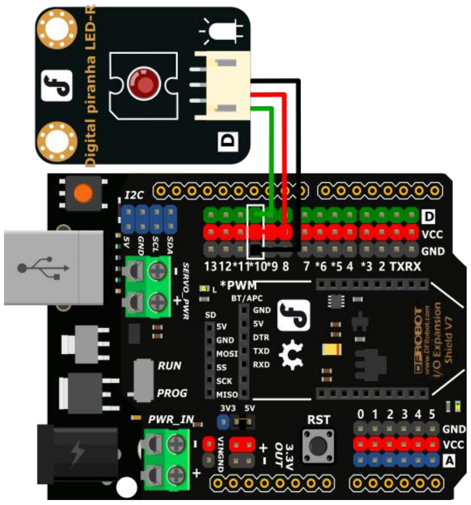
Command List

Module	Type	Description
 set serial-port baud rate to 9600	Baudrate setting	Set the baudrate for serial communication.
 convert number 0 to ASCII	Operator	Converts ASCII values to characters.

Hands-On

Hardware connection

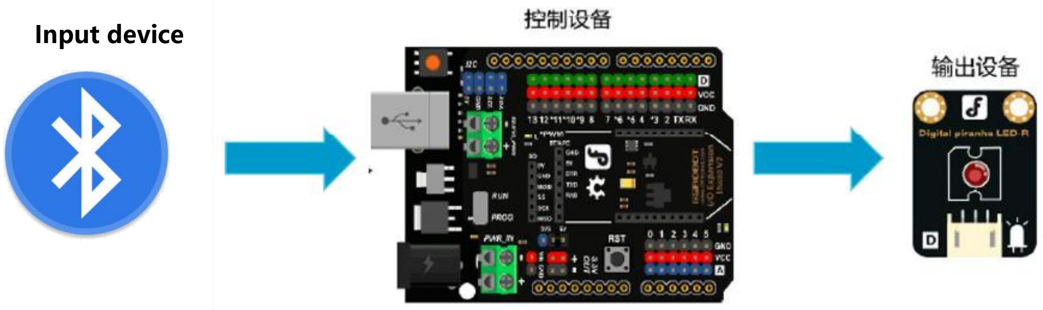
Plug the 3-pin interface of the LED module directly to the Digital Pin 10 of Arduino Uno.


Hardware connection (LED-10)

Please match the colors when plugging

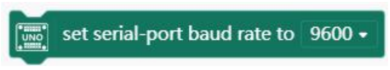
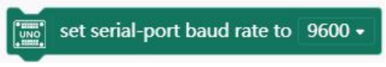
Programming

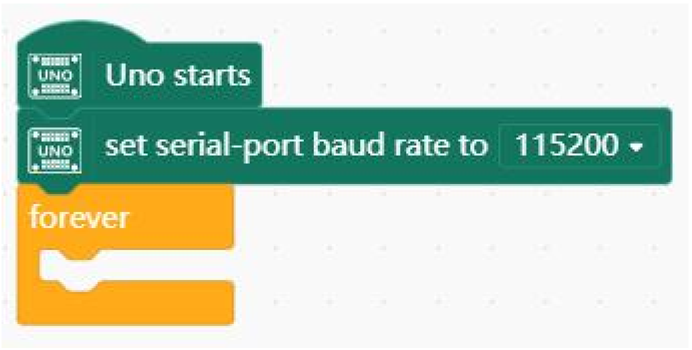
On the basis of controlling LEDs directly with UNO, we use the commands received by Bluetooth to control the LEDs, forming a relationship of "input-control-output", as shown below:



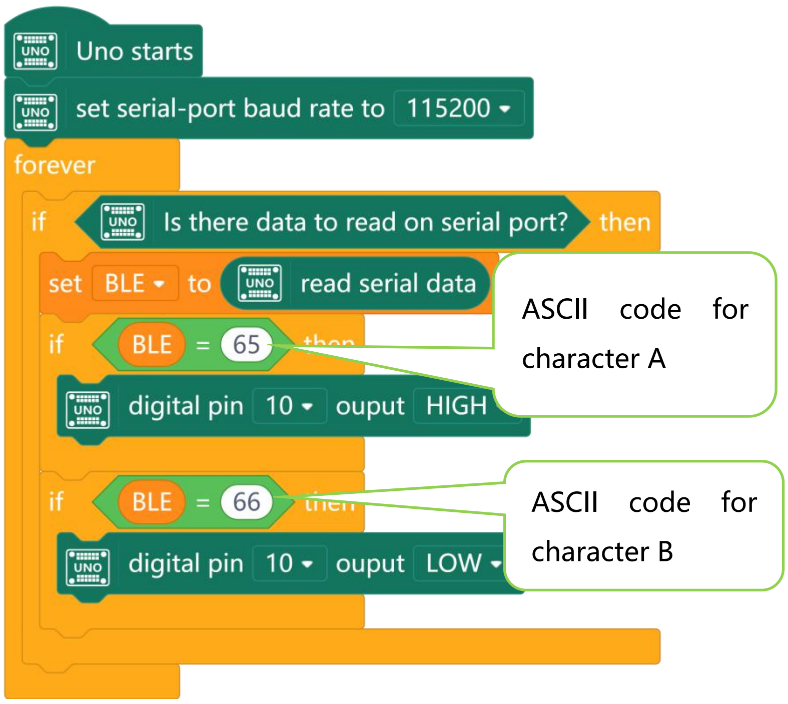
Previously, we learned how to control LEDs with button switches. The mechanism remains the same, but we are replacing the switches with Bluetooth for control.

Write the program

1) Find the "  set serial-port baud rate to 9600 " command in the "Arduino" module,  set serial-port baud rate to 9600 drag it to the editing pane on the right, and then change the baudrate value to 115200.


Uno starts
set serial-port baud rate to 115200
forever

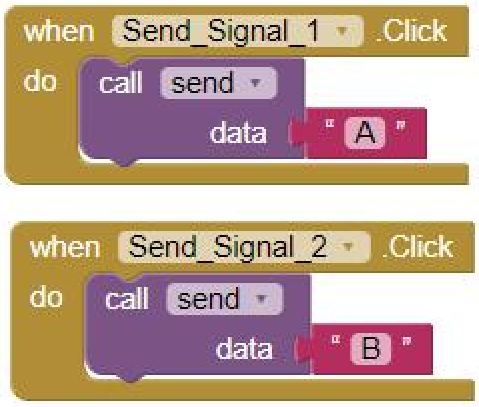
1. Reference program



```
Uno starts
set serial-port baud rate to 115200
forever
  if Is there data to read on serial port? then
    set BLE to read serial data
    if BLE = 65 then
      digital pin 10 ouput HIGH
    if BLE = 66 then
      digital pin 10 ouput LOW
```

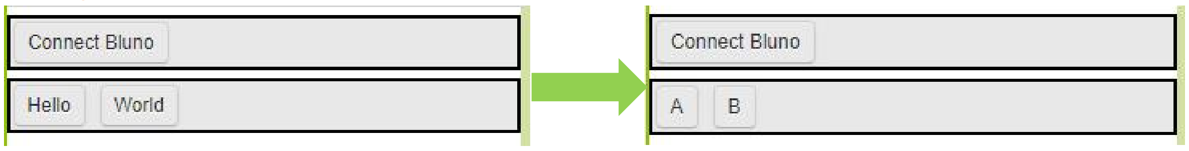
2. **Modify app**

Modify the sending signal in app Inventor and change the previous Hello and World to A and B.



```
when Send_Signal_1 .Click
do  call send
         data "A"

when Send_Signal_2 .Click
do  call send
         data "B"
```

Then adjust the text of the buttons in the interface.



3. **Program’ s effect**

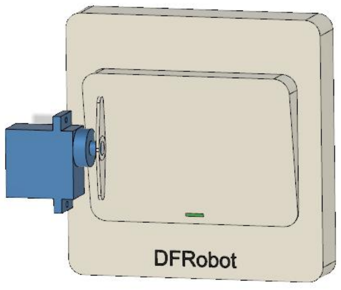
Press button A with the phone, the LED is on; press button B with the phone, the LED is off.

Further Exercise

Everyone has learned to control LED lights with Bluetooth. Can we use Bluetooth to control other actuators?

Project 23 Bluetooth Access Control

In the previous course, we learned to control the LED and other actuators using mobile phone Bluetooth. In this section, we will use phone’ s Bluetooth to control the rotation of the servo to make an access control.



Task Navigation

1. Use the phone app to control the servo.
2. Use phone to turn on and off lights in the house.

Key Points Analysis

1. Let the app send signals that can control the servo.

In the previous section, we made an app that turns on and off of the LED. It is performed by sending data A and B. Here we can use the app made in the previous section to control the servo action.

2. Adjusting the servo to angles that can successfully open and close the switch panel.

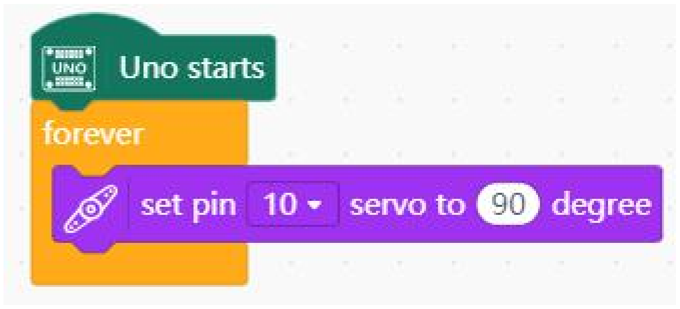
Different switches might have different button sizes, therefore different rotation angles by the servo might be needed. The angles in this chapter are only for reference and you should adjust the angles according to situation.

Hands-On

Arduino side Programming

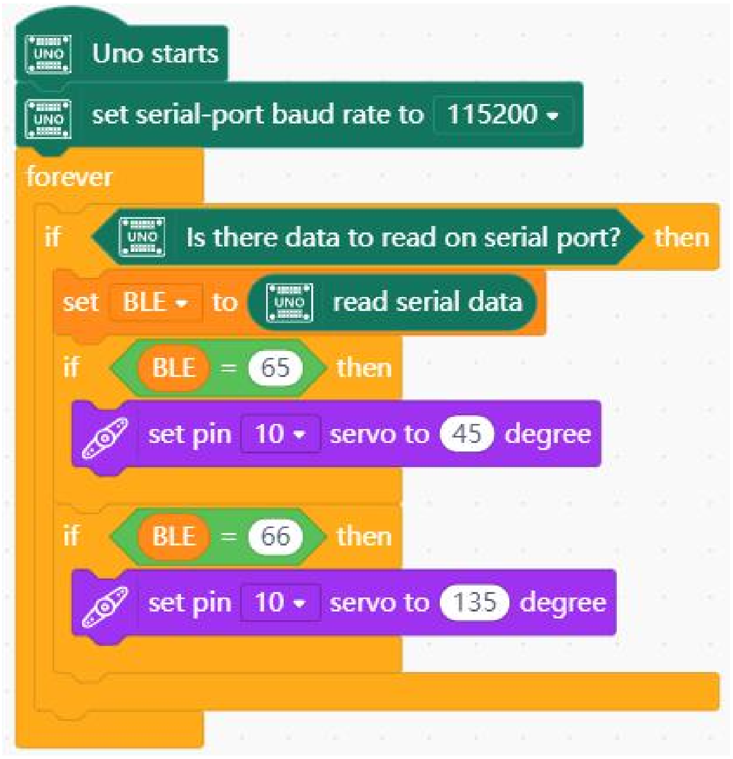
1. Write the program

1) Open the "Actuator" and find the servo module and drag it to the program. Change the pin to 10 and the angle to 90 degrees as the switch initial position. At this time, the angle of the servo is that when switch is not pressed.



```
Uno starts
forever
    set pin 10 ▾ servo to 90 degree
```

1. Reference program



```
Uno starts
set serial-port baud rate to 115200 ▾
forever
    if Is there data to read on serial port? then
        set BLE ▾ to read serial data
        if BLE = 65 then
            set pin 10 ▾ servo to 45 degree
        if BLE = 66 then
            set pin 10 ▾ servo to 135 degree
```

3. Program effect

After sending the A signal with the phone, the servo turns to 45 degrees; and after sending the B signal, the steering gear turns to 135 degrees. Adjust the switch status by turning the servo.

4. Program analysis

Different switches might have different button sizes, and servo' s position might vary as well. Therefore, different rotation angles by the servo might be needed.

Further Exercise

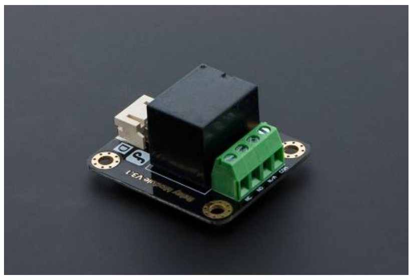
Now that you have mastered the use of the servo, try to apply the servo to other scenarios.

Chapter 11 Smart Home

Previously, we learned to send and receive data to and from the Arduino board through a mobile app, and use the mobile app to control LEDs, servos, etc. However the functions are simple and the interface of the app is relatively primitive. In this chapter we are going to make a feature-rich app with a nice-looking interface. Via which you can control your home appliances with your phone.

Project 24 Special Switch -- Relay

The Arduino UNO board we use can output 5V or 3.3V, and the voltage of common appliances in the home is generally 12V or higher. It’ s kind of unlikely to use Arduino to control these appliances. So in this chapter, we will use relay to solve this problem.

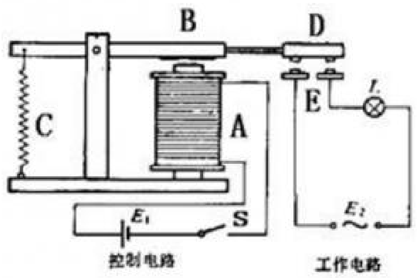


Task Navigation

1. Understanding relays
2. Using relay to control lights

Key Points Analysis

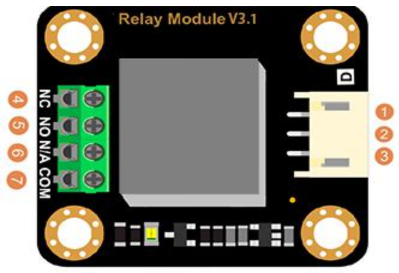
How the relay works.



When the relay is on, the electromagnet A is charged, and the armature is pulled down to make D and E contact, and the working circuit is closed. When the electromagnet loses power, it loses its magnetism. The spring pulls the armature and cuts open the working circuit. Therefore, the relay is a switch that controls the working circuit using an electromagnet. The advantages of using a relay to control the circuit is that: it uses low voltage to control the high voltage, and, so that we can use Arduino to control high voltage electrical appliances.

Note: the maximum power of the working circuit supported by this relay supports up to 250V / 10A AC devices or 28V / 10A DC devices. Do not exceed the limits during use of relay. Attention: when **connecting, if it is a 3.0-5.0V low-power electrical equipment, you can directly connect to the Arduino to supply power. If it is externally powered, pay special attention to your safety.**

Relay module interface description

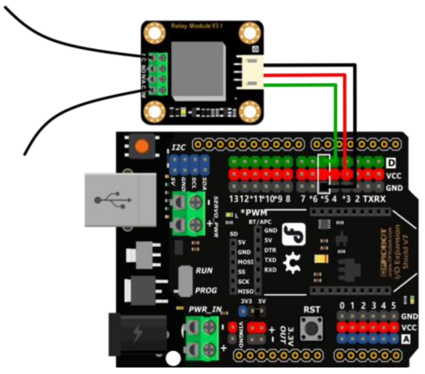


No.	Name	Description
1	−	Power -
2	+	Power +
3	D	Signal Control Input
4	NC	Normally Closed: disconnected usually, connect when signal comes(Power on)
5	NO	Normally Open: connected usually, disconnect when signal comes(Power down)
6	N/A	Empty pin
7	COM	Common Contact/Port

Hands-On

Hardware connection

Connect the relay module to UNO Digital Pin 5.



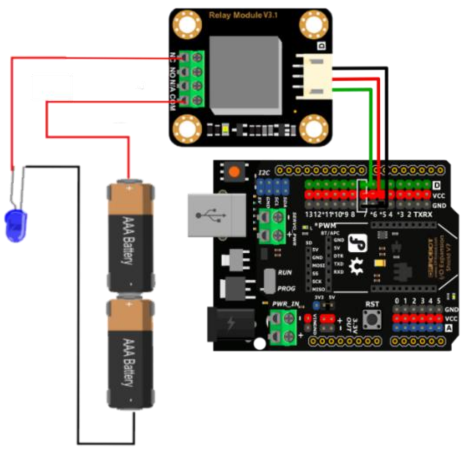
Hardware connection (relay-5)

Please match the colors when plugging

On the working circuit side of the relay, connect the lamp power to the NC and COM ports of the relay.

Note: lamp is off when wiring. Here we use a 12V1A DC power supply for the lamp.

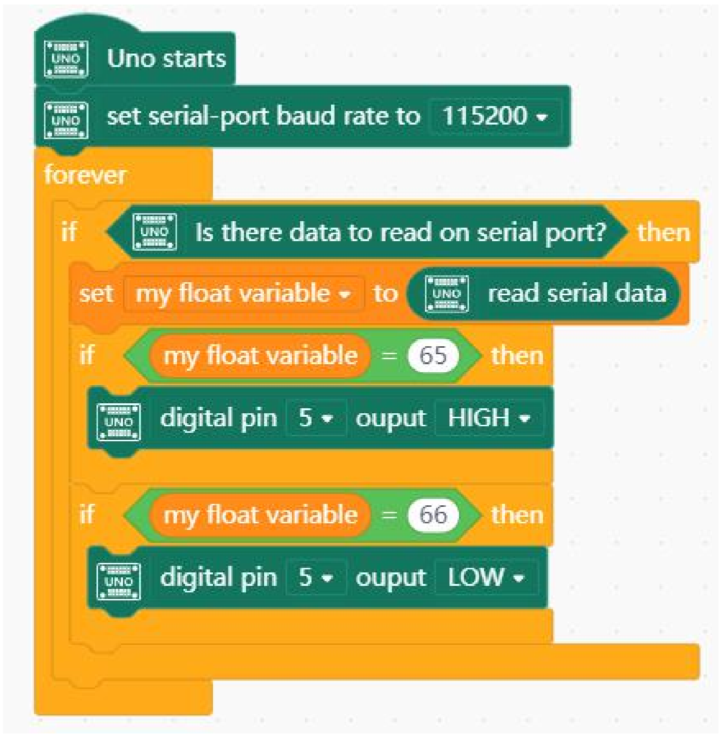
If we don't have a suitable table to use, we can use LED lights or other elements instead of the lamp for demonstration.



Programming

With the way a relay controls working circuit, you can close or open the working circuit by sending high or low level to the corresponding pins. Here, we connect the lamp to the port that is normally closed. When relay’ s pin has high level, the relay closes the circuit and the lamp is on.

1. Reference program



```
Uno starts
set serial-port baud rate to 115200
forever
  if Is there data to read on serial port? then
    set my float variable to read serial data
    if my float variable = 65 then
      digital pin 5 ouput HIGH
    if my float variable = 66 then
      digital pin 5 ouput LOW
```

2. Program analysis

Here we do not need to modify the app. When the app sends data "A", the relay is connected and the table lamp is on; when the app sends data "B", the relay is off and the table lamp is off.

Further Reading

Common applications of relays in life:

We might not be familiar with the name “relay” , and it seems pretty far from our daily life. That’ s simple not true. Relays are in almost every electrical appliance in our life.

Car turn signals and relays

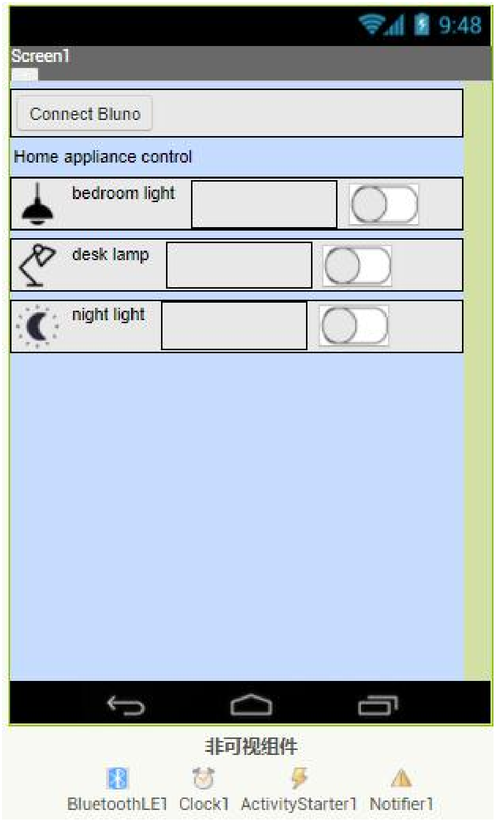
When we use the relay, we can hear a crisp "snap" sound when the relay is turned on. Now, we can try to think what appliances make this sound. (Different types of relays are used in different appliances, so you won' t hear exactly same sounds.) For example, the relay is used to flash in the car's turn signal. When the car has the turn signal on, the "tick" sound we hear is made by the relay. Relays are also used in the computers and heating plate in rickcookers.

Further Exercise

Now that you have mastered controlling a lamp through the relay, can you use relay to control other appliances? Pay special attention to your safety.

Project 25 Smart Home

We have already controlled all kinds of modules via app and Bluetooth in previous chapters. In this chapter, we will integrate all these functions and make a comprehensive smart home app.



Task Navigation

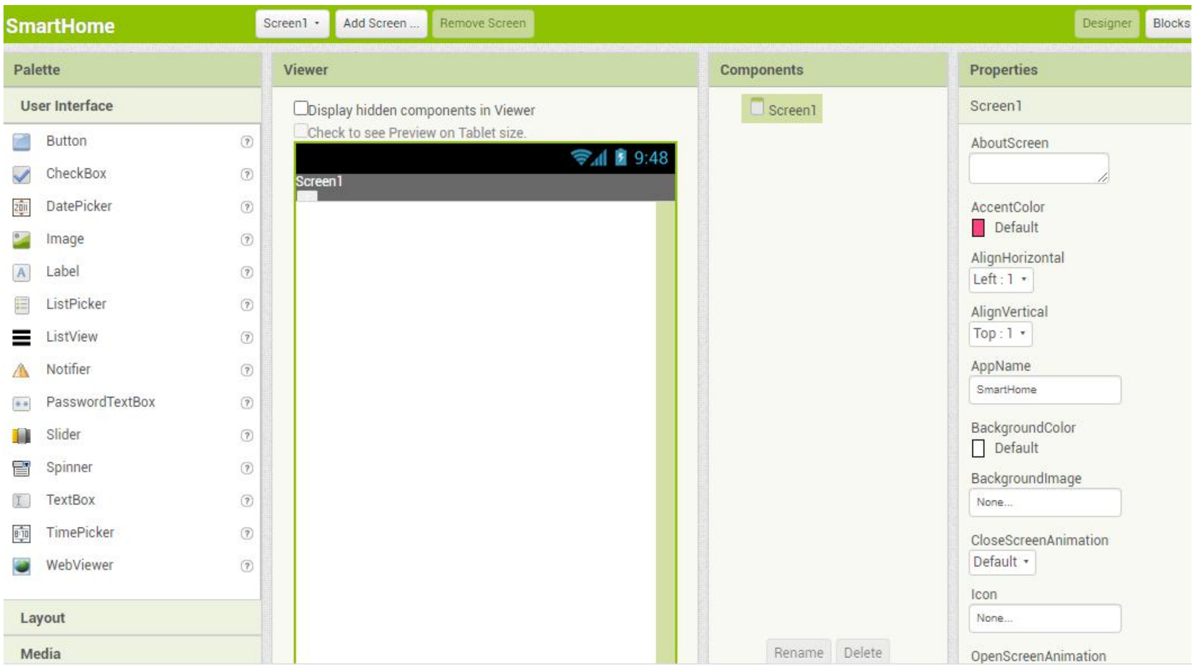
1. App function integration, and interface optimization.
2. Complete the smart home app

Hands-On

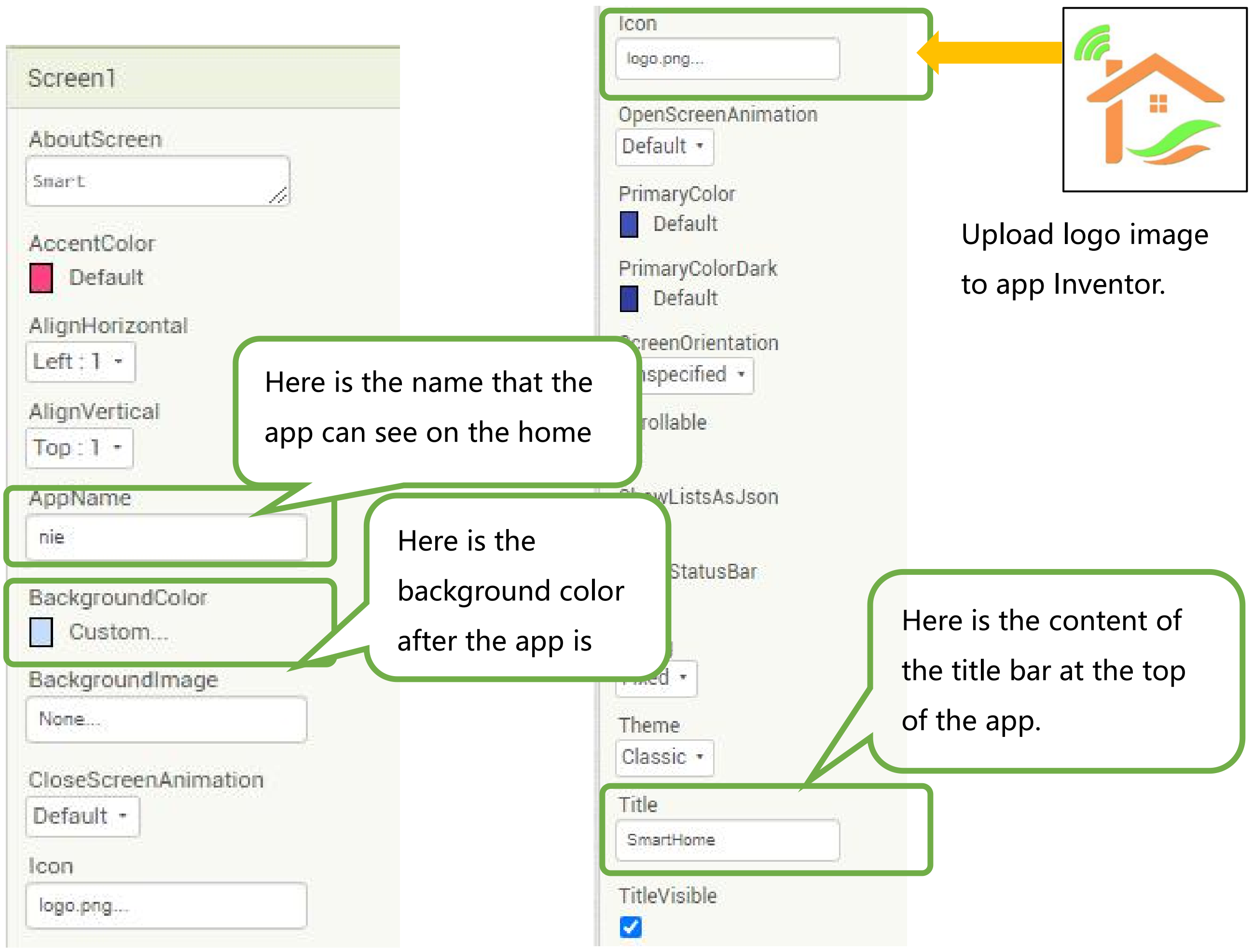
App production

1. Interface design

1) Open app inventor and create a new project.



2) Under " **Screen1** " component properties, we can edit some properties of the app to make it closer to common app design, such as: app icon, app name, background color, etc.

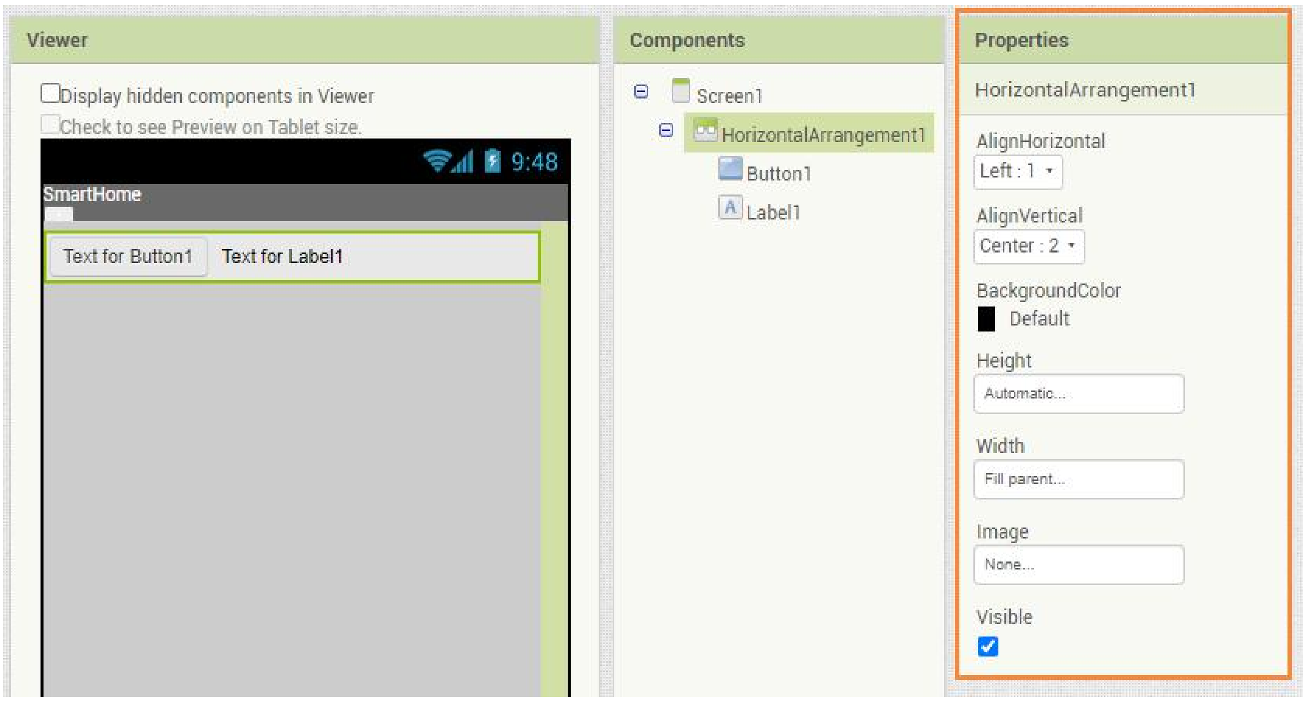

Here is the name that the app can see on the home
Here is the background color after the app is
Upload logo image to app Inventor.
Here is the content of the title bar at the top of the app.

Screen1

AboutScreen: Smart
AccentColor: Default
AlignHorizontal: Left : 1
AlignVertical: Top : 1
AppName: nie
BackgroundColor: Custom...
BackgroundImage: None...
CloseScreenAnimation: Default
Icon: logo.png...

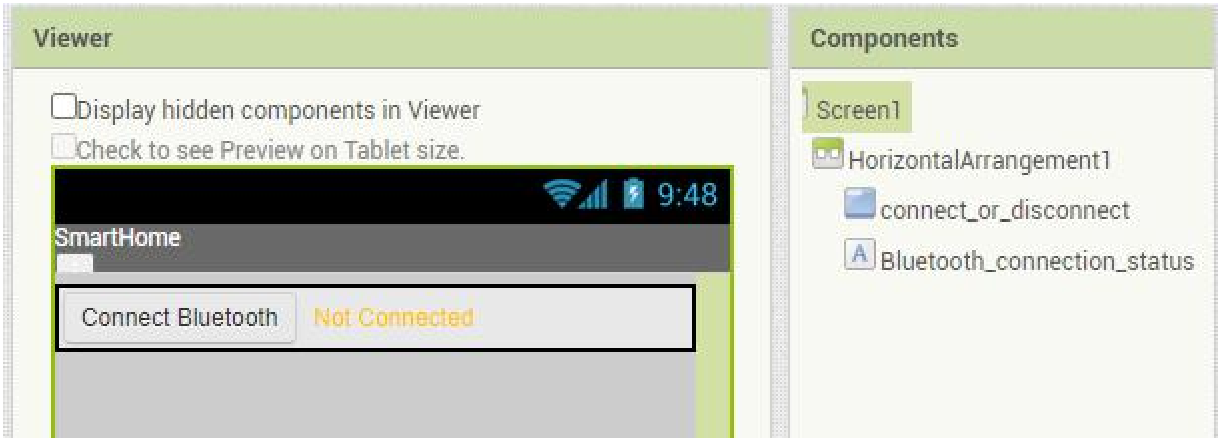
Icon: logo.png...
OpenScreenAnimation: Default
PrimaryColor: Default
PrimaryColorDark: Default
creenOrientation: nspecified
rollable
wListsAsJson
StatusBar
Theme: Classic
Title: SmartHome
TitleVisible: ✓

3) Now design the component for Bluetooth connection. First set **horizontal layout**, then change vertical align to center and change width to fill. Then add a **button** and **label** to the **horizontal layout**.

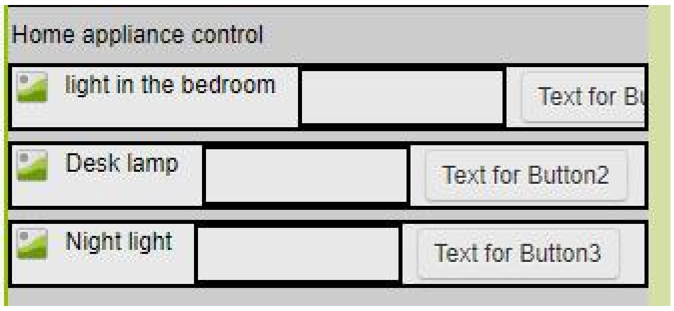


AlignHorizontal: Left : 1
AlignVertical: Center : 2
BackgroundColor: Default
Height: Automatic...
Width: Fill parent...
Image: None...
Visible: ✓

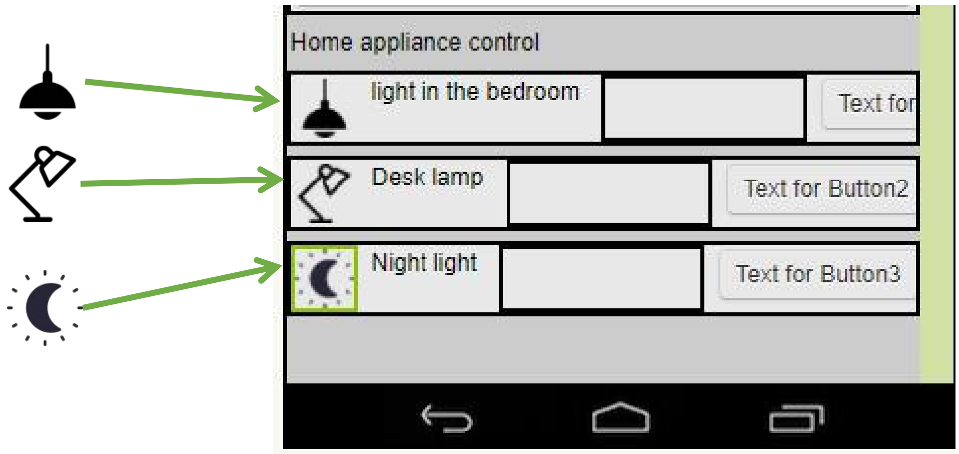
4) Change the display text of **Button 1** to **" Connect Bluetooth"** and the display text of **Label 1** to "**Not Connected**" (color changes to orange). Rename the component name of **button 1** to "**connect or disconnect**", and rename the component name of **label 1** to "**Bluetooth connection status**".



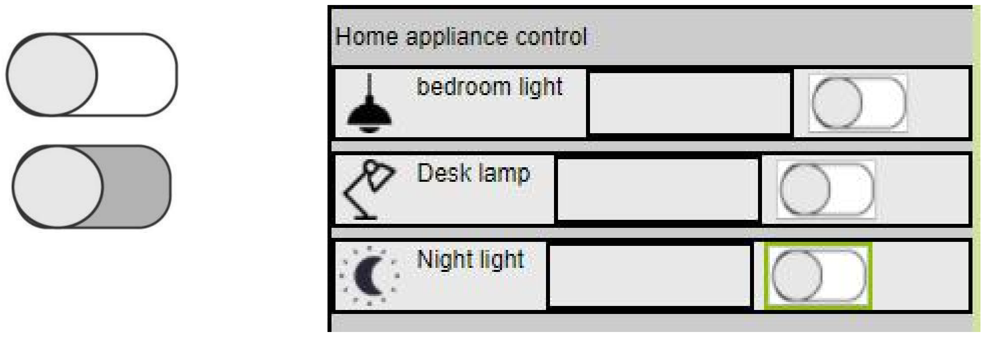
5) Then make the home appliance control part layout. Add three **horizontal layouts** below ;
After changing the width of the **horizontal layout** to Fill:
add an **image**, a **label**, a **horizontal layout** (the width and height are changed to Fill), and a **button**;
The three **horizontal layouts** are the same.
6) Set the label text as shown.



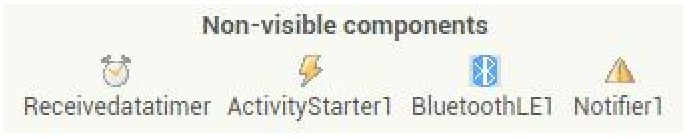
7) Upload images to appinventor to set the icons of appliances. Set the image size to 30*30 pixels.



8) Modify the **button** graphic; upload the switch status image to the app inventor, and set the button graphic to the state "Off". Delete the button display text.

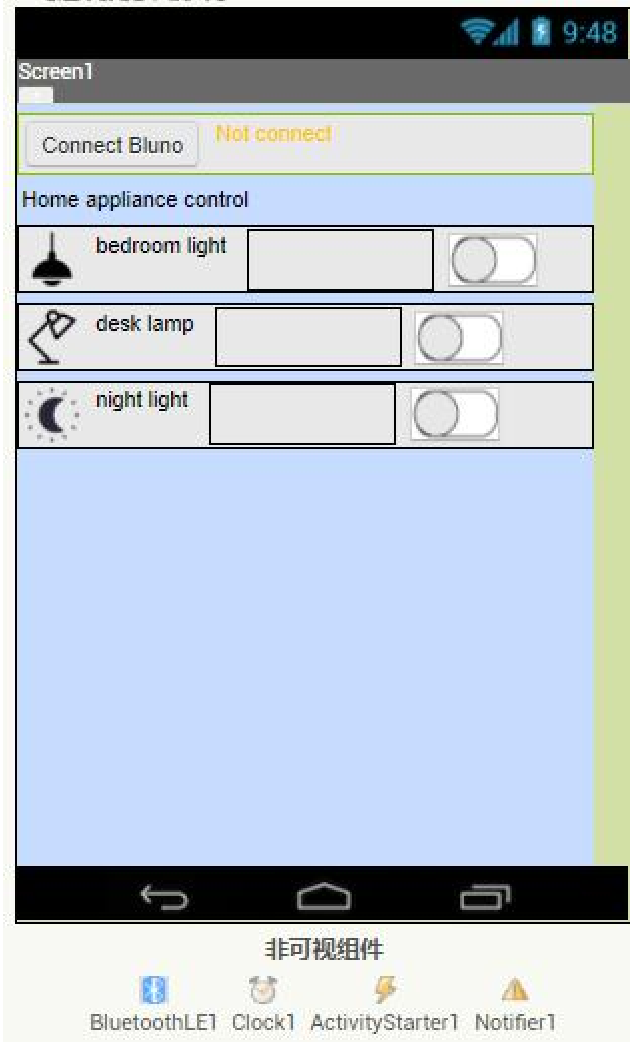


9) The interface layout is completed. Now add the non-visual components (the Bluetooth addition steps are the same as in Chapter IX; the aix file of the Bluetooth module needs to be uploaded). The timer needs to set the timing interval to 500 milliseconds; and the launcher needs to set the Action to: "android.bluetooth.adapter.action.REQUEST_ENABLE", the same as before.



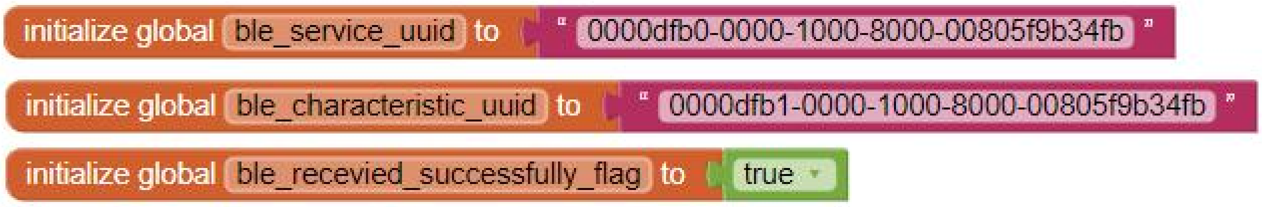
A dialog is added here to alert the user when Bluetooth is not successfully connected, "Bluetooth is not connected". The setting of it will be in the logic section. All other components have been used in previous cases.

10) Complete the interface design. (You can adjust the color and pattern of the icon / text / background according to your preferences.)



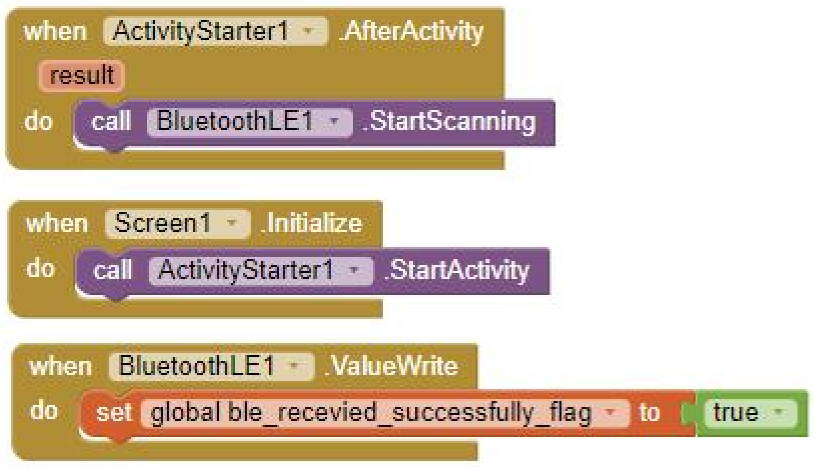
2. Logical design

1) Define global variables. The first two are the same as the previous case, as they fixed parameters. Here we add a sign for a successful connection. Because the app we use this time are capable of both sending and receiving. We need this sign to check status.



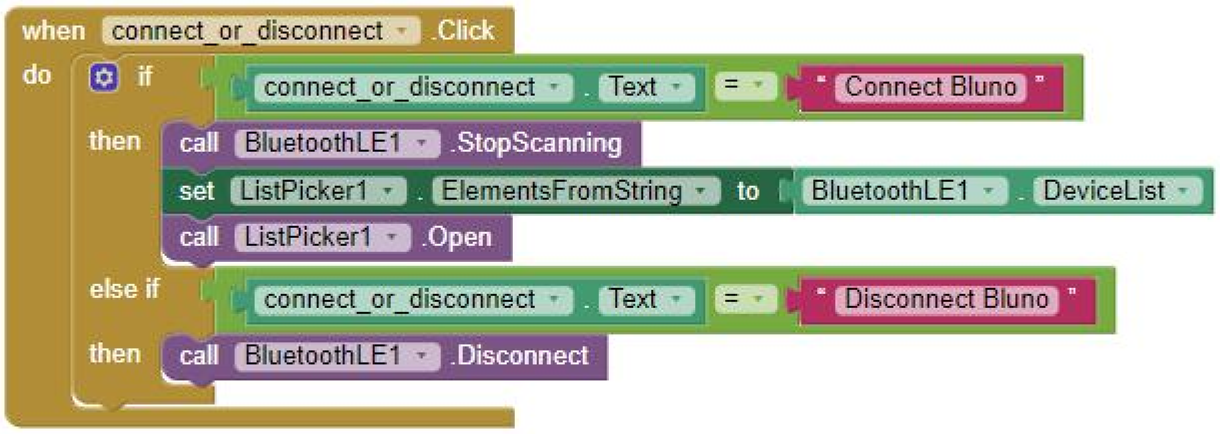
```
initialize global ble_service_uuid to " 0000dfb0-0000-1000-8000-00805f9b34fb "
initialize global ble_characteristic_uuid to " 0000dfb1-0000-1000-8000-00805f9b34fb "
initialize global ble_recevied_successfully_flag to true
```

2) Set the screen initialization parameters, which are as before. Use the launcher to start Bluetooth when the screen is initialized.

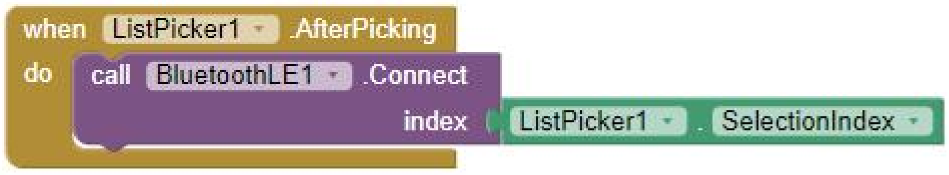


```
when ActivityStarter1 .AfterActivity
  result
do  call BluetoothLE1 .StartScanning

when Screen1 .Initialize
do  call ActivityStarter1 .StartActivity

when BluetoothLE1 .ValueWrite
do  set global ble_recevied_successfully_flag to true
```

3) Set the function to connect to Bluetooth when the Connect button is pressed, similar to previous cases.

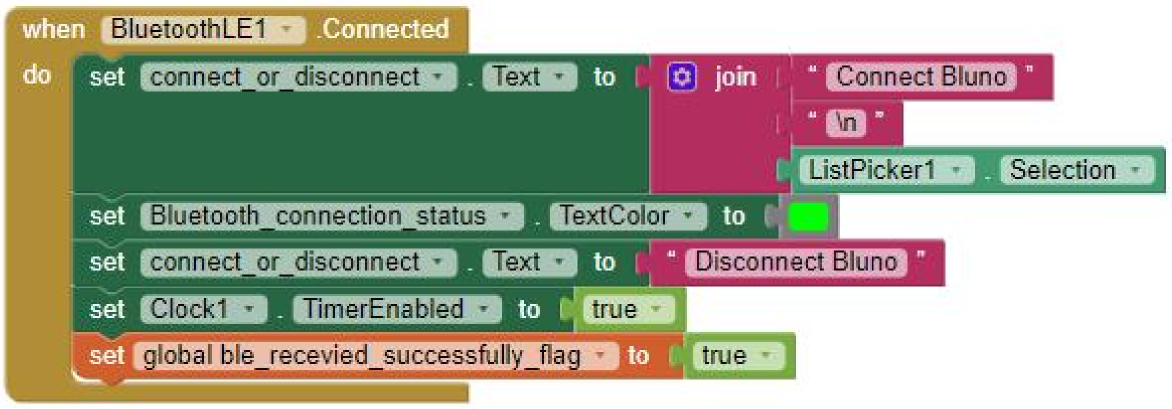


```
when connect_or_disconnect .Click
do  if      connect_or_disconnect . Text = " Connect Bluno "
    then    call BluetoothLE1 .StopScanning
            set ListPicker1 . ElementsFromString to BluetoothLE1 . DeviceList
            call ListPicker1 .Open
    else if connect_or_disconnect . Text = " Disconnect Bluno "
    then    call BluetoothLE1 .Disconnect
```

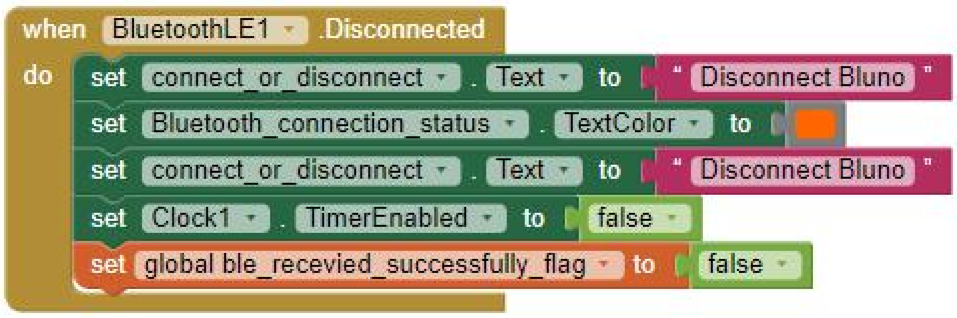


```
when ListPicker1 .AfterPicking
do  call BluetoothLE1 .Connect
                    index  ListPicker1 . SelectionIndex
```

4) Set the status display and checking when the Bluetooth connection is successful and disconnected.

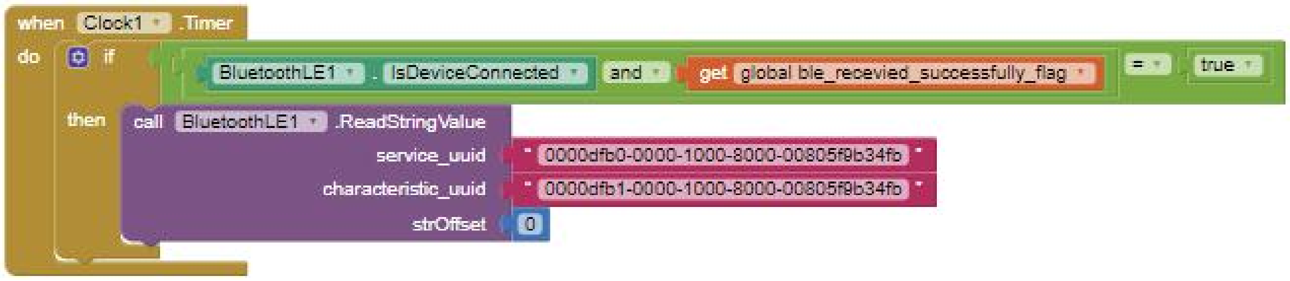


```
when BluetoothLE1 .Connected
do  set connect_or_disconnect . Text to  join  " Connect Bluno "
                                               " \n "
                                               ListPicker1 . Selection
    set Bluetooth_connection_status . TextColor to 🟩
    set connect_or_disconnect . Text to " Disconnect Bluno "
    set Clock1 . TimerEnabled to true
    set global ble_recevied_successfully_flag to true
```



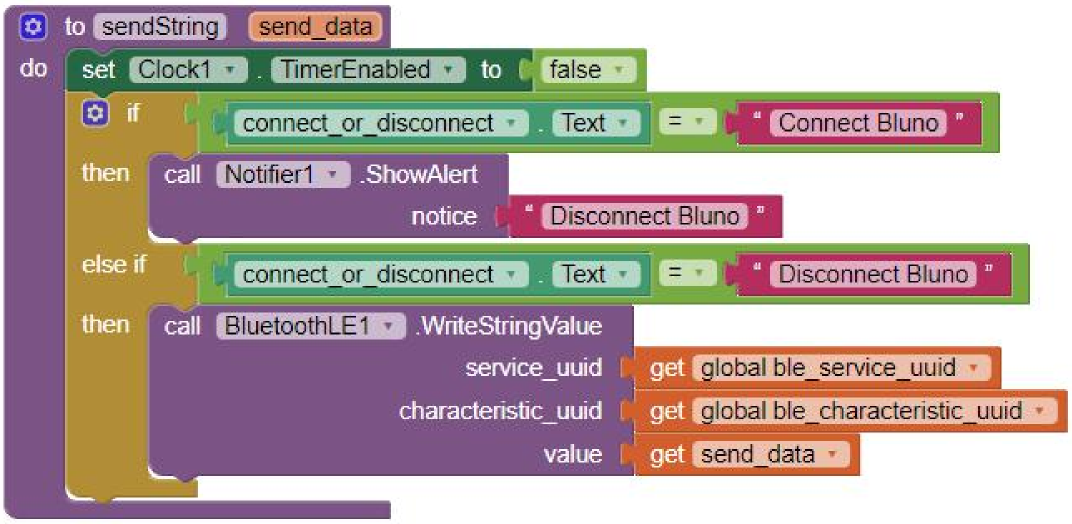
```
when BluetoothLE1 .Disconnected
do  set connect_or_disconnect . Text to " Disconnect Bluno "
    set Bluetooth_connection_status . TextColor to
    set connect_or_disconnect . Text to " Disconnect Bluno "
    set Clock1 . TimerEnabled to false
    set global ble_recevied_successfully_flag to false
```

5) Set a timer. When the timer reaches the time, it will check. If the current Bluetooth connection is successful, it will receive Bluetooth data. The timer we set is 500ms, so every 500ms it will scan for data received.



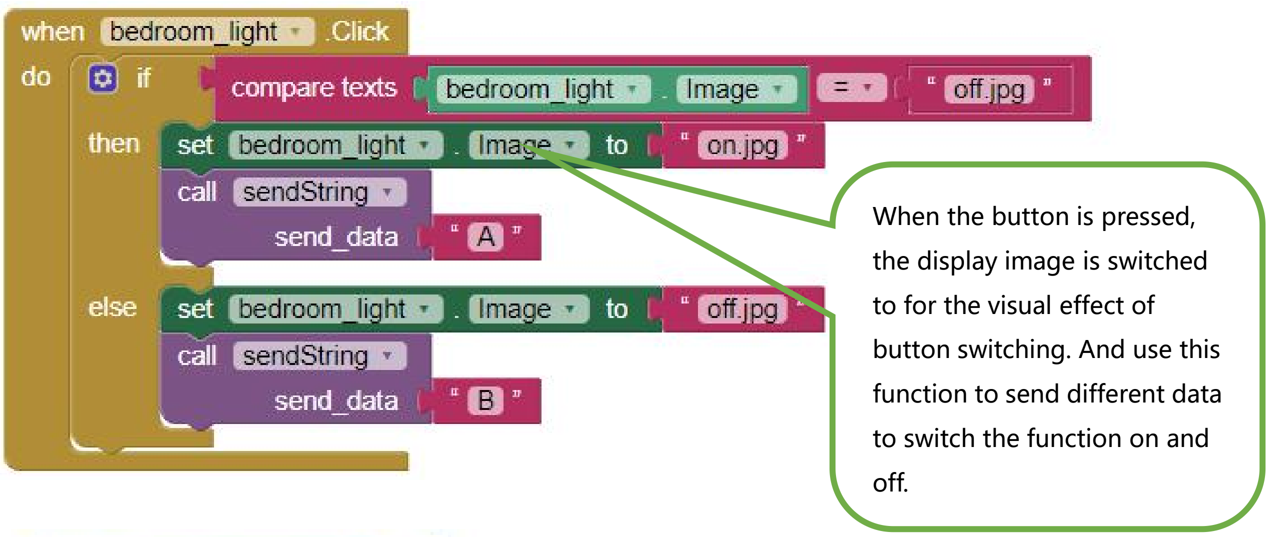
```
when Clock1 .Timer
do  if   BluetoothLE1 . IsDeviceConnected and get global ble_recevied_successfully_flag = true
    then call BluetoothLE1 .ReadStringValue
                   service_uuid " 0000dfb0-0000-1000-8000-00805f9b34fb "
            characteristic_uuid " 0000dfb1-0000-1000-8000-00805f9b34fb "
                      strOffset 0
```

6) Define a process or a function to send data. When the Bluetooth is connected, close the timer for receiving data, then send data. Re-enabled timer after completing data sending. Same with previous cases.



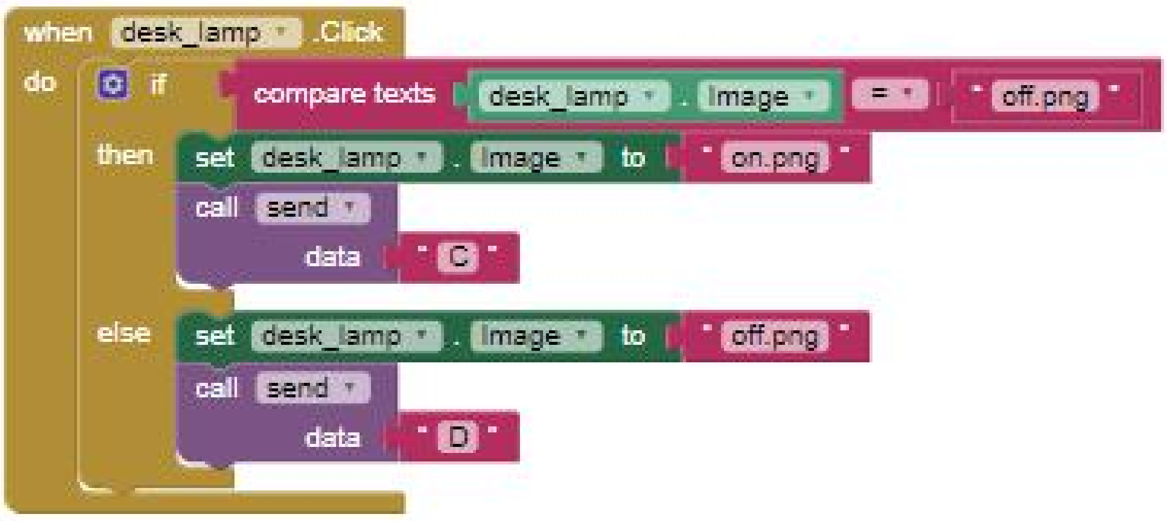
```
to sendString send_data
do  set Clock1 . TimerEnabled to false
    if      connect_or_disconnect . Text = " Connect Bluno "
    then    call Notifier1 .ShowAlert
                     notice " Disconnect Bluno "
    else if connect_or_disconnect . Text = " Disconnect Bluno "
    then    call BluetoothLE1 .WriteStringValue
                       service_uuid get global ble_service_uuid
                characteristic_uuid get global ble_characteristic_uuid
                              value get send_data
```

7) Set the function of several buttons.

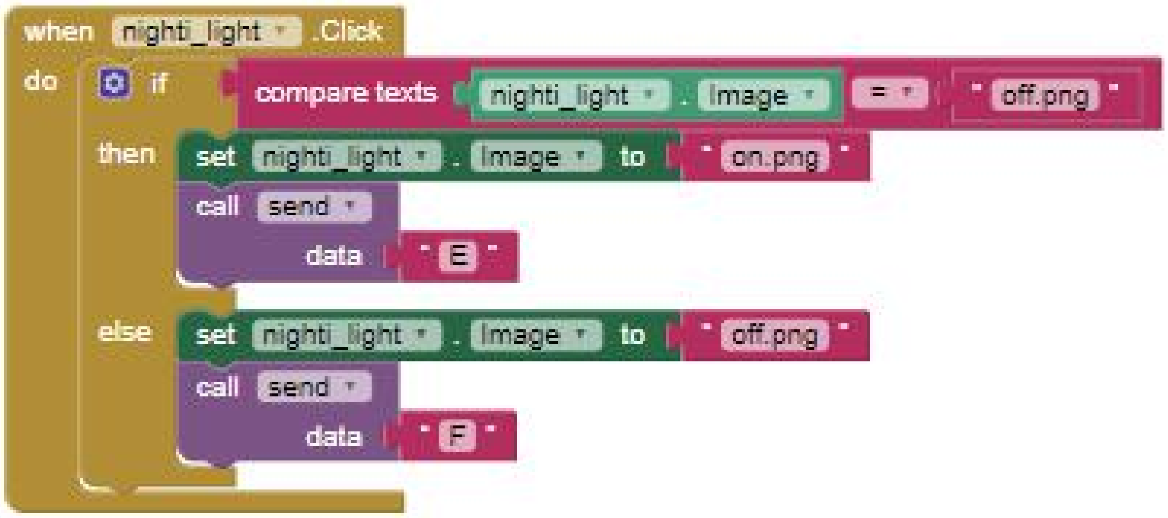


```
when bedroom_light .Click
do  if  compare texts  bedroom_light . Image  =  " off.jpg "
    then  set bedroom_light . Image to " on.jpg "
          call sendString
               send_data " A "
    else  set bedroom_light . Image to " off.jpg "
          call sendString
               send_data " B "
```

When the button is pressed, the display image is switched to for the visual effect of button switching. And use this function to send different data to switch the function on and off.

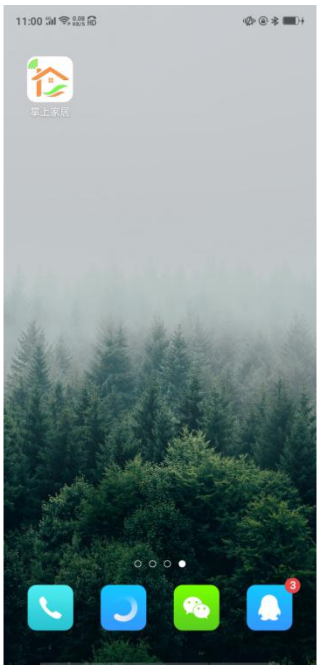
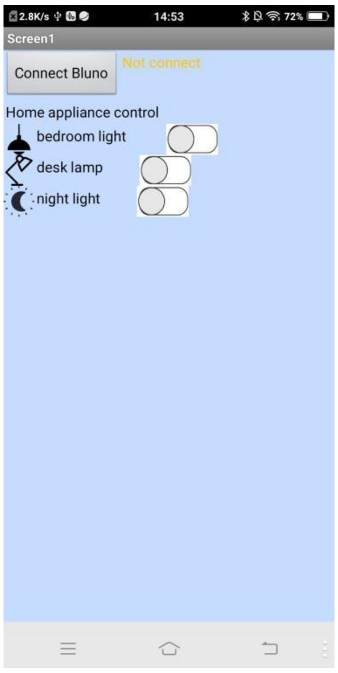


```
when desk_lamp .Click
do  if  compare texts  desk_lamp . Image  =  " off.png "
    then  set desk_lamp . Image to " on.png "
          call send
               data " C "
    else  set desk_lamp . Image to " off.png "
          call send
               data " D "
```



```
when nighti_light .Click
do  if  compare texts  nighti_light . Image  =  " off.png "
    then  set nighti_light . Image to " on.png "
          call send
               data " E "
    else  set nighti_light . Image to " off.png "
          call send
               data " F "
```

This completes the logic design.
Package and install it on your phone.

Screen1

Connect Bluno

Not connect

Home appliance control

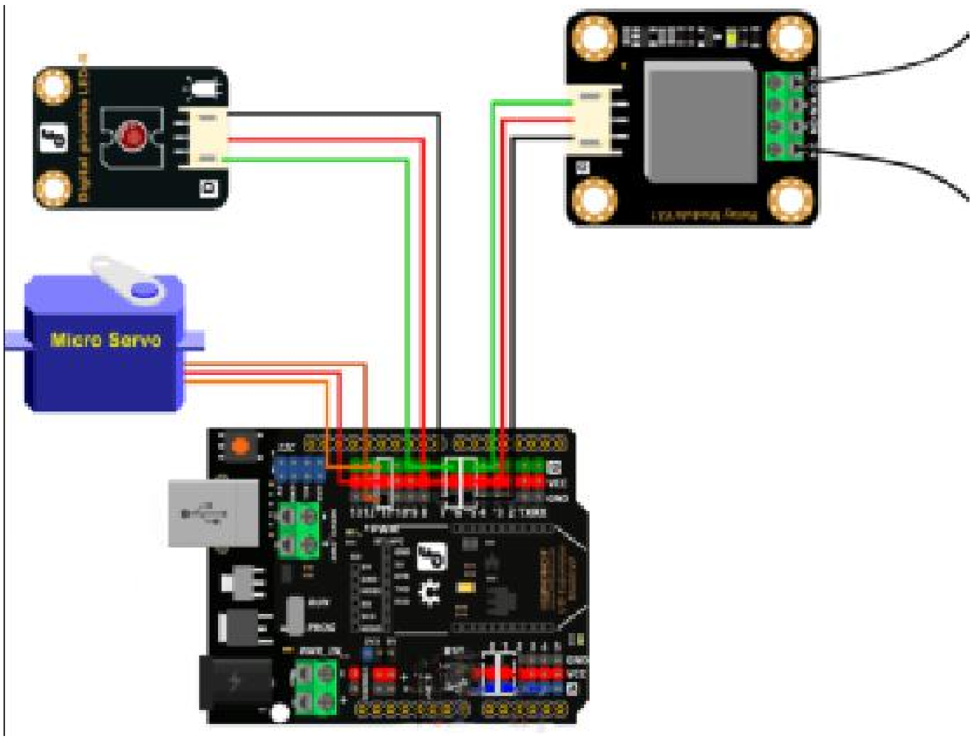
bedroom light

desk lamp

night light

Hardware connection

Connect the relay to Digital Pin 5 on the UNO board, the LED light module to Digital Pin 6, the servo to Digital Pin 11.



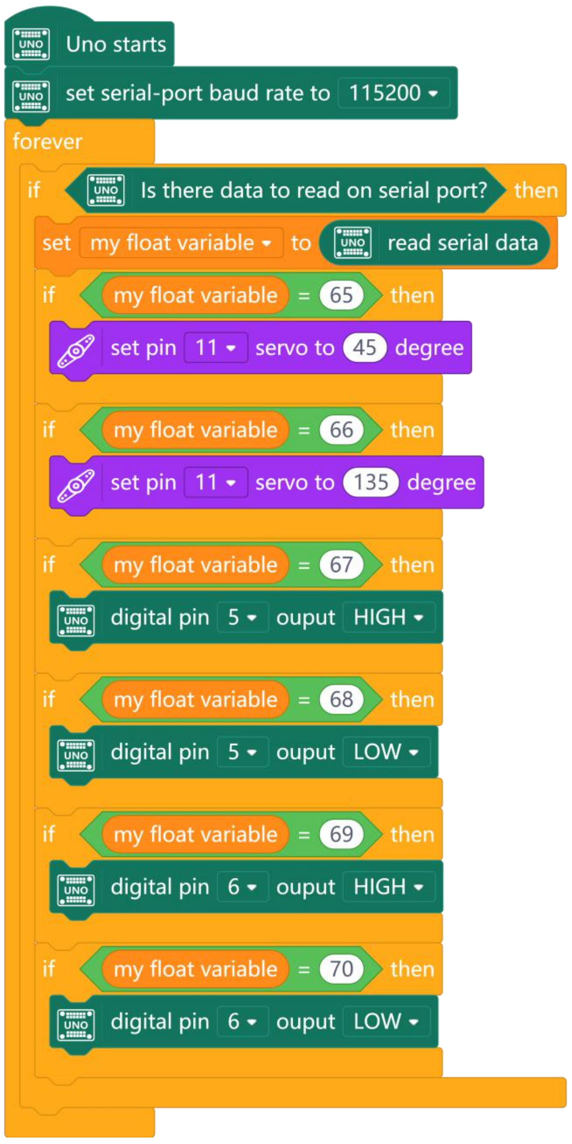
Hardware connection

(Relay -- 5, LED -- 6, Servo -- 11)

Please match the colors when plugging

Programming

1. Reference program



```
Uno starts
set serial-port baud rate to 115200
forever
    if Is there data to read on serial port? then
        set my float variable to read serial data
        if my float variable = 65 then
            set pin 11 servo to 45 degree
        if my float variable = 66 then
            set pin 11 servo to 135 degree
        if my float variable = 67 then
            digital pin 5 ouput HIGH
        if my float variable = 68 then
            digital pin 5 ouput LOW
        if my float variable = 69 then
            digital pin 6 ouput HIGH
        if my float variable = 70 then
            digital pin 6 ouput LOW
```

2. Program’ s effect:

The latest temperature, humidity, and light data can be obtained on the phone by refreshing, and the corresponding servo / relay / LED can be started by pressing the corresponding buttons.

Further Exercise

We are now mainly controlling lights in our homes. Can we actually use Bluetooth to control other home appliances?

There are a lot of fun projects in the DFRobot community, go search and find out! Community URL: www.dfrobot.com